MATHEMATICS OF COMPUTATION, VOLUME 27, NUMBER 122, APRIL, 1973

On a Method of Carasso and Laurent for Constructing Interpolating Splines

By M. J. Munteanu and L. L. Schumaker*

Abstract. Carasso and Laurent studied a method for computing natural polynomial splines interpolating simple data. We discuss several similar methods which can be applied to numerical construction of more general interpolating splines, including *Lg*-splines interpolating Extended-Hermite-Birkhoff data.

1. Introduction. Let $x_0 < x_1 < \cdots < x_{N+1}$. A natural polynomial spline of degree $2m - 1$ with knots at the $\{x_i\}_0^{N+1}$ and interpolating data $\{y_i\}_0^{N+1}$ at the knots is a function s with the properties

$$(1.1) \qquad s \text{ is a polynomial of degree } 2m-1 \text{ in } (x_i, x_{i+1}), \qquad i = 0, 1, \cdots, N;$$

$$(1.2) \qquad \text{(interpolation)} \quad s(x_i) = y_i, \qquad i = 0, 1, \cdots, N+1;$$

$$(1.3) \qquad \text{(smoothness)} \quad s^{(j)}(x_i+) = s^{(j)}(x_i-), \qquad j = 0, 1, \cdots, 2m-2;\ i = 1, 2, \cdots, N;$$

and

$$(1.4) \qquad \text{(end conditions)} \quad \begin{aligned} s^{(m)}(x_0+) &= \cdots = s^{(2m-2)}(x_0+) = 0, \\ s^{(m)}(x_{N+1}-) &= \cdots = s^{(2m-2)}(x_{N+1}-) = 0. \end{aligned}$$

There are a wide variety of computational schemes for numerical construction of such splines (see e.g. [1], [3], [5], [6], [9], [10], [11], [12], [15] and the references therein). Carasso and Laurent [3] proposed a certain factorization method which proceeds roughly as follows. Let $\bar{s}(x) = [s(x), \cdots, s^{(2m-1)}(x)]^T$ and $R_i = \bar{s}(x_i+)$, $L_i = \bar{s}(x_i-)$. Clearly, (1.2) and (1.4) provide m relations on the components of R_0. But, in view of (1.1), it is easy to obtain (using the Taylor expansion) m relations on the components of L_1 from those on R_0. Furthermore, the smoothness property (1.3) at x_1 and the interpolation condition permit the derivation of m relations on R_1 from those on L_1. Continuing this process, m relations can be found on all of the vectors $\{L_i\}_1^{N+1}$ and $\{R_i\}_0^N$. Similarly, starting with L_{N+1} and proceeding backwards, a separate set of m relations on each of these vectors can be derived. Finally, for each i, the sets of relations can be combined to obtain $2m$ equations for the $2m$ components of the L_i and R_i, which, of course, determines the spline in each interval.

The purpose of this paper is to discuss such factorization schemes in a general framework and to derive several explicit methods which can be utilized for the numer-

Received April 2, 1972.

AMS (MOS) subject classifications (1970). Primary 65D05; Secondary 41A15.

* Supported in part by AFOSR-69-1812-C.

ical construction of considerably more general interpolating splines, including, for example, Lg-splines interpolating Extended-Hermite-Birkhoff data.

2. Preliminaries. The basic structural properties underlying the factorization method outlined in Section 1 for polynomial splines are shared by a wide class of generalizations (e.g. g-splines, L-splines, and Lg-splines; see Section 4). It is convenient to abstract these properties.

Let $x_0 < x_1 < \cdots < x_{N+1}$, and let $\{u_i\}_0^{2m-1}$ be $2m$ functions such that the Wronskian $W(u_0, \cdots, u_{2m-1})(x) > 0$ for $x \in [x_0, x_{N+1}]$. We consider construction of a spline function $s(x)$ with the following properties (as before, we use the notation $\bar{s}(x) = [s(x), \cdots, s^{(2m-1)}(x)]^T$ and $L_i = \bar{s}(x_i-)$, $R_i = \bar{s}(x_i+)$):

(2.1) (*piecewise property*). s is a linear combination of $\{u_i\}_0^{2m-1}$ in each interval (x_i, x_{i+1}), $i = 0, 1, \cdots, N$.

