JOURNAL OF APPROXIMATION THEORY 36, 16–31 (1982)

On Recursions for Generalized Splines

LARRY L. SCHUMAKER

Department of Mathematics and Center for Approximation Theory, Texas A & M University, College Station, Texas 77843, U.S.A.

Communicated by E. W. Cheney

Received November 10, 1981

1. INTRODUCTION

One of the most important properties of the space of polynomial splines is that it has a basis of B-splines which can be computed efficiently and accurately by means of certain recursion relations. Recently it has been shown [7, 11] that some spaces of trigonometric and hyperbolic splines also have bases of B-splines which can be computed by analogous recursion relations.

The splines mentioned above are part of an extensive hierarchy of generalized splines which includes Tchebycheffian splines, L-splines, Lg-splines, and many others (cf. [10] and the references therein). Although bases of local-support functions have been constructed for a variety of these generalized spline spaces, the question of when these basis elements can be computed recursively has remained unanswered.

The purpose of this paper is to identify those classes of generalized splines which have B-spline bases which are computable by recursion relations analogous to those for polynomial, trigonometric, and hyperbolic splines. We shall see that, in addition to these three spaces, essentially the only other space of splines which admits of a basis satisfying an analogous recursion relation is a certain space of Tchebycheffian splines.

We begin the paper in Section 2 by reviewing the situation for polynomial splines. In Section 3 we introduce a general class of splines and discuss the kind of recursions we are looking for. In Sections 4 and 5 we treat two general classes of recursions and show that the first works only for polynomial, trigonometric, and hyperbolic splines, while the second works only for a special class of Tchebycheffian splines.

Section 6 is devoted to some results on transformed spline spaces. Here we identify a wide variety of generalized spline spaces with B-spline bases which do not satisfy any recursion relations, but which nevertheless can be dealt

0021-9045/82/090016–16$02.00/0

with numerically in terms of B-splines which do. We conclude the paper with a section containing several remarks and references.

2. Polynomial Splines

In this section we recall several well-known properties of the polynomial splines. First we need some notation. Suppose $a = x_0 < x_1 < \cdots < x_{k+1} = b$ is a partition of the interval $[a, b]$, and let $\Delta = \{x_i\}_1^k$. Given an integer m, let $\mathscr{P}_m = \operatorname{span}\{1, x, \ldots, x^{m-1}\}$ be the space of polynomials of order m, and let $\mathscr{M} = (m_1, \ldots, m_k)$ be a vector of positive integers with $m_i \leqslant m$, $i = 1, 2, \ldots, k$.

The space of *polynomial splines* of order m with knots at $x_1, \ldots, x_k$ of multiplicities $m_1, \ldots, m_k$ is defined by

$$\begin{aligned} \mathscr{S}(\mathscr{P}_m; \mathscr{M}; \Delta) = \{s\colon s|_{(x_i, x_{i+1})} \in \mathscr{P}_m,\ i = 0, 1, \ldots, k \text{ and} \\ D^j s(x_i^-) = D^j s(x_i^+),\ j = 0, \ldots, m - m_i - 1,\ i = 1, \ldots, k\}. \end{aligned} \tag{2.1}$$

This space is of dimension $n = m + \sum_1^k m_i$.

We now describe a basis for $\mathscr{S}$ which can be computed recursively. Associated with $\mathscr{M}$ and Δ, we define the *extended partition* $y_1 \leqslant y_2 \leqslant \cdots \leqslant y_{n+m}$ to be the points

$$\begin{gathered} y_1 = \cdots = y_m = a, \qquad b = y_{n+1} = \cdots = y_{n+m} \\ y_{m+1} \leqslant \cdots \leqslant y_n = \overbrace{x_1, \ldots, x_1}^{m_1}, \ldots, \overbrace{x_k, \ldots, x_k}^{m_k}. \end{gathered} \tag{2.2}$$

Associated with this extended partition, we now define

$$\begin{aligned} B_i^1(x) &= \frac{1}{y_{i+1} - y_i}, \qquad && y_i \leqslant x < y_{i+1} \\ &= 0 && \text{otherwise} \end{aligned} \tag{2.3}$$

for $i = 1, 2, \ldots, n-1$. Let B_n^1 be defined similarly, except that we require that its value be $1/(y_{n+1} - y_n)$ throughout the closed interval $[y_n, y_{n+1}]$. Finally, let $B_{n+i}^1 = 0$ for $i = 1, 2, \ldots, m-1$. These B's are called first-order B-splines.

To describe a basis for $\mathscr{S}$, we now introduce higher-order B-splines recursively. For each $r = 2, 3, \ldots, m$, define

$$\begin{aligned} B_i^r(x) &= \frac{(x - y_i) B_i^{r-1}(x) + (y_{i+r} - x) B_{i+1}^{r-1}(x)}{(y_{i+r} - y_i)}, \qquad && y_i < y_{i+r} \\ &= 0 && \text{otherwise,} \end{aligned} \tag{2.4}$$

$i = 1, 2, \ldots, n + m - r$. Then the following result is well known (cf. [10]):

THEOREM 2.1. *The B-splines $\{B_i^m\}_1^n$ form a basis for $\mathcal{S}(\mathcal{P}_m; \mathcal{M}; \Delta)$. Moreover,*

$$B_i^m(x) > 0 \qquad \text{for} \quad y_i < x < y_{i+m}, \tag{2.5}$$

$$B_i^m(x) = 0 \qquad \text{for} \quad x < y_i,\ y_{i+m} < x, \tag{2.6}$$

$i = 1, 2, \ldots, n$.

