SIAM J. NUMER. ANAL.
Vol. 23, No. 3, June 1986

012

MAKING THE OSLO ALGORITHM MORE EFFICIENT*

T. LYCHE† AND K. MØRKEN†

Abstract. The Oslo Algorithm is a general method for adding knots to a B-spline curve or tensor product B-spline surface. The method provides a framework in computer aided geometric design for both manipulating and rendering spine curves and surfaces, and is derived from properties of discrete B-splines. In this paper we prove that all discrete B-splines which are nonzero at a particular point can in general be considered as lower order discrete B-splines on a subset of the knots. We also give necessary and sufficient conditions for a discrete B-spline to be a continuous function of its parameters. These results are used to improve on the original Oslo algorithms.

Key words. divided differences, B-splines, subdivision

AMS(MOS) subject classifications. 41A15, 65D07

1. Introduction. The Oslo Algorithm is a general method for adding knots to a B-spline curve or tensor product surface. The method provides a framework in computer aided geometric design for both manipulating and rendering spline curves and surfaces.

The Oslo Algorithm was derived in [3] using the theory of discrete B-splines. In general, discrete B-splines occur as coefficients when expressing a kth order divided difference at some points as a sum of divided differences over a refinement of the original points. More specifically, suppose $t=(t_j)_{-\infty}^{\infty}$ is a nondecreasing, bi-infinite sequence of real numbers and k a positive integer. Let τ be a subsequence of t containing at least $k+1$ elements. Then for any suitable integer i and a sufficiently smooth function f we have

$$(1.1)\qquad (\tau_{i+k}-\tau_i)[\tau_i,\cdots,\tau_{i+k}]f=\sum_j \alpha_{i,k}(j)(t_{j+k}-t_j)[t_j,\cdots,t_{j+k}]f.$$

The existence of a formula like (1.1) with nonnegative weights $\alpha_{i,k}(j)=\alpha_{i,k,\tau,t}(j)$, goes back to [7, p. 7]. Equation (1.1) is a discrete analogue of the Peano integral representation for divided differences

$$(1.2)\qquad (\tau_{i+k}-\tau_i)[\tau_i,\cdots,\tau_{i+k}]f=\int f^{(k)}(x)B_{i,k}(x)\,dx/(k-1)!,$$

where $B_{i,k}$ is a B-spline of order k with knots $\tau_i,\cdots,\tau_{i+k}$. For this reason the name discrete B-spline was used for $\alpha_{i,k}$ in [1]. With an appropriate choice of f in (1.1) we obtain for all possible values of i,

$$(1.3)\qquad B_{i,k}=\sum_j \alpha_{i,k,\tau,t}(j)N_{j,k},$$

where $N_{j,k}$ is also a B-spline of order k but with knots $t_j,\cdots,t_{j+k}$. In knot refinement applications $\mathbf{t}$ is a new knot sequence obtained by adding knots to $\boldsymbol{\tau}$. Equation (1.3) gives the transformation from the basis $\{B_{i,k}\}$ for splines on $\boldsymbol{\tau}$ to the basis $\{N_{j,k}\}$ for splines on $\mathbf{t}$. The existence of a formula like (1.3) follows a priori from the fact that

$$\mathbf{S}_{k,\tau}=\operatorname{span}\{B_{i,k}\}\subseteq\mathbf{S}_{k,t}=\operatorname{span}\{N_{j,k}\}.$$

Discrete B-splines on a uniform $\mathbf{t}$-sequence were introduced in [9] and studied further in [6]. The generalization to an arbitrary $\mathbf{t}$-sequence was given in [1, p. 15].

* Received by the editors November 12, 1984, and in revised form May 6, 1985.

† Institute of Informatics, University of Oslo, Oslo 3, Norway.

The numerical value of $\alpha_{i,k}(j)$ can be computed by means of a recurrence relation similar to the recurrence relation for B-splines ([3, p. 97], see also [2] for the case of adding one knot to $\boldsymbol{\tau}$). Several other properties of B-splines carry over to the discrete case. In particular, discrete B-splines are nonnegative and $\sum_i \alpha_{i,k}(j)=1$. Total positivity properties were given in [5], and in [10, p. 355] for the case of a uniform $\mathbf{t}$-sequence. On the other hand, except for in the uniform case [6], [9], [10], the piecewise polynomial nature of $\alpha_{i,k}(j)$ is not clear.

In this paper we investigate more closely the dependence of $\alpha_{i,k}(j)$ on $\tau_i, \cdots, \tau_{i+k}$ and $t_j, \cdots, t_{j+k}$. We give necessary and sufficient conditions for $\alpha_{i,k}(j)$ to depend continuously on these parameters, and show that for a given j the discrete B-spline $\alpha_{i,k}(j)$ can in general be considered as a lower order discrete B-spline on a subset of the knots. This provides a basis for reducing the number of operations in the Oslo Algorithm.

There are in fact two algorithms which qualify for the name *the* "Oslo Algorithm". These are called Algorithm 1 and Algorithm 2 in [3]. Since both algorithms are useful, we will refer to them as "Oslo Algorithm 1" and "Oslo Algorithm 2".

We develop detailed improved versions of Oslo Algorithms 1 and 2. Algorithm 1 now uses only linear combinations of positive quantities with positive weights and is unconditionally stable. In the new Algorithm 2 only strict convex combinations[1] of B-spline coefficients are used. We also remove some problems which can occur near the ends of finite knot vectors.

The content of this paper is as follows. Section 2 contains the mathematical results mentioned above, while the improved Oslo Algorithms are given in § 3. In § 4 we collect some remarks.

2. Discrete *B*-splines. Let the positive integer k be given, and let $\mathbf{t}=(t_j)_{-\infty}^{\infty}$ be a nondecreasing bi-infinite sequence of real numbers with $t_j<t_{j+k}$ for all j. Let $\boldsymbol{\tau}$ be a subsequence of $\mathbf{t}$ containing at least $k+1$ elements. Let

$$\psi_{j,k}(y)=\begin{cases}(y-t_{j+1})\cdots(y-t_{j+k-1}) & \text{if } k>1;\\ 1 & \text{if } k=1;\end{cases}$$

and let for any $a_j\in[t_j, t_{j+k})$,

$$\Phi_{j,k}(y)=\begin{cases}\psi_{j,k}(y) & \text{if } y>a_j,\\ 0 & \text{otherwise.}\end{cases} \tag{2.1}$$

Let $B_{i,k}$ and $N_{j,k}$ denote B-splines on $\boldsymbol{\tau}$ and $\mathbf{t}$ respectively, right continuous and normalized to sum to 1. If $t_j<t_{j+k}$ for all j, then it was shown in [1] and [3] that (1.3) holds with

$$\alpha_{i,k,\tau,t}(j)=(\tau_{i+k}-\tau_i)[\tau_i, \cdots, \tau_{i+k}]\Phi_{j,k}. \tag{2.2}$$

For convenience we also define $\alpha_{i,k,\tau,t}(j)$ for $\tau_i=\tau_{i+k}$ and $t_j=t_{j+k}$,

$$\alpha_{i,k,\tau,t}(j)=\begin{cases}(\tau_{i+k}-\tau_i)[\tau_i, \cdots, \tau_{i+k}](y-t_j)_+^{k-1} & \text{if } t_j=t_{j+k} \text{ and } \tau_i<\tau_{i+k},\\ 0 & \text{if } \tau_i=\tau_{i+k}.\end{cases} \tag{2.3}$$

[1] A number x is a strict convex combination of two numbers a and b if $x=\lambda a+(1-\lambda)b$ for some weight λ with $0<\lambda<1$. It is known that negative weights can occur in Oslo algorithm 2. However, it can be shown that these negative weights do not contribute to the final result (see remark in [4]). Nevertheless, the possibility of negative weights is somewhat annoying.