(2.2) (*interpolating conditions*). For each $i = 1, 2, \cdots, N$, there exist $1 \leqq z_i \leqq m$, a z_i-vector γ_i, and a $z_i \times 2m$ (rank z_i) matrix A_i such that $A_i\bar{s}(x_i) = \gamma_i$.

(2.3) (*smoothness conditions*). For each $i = 1, 2, \cdots, N$, there exists a $(2m - z_i) \times 2m$ matrix S_i of rank $2m - z_i$ such that $S_iL_i = S_iR_i$.

(2.4) (*end conditions*). There exist $m \times 2m$ rank m matrices RF_0 and LB_{N+1} and m-vectors r_0 and l'_{N+1} such that $RF_0R_0 = r_0$, $LB_{N+1}L_{N+1} = l'_{N+1}$.

We need the following elementary fact.

LEMMA 2.1. *Given a real number y, suppose $s(x)$ is a linear combination of $\{u_j(x)\}_0^{2m-1}$ in some interval $[\alpha, \beta]$ containing y. Then, there exists a set of functions $\{u_j(y; x)\}_0^{2m-1}$ in $U = \operatorname{span}\{u_j(x)\}_0^{2m-1}$ such that, for every $x \in [\alpha, \beta]$,*

$$s(x) = \sum_{j=0}^{2m-1} s^{(j)}(y)u_j(y; x). \tag{2.5}$$

Proof. By the Wronskian assumption on the functions of U, the linear functionals $\{\lambda_j\}_0^{2m-1}$ given by $\lambda_j f = f^{(j)}(y)$ are linearly independent in the dual U^*. Theorem 2.5.1 of Davis [4] applies.

The functions $u_j(y; x)$ in the above lemma depend only on U and y, and not on α, β. The following immediate consequence of the lemma shows that, in each interval (x_i, x_{i+1}), a spline s is determined by either R_i or L_{i+1}.

COROLLARY 2.2. *For $k = 0, 1, \cdots, N+1$, let $\{u_j(x_k; x)\}_0^{2m-1}$ be the set of functions corresponding to $y = x_k$ in Lemma 2.1. Suppose s satisfies (2.1). Then*

$$\bar{s}(x) = W(u_0(x_i; \cdot), \cdots, u_{2m-1}(x_i; \cdot))(x)R_i, \tag{2.6}$$

$$\bar{s}(x) = W(u_0(x_{i+1}; \cdot), \cdots, u_{2m-1}(x_{i+1}; \cdot))(x)L_{i+1}, \tag{2.7}$$

for $x \in (x_i, x_{i+1})$, $i = 0, 1, \cdots, N$.

COROLLARY 2.3. *Let s satisfy (2.1). Then, there exist $2m \times 2m$ nonsingular matrices TF_i and TB_i such that*

$$L_{i+1} = TF_iR_i, \tag{2.8}$$

$$R_i = TB_iL_{i+1}. \tag{2.9}$$

Proof. For example, from Corollary 2.2, we see that

$$TF_i = W(u_0(x_i;\cdot), \cdots, u_{2m-1}(x_i;\cdot))(x_{i+1}).$$

As we remarked above, to construct a spline s satisfying (2.1)–(2.4) it suffices to find either L_i or R_i for each $i = 0, 1, \cdots, N+1$. To this end, we now seek to find $2m$ conditions on the components of each of the L_i and R_i.

First, starting with the end condition $RF_0R_0 = r_0$, we show how to find m conditions on each of the L_i and R_i. If we substitute $R_0 = TB_0L_1$ in $RF_0R_0 = r_0$, we get $LF_1L_1 = l_1$ with $LF_1 = RF_0TB_0$ and $l_1 = r_0$ (since RF_0 is $m \times 2m$ so is LF_1, and these are the desired m relations on L_1). To find conditions on R_1, consider

$$\left[\begin{array}{c|c} LF_i & 0 \\ \hline S_i & -S_i \end{array}\right]\begin{bmatrix} L_i \\ R_i \end{bmatrix} = \begin{bmatrix} l_i \\ 0 \end{bmatrix} \tag{2.10}$$