3. Generalized Splines

We now introduce the space of generalized splines of interest to us in this paper. Suppose that $\Delta = \{x_i\}_1^m$ is a partition of $[a, b]$ and $\mathcal{M}$ is a multiplicity vector as in Section 2. Let $\mathcal{U} = \operatorname{span}\{u_i\}_1^m \subseteq C^{m-2}[a, b]$. Then we call

$$\mathcal{S}(\mathcal{U}; \mathcal{M}; \Delta) = \{s\colon s|_{(x_i, x_{i+1})} \in \mathcal{U},\ i = 0, 1, \ldots, k \text{ and}$$
$$D^j s(x_i^-) = D^j s(x_i^+),\ j = 0, \ldots, m - m_i - 1,\ i = 1, 2, \ldots, k\}$$

the *space of generalized splines with knots at* $x_1, \ldots, x_k$ *of multiplicities* $m_1, \ldots, m_k$.

We can now state the main problem of the paper.

Problem 3.1. Under what conditions on $\mathcal{U}$ is it possible to find a basis $\{B_i^m\}_1^n$ for $\mathcal{S}(\mathcal{U}; \mathcal{M}; \Delta)$ which can be computed by recursions similar to (2.3)–(2.4)?

In order to make this problem more precise, we shall concentrate on recursions in which B_i^r is computed from B_i^{r-1} and B_{i+1}^{r-1}, and which lead to a basis of B-splines $B_1^m, \ldots, B_n^m$ satisfying properties (2.5)–(2.6). The form of the recursions (2.3)–(2.4) suggest two possibilities.

Algorithm 3.2. Suppose $\phi_1, \phi_2, \ldots, \phi_m$ are functions in $C^{m-2}[0, b-a]$. Define

$$B_i^1(x) = \frac{1}{\phi_1(y_{i+1} - y_i)}, \qquad y_i \leqslant x < y_{i+1}$$
$$= 0 \qquad \text{otherwise}, \tag{3.1}$$

$i = 1, 2, \ldots, n + m - 1$ (where B_n^1 is taken to be $1/\phi_1(y_{n+1} - y_n)$ on the closed interval $[y_n, y_{n+1}]$). For $r = 2, 3, \ldots, m$, let

$$B_i^r(x) = \frac{\phi_r(x - y_i)\, B_i^{r-1}(x) + \phi_r(y_{i+r} - x)\, B_{i+1}^{r-1}(x)}{\phi_r(y_{i+r} - y_i)}, \qquad y_i < y_{i+r}$$
$$= 0 \qquad \text{otherwise}, \tag{3.2}$$

$i = 1, 2, \ldots, n + m - r$.

Algorithm 3.3. Suppose $\psi_1,\ldots,\psi_m$ are functions in $C^{m-2}[a,b]$. Define

$$\begin{aligned} B_i^1(x) &= \frac{1}{\psi_1(y_{i+1})-\psi_i(y_i)}, \qquad & y_i \leqslant x < y_{i+1} \\ &= 0 & \text{otherwise,} \end{aligned} \tag{3.3}$$

$i = 1, 2,\ldots, n+m-1$ (with the usual modification for B_n^1). For $r = 2, 3,\ldots, m$, let

$$\begin{aligned} B_i^r(x) &= \frac{[\psi_r(x)-\psi_r(y_i)]\, B_i^{r-1}(x) + [\psi_r(y_{i+r})-\psi_r(x)]\, B_{i+1}^{r-1}(x)}{\psi_r(y_{i+r})-\psi_r(y_i)}, \quad & y_i < y_{i+r} \\ &= 0 & \text{otherwise,} \end{aligned} \tag{3.4}$$

$i = 1, 2,\ldots, n+m-r$.

We discuss these two schemes in detail in Sections 4 and 5, respectively.

4. Polynomial, Hyperbolic, and Trigonometric Splines

In this section we investigate what properties the ϕ's in Algorithm 3.2 must possess in order for it to produce B-splines satisfying the support properties (2.5)–(2.6). The main result of the section is Theorem 4.4 which shows that the only splines for which this algorithm works are the polynomial, trigonometric, and hyperbolic splines.

Clearly, in order for the B-splines generated by (3.1)–(3.2) to satisfy properties (2.5)–(2.6), we must have that $\phi_1(x) > 0$ for $x > 0$, and

$$\phi_i(0) = 0 \qquad \text{and} \qquad \phi_i(x) > 0 \qquad \text{for} \quad x > 0, \quad i = 2, 3,\ldots. \tag{4.1}$$

To obtain further necessary conditions on the ϕ's, we now examine the effect of requiring that the B-splines produced by (3.1)–(3.2) satisfy the usual smoothness associated with B-splines. In particular, if ξ is a μ-tuple knot of the B-spline B_i^r, then we want

$$D^j B_i^r(\xi+) = D^j B_i^r(\xi-), \qquad j = 0, 1,\ldots, r-\mu-1. \tag{4.2}$$

Lemma 4.1. *A necessary condition in order that for any arbitrary prescribed set of knots the B-spline B_i^m constructed recursively by Algorithm* 3.2 *will satisfy* (4.2) *is*

$$\phi_i(x) = c_i \phi_1(x), \qquad i = 2, 3,\ldots, m, \tag{4.3}$$

where $c_2,\ldots, c_m$ are constants.