Here

$$(y-t_j)_+^{k-1}=\begin{cases}(y-t_j)^{k-1} & \text{if } y>t_j,\\ 0 & \text{otherwise.}\end{cases}$$

We call $\alpha_{i,k,\tau,t}$ the ith discrete B-spline of order k on t with knots τ.

In the rest of this paper it is convenient to think of j as a fixed integer. Consider $t_{j+1}, \cdots, t_{j+k-1}$ on which $\psi_{j,k}$ in (2.1) depends. These knots can be divided into two groups, the "old" knots and the "new" knots. More precisely, suppose that

$$t_{j+1}, \cdots, t_{j+k-1}=\overbrace{z_1, \cdots, z_1}^{r_1 \text{ times}}, \cdots, \overbrace{z_h, \cdots, z_h}^{r_h \text{ times}} \tag{2.4}$$

where $z_1<\cdots<z_h$. If z_i occurs s_i times in the whole τ-sequence, then the integer ν and the new knots $\xi_1, \cdots, \xi_\nu$ are defined by

$$\xi_1, \cdots, \xi_\nu=\overbrace{z_1, \cdots, z_1}^{\nu_1 \text{ times}}, \cdots, \overbrace{z_h, \cdots, z_h}^{\nu_h \text{ times}} \tag{2.5}$$

where $\nu_i=\max(r_i-s_i, 0)$ for $i=1, \cdots, h$. The remaining knots among $t_{j+1}, \cdots, t_{j+k-1}$ are the old knots $\omega_1, \cdots, \omega_{k-1-\nu}$. We assume that the ξ's and ω's have been arranged in nondecreasing order.

Example 2.1. Suppose that $k=4$, and let the knot vectors τ and t be given by $(\tau_1, \cdots, \tau_8)=(0,0,0,0,1,2,2,2)$, and $(t_1, \cdots, t_{10})=(0,0,0,0,1,1,1,2,2,2)$. Find the new and old knots among $t_{j+1}, t_{j+2}, t_{j+3}$ for $j=1,2,\cdots,6$.

DISCUSSION. There are no new knots among $t_{j+1}, t_{j+2}, t_{j+3}$ for $j=1,2,6$. Thus, for these values of j, we have $\nu=0$ and $\omega_p=t_{j+p}$ for $p=1,2,3$. For $j=3$ we have $\nu=1$, $\xi_1=1$, $\omega_1=0$, and $\omega_2=1$. If $j=4$ then $\nu=2$, $\xi_1=\xi_2=1$, and $\omega_1=1$. Finally, for $j=5$ we have $\nu=1$, $\xi_1=\omega_1=1$, and $\omega_2=2$. □

For fixed j, let $\boldsymbol{\tau}'$ and $\mathbf{t}'$ denote the sequences obtained by removing the old knots $\omega_1, \cdots, \omega_{k-1-\nu}$ from $\boldsymbol{\tau}$ and $\mathbf{t}$ respectively. (In the example above we have $\omega_1=\omega_2=\omega_3=0$ for $j=1$, so that $\boldsymbol{\tau}'=(0,1,2,2,2)$ and $\mathbf{t}'=(0,1,1,1,2,2,2)$ for this j.)

We can now state the main result of this paper.

THEOREM 2.2. *Let j be a fixed integer and suppose that there are ν new knots $\xi_1, \cdots, \xi_\nu$ among $t_{j+1}, \cdots, t_{j+k-1}$. Then for all i*

$$\alpha_{i,k,\tau,t}(j)=\alpha_{i,\nu+1,\tau',t'}(j) \tag{2.6}$$

where τ' and t' are defined above. Moreover

$$\alpha_{i,k,\tau,t}(j)>0 \quad \text{for } i=\mu'-\nu, \mu'-\nu+1, \cdots, \mu',$$

and zero otherwise. Here μ' is the unique integer such that

$$\tau_{\mu'}=\tau'_{\mu'}\leqq t_j=t'_j<\tau'_{\mu'+1}. \tag{2.7}$$

For the proof of Theorem 2.2 it is convenient to define polynomials ψ_ω and ψ_ξ by

$$\psi_\omega(y)=(y-\omega_1)\cdots(y-\omega_{k-1-\nu}),$$

$$\psi_\xi(y)=(y-\xi_1)\cdots(y-\xi_\nu).$$

We also define for any $a_j\in[t_j, t_{j+k})$,

$$\Phi_\xi(y)=\begin{cases}\psi_\xi(y) & \text{if } y>a_j,\\ 0 & \text{otherwise,}\end{cases}$$

$$\pi_i(y)=(y-\tau_{i+1})\cdots(y-\tau_{i+k-1}).$$

Since $[t_j, t_{j+k}) = [t'_j, t'_{j+\nu+1})$, we have

$$\alpha_{i,\nu+1,\tau',t'}(j) = (\tau'_{i+\nu+1} - \tau'_i)[\tau'_i, \cdots, \tau'_{i+\nu+1}]\Phi_\xi$$

for any $a_j \in [t_j, t_{j+k})$.

LEMMA 2.3. *Equation* (2.6) *holds for all i such that ψ_ω is a factor of π_i.*

Proof. Suppose that i is such that ψ_ω is a factor of π_i. Then the numbers $\tau_i, \cdots, \tau_{i+k}$ can be arranged in some order $x_0, \cdots, x_k$ such that $x_p = \omega_{p+1}$ for $p = 0, 1, \cdots, k-2-\nu$. Since $\psi_\omega(y) = (y - x_0) \cdots (y - x_{k-2-\nu})$, we have by properties of divided differences

$$[x_0, \cdots, x_p]\psi_\omega = \delta_{p,k-1-\nu} \quad \text{for } p = 0, 1, \cdots, k.$$

Applying Leibniz' rule for divided differences we find for any $a_j \in [t_j, t_{j+k})$

$$\begin{aligned}[\tau_i, \cdots, \tau_{i+k}]\Phi_{j,k} &= [x_0, \cdots, x_k](\psi_\omega \Phi_\xi) \\ &= \sum_{p=0}^{k} [x_0, \cdots, x_p]\psi_\omega [x_p, \cdots, x_k]\Phi_\xi \\ &= [x_{k-1-\nu}, \cdots, x_k]\Phi_\xi = [\tau'_i, \cdots, \tau'_{i+\nu+1}]\Phi_\xi.\end{aligned}$$

By assumption τ' is obtained from τ by removing $k-1-\nu$ of $\tau_{i+1}, \cdots, \tau_{i+k-1}$. Therefore $\tau_{i+k} - \tau_i = \tau'_{i+\nu+1} - \tau'_i$, and we have

$$(\tau_{i+k} - \tau_i)[\tau_i, \cdots, \tau_{i+k}]\Phi_{j,k} = (\tau'_{i+\nu+1} - \tau'_i)[\tau'_i, \cdots, \tau'_{i+\nu+1}]\Phi_\xi$$

which is (2.6). □

In order to show that (2.6) also holds for those values of i for which ψ_ω is not a factor of π_i, we need several lemmas. These lemmas also give some properties of discrete B-splines that we need for the algorithms in the next section. We start by giving a recurrence relation for discrete B-splines similar to (cf. [3, p. 97])

$$\alpha_{i,k}(j) = (t_{j+k-1} - \tau_i)\beta_{i,k-1}(j) + (\tau_{i+k} - t_{j+k-1})\beta_{i+1,k-1}(j), \tag{2.8}$$

where $\beta_{i,k}(j)$ is defined below. (For an alternative proof of (2.8) see [8].)