with $i = 1$. The matrix

$$\begin{bmatrix} LF_i \\ S_i \end{bmatrix}$$

is $3m - z_i$ by $2m$. Suppose the rows of LF_i are rearranged so that the last e_i rows coupled with the $2m - z_i$ rows of S_i are linearly independent. Since S_i is of rank $2m - z_i$, $0 \leqq e_i \leqq z_i$. Now, using the $2m - z_i$ rows of the matrix S_i and the last e_i rows from LF_i, we can eliminate the first $2m$ variables from the first $m - z_i$ of the equations. This leaves $m - z_i$ equations on the components of R_i. Adding the z_i equations $A_iR_i = \gamma_i$, we have m equations on R_i which can be written $RF_iR_i = r_i$. These steps can be repeated to go from m conditions on R_1 to m conditions on L_2, then to R_2, and so on. We conclude there exist $m \times 2m$ matrices LF_i, RF_i and m-vectors l_i, r_i such that

$$LF_iL_i = l_i, \qquad i = 1, 2, \cdots, N+1, \tag{2.11}$$

$$RF_iR_i = r_i, \qquad i = 0, 1, \cdots, N. \tag{2.12}$$

In general, $LF_i = RF_{i-1}TB_{i-1}$ and $l_i = r_{i-1}$, $i = 1, 2, \cdots, N+1$.

Conditions (2.11) and (2.12) were constructed by starting with the left end condition and proceeding forward (the F in RF is to remind us of that). A similar procedure starting with $LB_{N+1}L_{N+1} = l'_{N+1}$ and proceeding backwards produces $m \times 2m$ matrices LB_i, RB_i and m-vectors l'_i, r'_i such that

$$LB_iL_i = l'_i, \qquad i = 1, 2, \cdots, N+1, \tag{2.13}$$

$$RB_iR_i = r'_i, \qquad i = 0, 1, \cdots, N. \tag{2.14}$$

Here, $RB_i = LB_{i+1}TF_i$ and $r'_i = l'_{i+1}$. The matrix LB_i is found from RB_i by eliminating the last $2m$ variables from the first $m - z_i$ rows of

$$\left[\begin{array}{c|c} 0 & RB_i \\ \hline S_i & -S_i \end{array}\right]\begin{bmatrix} L_i \\ R_i \end{bmatrix} = \begin{pmatrix} r'_i \\ 0 \end{pmatrix} \tag{2.15}$$

and adding the relations $A_iL_i = \gamma_i$. We assume for the remainder of the paper that if $S_iL_i = S_iR_i$, then $A_iR_i = \gamma_i$, if and only if $A_iL_i = \gamma_i$. The following lemma will be needed later.

LEMMA 2.4. *Suppose R_i^* is a vector satisfying* (2.12) *and* (2.14). *Then there exists* *a vector* L_i^* *satisfying* (2.11) *and* (2.13). *Conversely, given* L_i^* *satisfying* (2.11), (2.13) *there exists* R_i^* *satisfying* (2.12), (2.14).

Proof. Let R_i^* satisfy (2.12) and (2.14) .We show how to construct L_i^*. Let $\widetilde{LF}_i$ be a $z_i \times 2m$ matrix formed by augmenting the last e_i rows of LF_i (cf. the discussion following (2.10)) in such a way that the matrix in (2.16) below is nonsingular. Let L_i^* be the solution of

$$\begin{bmatrix} \widetilde{LF}_i \\ \hline S_i \end{bmatrix} L_i^* = \begin{bmatrix} 0 \\ \tilde{l}_i \\ S_i R_i^* \end{bmatrix} \tag{2.16}$$

where $\tilde{l}_i$ consists of the last e_i rows of l_i.

By the construction of (2.12), $RF_i R_i^* = r_i$ implies $A_i R_i^* = \gamma_i$. Since $S_i R_i^* = S_i L_i^*$, it follows that $A_i L_i^* = \gamma_i$. Now we claim $LF_i L_i^* = l_i$ and $LB_i L_i^* = l_i'$. Indeed, suppose $LF_i L_i^* \neq l_i$. Then

$$\left[\begin{array}{c|c} LF_i & 0 \\ \hline S_i & -S_i \end{array}\right] \begin{bmatrix} L_i^* \\ R_i^* \end{bmatrix} = \begin{bmatrix} l_i + \delta_i \\ 0 \end{bmatrix}$$

for some δ_i. By the construction of L_i^*, the last e_i δ_i's are zero. Suppose at least one of the first $m - e_i$ is not. Now, performing the same elimination used on (2.10) leads to a contradiction of $RF_i R_i^* = r_i$. To check that $LB_i L_i^* = l_i'$, note that (2.15) holds for L_i^* and R_i^*. Eliminating the last $2m$ variables in the first $m - z_i$ equations and using $A_i L_i^* = \gamma_i$ yields $LB_i L_i^* = l_i'$.