Proof. We first establish the assertion for $m=2$. To this end, we examine the B-spline B_0^2 associated with the knots $\{y_0, y_1, y_2\} = \{0, 1, z\}$. The recursions yield

$$B_0^2 = \frac{1}{\phi_2(z)} \begin{cases} \phi_2(x)/\phi_1(1), & 0 \leqslant x < 1 \\ \phi_2(z-x)/\phi_1(z-1), & 1 \leqslant x \leqslant z. \end{cases}$$

Now the requirement that B_0^2 be continuous across the knot at $y_1 = 1$ implies

$$\phi_2(1)/\phi_1(1) = \phi_2(z-1)/\phi_1(z-1).$$

Since this must be true for all $z > 1$, we get (4.3) with $c_2 = \phi_2(1)/\phi_1(1)$.

To prove the assertion for general m, we proceed by induction, assuming the result has already been established for $m-1$. Now we examine the B-spline B_0^m associated with the knots $\{y_0,..., y_m\} = \{0, 1,..., m-1, z\}$. Using the recursion (3.2), we may write the requirement that the $(m-2)$nd derivative be continuous across the knot at $y_1 = 1$ as

$$\begin{aligned} 0 &= \phi_m(y_m - y_0)[D^{m-2}B_0^m]_1 \\ &= [D^{m-2}(\phi_m(x-y_0)\, B_0^{m-1})]_1 + [D^{m-2}(\phi_m(y_m - x)\, B_1^{m-1})]_1, \end{aligned} \tag{4.4}$$

where for convenience we use the notation

$$[f]_\xi = f(\xi+) - f(\xi-).$$

Thinking of this as a function of z, clearly the first term in (4.4) is a constant (call it C_1), while the second term can be expanded as

$$\sum_{i=0}^{m-2} \binom{m-2}{i} D^i\phi_m(z-1)(-1)^i\, [D^{m-2-i}B_1^{m-1}]_1$$

which by the smoothness of the $(m-1)$st-order B-spline B_1^{m-1} at 1 reduces to

$$0 = C_1 + \phi_m(z-1)[D^{m-2}B_1^{m-1}]_1.$$

Now it is easily seen that

$$B_1^{m-1}(x) = [\phi_1(x-1)]^{m-2}/\phi_1(1) \cdots \phi_1(m-2)\,\phi_1(z-1), \qquad 1 \leqslant x < 2,$$

and thus (4.4) reduces to

$$0 = C_1 + C_2\phi_m(z-1)/\phi_1(z-1) \qquad (C_i \neq 0),$$

which implies the desired result. ∎

Lemma 4.1 implies that in further discussing Algorithm 3.2 we may assume that all of the ϕ_i's are constant multiples of one fixed ϕ. Since the introduction of constants into the recursions (3.1)–(3.2) only alters the resulting B-splines to the extent that they are multiplied by a non-zero constant, we may as well assume all of the ϕ_i's are equal to ϕ.

LEMMA 4.2. *A necessary condition in order that Algorithm* 3.2 (*with* $\phi_i = \phi$, $i = 1, 2, \ldots, m$) *will produce B-splines satisfying* (4.2) *is that*

$$2\phi(h)\,\phi'(h) = \phi'(0)\,\phi(2h) \qquad \text{all} \quad h > 0. \tag{4.5}$$

Proof. It will suffice to examine B_0^3 with knots $y_i = ih$, $i = 0, 1, 2, 3$. We easily compute

$$B_0^3(x) = \frac{1}{\phi(h)\,\phi(2h)\,\phi(3h)} \begin{cases} \phi(x)^2, & 0 \leqslant x < h \\ \phi(2h-x)\,\phi(x) + \phi(x-h)\,\phi(3h-x), & h \leqslant x < 2h \\ \phi(3h-x)^2, & 2h \leqslant x < 3h. \end{cases}$$

Now coupling the condition $\phi(0) = 0$ with the requirement that the derivative DB_0^3 must be continuous across the knot at $y_1 = h$, we obtain

$$\begin{aligned} &2\phi(x)\,\phi'(x)|_{x=h} \\ &\quad = [\phi(2h-x)\,\phi'(x) - \phi(x)\,\phi'(2h-x) + \phi'(x-h)\,\phi(3h-x)]|_{x=h}, \end{aligned}$$

which reduces to (4.5). ∎

The following lemma shows that identity (4.5) is only satisfied for a very restricted choice of ϕ.

LEMMA 4.3. *Suppose ϕ is a function with $\phi(0) = 0$ and $\phi'(0) = 1$, and suppose that ϕ has a power series expansion in a neighborhood of* 0. *Then the only choices of ϕ which satisfy* (4.5) *are $\phi(x) = x$, $\phi(x) = \sin(\alpha x)/\alpha$ or $\phi(x) = \sinh(\alpha x)/\alpha$, where $\alpha \neq 0$.*

Proof. Suppose that ϕ has the power series expansion

$$\phi(x) = x + \sum_{i=2}^{\infty} a_i x^i. \tag{4.6}$$

Then computing $\phi'(x)$ and $\phi(2x)$ and combining the power series, we see that (4.5) reduces to

$$(k+1)\,a_k + \sum_{i=2}^{k-1} (k-i+1)\,a_i a_{k-i+1} = 2^{k-1} a_k, \qquad k = 3, 4, \ldots$$

with $a_2 = 0$. Solving for a_k, we obtain

$$a_k = \frac{1}{2^{k-1} - k - 1} \sum_{i=2}^{k-1} (k - i + 1)\, a_i a_{k-i+1}, \qquad k = 3, 4, \ldots. \tag{4.7}$$

Since $a_2 = 0$ while at least one of the indices i or $(2j - i + 1)$ is always even, we conclude that $a_{2j} = 0$ for all $j = 1, 2, \ldots$.