LEMMA 2.4. *Suppose that ξ is a new knot among $t_{j+1}, \cdots, t_{j+k-1}$ and that $t_j < t_{j+k}$. Let $\hat{\mathbf{t}}$ denote the knot vector obtained from $\mathbf{t}$ by reducing the number of occurrences of ξ by one. Then for all i*

$$\alpha_{i,k,\tau,t}(j) = (\xi - \tau_i)\beta_{i,k-1,\tau,\hat{t}}(j) + (\tau_{i+k} - \xi)\beta_{i+1,k-1,\tau,\hat{t}}(j), \tag{2.9}$$

where for $k \geqq 2$

$$\beta_{i,k-1,\tau,\hat{t}}(j) = \begin{cases} \alpha_{i,k-1,\tau,\hat{t}}(j)/(\tau_{i+k-1} - \tau_i) & \text{if } \tau_{i+k-1} > \tau_i, \\ 0 & \text{otherwise;} \end{cases} \tag{2.10}$$

and

$$\alpha_{i,1,\tau,\hat{t}}(j) = \begin{cases} 1 & \text{if } \tau_i \leqq \hat{t}_j < \tau_{i+1}, \\ 0 & \text{otherwise.} \end{cases}$$

Proof. We apply Leibniz' rule for divided differences to the product $\Phi_{j,k}(y) = (y - \xi)\hat{\Phi}_{j,k}(y)$ where $\hat{\Phi}_{j,k}(y)$ is obtained from $\Phi_{j,k}(y)$ by removing the factor $y - \xi$. Arguing as in [3, p. 97] we obtain (2.9) and (2.10). □

The assumption that ξ is a new knot ensures that τ will be a subsequence of $\hat{t}$.

LEMMA 2.5. *Suppose $t_{j+1}, \cdots, t_{j+k-1}$ are all new knots. Then*

$$\tau_\mu < t_{j+1} \leqq \cdots \leqq t_{j+k-1} < \tau_{\mu+1} \tag{2.11}$$

for some integer μ. Moreover $\alpha_{i,k,\tau,t}(j) > 0$ for $i = \mu - k + 1, \cdots, \mu$, and zero otherwise.

Proof. Clearly $t_{j+p} \neq \tau_q$ for $p = 1, 2, \cdots, k-1$ and any integer q, for otherwise $t_{j+1}, \cdots, t_{j+k-1}$ could not all be new knots in the above sense. Since also $\boldsymbol{\tau}$ is a subsequence of $\mathbf{t}$, the inequalities (2.11) follow.

For the proof of the positivity we use (2.9) with $\xi = t_{j+k-1}$ and induction on k. If $k = 1$ and $t_j < t_{j+1}$ then $\alpha_{i,1}(j) = 1 > 0$ for $i = \mu$, and zero otherwise. Let as before $\hat{\mathbf{t}}$ denote the sequence obtained from $\mathbf{t}$ by reducing the number of occurrences of ξ by one. Since

$$\tau_\mu < \hat{t}_{j+1} \leqq \cdots \leqq \hat{t}_{j+k-2} < \tau_{\mu+1} \quad \text{and} \quad \hat{t}_j = t_j < t_{j+k} = \hat{t}_{j+k-1}$$

we have by the induction hypothesis that $\beta_{i,k-1,\tau,\hat{t}}(j) > 0$ for $i = \mu - k + 2, \cdots, \mu$. Since $t_{j+k-1} > \tau_i$ for $i \leqq \mu$ and $\tau_{i+k} > t_{j+k-1}$ for $i \geqq \mu - k + 1$, we obtain the positivity result by (2.9). □

Let μ be an integer such that

$$\tau_\mu \leqq t_j < \tau_{\mu+1}. \tag{2.12}$$

In the next lemma we relate μ to the integer μ' in Theorem 2.2. We also show that the old knots among $t_{j+1}, \cdots, t_{j+k-1}$ can be identified as $\tau_{\mu'+1}, \cdots, \tau_{\mu'+k-1-\nu}$.

LEMMA 2.6. *Let μ and μ' be integers given by* (2.12) *and* (2.7) *respectively. Then*

$$\mu' = \begin{cases} \mu & \text{if } t_j < t_{j+1}, \\ \mu - \min(r_1, s_1) & \text{if } t_j = t_{j+1}, \end{cases} \tag{2.13}$$

where r_1 is given by (2.4) *and s_1 is the number of occurrences of t_{j+1} in τ. Moreover*

$$\psi_\omega(y) = (y - \tau_{\mu'+1}), \cdots (y - \tau_{\mu'+k-1-\nu}) \tag{2.14}$$

and

$$\tau_{\mu'} = \tau'_{\mu'} < \xi_r < \tau'_{\mu'+1} = \tau_{\mu'+k-\nu} \quad \text{for } r = 1, \cdots, \nu. \tag{2.15}$$

Proof. We consider two cases.

(i) $r_1 \geqq s_1$. In this case we have $r_1 - s_1$ new knots at z_1, and we obtain τ' by removing all the s_1 occurrences of z_1 from τ. But then $\mu' = \max\{p | \tau_p < z_1\}$ and we remove $\tau_{\mu'+1}, \cdots, \tau_{\mu'+s_1}$ from τ. Since $\omega_1, \cdots, \omega_{k-1-\nu}$ are consecutive elements of τ, equation (2.14) hold. Suppose $t_j = t_{j+1}$. Then $\mu = \max\{p | \tau_p = z_1\}$ and $\mu' = \mu - r_1$ since we remove then $\mu = \mu'$. Thus (2.13) holds in this case.

(ii) $r_1 < s_1$. There are no new knots at z_1. If $t_j < t_{j+1}$ then $\mu = \mu'$, and (2.13) and (2.14) hold. Suppose $t_j = t_{j+1}$. Then $\mu = \max\{p | \tau_p = z_1\}$ and $\mu' = \mu - r_1$ since we remove r_1 knots equal to z_1 from τ. Thus (2.13) and (2.14) follow.

