CONSTRUCTIVE THEORY
OF FUNCTIONS, Varna '91
Sofia, 1992, pp. 173-181

TRIANGULAR CONSTRUCTIVE ALGORITHMS

Ljubiša M. Kocić, Sladjana D. Marinković

Faculty of Electronic Engineering, Department of Mathematics,
P.O.Box 73, 18000 NIŠ, YUGOSLAVIA

Abstract. The common characteristic of Bernstein polynomials, Lagrange interpolants and B-spline functions is the existence of a recursive triangular algorithm that allow both to evaluate these functions in a computer oriented way and to use it in theoretical considerations. Here, the general triangular scheme is proposed and its main properties are studied. The application in Computer Aided Geometric Design is specially stressed.

1. Triangular Algorithms

It is well known that some types of interpolating or approximating functions can be easily computed by recursive algorithms that have form of a triangular array. Classical examples are de Casteljau algorithm for Bernstein polynomials , Aitken-Neville recursion for Lagrange interpolants and De Boor algorithm for B-spline functions. Typically, such algorithms construct the values of some basic functions $\{b_i^n(t)\}_{i=0}^n$ using

$$
\begin{aligned}
&b_0^0(t) = b_0(t) , \\
&b_i^{r+1}(t) = [1-\lambda_i^r(t)]b_i^r(t) + \lambda_{i-1}^r(t)b_{i-1}^r(t) , \\
&i = 0, \dots , n-r, \quad r = 1, \dots , n ,
\end{aligned}
\tag{1}
$$

where b_{-j}^0 and b_j^0 $(j = 1, 2, \dots)$ are supplied conveniently.

In three classical cases, the blending function λ_i^r is an affine function of t which results in polynomial basis over I. Using a general form of an affine function, suggested by urn models, Goldman [2], [3] introduced other bases and got a class of generalized B-splines, β_1-splines, Pólya curves and so on. Further generalizations are introduced and studied in [4], [7] and [8]. It is easy to see that if $p(t)$ is a span of the basis $\{b_0^n, \dots , b_n^n\}$,

$$
p(t) = \sum_{i=0}^{n} b_i^n(t)P_i, \quad t \in I , \tag{2}
$$

where $P_0, \dots P_n$ are some data values, the recursive algorithm takes place. This algorithm begins with the set of data $P = \{P_0, \dots P_n\}$, and using the recurrence

$$
\begin{aligned}
&P_i^0(t) = P_i , \quad i = 0, \dots , n , \\
&P_i^r(t) = [1 - \lambda_i^{n-r}(t)]P_i^{r-1}(t) + \lambda_i^{n-r}(t)P_{i+1}^{r-1}(t) , \\
&i = 0, \dots , n-r, \quad r = 1, \dots , n ,
\end{aligned}
\tag{3}
$$

yields the value $P_0^n(t) = p(t), t \in I$. The recursion (1) products a triangular array of functions with b_0^0 as an apex, and $\{b_0^n, \dots, b_n^n\}$ as the bottom, while (3) starts from the bottom $P_0, \dots, P_n$ and finishes in $P_0^n(t)$.

The special importance of triangular algorithms is motivated by their broad application, both theoretical and practical. If the data P contained the multidimensional points, $p(t)$ is a parametricaly defined multidimensional curve being the main tool in Geometric Modeling and CAGD. The elegant and constructive manner of algorithm (3) lead to a nice theory of polar forms (blossoming) exhaustively studied by Ramshaw (see for example [14]). In [8] and [15] this theory was applied for further study of triangular algorithms.

But, there are also other schemes of the form (1) and (3) with blending functions λ_i^r that are not affine, like in the case of Chebyshevian B-splines [13]. Also, there are many examples where it is necessary to change one basis to another. Having the same triangular form as (3), this procedure defines the element in the next level as linear combination of two or more elements from the preceeding stage. The aim of this paper is to establish and examine the basic properties of such algorithms.