It remains to examine the odd numbered coefficients. Let $a_3 = \beta/3!$. Then it is easily shown by induction that $a_{2j+1} = \beta^j/(2j+1)!$. Indeed, assuming the result for $j - 1$, we have

$$\begin{aligned} a_{2j+1} &= \frac{1}{2^{2j} - 2j - 2} \sum_{i=1}^{j-1} [2(j-i)+1] \frac{\beta^i}{(2i+1)!} \frac{\beta^{j-i}}{(2(j-i)+1)!} \\ &= \frac{\beta^j}{2^{2j} - 2j - 2} \sum_{i=1}^{j-1} \frac{1}{[2(j-i)]!\,(2i+1)!}. \end{aligned}$$

Some elementary manipulations show that the sum is equal to $[2^{2j} - 2j - 2]/(2j+1)!$, and it follows that $a_{2j+1} = \beta^j/(2j+1)!$ as asserted.

Now substituting in the expansion ϕ, we obtain

$$\phi(x) = \sum_{j=0}^{\infty} \frac{\beta^j x^{2j+1}}{(2j+1)!}.$$

If $\beta = 0$, this is simply the function $\phi(x) = x$. If $\beta \neq 0$, then we may rewrite this as

$$\phi(x) = \frac{1}{\sqrt{\beta}} \sum_{j=0}^{\infty} \frac{(\sqrt{\beta}x)^{2j+1}}{(2j+1)!} = \frac{\sinh(\sqrt{\beta}x)}{\sqrt{\beta}}.$$

Now the choice $\beta = -\alpha^2$ produces $\phi(x) = \sin(\alpha x)/\alpha$, while the choice $\beta = \alpha^2$ produces $\phi(x) = \sinh(\alpha x)/\alpha$. ∎

Before stating the main result of this section, we need some additional notation. For any positive integer m and any $\alpha > 0$, let

$$\begin{aligned} \mathscr{E}_m^\alpha &= \operatorname{span}\{\cos(\alpha x), \sin(\alpha x), \ldots, \cos((2r-1)\,\alpha x), \sin((2r-1)\,\alpha x)\}, \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad m = 2r \\ &= \operatorname{span}\{1, \sin(2\alpha x), \cos(2\alpha x), \ldots, \sin(2r\alpha x), \cos(2r\alpha x)\}, \qquad m = 2r+1, \end{aligned} \tag{4.8}$$

$$\begin{aligned} \mathscr{H}_m^\alpha &= \operatorname{span}\{\cosh(\alpha x), \sinh(\alpha x), \ldots, \cosh((2r-1)\,\alpha x), \sinh((2r-1)\,\alpha x)\}, \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad m = 2r \\ &= \operatorname{span}\{1, \sinh(2\alpha x), \cosh(2\alpha x), \ldots, \sinh(2r\alpha x), \cosh(2r\alpha x)\}, \quad m = 2r+1. \end{aligned} \tag{4.9}$$

THEOREM 4.4. *The only classes of splines with a B-spline basis* $\{B_i^m\}_1^n$ *which can be computed by Algorithm* 3.2 *are the polynomial, trigonometric, and hyperbolic splines*; *i.e.*, $\mathscr{S}(\mathscr{U};\mathscr{M};\Delta)$ *with* $\mathscr{U}=\mathscr{P}_m$, $\mathscr{E}_m^\alpha$ *or* $\mathscr{H}_m^\alpha$, *respectively.*

Proof. Lemmas 4.1–4.3 show that Algorithm 3.2 can succeed only in the case where all the ϕ_i's are equal to one of the functions

$$\begin{aligned}\phi(x)&=x,\\ \phi(x)&=\sin(\alpha x),\\ \phi(x)&=\sinh(\alpha x)\end{aligned}\tag{4.10}$$

with $\alpha>0$. In the first case we clearly obtain polynomial splines. In the other two cases it is known (cf. [7, 11]) that the algorithm produces trigonometric or hyperbolic splines, respectively. (Strictly speaking, the authors of [7] used $\alpha=\frac{1}{2}$ while, in [11], $\alpha=1$ was used, but the result for general α follows after a simple change of variables.) ∎

While Theorem 4.4 asserts that only the polynomial, trigonometric, and hyperbolic splines have B-spline bases which can be computed by Algorithm 3.2, the following theorem shows that there are some closely related spaces which have B-spline bases which can be generated by a minor modification of Algorithm 3.2.

THEOREM 4.5. *Let* ω *be a positive function in* $C^{m-2}[a,b]$, *and let* $\tilde{\mathscr{U}}=\{\omega u\colon u\in\mathscr{U}\}$, *where* $\mathscr{U}$ *is one of the spaces* $\mathscr{P}_m$, $\mathscr{E}_m^\alpha$, *or* $\mathscr{H}_m^\alpha$. *Then* $\tilde{\mathscr{S}}=\mathscr{S}(\tilde{\mathscr{U}};\mathscr{M};\Delta)$ *has a basis of B-splines* $\tilde{B}_1^m,\dots,\tilde{B}_n^m$ *which can be computed by Algorithm* 3.2 *provided that the* B_i^1 *defined in* (3.1) *are replaced by* $\tilde{B}_i^1(x)=\omega(x)\,B_i^1(x)$, $i=1,2,\dots,n+m-1$.