It remains to prove (2.15). Since we obtain $\boldsymbol{\tau}'$ from $\boldsymbol{\tau}$ by removing $\tau_{\mu'+1}, \cdots, \tau_{\mu'+k-1-\nu}$, we have $\tau'_{\mu'} = \tau_\mu$ and $\tau'_{\mu'+1} = \tau_{\mu'+k-\nu}$. Also $\xi_p = t'_{j+p}$ for $p = 1, \cdots, \nu$. But then (2.15) follows from Lemma 2.5. □

We define

$$r_t(j) = \max\{p | t_{j+p-1} = t_j\},$$
$$l_t(j) = \max\{p | t_{j-p+1} = t_j\}.$$

The integers $r_t(j)$ and $l_t(j)$ are called respectively the right and left multiplicity of t_j. In the next lemma we want to show that $\alpha_{i,k}(j) = 0$ for $i < \mu' - \nu$ and $i > \mu'$. Consider equation (1.3). From local support, linear independence, and continuity properties of B-splines, it follows that $\alpha_{i,k}(j) = \alpha_{i,k,\tau,t}(j) = 0$ unless the support $[t_j, t_{j+k}]$ of $N_{j,k}$ is *properly contained* in the support $[\tau_i, \tau_{i+k}]$ of $B_{i,k}$. By definition, this means that the following four conditions must be satisfied:

(2.16) $$t_j \geqq \tau_i,$$

(2.17) $$t_j = \tau_i \Rightarrow r_t(j) \leqq r_\tau(i),$$

(2.18) $$t_{j+k} \leqq \tau_{i+k},$$

(2.19) $$t_{j+k} = \tau_{i+k} \Rightarrow l_t(j+k) \leqq l_\tau(i+k).$$

Equations (2.17) and (2.19) say that if $t_j = \tau_i (t_{j+k} = \tau_{i+k})$ then $N_{j,k}$ should have at least as many continuous derivatives at $t_j(t_{j+k})$ as $B_{i,k}$.

LEMMA 2.7. *Let μ' be given by* (2.13). *If $i < \mu' - \nu$ or $i > \mu'$ then $\alpha_{i,k,\tau,t}(j) = 0$.*

Proof. It is enough to show that for $i = \mu' + 1$ and $i = \mu' - \nu - 1$ the support of $N_{j,k}$ is not properly contained in the support of $B_{i,k}$. Consider $i = \mu' + 1$. Since by (2.14) we have $t_j \leqq \omega_1 = \tau_{\mu'+1}$, it is enough to show that $t_j = \tau_{\mu'+1}$ implies $r_t(j) > p = r_\tau(\mu'+1)$. By (2.14) we have $t_{j+h} = \omega_h = \tau_{\mu'+h}$ for $h = 1, \cdots, p$. But then $t_j = \cdots = t_{j+p}$ so that $r_t(j) > p$. The proof for $i = \mu' - \nu - 1$ is similar. □

Proof of Theorem 2.2. In Lemma 2.4 we proved (2.6) for all i such that ψ_ω is a factor π_i. By (2.14), the polynomial ψ_ω is a factor of π_i for $i = \mu' - \nu, \cdots, \mu'$. By Lemma 2.5 and Lemma 2.7, both sides of (2.6) are zero for $i > \mu'$ or $i < \mu' - \nu$. Thus equation (2.6) holds for all i. Since $t'_{j+p} = \xi_p$ for $p = 1, \cdots, \nu$, we have by (2.15) $\tau'_{\mu'} < t'_{j+1} \leqq \cdots \leqq t'_{j+\nu} < \tau'_{\mu'+1}$. Therefore $\alpha_{i,\nu+1,\tau',t'}(j) > 0$ for $i = \mu' - \nu, \cdots, \mu'$ by Lemma 2.5. Equation (2.6) now gives the positivity result. □

Let η be a real number, and let $\tau \cup \{\eta\}$ denote the sequence obtained by adding η to τ. If η is already an element of τ, the number of occurrences of η in τ is increased by one in $\tau \cup \{\eta\}$. Similarly, if η is an element of τ, let $\tau \backslash \{\eta\}$ denote the sequence obtained by decreasing the number of occurrences of η by one. With this convenient abuse of notation we may state the following corollary.

COROLLARY 2.8. *Let j be a given integer, and let η be any real number in $[t_j, t_{j+k}]$. Then for all i*

(2.20) $$\alpha_{i,k,\tau,t}(j) = \alpha_{i,k+1,\tau\cup\{\eta\},t\cup\{\eta\}}(j).$$

If ξ is a new knot among $t_{j+1}, \cdots, t_{j+k-1}$ then

(2.21) $$\alpha_{i,k,\tau\cup\{\xi\},t}(j) = \alpha_{i,k-1,\tau,t\backslash\{\xi\}}(j).$$

Proof. By Theorem 2.2 we have

$$\alpha_{i,k,\tau,t}(j) = \alpha_{i,\nu+1,\tau',t'}(j),$$

where as before, the old knots $\omega_1, \cdots, \omega_{k-1-\nu}$ have been removed from $\boldsymbol{\tau}$ and $\mathbf{t}$ in order to obtain $\boldsymbol{\tau}'$ and $\mathbf{t}'$. Let $\hat{\mathbf{t}} = \mathbf{t} \cup \{\eta\}$ and $\hat{\boldsymbol{\tau}} = \boldsymbol{\tau} \cup \{\eta\}$. Since $\eta \in [t_j, t_{j+k}]$, the old knots among $\hat{t}_{j+1}, \cdots, \hat{t}_{j+k}$ are $\omega_1, \cdots, \omega_{k-1-\nu}, \eta$. Therefore $\hat{\mathbf{t}}' = \mathbf{t}'$ and $\hat{\boldsymbol{\tau}}' = \boldsymbol{\tau}'$ so that

$$\alpha_{i,k+1,\hat{\tau},\hat{t}}(j) = \alpha_{i,\nu+1,\tau',t'}(j) = \alpha_{i,k,\tau,t}(j)$$

for all i. Equation (2.21) is a simple consequence of (2.20). □

Let τ and t be as before except that we may have $t_j = t_{j+k}$ for one or more values of j. It is of interest to determine whether the problem of computing $\alpha_{i,k}(j)$ given by (2.2) and (2.3) is properly posed. In the next theorem we give the exact conditions under which $\alpha_{i,k}(j)$ depends continuously on the sequences $\boldsymbol{\tau}$ and $\mathbf{t}$.

THEOREM 2.9. *Let i and j be given integers. The value of $\alpha_{i,k}(j)$ depends continuously on $\tau_i, \cdots, \tau_{i+k}$ and $t_j, \cdots, t_{j+k}$ if and only if the following condition is satisfied*:

(2.22) *If $t_j = t_{j+k} = x$ then x occurs at most $k-1$ times in the sequence $\tau_i, \cdots, \tau_{i+k}$.*

Proof. We first show that if (2.22) holds then $\alpha_{i,k}(j)$ is a continuous function of $\boldsymbol{\tau}$ and $\mathbf{t}$. Let $(\mathbf{t}^p)_1^\infty$ be a sequence of nondecreasing bi-infinite sequences $\mathbf{t}^p=(t_j^p)_{j=-\infty}^\infty$ such that $\lim_{p\to\infty} t_j^p = t_j$ for each j. For each p, let $\boldsymbol{\tau}^p$ be a subsequence of $\mathbf{t}^p$ containing at least $k+1$ elements and sucn that $\lim_{p\to\infty}\tau_i^p=\tau_i$ for each i. For every pair of integers i and j we have a sequence of discrete B-spline values $(\alpha_{i,k}^p(j))_{p=1}^\infty$, where $\alpha_{i,k}^p(j)=\alpha_{i,k,\tau^p,t^p}(j)$. Given i and j satisfying (2.22), we want to show that $(\alpha_{i,k}^p(j))$ converges to $\alpha_{i,k}(j)=\alpha_{i,k,\tau,t}(j)$. We distinguish between two cases:

