M2AN. MATHEMATICAL MODELLING AND NUMERICAL ANALYSIS
- MODÉLISATION MATHÉMATIQUE ET ANALYSE NUMÉRIQUE

H.-P. SEIDEL

New algorithms and techniques for computing with geometrically continuous spline curves of arbitrary degree

M2AN. Mathematical modelling and numerical analysis - Modélisation mathématique et analyse numérique, tome 26, n° 1 (1992), p. 149-176

<http://www.numdam.org/item?id=M2AN_1992__26_1_149_0>

MATHEMATICAL MODELLING AND NUMERICAL ANALYSIS
MODÉLISATION MATHÉMATIQUE ET ANALYSE NUMÉRIQUE

(Vol 26, n° 1, 1992, p 149 à 176)

NEW ALGORITHMS AND TECHNIQUES FOR COMPUTING WITH GEOMETRICALLY CONTINUOUS SPLINE CURVES OF ARBITRARY DEGREE (*)

by H.-P. SEIDEL (1)

Abstract — *The concept of universal splines provides new techniques for computing with geometrically continuous spline curves of arbitrary degree These techniques lead to new algorithms for computing both the spline control points and the Bezier points, for computing locally supported basis functions, and for knot insertion As a result we obtain a generalization of polar forms to geometrically continuous spline curves The presented algorithms have been coded in Maple and concrete examples illustrate the approach Maple output can be stored in look-up tables and allows the inclusion of geometrically continuous spline curves in interactive applications*

Categories and Subject Descriptors : I.3.5 [Computer Graphics] Computational Geometry and Object Modelling - curve, surface, solid, and object representations

General Terms : Algorithms, Design

Additional Key Words and Phrases : Bézier point, blossom, de Boor algorithm, B-spline, β-spline, connection matrix, control point, geometric continuity, knot insertion, knot vector, osculating flat, polar form, spline control point, universal spline.

Résumé — Nouveaux algorithmes et techniques pour calculer avec des courbes splines géométriquement continues de degrés arbitraires *La notion de Spline universel fournit de nouvelles techniques pour le calcul avec des courbes splines géométriquement continues de degrés arbitraires Ces techniques mènent à de nouveaux algorithmes pour calculer les points de contrôle du spline et ceux de Bézier, les fonctions de bases à support local et les insertions de nœuds Comme conséquence nous obtenons une généralisation des formes polaires aux courbes splines à continuité géométrique Les algorithmes exposés ont été codés en « Maple » et des exemples concrets illustrent la démarche Les sorties Maple peuvent être rangées dans des tableaux de recherche et permettent l'insertion de spline géométriquement continues dans des applications interactives*

(*) This work has been partly supported by the Natural Sciences and Engineering Research Council of Canada through Strategic Operating Grant STR0040527

(1) Computer Graphics Laboratory, Department of Computer Science University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada

1. INTRODUCTION

During the last decade, geometrically continuous spline curves have received considerable attention among the graphics and CAGD communities. However, up to now, algorithms for manipulating geometrically continuous spline curves of arbitrary degree have not been available. While it is known that B-spline-like basis functions exist for geometrically continuous spline curves of arbitrary degree [27], [37], this lack of algorithms has so far prevented their use in practical applications. In particular, algorithms for constructing the Bézier points from the given spline control points, and algorithms for knot insertion, have been missing.

Recently, such algorithms have been developed in [62]. The development is based on the new concept of universal splines and yields geometric constructions for both the spline control points and the Bézier points, as well as new algorithms for constructing locally supported basis functions and for knot insertion. As a result of this development one obtains a generalization of the polar form of a *B*-spline to geometrically continuous spline curves.

This paper reviews the techniques and algorithms given in [62] and augments the presentation in [62] by a more detailed discussion of some implementational issues in computing the Bézier points of a geometrically continuous spline curve from the given control points. The paper is organized as follows : Section 2 gives a brief introduction to geometric continuity and sets up our notation. Section 3 introduces the concept of universal splines, which is essential for the constructions that follow. Section 4 presents a geometric construction for the spline control points of a geometrically continuous spline curve and generalizes the polar form of a *B*-spline to spline curves with geometric continuity. Section 5 shows how to compute the Bézier points of a geometrically continuous spline from the given control points and how to compute locally supported basis functions. Section 6 presents an algorithm for knot insertion and generalizes the de Boor algorithm for the evaluation of a *B*-spline to geometrically continuous spline curves. Section 7 discusses some details of our Maple implementation. Section 8 contains concluding remarks and points out directions for further research.