Proof. It is clear that if Algorithm 3.2 is carried out with B_i^1's replaced by $\tilde{B}_i^1$'s, it will produce the functions

$$\tilde{B}_i^m(x)=\omega(x)\,B_i^m(x),\tag{4.11}$$

where $B_1^m,\dots,B_n^m$ is the B-spline basis for $\mathscr{S}(\mathscr{U};\mathscr{M};\Delta)$. Clearly the $\tilde{B}_1^m,\dots,\tilde{B}_n^m$ satisfy (2.5)–(2.6), and are piecewise in $\tilde{\mathscr{U}}$. To verify that they are B-splines in $\tilde{\mathscr{S}}$, it remains only to check that they satisfy the smoothness condition (4.2). But

$$D^j\tilde{B}_i^m(x)=\sum_{\nu=0}^{j}\binom{j}{\nu}D^\nu B_i^m(x)\,D^{j-\nu}\omega(x),\tag{4.12}$$

and (4.2) follows from the smoothness of B_i^m coupled with that of ω. ∎

We give one simple example to illustrate how this theorem can be used in practice.

EXAMPLE 4.6. Let $\mathscr{S}(\mathscr{P}_m; \mathscr{M}; \Delta)$ be a class of polynomial splines defined on $[0, 1]$, and let $\omega(x) = x^{1/2}$.

Discussion. In this case it is clear that $\mathscr{S}(\tilde{\mathscr{U}}; \mathscr{M}; \Delta)$ is made up of functions in $\tilde{\mathscr{U}} = \text{span}\{x^{1/2}, x^{3/2}, \ldots, x^{m-1/2}\}$. This unusual type of spline space can be of numerical importance; e.g. see [8] where it arises in connection with the solution of singular two-point boundary value problems. ∎

5. TCHEBYCHEFFIAN SPLINES

In this section we examine Algorithm 3.3. The main result is Theorem 5.2 which shows that the only splines for which this algorithm works are certain classes of Tchebycheffian splines.

It is clear from the form of the recursions (3.3)–(3.4) that if we want the B-splines produced by Algorithm 3.3 to have properties (2.5)–(2.6), then we must require that each of the functions $\psi_1, \ldots, \psi_m$ be monotone increasing on $[a, b]$. Now by the assumption that these functions lie in $C^{m-2}[a, b]$, we conclude that each of these functions can be written as an integral of a positive weight function:

$$\psi_i(x) = \psi_i(a) + \int_a^x \omega_i(t)\, dt, \qquad \omega_i(t) > 0 \qquad \text{for} \quad a \leqslant t \leqslant b.$$

We now establish the analog of Lemma 3.1 to show that Algorithm 3.2 will produce B-splines which have the requisite smoothness only if the ψ_i's are all linear combinations of ψ_1 and 1.

LEMMA 5.1. *A necessary condition in order that for any arbitrary prescribed set of knots the B-spline B_i^m constructed recursively from Algorithm* 3.3 *will satisfy* (4.2) *is*

$$\psi_i = a_i + c_i \psi_1, \qquad i = 2, 3, \ldots, m, \tag{5.1}$$

where a_i, c_i are constants.

Proof. The proof is very much like the proof of Lemma 4.1. First we establish the assertion for $m = 2$ by examining the B-spline B_0^2 associated with the knots $\{y_0, y_1, y_2\} = \{0, 1, z\}$. The recursions yield

$$B_0^2(x) = \frac{1}{\int_0^z \omega_2(t)\, dt} \begin{cases} \int_0^x \omega_2(t)\, dt \Big/ \int_0^1 \omega_1(t)\, dt, & 0 \leqslant x < 1 \\ \int_x^z \omega_2(t)\, dt \Big/ \int_1^z \omega_1(t)\, dt, & 1 \leqslant x \leqslant z. \end{cases}$$

Now the requirement that B_0^2 be continuous across the knot at $y_1 = 1$ implies

$$\int_1^z \omega_2(t)\,dt = \int_1^z \omega_1(t)\,dt \left[\int_0^1 \omega_2(t)\,dt \bigg/ \int_0^1 \omega_1(t)\,dt\right].$$

Since this must hold for all $z > 1$, it follows that $\psi_2 = a_2 + c_2\psi_1$.

To prove the result for general m, we proceed by induction, assuming the result has already been established for $m-1$. We now examine the B-spline B_0^m associated with the knots $\{y_0,..., y_m\} = \{0, 1,..., m-1, z\}$. Then proceeding as in the proof of Lemma 4.1, we obtain

$$0 = C_1 + [D^{m-2}B_1^{m-1}]_1 \int_1^z \omega_m(t)\,dt.$$

On the other hand, by the recursion, we know that

$$B_1^m(x) = \frac{[\int_1^x \omega_1(t)\,dt]^{m-1}}{\int_1^2 \omega_1(t)\,dt \cdots \int_1^{m-1} \omega_1(t)\,dt \int_1^z \omega_1(t)\,dt}, \qquad 1 \leqslant x < 2,$$

and substituting in the above, we obtain

$$0 = C_1 + C_2 \int_1^z \omega_m(t)\,dt \bigg/ \int_1^z \omega_1(t)\,dt \qquad \text{all} \quad z > 1.$$

Arguing as before, we find that $\psi_m = a_m + c_m\psi_1$. ∎

Lemma 5.1 implies that in our further discussion of Algorithm 3.3, we may assume that all of the ψ_i's are integrals of one fixed positive weight function w (the constants a_i appearing in (5.1) drop out in the algorithm since we are always taking differences of the ψ_i's at two points). (The constants c_i appearing in (5.1) can be ignored since they simply alter the B-splines by constant multiples.)