(i) $t_j<t_{j+k}$. Choose $s\in\{j,j+1,\cdots,j+k-1\}$ such that $t_s<t_{s+1}$. Since discrete B-splines are nonnegative and sum to one, they are uniformly bounded. We can therefore find subsequences $(\mathbf{u}^p)_1^\infty$ and $(\mathbf{v}^p)_1^\infty$ of $(\tau^p)_1^\infty$ and $(\mathbf{t}^p)_1^\infty$ respectively, such that

$$\lim_{p\to\infty}\alpha_{i,k,u^p,v^p}(q)=\rho_q$$

exists for $q=s+1-k,\cdots,s$. Choosing any $x\in(t_s,t_{s+1})$, we have by (2.6) and [10, Thm. 4.26]

$$\begin{aligned}\sum_{q=s+1-k}^{s}\alpha_{i,k}(q)N_{q,k}(x)=B_{i,k}(x)&=\lim_{p\to\infty}B_{i,k,u^p}(x)\\&=\lim_{p\to\infty}\sum_{q=s+1-k}^{s}\alpha_{i,k,u^p,v^p}(q)N_{q,k,v^p}(x)\\&=\sum_{q=s+1-k}^{s}\rho_qN_{q,k}(x).\end{aligned}$$

Here B_{i,k,u^p} and N_{q,k,v^p} denote normalized B-splines on the knot sequences $\mathbf{u}^p$ and $\mathbf{v}^p$ respectively. By linear independence we have $\alpha_{i,k}(q)=\rho_q$ for $q=s+1-k,\cdots,s$. In particular $\rho_j=\alpha_{i,k}(j)$ for any limit point ρ_j of $(\alpha_{i,k}^p(j))_{p=1}^\infty$.

(ii) $t_j=t_{j+k}=x$. We use induction on k. The theorem clearly holds for $k=1$ since then $x\neq\tau_i$ and $x\neq\tau_{i+1}$.

Suppose that $k\geqq 2$. Consider first the case where x occurs at most $k-2$ times among $\tau_i,\cdots,\tau_{i+k}$. By the recurrence relation (2.8) we have

$$\alpha_{i,k}^p(j)=(t_{j+k-1}^p-\tau_i^p)\beta_{i,k-1}^p(j)+(\tau_{i+k}^p-t_{j+k-1}^p)\beta_{i+1,k-1}^p(j).$$

Since $\tau_i<\tau_{i+k-1}$ and $\tau_{i+1}<\tau_{i+k}$ we obtain by induction

$$\begin{aligned}\lim_{p\to\infty}\alpha_{i,k}^p(j)&=(t_{j+k-1}-\tau_i)\beta_{i,k-1}(j)+(\tau_{i+k}-t_{j+k-1})\beta_{i+1,k-1}(j)\\&=\alpha_{i,k}(j).\end{aligned}\tag{2.23}$$

If x occurs exactly $k-1$ times among $\tau_i,\cdots,\tau_{i+k}$ there are three possibilities.

(a) $x=\tau_i=\tau_{i+k-2}<\tau_{i+k-1}\leqq\tau_{i+k}$. In this case $t_{j+k-1}^p-\tau_i^p\to 0$ and $\beta_{i,k-1}^p(j)$ remains bounded since $\tau_i<\tau_{i+k-1}$, while $\beta_{i+1,k-1}^p(j)\to\beta_{i+1,k-1}(j)$ by induction. Hence (2.23) holds even in this case.

(b) $\tau_i<x=\tau_{i+1}=\tau_{i+k}-1<\tau_{i+k}$. By (2.3) we have $\alpha_{i,k,\tau,t}(j)=B_{i,k}(x)=1$. It will therefore be enough to show that

$$\lim_{p\to\infty}\sum_{q=i}^{i+k-r}\alpha_{q,r}^p(j)=1\quad\text{for } r=1,2,\cdots,k.\tag{2.24}$$

Without loss of generality we may assume that $\tau_q^p, t_s^p\in(\tau_i^p,\tau_{i+k}^p)$ for $q=i+1,\cdots,i+k-1$ and $s=j,\cdots,j+k$. Then precisely one of $\alpha_{i,1}^p(j),\cdots,\alpha_{i+k-1,1}^p(j)$

is equal to 1. Thus (2.24) holds for $r=1$. Suppose that $r\geqq 2$. By (2.8) we have

$$\sum_{q=i}^{i+k-r} \alpha^p_{q,r}(j) = \sum_{q=i}^{i+k-r} [(t^p_{j+r-1}-\tau^p_q)\beta^p_{q,r-1}(j)+(\tau^p_{q+r}-t^p_{j+r-1})\beta^p_{q+1,r-1}(j)]$$

$$=\left[(t^p_{j+r-1}-\tau^p_i)\beta^p_{i,r-1}(j)+(\tau^p_{i+k}-t^p_{j+r-1})\beta^p_{i+k-r+1,r-1}(j)+\sum_{q=i+1}^{i+k-r}\alpha^p_{q,r-1}(j)\right].$$

Therefore, since $t^p_{j+r-1}-\tau^p_i \to \tau_{i+r-1}-\tau_i$ and $\tau^p_{i+k}-t^p_{j+r-1}\to \tau_{i+k}-\tau_{i+k-r+1}$ we find

$$\lim_{p\to\infty}\sum_{q=i}^{i+k-r}\alpha^p_{q,r}(j)=\lim_{p\to\infty}\sum_{q=i}^{i+k-r+1}\alpha^p_{q,r-1}(j)$$

and (2.24) follows by induction.

(c) $\tau_i \leqq \tau_{i+1} < x = \tau_{i+2} = \tau_{i+k}$. This case is similar to case (a).

In order to complete the proof of the theorem, we give examples showing that $\alpha_{i,k}(j)$ does not depend continuously on τ and t if condition (2.22) is violated, i.e., if the number $x=t_j=t_{j+k}$ occurs k or $k+1$ times in the sequence $\tau_i, \cdots, \tau_{i+k}$. Again there are three cases:

(1) $\tau_i < \tau_{i+1} = \tau_{i+k} = x$. We have $\alpha_{i,k}(j) = B_{i,k}(x) = 0$ since $B_{i,k}$ is right continuous. Choose $\boldsymbol{\tau}^p = \boldsymbol{\tau}$ and $\mathbf{t}^p$ such that $t^p_j < x$ and $t^p_{j+1} = x$. Then by Theorem 2.2 $\alpha^p_{i,k}(j)=1$ for all p.

(2) $\tau_i = \tau_{i+k-1} = x < \tau_{i+k}$. This case is similar to case (i).