The converse assertion is proved analogously.

3. Constructive Methods. In this section, we discuss three methods for constructing a spline s satisfying (2.1)–(2.4). The methods are based on computing L_i and/or R_i for each $i = 0, 1, \cdots, N+1$ from the information on s inherent in the relations (2.11)–(2.14).

Method 1. For $i = 1, 2, \cdots, N+1$, solve

$$\begin{bmatrix} LF_i \\ LB_i \end{bmatrix} L_i = \begin{bmatrix} l_i \\ l_i' \end{bmatrix}. \tag{3.1}$$

Method 2. For $i = 1, 2, \cdots, N$, solve

$$\left[\begin{array}{c|c} LF_i & 0 \\ \hline 0 & RB_i \\ \hline A_i \mid 0 & 0 \\ \hline S_i & -S_i \end{array}\right] \begin{bmatrix} L_i \\ R_i \end{bmatrix} = \begin{bmatrix} l_i \\ r_i \\ \gamma_i \\ 0 \end{bmatrix}. \tag{3.2}$$

Method 3. Solve

$$\begin{bmatrix} LF_{N+1} \\ LB_{N+1} \end{bmatrix} L_{N+1} = \begin{bmatrix} l_{N+1} \\ l_{N+1}' \end{bmatrix}. \tag{3.3}$$

For $i = N, N - 1, \cdots, 1$, find

$$R_i = TB_i L_{i+1} \tag{3.4}$$

and solve

$$\begin{bmatrix} \widetilde{LF}_i \\ S_i \end{bmatrix} L_i = \begin{bmatrix} 0 \\ \tilde{l}_i \\ S_i R_i \end{bmatrix}, \tag{3.5}$$

where $\widetilde{LF}_i$ and $\tilde{l}_i$ are as in the proof of Lemma 2.4 (see (2.16)).

THEOREM 3.1. *Suppose there is a unique spline function s satisfying* (2.1)–(2.4). *Then Methods* 1–3 *are well defined; i.e., the systems in* (3.1)–(3.3) *and* (3.5) *are nonsingular.*

Proof. We consider first Method 1. Suppose (3.1) is singular for some k, $1 \leqq k \leqq N + 1$. Then, there exists $L_k^* \neq 0$ with

$$\begin{bmatrix} LF_k \\ LB_k \end{bmatrix} L_k^* = 0.$$

For $x \in (x_{k-1}, x_k)$ define

$$s^*(x) = \sum_{j=0}^{2m-1} L_k^*(j) u_j(x_k; x).$$

Clearly, $R_{k-1}^* = \bar{s}^*(x_{k-1}+)$ satisfies $RF_{k-1}R_{k-1}^* = RB_{k-1}R_{k-1}^* = 0$. Now, by Lemma 2.4, we can find L_{k-1}^* satisfying

$$\begin{bmatrix} LF_{k-1} \\ LB_{k-1} \end{bmatrix} L_{k-1}^* = 0.$$

This process can be continued to define $s^*(x)$ in $[x_0, x_k]$. A similar process starting with L_k^* and proceeding forward can be used to define $s^*(x)$ throughout $[x_0, x_{N+1}]$ with the property that (2.11)–(2.14) all hold with zero right-hand sides. Since

$$\begin{bmatrix} LF_i \\ LB_i \end{bmatrix} L_i^* = 0$$

implies $A_i = 0$, we conclude that s^* is a nontrivial spline satisfying (2.1)–(2.4) with homogeneous data. This contradicts the uniqueness assumption.

For Method 2, suppose (3.2) is singular. Then, for some k, $1 \leqq k \leqq N$, there exist nontrivial L_k^* and R_k^* with

$$\left[\begin{array}{c|c} LF_k & 0 \\ \hline 0 & RB_k \\ \hline A_k \mid 0 & 0 \\ \hline S_k & -S_k \end{array}\right] \begin{bmatrix} L_k^* \\ R_k^* \end{bmatrix} = 0. \tag{3.6}$$

Moreover, performing elimination on (3.6), we conclude that $RF_kR_k^* = LB_kL_k^* = 0$. Now, $s^*(x)$ can be defined in $[x_{k-1}, x_k]$ and $[x_k, x_{k+1}]$ in terms of L_k^* and R_k^*, respectively. Lemma 2.4 can be used to extend s^* to $[x_0, x_{N+1}]$ such that (2.11)–(2.14) hold with 0 right-hand sides. This is again a contradiction of the uniqueness of spline interpolation, as $s^* \not\equiv 0$.