2. Definition and properties

Let $b_n(t) = [b_0^n(t) \dots b_n^n(t)]^T$ be the vector of basic functions from some linear space $S_n(I)$ of functions defined on I, and $P = [P_0 \dots P_n]^T$ be the data vector. Then, in matrix notation, the curve (2) is given by

$$p(t) = b_n^T(t)P\ ,\quad t \in I\ . \tag{4}$$

Let $A_k(t) = [a_{ij}^k(t)]_{i=0,j=0}^{k-1,k}$ be a rectangular matrix of functions a_{ij}^k, defined on I. Then,

Theorem 1. *The following three statements are equivalent:*

a) The basis vector allows decomposition

$$b_n(t) = A_n^T(t) \cdot \dots \cdot A_1^T(t)\ ,\quad t \in I; \tag{5}$$

b) The basis $b_n(t)$ is defined by the triangular constructive algorithm

$$b_0(t) = [1],\quad b_r(t) = A_r^T(t) b_{r-1}(t),\ \ r = 1, \dots n, t \in I; \tag{6}$$

c) The value of the curve for $t \in I$ is given by $p(t) = P_0^n(t)$, where $P_0^n(t)$ is defined by the triangular recursion

$$\begin{aligned} &P^0(t) = [P_0 \dots P_n]^T\ , \\ &P^{r+1}(t) = A_{n-r}(t)P^r(t),\ \ r = 0, ..., n-1. \end{aligned} \tag{7}$$

PROOF. Let $A_0(t) = [1]$, and $b_r(t) = A_r^T(t) \cdot \dots \cdot A_0^T(t)$. Then, (6) is the consequence of (5) and vice versa; It follows from (6) that $b_n = A_n^T \cdot \dots \cdot A_{m+1}^T b_m$ (the argument is omitted), wich yields, after insertion in (4) $p = b_n^T P = (A_n^T \cdot \dots \cdot A_{n-r+1}^T b_{n-r})^T P^0 = b_{n-r}^T (A_{n-r+1} \cdot \dots \cdot A_n P^0) = b_{n-r}^T P^r$, where $P^r = A_{n-r+1} \cdot \dots \cdot A_n P^0$ $r = 0, \dots, n$, which gives $P^{r+1} = A_{n-r} P^r$ and, for $r = n$, $P^n(t) = p(t), t \in I$. On the contrary, if (7) is valid, then (5) is true, so (5), (6) and (7) are equivalent statements. □

Theorem 2. *If $b = [b_0^n \dots b_n^n]^T$ be any basis in some linear space $S_n(I)$ with the known recursion of type (6). Than, any other basis β allows the recursive algorithm of the same type.*

PROOF. For any basis β from $S_n(I)$, there exists a square matrix M so that $\beta = Mb$. Since b has the recursive algorithm (6), then, by Theorem 1, the decomposition (5) takes place, so that

(8) $$\beta = MA_n^T \cdot \dots \cdot A_1^T = B_n^T \cdot \dots \cdot B_1^T,$$

where $B_n = A_n M^T$, $B_i = A_i$, $i = 1, \dots, n-1$. □

Corollary 1. *Any polynomial basis has the triangular recursive algorithm (6).*

PROOF. In the space of polynomials of order $\leq n$, on $[a, b]$, one of the bases is the Bernstein one $b_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$ that satisfies

(9) $$b_i^0(t) = \delta_{i0};\ b_i^{r+1}(t) = \frac{t-a}{b-a} b_{i-1}^r(t) + \frac{b-t}{b-a} b_i^r(t),\ r = 0, \dots, n-1,\ i \in Z,$$

which fully determines the matrices $A_1, \dots, A_n$. So, if $\beta = Mb$, then β is given by the same recursion (9), for $r = 0, \dots, n-2$, while for $r = n-1$ it is determined by the matrix $A_n M^T$. □

For example, using the Bernstein basis, the cubic mononial basis $\beta_3^T = [1 \quad t \quad t^2 \quad t^3]$ can be decompsed as

$$\beta_3^T = \frac{1}{3} [1-t \quad t] \begin{bmatrix} 1-t & t & 0 \\ 0 & 1-t & t \end{bmatrix} \begin{bmatrix} 3 & t & 0 & 0 \\ 3 & t+1 & t & 0 \\ 3 & t+2 & 2t+1 & 3t \end{bmatrix},$$

while if the Lagrange cardinal basis is used, one gets

$$\beta_3^T = \frac{1}{6} [3-t \quad t] \begin{bmatrix} 2-t & t & 0 \\ 0 & 3-t & t-1 \end{bmatrix} \begin{bmatrix} 1 & t & t & t \\ 1 & t & 3t-2 & 7t-6 \\ 1 & t & 5t-6 & 19t-30 \end{bmatrix}.$$

Two more decompositions are

(10) $$\beta_3^T = [1 \quad t] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & t \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & t \end{bmatrix},$$

and

(11) $$\beta_3^T = [1 \quad t] \begin{bmatrix} 1 & t/2 & 0 \\ 0 & 1/2 & t \end{bmatrix} \begin{bmatrix} 1 & t/2 & 0 & 0 \\ 0 & 1/2 & t/2 & 0 \\ 0 & 0 & 1/2 & t \end{bmatrix}.$$

Note that (10) products the Horner algorithm.