We can now state the main result of this section.

THEOREM 5.2. *The only classes of splines with a B-spline basis* $\{B_i^m\}_1^n$ *which can be computed by Algorithm* 3.3 *are the splines* $\mathscr{S}(\mathscr{U}; \mathscr{M}; \Delta)$ *with* $\mathscr{U} = \operatorname{span}\{u_i\}_1^m$, *where*

$$u_i(x) = \left[\int_a^x w(t)\,dt\right]^{i-1} \bigg/ (i-1)!, \qquad i = 1, 2,..., m, \tag{5.2}$$

and w *is a positive function in* $C^{m-2}[a, b]$.

Proof. By the above discussion, we may restrict our attention to the case where $\psi_i(x) = \int_a^x w(t)\,dt$, $i = 1, 2,..., m$. We claim that in this case Algorithm 3.3 produces B-splines $B_1^m,..., B_n^m$ which are in fact identifiable with certain polynomial B-splines. To see this, let $Q_1^m,\ldots,Q_n^m$ be the polynomial B-splines associated with the extended partition $\rho(y_1),\ldots,\rho(y_{n+m})$, where

$$\rho(x)=a+\frac{(b-a)\int_a^x w(t)\,dt}{\int_a^b w(t)\,dt}.$$

Clearly ρ is a monotone 1–1 mapping of the interval $[a,b]$ onto itself. Now if we compare Algorithm 3.3 with the recursions (2.3)–(2.4) for generating the polynomial B-splines, we see immediately that

$$B_i^m(x)=Q_i^m(\rho(x)), \qquad i=1,2,\ldots,n. \tag{5.3}$$

It follows that since the Q_i^m are piecewise polynomials, the B_i^m must be piecewise in the space spanned by $u_1,\ldots,u_m$.

It remains to show that if Algorithm 3.3 is carried out in this case that the functions B_i^m really are splines in $\mathscr{S}(\mathscr{U};\mathscr{M};\Delta)$. We have already seen that they have the correct piecewise nature. We must now show that they have the required smoothness; i.e., that they satisfy (4.2). But this follows immediately from (5.3) and the fact that both ρ and Q_i^m have the required smoothness. ∎

We can now state an analog of Theorem 4.5 which shows that a slight modification of Algorithm 3.3 can be used to compute B-splines for a large class of splines (which we shall see below are precisely the Tchebycheffian splines associated with an ECT-system with equal weights).

THEOREM 5.3. *Let ω be a positive function in $C^{m-2}[a,b]$, and let $\tilde{\mathscr{U}}=\{\omega u\colon u\in\mathscr{U}\}$, where $\mathscr{U}$ is the space in Theorem 5.2. Then $\mathscr{S}(\tilde{\mathscr{U}};\mathscr{M};\Delta)$ has a basis of B-splines $\tilde{B}_1^m,\ldots,\tilde{B}_n^m$ which can be computed by Algorithm* 3.3 *provided that the B_i^1 defined in* (3.3) *are replaced by $\tilde{B}_i^1(x)=\omega(x)\,B_i^1(x)$, $i=1,2,\ldots,n+m-1$.*

We turn now to a discussion of Tchebycheffian spline spaces; i.e., spline spaces of the form $\mathscr{S}(\mathscr{U};\mathscr{M};\Delta)$ where $\mathscr{U}$ is spanned by an Extended Complete Tchebycheff (ECT-) system (see [10]). It is known (cf. [10, p. 364]) that $\mathscr{U}$ is an ECT-system if and only if it has a basis of functions $\{u_i\}_1^m$ of the form

$$\begin{aligned} u_1(x)&=w_1(x),\\ u_2(x)&=w_1(x)\int_a^x w_2(t)\,dt,\\ &\ \vdots\\ u_m(x)&=w_1(x)\int_a^x w_2(s_2)\int_a^{s_2}\cdots\int_a^{s_{m-1}} w_m(s_m)\,ds_m\cdots ds_2, \end{aligned} \tag{5.4}$$

where $w_1, \dots, w_m$ are positive functions on $[a, b]$ with $w_i \in C^{m-i}[a, b]$, $i = 1, 2, \dots, m$.

LEMMA 5.4. *Suppose that $w_2 = w_3 = \cdots = w_m = w$, where w is a positive function in $C^{m-2}[a, b]$. Then the canonical* ECT-*system defined in* (5.4) *is given by*

$$u_i(x) = w_1(x) \frac{[\int_a^x w(t)\, dt]^{i-1}}{(i-1)!}, \qquad i = 1, 2, \dots, m. \tag{5.5}$$

Proof. Clearly the assertion holds for $i = 1, 2$. It will follow for $i = 3, \dots, m$ by induction provided that we can show that $v_i = u_i / w_1$ satisfy

$$v_i(x) = \frac{[\int_a^x w(t)]\, v_{i-1}(x)}{(i-1)}, \qquad i = 3, \dots, m. \tag{5.6}$$