Finally, for Method 3, we note that the system (3.3) cannot be singular (by the same proof as for Method 1). The matrix (3.5) is nonsingular by construction (cf. the proof of Lemma 2.4).

The matrix in (3.5) was obtained from LF_i and S_i by an augmentation process. In view of the uniqueness of the interpolating spline, we see that the solution of (3.5) must be unique, and so any augmentation must give the same solution. But this means that there is no freedom to choose different augmentations and we conclude that $e_i = z_i$ in the proofs of Corollary 2.3 and Lemma 2.4. Thus no augmentation is ever necessary in practice.

4. Applications. We recall the definition of Lg-splines [8]. Let L be an mth order differential operator, $\Lambda = \{\lambda_i\}_1^n$ a set of linearly independent linear functionals on H_2^m, and $\{d_i\}_1^n$ a set of real numbers. An Lg-spline interpolating $\{d_i\}_1^n$ with respect to Λ is a solution of

$$\|Ls\|_2 = \min_{f \in U(d)} \|Lf\|_2, \tag{4.1}$$

$$U(d) = \{f \in H_2^m \colon \lambda_j f = d_j,\ j = 1, 2, \cdots, n\}.$$

The methods of Section 3 can be used to construct such splines when Λ corresponds to EHB data.

To describe EHB interpolation, let $x_0 < x_1 < \cdots < x_{N+1}$ and let $1 \leqq z_i \leqq m$ with $\sum_{i=0}^{N+1} z_i = n$. For $0 \leqq i \leqq N+1$, $1 \leqq j \leqq z_i$, let $\alpha(i, j) = (\alpha_0(i, j), \cdots, \alpha_{m-1}(i, j))$ be vectors such that for each i the vectors in the set $\{\alpha(i, j)\}_{j=1}^{z_i}$ are linearly independent. Λ generates an EHB interpolation problem if the n linear functionals in Λ are

$$\sum_{k=0}^{m-1} \alpha_k(i, j)\, D_{x_i}^k, \qquad j = 1, 2, \cdots, z_i;\ i = 0, 1, \cdots, N+1, \tag{4.2}$$

where $D_{x_i}^k f = f^{(k)}(x_i)$.

We now show that an Lg-spline s interpolating EHB data satisfies the conditions (2.1)–(2.4). At each x_i, there are z_i interpolating conditions which can be written in the form of (2.2) with

$$A_i = \left[\begin{array}{c|c} \alpha(i, 1) & \\ \vdots & 0 \\ \alpha(i, z_i) & \end{array}\right], \qquad i = 1, 2, \cdots, N.$$

Furthermore (see [8, Theorem 3.6]), s belongs to C^{m-1} and at each x_i, $1 \leqq i \leqq N$, there are additional $m - z_i$ smoothness conditions. These are given by $R_j^{(x_i)}s(x_i+) = R_j^{(x_i)}s(x_i-)$, $j = z_i + 1, \cdots, m$, where $R_j^{(x_i)}s(x) = \sum_{k=m}^{2m-1} \beta_k(i, j)s^{(k)}(x)$, with certain (real) coefficients $\beta_k(i, j)$. Thus, the smoothing matrices of (2.3) are

$$S_i = \left[\begin{array}{c|c} I_m & 0 \\ \hline & \beta(i, z_i + 1) \\ 0 & \vdots \\ & \beta(i, m) \end{array}\right]. \tag{4.3}$$

Finally, the end conditions on s are given by (2.4) with

$$RF_0 = \left[\begin{array}{c|c} \alpha(0, 1) & \\ \vdots & 0 \\ \alpha(0, z_0) & \\ \hline & \beta(0, z_0 + 1) \\ 0 & \vdots \\ & \beta(0, m) \end{array}\right], \qquad r_0 = \begin{bmatrix} d_1 \\ \vdots \\ d_{z_0} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

and

$$LF_{N+1} = \left[\begin{array}{c|c} \alpha(N + 1, 1) & \\ \vdots & 0 \\ \alpha(N + 1, z_{N+1}) & \\ \hline & \beta(N + 1, z_{N+1} + 1) \\ 0 & \vdots \\ & \beta(N + 1, m) \end{array}\right], \qquad l'_{N+1} = \left[\begin{array}{c} d_{n-z_{N+1}+1} \\ \vdots \\ d_n \\ \hline 0 \end{array}\right].$$