This example show that the decomposition is not unique, and there are infinitely many triangular algorithms of the type (7) for evaluating a polynomial curve $p(t)$. Choice of the algorithm is inportant question and depands on the particular application. If for some approximation or interpolation the piecewise polynomial is applied, it is desirable to know the Bézier points of each polynomial segment. The Bézier points are the (vector valued) coeffitients of this polynomial being represented via Bernstein basis. Using the blossoming technique [15], determination of these coeffitients is not complicated. Typicaly, the vector of Bézier points B is connected with the vector of data V through $B = M^T P$, where M is a matrix that incorporates into de Casteljau algorithm so as it gets the form (7). For ex. if one works with the cubic parametric segment $p(t)$, $t \in [0,1]$ given by $p(0), p''(0), p''(1), p(1)$ the 'natural' basis is $[1-t \quad -\frac{1}{6}t(t^2-3t+2) \quad \frac{1}{6}t(t^2-1) \quad t]$. Evaluation of $p(t)$ accomplishes by the algorithm (7) for $n = 3$, where A_1 , A_2 are as in de Casteljau algorithm and A_3 is the matrix derived by multiplication of the de Casteljau matrix of the same format and the matrix M. Other examles from CAGD literature includes Ferfuson curve, Ball cubic curve, non-uniform B-spline segment [13], Timmer curve, '4-point' cubic [17], non-uniform β-spline segment [9], q-spline segment [12], and so on.

It is desirable for a triangular algorithm to preserve some good properties that the classical algorithms possess [1], [6]. In the sequel, the conditions are given for some of these properties to be preserved by the algorithm (7).

Theorem 3. *If the matrix sequence $A_r = [a^r_{ij}]$ in (7) satisfies*

$$\sum_{j=0}^{r} a^r_{ij}(t) = 1, \; t \in I, \; i = 0, \dots, r-1, r = 1, \dots, n \,, \tag{12}$$

then the curve $p(t)$ lies in the affine hull of the data points P. If, in addition

$$a^r_{ij}(t) \geq 0 \,, \quad t \in I, \; i = 0, \dots, r-1, \; j = 0, \dots, r, \; r = 1, \dots, n \,, \tag{13}$$

the curve $p(t)$ lies in the convex hull of P.

Proof. By the definition of the algorithm, if (12) is satisfied, than, for every $t \in I$ $P^r_j(t) \in \text{aff}\{P^{r-1}_j(t), P^{r-1}_{j+1}(t)\}$–the affine hull of the points P^{r-1}_j and P^{r-1}_{j+1}. It follows by induction that $P^0_n(t) = p(t) \in \text{aff}\{P_0, \dots, P_n\}$. □

Corollary 2. *If (12) and (13) is fulfiled, the basic functions b^n_i satisfy*

$$\sum_{i=0}^{n} b^n_i(t) = 1 \,, \tag{14}$$

$$0 \leq b^n_i(t) \leq 1 \,, \quad i = 0, \dots, n \,, \quad t \in I \,. \tag{15}$$

PROOF. The first relation follows from Theorem 3, for the choice $P_i = 1$. Then, $p(t) = P_0^n(t) = 1$ as the convex hull of the sequence $1, \ldots, 1$. Further, if one selects $P_i = \delta_{ij}$, $j = 0, \ldots, n$, then $b_i^n(t) = p(t) \in \mathrm{conv}\{\delta_{i0}, \ldots, \delta_{in}\} = [0, 1]$, so $0 \leq b_i^n(t) \leq 1$, $i = 0, \ldots, 1$. □

The curve $p(t)$, being a span of some basis that satisfy (14) is invariant under translation and rotation of the coordinate system [1]. Also, the property of being in the convex hull of the data points is of great importance for $p(t)$ in applications.