To establish (5.6), we note that (assuming (5.6) for $i - 1$),

$$\begin{aligned} v_i(x) &= \int_a^x w(t)\, v_{i-1}(t) = \int_a^x w(t) \int_a^t w(s)\, v_{i-2}(s) \\ &= \int_a^x w(t) \int_a^x w(s)\, v_{i-2}(s) - \int_a^x w(t) \int_t^x w(s)\, v_{i-2}(s) \\ &= \int_a^x w(t) \int_a^x w(s)\, v_{i-2}(s) - \int_a^x w(t)\, v_{i-2}(t) \int_a^t w(s) \\ &= \int_a^x w(t) \int_a^x w(s)\, v_{i-2}(s) - \int_a^x (i-2)\, w(t)\, v_{i-1}(t). \end{aligned}$$

This implies

$$(i-1)\, v_i(x) = \left[\int_a^x w(t)\right] \left[\int_a^x w(t)\, v_{i-2}(t)\right] = \left[\int_a^x w(t)\right] v_{i-1}(x),$$

which is (5.6). ∎

We conclude this section by giving two specific examples of Tchebycheffian spline spaces corresponding to an ECT-system with equal weights.

EXAMPLE 5.5. Let $w_1(x) = 1$ and $w(x) = \cosh(x)$.

Discussion. In this case the splines in $\mathscr{S}$ are piecewise members of the space

$$\mathscr{U} = \operatorname{span} \left\{ \frac{(\sinh x)^{i-1}}{(i-1)!} \right\} = \operatorname{span}\{1, \sinh(x), \cosh(2x), \sinh(3x), \dots\}.$$

Although this space can be regarded as a space of hyperbolic splines, it is not the same as the space of hyperbolic splines discussed in Section 4. ∎

EXAMPLE 5.6. Let $w_1(x) = 1$ and $w(x) = \cos(x)$.

Discussion. The piecewise nature of the splines in $\mathscr{S}$ in this case are now determined by

$$\mathscr{U} = \text{span}\left\{\frac{(\sin x)^{i-1}}{(i-1)!}\right\} = \text{span}\{1, \sin(x), \cos(2x), \sin(3x),...\}.$$

This space can be thought of as a space of trigonometric splines, but in general it is not the same as the one discussed in Section 4. ∎

6. TRANSFORMED SPLINE SPACES

The techniques of Section 5 suggest that it may be of interest to examine spaces of splines which are obtained from others by multiplication by a positive function and/or a monotone change of variables. Our aim in this section is to show that such transformations can be applied to the classes of splines with recursively computable B-spline bases discussed in Sections 4 and 5 to obtain still wider classes of non-polynomial splines which can be dealt with numerically in terms of B-splines which are recursively computable.

First we need some notation. Throughout this section, we suppose that $[a, b]$ and $[c, d]$ are closed intervals, and that $\rho \in C^{m-2}[c, d]$ is a monotone increasing function mapping $[c, d]$ onto $[a, b]$. In addition, we suppose that $\omega \in C^{m-2}[c, d]$ is a positive function on $[c, d]$. Now given a set of functions $\{u_i\}_1^m$ and a partition $\Delta = \{x_i\}_1^k$ of $[a, b]$ as in Section 2, we define

$$\tilde{\mathscr{U}} = \text{span}\{\tilde{u}_i\}_1^m, \qquad \text{where} \quad \tilde{u}_i(x) = \omega(x)\, u_i(\rho(x)), \quad i = 1, 2,..., m, \tag{6.1}$$

$$\tilde{\Delta} = \{\tilde{x}_i\}_1^k, \qquad \text{where} \quad \tilde{x}_i = \rho^{-1}(x_i), \quad i = 1, 2,..., k. \tag{6.2}$$

Our first theorem in this section shows what happens to a space of splines when we transform it via a scale factor and a change of variables.

THEOREM 6.1. *If* $\tilde{\mathscr{U}}$ *and* $\tilde{\Delta}$ *are as in* (6.1)–(6.2), *then*

$$\mathscr{S}(\tilde{\mathscr{U}}; \mathscr{M}; \tilde{\Delta}) = \{\tilde{s}(x) = \omega(x)\, s(\rho(x)) \colon s \in \mathscr{S}(\mathscr{U}; \mathscr{M}; \Delta)\}.$$

Proof. Clearly, $\tilde{s}(x) = \omega(x)\, s(\rho(x))$ belongs to $\tilde{\mathscr{U}}$ piecewise, and the knots are located at $\tilde{x}_1, \ldots, \tilde{x}_k$. It remains to check the smoothness of $\tilde{s}$ at a knot. Suppose that ξ is a knot of s of multiplicity μ. Then

$$D^j s(\xi+) = D^j s(\xi-), \qquad j = 0, 1, \ldots, m - \mu - 1. \tag{6.3}$$

We must check the smoothness of $\tilde{s}$ at the knot $\tilde{\xi} = \rho^{-1}(\xi)$ which is clearly also of multiplicity μ. Now since

$$D^j \tilde{s}(x) = \sum_{\nu=0}^{j} \binom{j}{\nu} D^\nu s(\rho(x))\, D^{j-\nu}\omega(x),$$

it follows from the smoothness properties of ω, ρ, and s that $\tilde{s}$ satisfies (6.3) at $\tilde{\xi}$. ∎