2. GEOMETRIC CONTINUITY

Consider a strictly increasing sequence $\xi = (x_j)_{j=0}^{\ell+1}$ of real numbers. A *spline F* of degree n over ξ is a continuous piecewise polynomial of degree n on the interval $[x_0, x_{\ell+1}]$ with breakpoints x_j such that the derivatives from the left and the derivatives from the right at x_j are related to each other. One way to specify this relationship between right and left derivatives is by means of *connection matrices* : Let :

$$D_+^k F(u) := (F'_+(u), F''_+(u), \ldots, F_+^{(k)}(u))^t \tag{1}$$

and

$$D_-^k F(u) := (F'_-(u), F''_-(u), \ldots, F_-^{(k)}(u))^t \tag{2}$$

be the column vectors that contain the first k derivatives of F from the right and from the left, respectively. The equation

$$D_+^k F_j(x_j) = C_j \,.\, D_-^k F_{j-1}(x_j) \tag{3}$$

sets up a linear connection between the right and left derivatives of the spline curve F at the breakpoints x_j. The $k \times k$-matrix C_j is therefore called a *connection matrix.*

If the connection matrix

$$C = \begin{pmatrix} 1 & & \\ & 1 & \\ & & 1 \\ & \cdots & \end{pmatrix} \tag{4}$$

is the identity matrix then F is parametrically C^k-continuous. If

$$C = \begin{pmatrix} \beta_1 & & \\ \beta_2 & \beta_1^2 & \\ \beta_3 & 3\beta_1\beta_2 & \beta_1^3 \\ & \cdots & \end{pmatrix}, \quad \beta_1 > 0 \tag{5}$$

is a so-called β-matrix [1], [3], [5], [22], [23], [33] then F is geometrically G^k-continuous. In other words : F can be reparametrized to obtain a C^k-parametrization without altering the shape of the curve. If

$$C = \begin{pmatrix} c_{1,1} & & \\ c_{2,1} & c_{1,1}^2 & \\ c_{3,1} & c_{3,2} & c_{1,1}^3 \\ & \cdots & \end{pmatrix}, \quad c_{1,1} > 0 \tag{6}$$

is lower triangular with $c_{i,i} = c_{1,1}^i$ then F is called Frenet-frame or F^k-continuous. Additional information on connection matrices is given in [1], [23], [27], [34], [32], [37], [38], [40], [42], [44], [53]. Following the standard convention [27], [37] we will assume throughout that the lower triangular connection matrices C_j are *nonsingular* and *totally positive,* but otherwise arbitrary.

In order to specify the order of continuity k at the breakpoints, knot multiplicities are introduced: If a breakpoint x_j is listed with multiplicity μ_j we require that the first $(n - \mu_j)$ derivatives of F from the left and from the right are constrained by C_j. In other words: C_j is an $(n - \mu_j) \times (n - \mu_j)$-matrix. The complete sequence of breakpoints, including multiplicities, is called the *knot vector* T. Using the standard convention of $(n + 1)$-fold end knots it is easy to see that the knot vector

$$T = (\underbrace{x_0, \dots, x_0}_{n+1}, \underbrace{x_1, \dots, x_1}_{\mu_1}, \dots, \underbrace{x_\ell, \dots, x_\ell}_{\mu_\ell}, \underbrace{x_{\ell+1}, \dots, x_{\ell+1}}_{n+1}) = (t_i)_{i=0}^{n+m+1} \tag{7}$$

can be indexed by i from 0 to $n + m + 1$ with

$$m := n + \sum_{j=1}^{\ell} \mu_j \,. \tag{8}$$

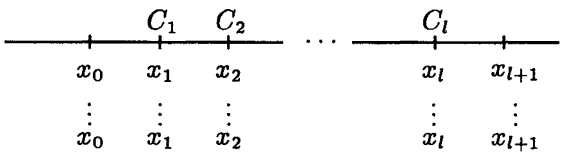

Figure 1. — Breakpoints, knots, and connection matrices.

If a knot vector $T = (t_i)_{i=0}^{m+n+1}$ and a sequence of connection matrices $(C_j)_{j=1}^{\ell}$ at the interior breakpoints $x_1, \dots, x_\ell$ are given (see *fig.* 1) we will denote by

$$\mathcal{S}_d(T, C) = \mathcal{S}_d((t_i)_{i=0}^{m+n+1}, (C_j)_{j=1}^{\ell}) \tag{9}$$

the corresponding space of spline curves in $\mathbb{R}^d$. For $d = 1$ we get e.g. real-valued splines, for $d = 2$ we get splines in the plane, for $d = 3$ we get splines in $\mathbb{R}^3$, etc.