(3) $\tau_i = \tau_{i+k} = x$. By definition $\alpha_{i,k}(j)=0$ in this case. Choose τ^p and t^p such that $\tau^p_i < \tau^p_{i+k}$ and $\tau^p_{i+q} = t^p_{j+q}$ for $q=0, \cdots, k$. Then $\alpha^p_{i,k}(j)=1$ for all p. □

The use of [10, Thm. 4.26] can be avoided if part (i) of the foregoing proof is based on (1.1) instead of (1.3). We note that Theorem 2.9 assures the continuity of $\alpha_{i,k}(j)$ in the cases $\tau_i=\tau_{i+k-1}$, $\tau_{i+1}=\tau_{i+k}$, and even $\tau_i=\tau_{i+k}$, as long as $t_j<t_{j+k}$.

3. Algorithms. In this section the knot vectors $\boldsymbol{\tau}$ and $\mathbf{t}$ will be finite sequences. Let k, m_1, and m_2 be given integers with k positive and $m_1 \leqq m_2$, and let $\mathbf{t}=(t_i)_{i=m_1}^{m_2+k}$ be a nondecreasing sequence of real numbers with $t_j<t_{j+k}$ for $j=m_1, m_1+1, \cdots, m_2$. Let $\boldsymbol{\tau}=(\tau_i)_{i=n_1}^{n_2+k}$ be a subsequence of $\mathbf{t}$ so that $n_2-n_1\leqq m_2-m_1$.

Let $j\in\{m_1, m_1+1, \cdots, m_2\}$ be a fixed integer. We first want to give an algorithm to compute

$$\alpha_{i,k,\tau,t}(j) \quad \text{for } i=i1, i1+1, \cdots, i2$$

where

$$i1=\max(\mu'-\nu, n_1), \quad i2=\min(\mu', n_2), \tag{3.1}$$

and μ' and ν are given by (2.7) and (2.5) respectively. These are the discrete B-splines of order k which are nonzero for a given j.[2]

In any of the four situations

$$t_j<\tau_{n_1}, \tag{3.2}$$

$$t_j=\tau_{n_1} \text{ and } r_t(j)>r_\tau(n_1), \tag{3.3}$$

$$t_{j+k}>\tau_{n_2+k}, \tag{3.4}$$

$$t_{j+k}=\tau_{n_2+k} \text{ and } l_t(j+k)>l_\tau(n_2+k), \tag{3.5}$$

[2] In order to handle the complications near the beginning and end of the knot vectors, it is convenient to first extend $\boldsymbol{\tau}$ and $\mathbf{t}$ to bi-infinite sequences, apply the results in § 3, and then restrict the range of the indices to the original, finite sequences.

the support of $N_{j,k}$ is not properly contained in the support of $B_{i,k}$ for any $i \in \{n_1, n_1+1, \cdots, n_2\}$. Therefore $\alpha_{i,k,\tau,t}(j) = 0$ for all i in these cases. For other values of j we use Theorem 2.2. For $p \leqq \nu+1$ and for fixed j, we define

$$\alpha'_{i,p} = \alpha'_{i,p}(j) = \alpha_{i,p,\tau',t'}(j)$$

where $\boldsymbol{\tau}' = (\tau'_i)_{i=n_1}^{n_2+\nu+1}$ and $\mathbf{t}' = (t'_j)_{j=m_1}^{m_2+\nu+1}$ are obtained from $\boldsymbol{\tau}$ and $\mathbf{t}$ respectively by removing the old knots $\tau_{\mu'+1}, \cdots, \tau_{\mu'+k-1-\nu}$ among $t_{j+1}, \cdots, t_{j+k-1}$. Using induction on p and the recurrence relation (2.9) on $\boldsymbol{\tau}'$, $\mathbf{t}'$, we find that for $p = 1, \cdots, \nu+1$

$$\alpha'_{i,p,\tau' t'}(j) > 0 \quad \text{for } i = \max(\mu'-p+1, n_1), \cdots, \min(\mu', n_2+\nu+1-p), \tag{3.6}$$

and zero otherwise. These positive α's can be arranged in a polygonal shaped scheme. If $n_1 + \nu \leqq \mu' \leqq n_2$ the scheme is triangular,

$$\begin{array}{ccc} & & \alpha'_{\mu',1} \\ & \alpha'_{\mu'-1,2} & \alpha'_{\mu',2} \\ & \vdots & \vdots \\ \alpha'_{\mu'-\nu,\nu+1} & \cdots & \alpha'_{\mu',\nu+1} \end{array} \tag{3.7}$$

where by (2.9)

$$\alpha'_{i,p+1} = \alpha'_{i,p+1}(j) = \begin{cases} \delta_{i+1,p}\alpha'_{i+1,p}(j), & \text{if } i = \mu'-p; \\ \gamma_{i,p}\alpha'_{i,p}(j) + \delta_{i+1,p}\alpha'_{i+1,p}(j), & \text{if } \mu'-p < i < \mu'; \\ \gamma_{i,p}\alpha'_{i,p}(j), & \text{if } i = \mu'; \end{cases} \tag{3.8}$$

where

$$\gamma_{i,p} = \frac{\xi_p - \tau'_i}{\tau'_{i+p} - \tau'_i} \quad \text{and} \quad \delta_{i,p} = \frac{\tau'_{i+p} - \xi_p}{\tau'_{i+p} - \tau'_i}. \tag{3.9}$$

Since

$$\tau'_i = \begin{cases} \tau_i & \text{if } i \leqq \mu', \\ \tau_{i+k-\nu} & \text{if } i > \mu', \end{cases} \tag{3.10}$$

we can replace τ'_{i-1} by τ_{i-1} and τ'_{i+p} by $\tau_{i+p+k-\nu-1}$ in (3.9). The following detailed algorithm can now be given to compute

$$\alpha_{i,k,\tau,t}(j) = \alpha'_{i,\nu+1} \quad \text{for } i = i1, \cdots, i2.$$

ALGORITHM 1. Let $j \in \{m_1, \cdots, m_2\}$. Then no $\alpha_{i,k}(j)$ is nonzero if any of (3.2)–(3.5) hold. Otherwise, let μ be such that $\tau_\mu \leqq t_j < \tau_{\mu+1}$. By performing the following steps the entries of (3.7) are computed.

1. $i = j+1$; $\mu' := \mu$;
2. **while** $t(i) = \tau(\mu')$ **and** $i < j+k$ **do** ($i := i+1$; $\mu' := \mu'-1$;)
3. $ih := \mu'+1$; $\nu := 0$;
4. **for** $p := 1, 2, \cdots, k-1$
   1. if $t(j+p) = \tau(ih)$ **then** $ih := ih+1$
      else ($\nu := \nu+1$; $\xi(\nu) := t(j+p)$;)
5. $ah(k, 1) := 1$;
6. **for** $p := 1, 2, \cdots, \nu$
   1. $\beta 1 := 0$; $tj := \xi(p)$;
   2. **if** $p \geqq \mu'$ **then**
      $\beta 1 := (tj - \tau(n1)) * ah(1+k-\mu', p)/(\tau(p+k-\nu) - \tau(n1))$;
   3. $il := \max(n1+1, \mu'-p+1)$; $iu := \min(\mu', n2+\nu-p)$;