An important subclass of triangular constructive algorithms (7) obtains when the matrices $A_k = [a_{ij}^k]_{0,0}^{k-1,k}$ are restricted to be *one-banded*. A matrix $A = [a_{ij}]_{0,0}^{m,n}$ is one-banded, provided that one of the following two possibilities hold [11]:

i) $m = n$ or $n + 1$ and $a_{ij} = 0$ unless $i = j$ or $j + 1$;

ii) $m = n$ or $n - 1$ and $a_{ij} = 0$ unless $i = j$ or $j - 1$;

For example, all matrics in (10) and (11) are one-banded. The notion of one-banded matrices is important for the question of shape preserving property of the curve p.

The property of preserving the shape of the data polygon (a polygon connecting the data points $[P_0 \ldots P_n]$) , is closely connected with the total poitivity of the matrices A_k that define the triangular algorithms (7). On the other hand this property is very desirable for any approximating or interpolating curve. The following theorem elaborates this question:

Theorem 4. *If the matrices $A_i(t)$, $i = 2, \ldots, n$ in (6) are totally positive for all $t \in I$, and $A_1(t) = [b_0(t) \quad b_1(t)]$, where b_0, b_1 is the sequence of totally positive functions on I, then the algorithm (6) yields the sequence of totally positive functions $b_0^n, \ldots, b_n^n$.*

PROOF. It is known that the product of two totally positive (TP) matrices is again a TP matrix, and the transpose of a TP matrix is again TP. So, if $A_2, \ldots, A_n$ are TP, then its product $A = A_n^T \cdot \ldots \cdot A_2^T$ is again a TP matrix. Note that the matrix A_1 a TP matrix be cause it is a positive one-banded matrix. Also, the sequence $f_0, \ldots, f_k$ of real functions on I is totally positive if for any partition $t_1 < \ldots < t_m$ in I, the collocation matrix $[f_i(t_j)]$ is TP [11]. The collocation matrix for the sequence b_0, b_1 is

$$B^1 = \begin{bmatrix} b_0(t_1) & \ldots & b_0(t_m) \\ b_1(t_1) & \ldots & b_1(t_m) \end{bmatrix}.$$

By the supposition of the theorem, B^1 is a TP matrix. The collocation matrix for the sequence of functions $b_0^n, \ldots, b_n^n$, produced by the algorithm (6) is, then given by

$$\begin{bmatrix} b_0^n(t_1) & \ldots & b_0^n(t_m) \\ \vdots & & \\ b_n^n(t_1) & \ldots & b_n^n(t_m) \end{bmatrix} = A \cdot B^1,$$

i.e. being the product of two TP matrices, it is TP. Cosequently, the sequence of functions $b_0^n, \ldots, b_n^n$ is totally positive. □

Corollary 3. *Under the assumptions of Theorem 4, the algorithm (7) is variation diminishing, i.e., if $v(f)$ is the sign variation of a function f and $V(c_0,\dots,c_n)$ the sign variation of the sequence, then*

$$v(p) = v(c_0 b_0^n + \dots + c_n b_n^n) \le V(c_0,\dots,c_n) .$$

An immediatelly consequence of this Corollary is the shape preserving of the curve $p(t)$. For example, if the control polygon is convex, the curve is also convex.

An important special case of Theorem 4 occurs when the matrices $A_2,\dots,A_n$ are two-banded matrices with nonnegative items. Then, they are TP, and the statement of the theorem is valid. As an illustration we mention de Casteljau scheme, de Boor-Cox-Mansfield algorithm, and the recursion of Barry and Goldman for Pólya cuves [7], [8] and progressive curves [9], the algorithm of Lyche [16] for Chebyshevian B-splines, scheme of Seidel [18]. Actually, two-banded matrices in (7) means a two-term recursion algorithm. On the other hand, the Aitken-Neville scheme does not fulfil the condition of Theorem 4, so the Lagrange basis it produces is not a totally positive sequence of functions.