In view of Theorem 6.1, it is clear that we can get a basis for the transformed space $\mathscr{S}(\tilde{\mathscr{U}}; \mathscr{M}; \tilde{\Delta})$ simply by taking

$$\tilde{B}_i(x) = \omega(x)\, B_i(\rho(x)), \qquad i = 1, 2, \ldots, n, \tag{6.4}$$

where $B_1, \ldots, B_n$ is any basis for $\mathscr{S}(\mathscr{U}; \mathscr{M}; \Delta)$. It is also clear that if the B's satisfy (2.5)–(2.6), then for all $i = 1, 2, \ldots, n$

$$\tilde{B}_i(x) > 0 \qquad \text{for} \quad \tilde{y}_i < x < \tilde{y}_{i+m}, \tag{6.5}$$

$$\tilde{B}_i(x) = 0 \qquad \text{for} \quad x < \tilde{y}_i \quad \text{and} \quad \tilde{y}_{i+m} < x, \tag{6.6}$$

where

$$\tilde{y}_i = \rho^{-1}(y_i), \qquad i = 1, 2, \ldots, n + m. \tag{6.7}$$

As we saw in Section 5, if the space $\tilde{\mathscr{S}} = \mathscr{S}(\tilde{\mathscr{U}}; \mathscr{M}; \Delta)$ is constructed by transforming a polynomial spline space, then the transformed B-splines (6.4) also satisfy recursion relations and can be computed by Algorithm 3.3. If we start with any other spline space, this is not generally true. On the other hand, since each $\tilde{s} \in \tilde{\mathscr{S}}$ can be written in the form

$$\tilde{s}(x) = \sum_{i=1}^{n} c_i\, \omega(x)\, B_i(\rho(x)), \tag{6.8}$$

we can work with $\tilde{s}$ by working with the B-splines $B_1, \ldots, B_n$ forming a basis for $\mathscr{S} = \mathscr{S}(\mathscr{U}; \mathscr{M}; \Delta)$.

The expansion (6.8) is valid for any splines obtained by transformation. It is most useful, however, in the cases where the original spline space $\mathscr{S}$ has a B-spline basis which is recursively computable; i.e., when $\mathscr{U}$ is a space of polynomial, trigonometric, hyperbolic, or (constant weight) Tchebycheffian splines.

We conclude this section with two examples of the kinds of spline spaces which can be handled via transformation.

EXAMPLE 6.2. Let $\tilde{\mathscr{U}} = \operatorname{span}\{1, \sinh(x^2), \cosh(x^2), \sinh(2x^2), \cosh(2x^2)\}$ on $[0, 1]$.

Discussion. The space of splines $\mathscr{S}(\tilde{\mathscr{U}}; \mathscr{M}; \Delta)$ is obtained by transforming the space of hyperbolic splines $\mathscr{S}(\mathscr{H}_5^{1/2}; \mathscr{M}; \Delta)$.

EXAMPLE 6.3. Let $\tilde{\mathscr{U}} = \operatorname{span}\{1, \sinh(x^2), \cosh(2x^2), \sinh(3x^2)\}$ on $[0, 1]$.

Discussion. In this case $\mathscr{S}(\tilde{\mathscr{U}}; \mathscr{M}; \Delta)$ is obtained by transforming the Tchebycheffian spline space $\mathscr{S}(\mathscr{U}; \mathscr{M}; \Delta)$ where $\mathscr{U} = \{u_i\}_1^4$ is the ECT-system in Example 5.5. ▮

7. REMARKS

(1) Throughout this paper we have concentrated on recursions whose form closely models that of the recursion for polynomial B-splines. Thus, while we have shown that only a very restricted collection of splines satisfy these kinds of recursions, there may well be other spaces of splines which satisfy completely different kinds of recursions.

(2) The two types of recursions discussed in Algorithms 3.2 and 3.3 are essentially different. They coincide only when $\phi_i = \psi_i = x$, in which case they both produce polynomial splines.

(3) The three basic splines spaces—polynomial, trigonometric, and hyperbolic—cannot be obtained from each other by (real) transformations.

ACKNOWLEDGMENTS

I would like to thank Professors C. K. Chui and J. D. Ward for several informative conversations, and in particular for providing Lemma 4.3.

REFERENCES

1. C. DE BOOR, On calculating with B-splines, *J. Approx. Theory* **6** (1972), 50–62.
2. M. G. COX, The numerical evaluation of B-splines, *J. Inst. Math. Appl.* **10** (1972), 134–149.
3. J. W. JEROME, On uniform approximation by certain generalized spline functions, *J. Approx. Theory* **7** (1973), 143–154.

4. J. W. JEROME AND L. L. SCHUMAKER, Local bases and computation of g-splines, *Methoden Verfahren Math. Phys.* **5** (1971), 171–199.
5. J. W. JEROME AND L. L. SCHUMAKER, Local support bases for a class of spline functions, *J. Approx. Theory* **16** (1976), 16–27.
6. S. KARLIN, "Total Positivity, Volume I," Stanford Univ. Press, Stanford, 1968.
7. T. LYCHE AND R. WINTHER, A stable recurrence relation for trigonometric B-splines, *J. Approx. Theory* **25** (1979), 266–279.
8. G. W. REDDIEN AND L. L. SCHUMAKER, On a collocation method for singular two point boundary value problems, *Numer. Math.* **25** (1976), 427–432.
9. K. SCHERER AND L. L. SCHUMAKER, A dual basis for Tchebycheffian B-splines and applications, *J. Approx. Theory* **29** (1980), 151–169.
10. L. L. SCHUMAKER, "Spline Functions: Basic Theory," Interscience, New York, 1981.
11. L. L. SCHUMAKER, On hyperbolic splines, *J. Approx. Theory*, to appear.