4. **for** $i := il, il+1, \cdots, iu$
   1. $d1 := tj - \tau(i)$; $d2 := \tau(i+p+k-\nu-1) - tj$;
   2. $\beta := ah(i+k-\mu', p)/(d1+d2)$;
   3. $ah(i+k-\mu'-1, p+1) := d2 * \beta + \beta 1$;
   4. $\beta 1 := d1 * \beta$;
5. $ah(iu+k-\mu', p+1) := \beta 1$;
6. **if** $iu < \mu'$ **then**
   $ah(iu+k-\mu', p+1) := \beta 1 + (\tau(n2+k) - tj) * ah(iu+k-\mu'+1, p)$
   $/(\tau(n2+k) - \tau(iu+1))$;

Algorithm 1 requires two arrays $\xi(1:k-1)$ and $ah(1:k, 1:k)$ in addition to $\boldsymbol{\tau}$ and $\mathbf{t}$. We have for $p \leqq \nu+1$

$$ah(i+k-\mu', p) = \alpha_{i,p,\tau',t'}(j) \text{ for } i = \max(\mu'-p+1, n_1), \cdots, \min(\mu', n_2+\nu+1-p).$$

If only the bottom line of (3.7) is of interest, it is possible to use a one-dimensional array $ah(1:k)$, by simply omitting the second subscript in all references to ah. By Theorem 2.2 we then have

$$ah(i+k-\mu') = \alpha_{i,\nu+1,\tau',t'}(j) = \alpha_{i,k,\tau,t}(j) \quad \text{for } i = i1, \cdots, i2,$$

where $i1$ and $i2$ are given by (3.1).

We note that by (2.15) the quantities $\delta_{i,p}$ and $\gamma_{i,p}$ given by (3.9) satisfy $0 < \delta_{i,p} < 1$ and $0 < \gamma_{i,p} < 1$ for all values of i and p in Algorithm 1. Since also the discrete B-spline values involved are positive by (3.6), the algorithm is unconditionally stable. Moreover, division by zero in statement 6.4.2 can never occur.

It may be of interest to try to relate the entries in (3.7) to those computed by Algorithm 1 in [3]. There, the triangular scheme

$$\begin{array}{ccc} & & \alpha_{\mu,1} \\ & \alpha_{\mu-1,2} & \alpha_{\mu,2} \\ & \vdots & \vdots \\ \alpha_{\mu-k+1,k} & \cdots & \alpha_{\mu,k} \end{array} \tag{3.11}$$

was computed, where $\alpha_{i,p} = \alpha_{i,p,\tau',t}(j)$ and μ is given by (2.12). If $t_{j+1}, \cdots, t_{j+k-1}$ are all new knots, the two schemes (3.11) and (3.7) are identical. Consider now the general case. In order to express $\alpha'_{i,p} = \alpha_{i,p,\tau',t'}(j)$ as discrete B-splines with $\boldsymbol{\tau}$ as knot vector, we first note that

$$\alpha_{i,p,\tau',t'\tau',t'}(j) = \alpha_{i,p,\tau',t'\setminus\{\xi_p,\cdots,\xi_\nu\}}(j) \tag{3.12}$$

where $\xi_l = t'_{j+l}$ for $l = 1, \cdots, \nu$. This follows since the left-hand side is independent of $t'_{j+p}, \cdots, t'_{j+\nu}$. Applying Corollary 2.8 and (3.12) to $\alpha_{i,p,\tau',t'}(j)$ we obtain

$$\alpha'_{i,p} = \alpha_{i,p+k-1-\nu,\tau,t\setminus\{\xi_p,\cdots,\xi_\nu\}}(j). \tag{3.13}$$

Thus $\alpha'_{i,p}$ is a discrete B-spline of order $p+k-1-\nu$ on $\mathbf{t}^p = \mathbf{t}\setminus\{\xi_p, \cdots, \xi_\nu\}$ (but not necessarily on $\mathbf{t}$) with knots $\boldsymbol{\tau}$. Therefore we cannot in general recover (3.7) as a subtriangle of (3.11).

By (3.13) Algorithm 1 can be interpreted as follows. We start with $\alpha'_{i,1} = \alpha_{i,k-\nu,\tau,t^1}(j) = \delta_{\mu',i}$, where $\mathbf{t}^1$ is obtained from $\mathbf{t}$ by removing all the new knots among $t_{j+1}, \cdots, t_{j+k-1}$. Using (2.9) we then compute $\alpha'_{i,p+1} = \alpha_{i,p+k-\nu,\tau,t^{p+1}}(j)$ from $\alpha'_{i,p}$ and $\alpha'_{i+1,p}$ by adding ξ_p to $\mathbf{t}^p$ for $p = 1, \cdots, \nu$. When $p = \nu$ we have added all the new knots and $\mathbf{t}^{\nu+1} = \mathbf{t}$.

Algorithm 1 may be used to compute

$$d_j = \sum_{i=i1}^{i2} \alpha_{i,k,\tau,t}(j)c_i = \sum_{i=i1}^{i2} \alpha'_{i,\nu+1}(j)c_i \tag{3.14}$$

where $c_1, \cdots, c_n$ are given numbers and $i1$ and $i2$ are given by (3.1). Alternatively, we can follow a well-known procedure (see e.g. [3, p. 99]) and generate a triangular scheme

$$\begin{array}{cccc} c^{[1]}_{\mu'-\nu} & c^{[1]}_{\mu'-\nu+1} & \cdots & c^{[1]}_{\mu'} \\ & c^{[2]}_{\mu'-\nu+1} & \cdots & c^{[2]}_{\mu'} \\ & & & \vdots \\ & & & c^{[\nu+1]}_{\mu'} \end{array} \tag{3.15}$$

where $c_i^{[1]} = c_i$, and where by (3.9), (3.10), and (3.6)

$$c_i^{[p+1]} = \frac{(\xi_p - \tau_i)c_i^{[p]} + (\tau_{i+k-p} - \xi_p)c_{i-1}^{[p]}}{\tau_{i+k-p} - \tau_i}.$$

We have $d_j = c_{\mu'}^{[\nu+1]}$. (Note that $\tau_{i+k-p} = \tau'_{i+\nu-p+1}$.)

A detailed algorithm may be as follows.

ALGORITHM 2. Let $j \in \{m_1, m_1+1, \cdots, m_2\}$. If any of (3.2)–(3.5) hold then $d_j = 0$. Otherwise, let μ be such that $\tau_\mu \leqq t_j < \tau_{\mu+1}$. In order to compute d_j, perform the following steps.