For the smoothness matrices of (4.3), the condition $S_iL_i = S_iR_i$ clearly implies the first m components of L_i and R_i are the same. Thus, the hypothesis needed for Lemma 2.4 that $A_iL_i = \gamma_i$ iff $A_iR_i = \gamma_i$ holds for these splines.

The matrices A_i, S_i, RF_0, and LF_{N+1} are considerably simplified for less general forms of data. For example, for HB data (see [8]) each of the vectors $\alpha(i, j)$ and $\beta(i, k)$ has only one nonzero coefficient. For Hermite data,

$$A_i = [I_{z_i} \mid 0], \qquad S_i = \left[\begin{array}{c|c} I_m & 0 \\ \hline 0 & I_{m-z_i} \end{array}\right]$$

and

$$RF_0 = \left[\begin{array}{c|c} I_{z_0} & 0 \\ \hline 0 & I_{m-z_0} \end{array}\right], \qquad LF_{N+1} = \left[\begin{array}{c|c} I_{z_{N+1}} & 0 \\ \hline 0 & I_{m-z_{N+1}} \end{array}\right].$$

When $L = D^m$, the Lg-splines are just polynomial splines. In this case, $u_j(x) = x^j$, $j = 0, 1, \cdots, 2m - 1$, and the functions of Corollary 2.2 are $u_j(x_i; x) = (x - x_i)^j$, $j = 0, 1, \cdots, 2m - 1$.

5. Remarks. 1. If Method 1 is applied to the computation of a natural polynomial spline interpolating simple data ($z_0 = z_1 = \cdots = z_{N+1} = 1$), we have the method introduced by Carasso and Laurent [3]. Method 2 was introduced for type I L-splines by Munteanu [12].

2. There are a variety of methods using bases for computing polynomial interpolating splines with Hermite data (cf. the discussions in [1], [3], [5], [6], [8], [9], [11], [15]). These fall into two classes: methods where bases for the space of splines $\mathcal{S}$ are used and those where bases for the space $L\mathcal{S}$ are used (called projection methods). For polynomial splines with Hermite data, convenient local-support bases have been found ([5], [7], [11]). Similar bases have been found for g-splines with Hermite-Birkhoff data [9], but to date not for EHB data or for more general operators L. The projective methods [1], [3], [6] can be used for L-splines with Hermite data [12]. (Projective methods have the drawback that bases (even local support bases) for $L\mathcal{S}$ lead to nonlocal bases for $\mathcal{S}$, with attendant conditioning problems.)

3. Methods for computing Lg-splines have a variety of practical applications; e.g. in constructing optimal quadrature rules (see e.g. Karlin [10, p. 83] and references therein) and solving two-point-boundary-value problems [2], [6]. An optimal method for boundary value problems of Golomb [6] requires the construction of Lg-spline corresponding to EHB data.

4. There are some obvious dual versions of Methods 1 and 3. Method 1′ would involve solving

$$\begin{bmatrix} RF_i \\ RB_i \end{bmatrix} R_i = \begin{bmatrix} r_i \\ r_i' \end{bmatrix}, \qquad i = 0, 1, \cdots, N.$$

A Method 3′ can be constructed by starting with x_0; we compute R_0 and proceed forward. It should be cautioned that the natural dual to Method 2 (that is, the use of RF_i and LB_i to determine L_i and R_i) would not be well defined. The reason is that both sets of m relations are obtained by adding the interpolating conditions to other sets of $m - z_i$ relations.

5. The methods of Section 3 apply equally well to type III L-splines [14]. The periodic case (type IV) is not covered. It may be possible to develop factorization methods for the case of splines satisfying inequalities (see [8]) rather than interpolating data exactly, and for smoothing splines (see [11]).

The methods of Section 3 apply equally well to type III L-splines [14]. The periodic case (type IV) is not covered. It may be possible to develop factorization methods for the case of splines satisfying inequalities (see [8]) rather than interpolating data exactly, and for smoothing splines (see [11]).