3. Rational extensions, compositions and blending

Using the theory of projective maps, it is shown by Farin [12], that de Boor algorithm can be modified to produce a rational B-spline curve $p^w = \sum w_i N_i^k(t) / \sum w_i N_i^k(t)$, where N_i^k are B-splines and w_i are the weights. Following the Farin's technique, and using our matrix notation of the algorithm (7), the rational extension of the curve p can be introduced. Actually, let $[w_0 \dots w_r] = [w_0^0 \dots w_r^0]$ be the vector of prescribed weights, associated with the data $P = [P_0 \dots P_n]$. For a given matrix sequence $A_1,\dots,A_n$, application of the algorithm (7) on w^0, will product the triangular array $[w_{ij}^r]$ by

$$\text{(16)} \qquad \begin{aligned} w^0 &= [w_0^0 \dots w_r^0]^T , \\ w^{r+1}(t) &= A_{n-r}(t) w^r(t), r = 0,\dots,n-1 . \end{aligned}$$

Then, if introduce the new matrix sequence $A_1^w,\dots,A_n^w$, where

$$\text{(17)} \qquad A_k^w = [\frac{w_i^{k-1}}{w_i^k} a_{ij}^k] ,$$

the following theorim is valid

Theorem 5. *The constructive triangular algorithm*

$$P^{r+1}(t) = A_{n-r}^w(t) P^r(t) \; , \; r = 0,\dots,n-1 \; ,$$

where w^r, and A_k^w is given by (16) and (17), defines the rational extension

$$p^w(t) = \sum_{i=0}^{n} w_i b_i^n(t) P_i \Big/ \sum_{i=0}^{n} w_i b_i^n(t) \; , \; t \in I.$$

PROOF. Using the process of inhomogenization of the points produced by the algorithm (7), in the space which dimension is greater than 1 than the dimension of the data (see [12]). □

Note that for w_i =Const., $i = 0, \dots, n$, the rational curve p^w reduces on the non-rational one.

At the end, two operations over algorithms A and B of the form (7) will be discussed. Let two curves is defined by

$$A: \quad p_A = A_1 \cdot \ldots \cdot A_n \cdot P ,$$

$$B: \quad p_B = B_1 \cdot \ldots \cdot B_n \cdot P .$$

The *composition* of A and B is given by

$$p_{AB} = A_1 \cdot \ldots \cdot A_k B_{k+1} \cdot \ldots \cdot B_n \cdot P$$

where $P = [P_0 \dots P_n]$. The composite curve, p_{AB} inherites the properties of both curves: if $k < [n/2]$, the properties of the algorithm B will preveal, while in the opposite case the features of the algorithm A will predominate. The example of such composite algorithm is the triangular scheme, developed in [5] for the class of Catmull-Rom splines.

The algorithms A and B can be blend together. The linear blend of two intervals $I = [a, b]$ and $J = [c, d]$ is the interval

$$K = (1-\alpha)I + \alpha J = [(1-\alpha)a + \alpha c, (1-\alpha)b + \alpha d] .$$

Let two algorithms, A and B be given, and let

$$P_i^{r+1} = [a_{i0}^{n-r} \dots a_{i,n-r}^{n-r}] \cdot P^r \ , t \in I_i^{n-r} \ ,$$

be the $(r+1)$-st stage of A, while

$$Q_i^{r+1} = [b_{i0}^{n-r} \dots b_{i,n-r}^{n-r}] \cdot Q^r \ , \ t \in J_i^{n-r},$$

be the same for B with $Q^0(t) = P$. Let the blend with the coeffitient α of I_i^k and J_i^k be K_i^k. Denote with u and v the local parameters of the intervals I_i^k and J_i^k respectively. Then $w = (1-\lambda_i^k)u + \lambda_i^k v$ will be the local parameter for K_i^k, under condition that

$$\lambda_i^k = \lambda_i^k(\alpha) = \frac{\alpha |J_i^k|}{(1-\alpha)|I_i^k| + \alpha |J_i^k|} \ ,$$

where $|I|$ denotes the lenght of I. Now, the sequence of matrices $C_k(\alpha, t) = [c_{ij}^k]_{0,0}^{k-1,k}$ defined by

$$c_{ij}^k(t) = (1 - \alpha_i^k(\alpha)) a_{ij}^k(t) + \lambda_i^k(\alpha) b_{ij}^k \ ,$$

defines the blended algorithm

$$C: \quad p(\alpha, t) = C_1(\alpha, t) \dots C_n(\alpha, t) \cdot P . \tag{19}$$

For $\alpha = 0$, (19) reduces on the algorithm A, while for $\alpha = 1$ on B. The examle is given in [11], where the blend of A : de Casteljau and B : Neville-Aitken algorithm is studied. For $\alpha = 0$ this blend yields the Bézier curve, while for $\alpha = 1$, the Lagrange interpolant is obtained.