1. $i := j+1$; $\mu' := \mu$;
2. **while** $t(i) = \tau(\mu')$ **and** $i < j+k$ **do** ($i := i+1$; $\mu' := \mu'-1$;)
3. $ch(k) :=$ if $\mu' \leqq n2$ **then** $c(\mu')$ **else** 0;
4. $ih := \mu'+1$; $\nu := 0$;
5. **for** $p := 1, 2, \cdots, k-1$
   1. **if** $t(j+p) = \tau(ih)$ **then** $ih := ih+1$
      else begin
      1. $\nu := \nu+1$; $\xi(\nu) := t(j+p)$;
      2. $k\nu := k-\nu$; $s := 0$;
      3. $ch(k\nu) :=$ if $\mu'-\nu < n1$ **or** $\mu'-\nu > n2$
         then 0 **else** $c(\mu'-\nu)$;
      4. $\tau 2 :=$ **if** $\mu'-\nu \leqq n2$ **then** $\tau(\mu'+k\nu)$ **else** $\tau(n2+k)$;
      5. $il := \max(\mu'-\nu+1, n1)$; $iu := \min(\mu', n2+\nu)$;
      6. **for** $i := il, il+1, \cdots, iu$
         1. $s := s+1$; $k\nu := k\nu+1$;
         2. $d1 := \xi(s) - \tau(i)$; $d2 := \tau 2 - \xi(s)$;
         3. $ch(k\nu) := (d1 * ch(k\nu) + d2 * ch(k\nu-1))/(d1+d2)$;

      end;
6. $dj := ch(k)$;

Algorithm 2 generates the entries of (3.15) diagonal-wise from right to left in a vector $ch(1:k)$ with $d_j = ch(k)$ at the end of the algorithm.

It follows from (2.15) that only strict convex combinations are used in generating (3.15) by Algorithm 2. Moreover, division by zero in statement 5.1.6.3 can never occur.

The two algorithms are reasonably robust. They may fail if the computed μ' is smaller than the exact μ'. As an example of this, suppose that $k=4$, $m=5$, and $\boldsymbol{\tau} = \mathbf{t} = (1, 1, 1, 1, 2, 3, 3, 3, 3)$. Suppose that the machine representation $\hat{\tau}_5$ of τ_5 is greater than the machine representation $\hat{t}_5$ of t_5. For $j=5$ we find $\hat{\mu}' = 4$ instead of $\mu' = 5$ and the computed value of d_5 will be incorrect. Note that Algorithms 1 and 2 in [3] will also fail if μ is computed as 4.

Let $\hat{\boldsymbol{\tau}}$ and $\hat{\mathbf{t}}$ be the machine representations of $\boldsymbol{\tau}$ and $\mathbf{t}$ respectively. In order to avoid such problems as the above, one should make sure that $\hat{\boldsymbol{\tau}}$ is a subsequence of $\hat{\mathbf{t}}$ and that $\hat{\mathbf{t}}$ is nondecreasing.

We have assumed that $t_j < t_{j+k}$ for $j = m_1, \cdots, m_2$. It is possible that this condition will not be satisfied for $\hat{\mathbf{t}}$. In order to make the algorithms produce the value given by (2.3), for $t_j = t_{j+k}$, we have added a test $i < j+k$ in statement 2 of both algorithms. Thus, both algorithms work even if $\hat{t}_j = \hat{t}_{j+k}$ for one or more values of j. We do, however, assume that $\hat{\tau}_i < \hat{\tau}_{i+k}$.

4. Remarks. 1. In [2] a method is given for adding one (possibly multiple) knot to a B-spline curve. By sequentially adding one knot at a time an alternative method to Algorithm 2 is obtained. Our Algorithm 2 is similar to Böhm's method when $\mathbf{t}$ is obtained from $\boldsymbol{\tau}$ by adding one knot.

2. Algorithms 1 and 2 are amenable to parallel implementations.

3. Algorithms 1 and 2 can both be applied to computing d_j given by (3.14). In general, Algorithm 1 requires fewer arithmetic operations than Alorithm 2 when the spline coefficients are vectors. It is even more advantageous to use Algorithm 1 when dealing with a tensor product B-spline surface.

4. In [5] it was shown that $\alpha_{i,k,\tau,t}(j) > 0$ if and only if the support of $N_{j,k}$ is properly contained in the support of $B_{i,k}$. Theorem 2.2 gives an alternative formulation and proof of this result.

5. Algorithms 1 and 2 reduce to standard B-spline algorithms in special cases. Suppose $\tau_\mu \leqq x < \tau_{\mu+1}$. Algorithm 1 can be used to compute the values of all nonzero B-splines at x, of order $\leqq k$. This is achieved by forming $\mathbf{t}$ from $\boldsymbol{\tau}$ by making x occur precisely $k-1$ times in the $\mathbf{t}$ sequence. Let j be such that $t_{j+1} = \cdots = t_{j+k-1} = x$. As before, let ν be the number of times we added x to $\boldsymbol{\tau}$ to obtain $\mathbf{t}$, and let $\mu' = \mu - k + 1 + \nu$. By the definition of discrete B-splines (2.2), and (3.13), we find that

$$\alpha'_{i,p} = B_{i,k-1-\nu+p}(x) \quad \text{for } p = 1, 2, \cdots, \nu+1.$$

The scheme (3.7) therefore reduces to

$$\tag{4.1} \begin{array}{ccc} & & B_{\mu',k-\nu}(x) \\ & B_{\mu'-1,k-\nu+1}(x) & B_{\mu',k-\nu+1}(x) \\ & \vdots & \vdots \\ \beta_{\mu'-\nu,k}(x) & \cdots & B_{\mu',k}(x) \end{array}$$

in this case. If $\nu = k-1$ and $n_1 + k - 1 \leqq \mu \leqq n_2$ then Algorithm 1 is equivalent to [10, Algorithm 5.5, p. 192]. If $\nu < k-1$ then $B_{\mu+1-r,r}(x) = 1$ for $r = 1, 2, \cdots, k-\nu-1$. These numbers plus the ones in (4.1) are precisely all the nonzero B-spline values at x.

Similarly, Algorithm 2 contains [10, Algorithm 5.8, p. 194] as a special case.

Acknowledgment. We would like to thank E. Cohen and R. Riesenfeld for valuable comments.

REFERENCES

[1] C. DE BOOR, *Splines as linear combinations of B-splines. A survey*, in Approximation Theory II, G. G. Lorentz, C. K. Chui and L. L. Schumaker, eds., Academic Press, New York, 1976, pp. 1–47.

[2] W. BÖHM, *Inserting new knots into B-spline curves*, Computer Aided Deisgn, 12 (1980), pp. 199–201.

[3] E. Cohen, T. Lyche and R. Riesenfeld, *Discrete B-splines and subdivision techniques in computer-aided geometrc design and computer graphics*, Computer Graphics and Image Processing, 14 (1980), pp. 87–111.

[4] E. Cohen, T. Lyche and L. L. Schumaker, *Degree raising for splines*, Research Report CAT #44, Texas A & M University, November, 1983.

[5] Rong-Qing Jia, *Total positivity of the discrete spline collocation matrix*, J. Approx. Theory, 39 (1983), pp. 11–23.

[6] T. Lyche, *Discrete polynomial spline approximation methods*, Dissertation, Univ. Texas, Austin, 1975.

[7] T. Popoviciu, *Sur quelques propriétés des fonctions d'une ou de deux variables reelles*, Mathematica, 8 (1934), pp. 1–85.

[8] H. Prautzsch, *A short proof of the Oslo algorithm*, Computer Aided Geometric Design, 1 (1984), pp. 95–96.

[9] L. L. Schumaker, *Constructive aspects of discrete polynomial spline functions*, in Approximation Theory, G. G. Lorentz, ed. in cooperation with H. Berens, E. W. Cheney, and L. L. Schumaker, Academic Press, New York, 1973, pp. 469–476.

[10] ———, *Spline Functions: Basic Theory*, John Wiley, New York, 1981.