6. The methods of Section 3 can be compared with each other and with other methods in the literature for efficiency. Without examining operation counts, it is clear that Method 3 requires only half the effort to set up that the other two do as the $\{LF_i\}_1^N$ and $\{RB_i\}_1^N$ are not needed. The actual operation counts for finding $\{LF_i\}_1^{N+1}$ depend on the complexity of the interpolation and smoothing conditions. For polynomial splines, the total operation count for any of the algorithms is of the order con Nm^3, where con does not depend on N or m (but does depend on the algorithm). The storage required for the factorization method is of order Nm^2. Local support basis methods use on the order of Nm storage if the band structure is accounted for, and N^2 otherwise.

7. Method 3 has been tested for a wide variety of simple, Hermite, and Hermite-Birkhoff interpolation problems for polynomial splines. For complete FORTRAN subroutines and a discussion of the numerical experience with them, see [Eidson, H. L. and Schumaker, L. L., *Computation of g-Splines via a Factorization Method*, CNA Report 60, Center for Numerical Analysis, The University of Texas, Austin, 1972].

Department of Mathematics
University of Maryland
Baltimore, Maryland 21228

Department of Mathematics and Center for Numerical Analysis
The University of Texas
Austin, Texas 78712

1. P. M. ANSELONE & P. J. LAURENT, "A general method for the construction of interpolating or smoothing spline-functions," *Numer. Math.*, v. 12, 1968, pp. 66–82. MR **40** #3145.

2. G. D. BYRNE & D. N. H. CHI, "Linear multistep formulas based on *g*-splines," *SIAM J. Numer. Anal.*, v. 9, 1972, pp. 316–324.

3. C. CARASSO & P. J. LAURENT, *On the Numerical Construction and the Practical Use of Interpolating Spline-Functions,* Proc. IFIP Congress Information Processing 68 (Edinburgh, 1968), vol. 1, Mathematics, Software, North-Holland, Amsterdam, 1969, pp. 86–89. MR **40** #8219.

4. P. J. DAVIS, *Interpolation and Approximation,* Blaisdell, Waltham, Mass., 1963. MR **28** #393.

5 C. DE BOOR, "On calculating with *B*-splines," *J. Approximation Theory*, v. 6, 1972, pp. 50–62.

6. M. GOLOMB, *Spline Approximations to the Solutions of Two-Point Boundary-Value Problems,* MRC 1066, Univ. of Wisconsin, Madison, Wis., 1970.

7. T. N. E. GREVILLE, *Introduction to Spline Functions,* Proc. Sem. Theory of Applications of Spline Functions (Math. Res. Center, Univ. of Wisconsin, Madison, Wis., 1968), Academic Press, New York, 1969, pp. 1–35. MR **39** #1868.

8. J. W. JEROME & L. L. SCHUMAKER, "On *Lg*-splines," *J. Approximation Theory*, v. 2, 1969, pp. 29–49. MR **39** #3201.

9. J. W. JEROME & L. L. SCHUMAKER, "Local bases and computation of *g*-splines," *Methoden und Verfahren der Mathematische Physik*, v. 5, 1971, pp. 171–199.

10. S. KARLIN, "Best quadrature formulas and splines," *J. Approximation Theory,* v. 4, 1971, pp. 59–90. MR **43** #2403.

11. T. LYCHE & L. L. SCHUMAKER, "Computation of smoothing and interpolating natural splines via local bases," *SIAM J. Numer. Anal.* (To appear.)

12. M. J. MUNTEANU, *Contributions à la Théorie des Fonctions Splines à une et à Plusieurs Variables,* Dissertation, Louvain, 1970.

13. I. J. SCHOENBERG, "On the Ahlberg-Nilson extension of spline interpolation: the *g*-splines and their optimal properties," *J. Math. Anal. Appl.*, v. 21, 1968, pp. 207–231. MR **36** #6849.

14. M. H. SCHULTZ & R. S. VARGA, "*L*-splines," *Numer. Math.*, v. 10, 1967, pp. 345–369. MR **37** #665.

15. L. L. SCHUMAKER, *Some Algorithms for the Computation of Interpolating and Approximating Spline Functions,* Proc. Sem. Theory and Applications of Spline Functions (Math. Res. Center, Univ. of Wisconsin, Madison, Wis., 1968), Academic Press, New York, 1969, pp. 87–102. MR **39** #687.