4. Concluding remarks

The general mathematical model for a triangular constructive algorithm is proposed. It is quite general to cover all known triangular schemes and still to flexible to preserve a number of good properties of the classical algorithms (de Casteljau, de Boor and Aitken-Neville). Here, only the main characteristics of proposed algorithm are listed, like the existence, the convex hull property, variation diminution as well as the possibility of rational extensions, composition and blend of two algorithms. Many other properties like end-points interpolation, reparametrization and shape parameters incorporation, has not been discussed for sake of brevity.

However, a number of questions remains open: linear independancy of the functions b_i^n, symmetry, subdivision algorithms, numerical stability and others.

References

1. R. N. Goldman: *Pólya's Urn Model and Computer Aided Geometric Design*, SIAM J. Alg. Disc. Meth. **6** (1985), 1-28.
2. R. N. Goldman: *Urn Models, Approximation, and Splines*, J. Approx. Theory **54** (1988), 1-66.
3. R. N. Goldman: *Urn Models and B-Splines*, Contr. Approx. **4** (1988), 265-288.
4. P. J. Barry: *Urn models, recursive curve schemes, and computer aided geometric design*, Ph.D. dissertation, Dept. of Math., Univ. of Utah, Salt Lake City 1987.
5. P. J. Barry, R. N. Goldman: *A Recursive Evaluation Algorithm for a Class of Catmull-Rom Splines*, Computer Graphics **22** (1988), 199- 204.
6. P. J. Barry, R. N. Goldman: *What is the Natural Generalization of a Bézier Curve?* In: Mathematical Methods in Computer Aided Geometric Design (T. Lyche, L. L. Schumaker eds.) pp. 573-581, Academic Press, Boston 1989.
7. P. J. Barry, R. N. Goldman: *Recursive Polynomial Curve Schemes and Computer-Aided Geometric Design*, Constr. Approx. **6** (1990) , 65-96.
8. P. J. Barry, R. N. Goldman: *Interpolation and Approximation of Curves and Surfaces Using Pólya Polynomials*, to appear in Computer Vision, Graphics and Image Processing.
9. P. J. Barry, R. N. Goldman: *Algorithms for Progressive Curves: Extending B-Spline and Blossoming Techniques to the Monomial, Power, and Newton Dual Bases* (preprint).

10. P. Dierckx, B. Tytgat: *Generating the Bézier points of a β-spline curve*, Comput. Aided Geom. Des.**6** (1989), 279-291.

11. G. Farin, P. J. Barry: *Link between Bézier and Lagrange curve and surface schemes*, Computer Aided Design**18** (1986), no. 10, 525-528.

12. G. Farin: *Rational Curves and Surfaces*, in: Mathematical Methods in Computer Aided Geometric Design (T. Lyche, L. L. Schumaker eds.) pp. 215-238, Academic Press, Boston 1989.

13. T. N. T. Goodman: *Shape Preserving Representations*, in: Mathematical Methods in Computer Aided Geometric Design (T. Lyche, L. L. Schumaker eds.) pp. 333-351, Academic Press, Boston 1989.

14. R. Kulkarni, P-J. Laurent: *Q-Splines*, Numerical Algorithms **1** (1991), 45-74.

15. C-H. Lin: *Generalized Bernstein-Bézier curves and surfaces*, Computer Aided Design**20** (1988), no. 5, 259-262.

16. T. Lyche: *A Recurrence Relation for Chebyshevian B-Splines*, Constr. Approx. **1** (1985), 155-173.

17. L. Ramshaw: *Blossoming: A Connect the Dots Approach to Splines*, Digital Systems Reasearch Center, Report 19, Palo Alto, Ca. 1987.

18. H.-P. Seidel: *Symmetric triangular algorithms for curves*, Comput. Aided Geom. Des.**7** (1990), 57-67.

19. H. G. Timmer: *Alternative representation for parametric cubic curves and surfaces*, Computer Aided Design**12** (1980), no. 1, 25-28. Constr. Approx. **1** (1985), 155-173.