As shown in [27] and [37], the total positivity of the connection matrices $(C_j)_{j=1}^{\ell}$ implies the existence of B-spline-like basis functions $N_i^n(u)$, $i = 0, \dots, m$ satisfying the following properties:

$$N_i^n(u) = 0 \quad \text{for} \quad u \notin (t_i, t_{i+n+1}) \text{ (minimal support)} \tag{10}$$

$$N_i^n(u) > 0 \quad \text{for} \quad u \in (t_i, t_{i+n+1}) \text{ (positivity)} \tag{11}$$

$$\sum_{i=0}^{m} N_i^n(u) = 1 \quad \text{(partition of unity)} \,. \tag{12}$$

Hence, every spline curve $F \in \mathscr{S}_d(T, C)$ has a unique representation

$$F(u) = \sum_{i=0}^{m} N_i^n(u) \cdot d_i \,. \tag{13}$$

The coefficients $d_i \in \mathbb{R}^d$ are called *control points.*

Setting higher coordinates to 0 we obtain a natural inclusion

$$\mathscr{S}_1(T, C) \subset \mathscr{S}_2(T, C) \subset \mathscr{S}_3(T, C) \subset \dots , \tag{14}$$

and we denote by $\mathscr{S}(T, C)$ the union of all these spaces, i.e.

$$\mathscr{S}(T, C) = \bigcup_{d \geq 1} \mathscr{S}_d(T, C) \,. \tag{15}$$

Thus $\mathscr{S}(T, C)$ contains all spline curves over a given knot vector T with a given sequence of connection matrices, no matter what dimension space these curves lie in.

3. UNIVERSAL SPLINES

In this section we show that the study of the whole spline space $\mathscr{S}(T, C)$ can be reduced to the study of a single spline curve $\hat{F}$ in $\mathscr{S}(T, C)$. At first it seems surprising that the study of the infinitely many curves in $\mathscr{S}(T, C)$ can be reduced to the study of the single curve $\hat{F}$. The fundamental insight arises from the observation that $\mathscr{S}(T, C)$ is closed under affine maps : If F is an arbitrary spline curve in $\mathscr{S}(T, C)$, then every image of F under an affine map Φ will again be a spline in $\mathscr{S}(T, C)$.

All we have to do, therefore, is find a spline $\hat{F}$ in $\mathscr{S}(T, C)$ with the property that any other spline F in $\mathscr{S}(T, C)$ is an image of $\hat{F}$ under a unique affine map Φ, i.e. that there exists a unique affine map

$$\Phi : \operatorname{Aff}(\hat{F}) \to \operatorname{Aff}(F) \tag{16}$$

satisfying

$$F(u) = \Phi(\hat{F}(u)) \,, \tag{17}$$

where

$$\operatorname{Aff}(F) = \operatorname{span}\{F(u) \mid u \in [x_0, x_{\ell+1}]\} \tag{18}$$

denotes the affine space that is spanned by the points on the curve

F. Such an $\hat{F}$ is called a *universal spline* for $\mathscr{S}(T, C)$. It will turn out that universal splines always exist and are essentially unique. In order to gain insight into their construction we first look at polynomials :

EXAMPLE 3.1 (Normal Curve) : *Let* $\mathscr{P}^n$ *be the space of all degree n polynomials. We consider the Bézier curve*

$$\hat{F}(u) = \sum_{k=0}^{n} B_k^n(u) \, . \, \hat{b}_k \tag{19}$$

in $\mathbb{R}^{n+1}$ *whose Bézier points* $\hat{b}_k$ *are given by the unit vectors in* $\mathbb{R}^{n+1}$, i.e.

$$\hat{b}_k := (0, \dots, 0, \underset{\substack{| \\ k}}{1}, 0, \dots, 0), \quad k = 0, \dots, n . \tag{20}$$

Then $\hat{F}$ *is a polynomial curve of degree n in* $\mathbb{R}^{n+1}$ *with the property that any other degree n polynomial F is an image of* $\hat{F}$ *under a unique affine map* Φ *: If the Bézier points of F are denoted by* b_k, *the affine map* Φ *:* $\mathrm{Aff}(\hat{F}) \to \mathrm{Aff}(F)$ *is given by*

$$\Phi(\hat{b}_k) = b_k , \quad k = 0, \dots, n . \tag{21}$$

Therefore $\hat{F}$ *is universal for the space* $\mathscr{P}^n$ *of all degree n polynomials.* ■

This observation has been exploited in a different context in [63] for a geometric characterization of cubics in the plane. The following algorithm generalizes the above construction from polynomials to splines :

ALGORITHM 3.2 (Universal spline in Bézier form) : *Given a strictly increasing sequence* $(x_j)_{j=0}^{\ell+1}$ *of breakpoints, a series* $(\mu_j)_{j=1}^{\ell}$ *of multiplicities, the corresponding knot vector*

$$T = (\underbrace{x_0, \dots, x_0}_{n+1}, \underbrace{x_1, \dots, x_1}_{\mu_1}, \dots, \underbrace{x_\ell, \dots, x_\ell}_{\mu_\ell}, \underbrace{x_{\ell+1}, \dots, x_{\ell+1}}_{n+1}) = (t_i)_{i=0}^{n+m+1} \tag{22}$$

with $m = n + \sum_{j=1}^{\ell} \mu_j$ *as given by* (8), *and a series* $(C_j)_{j=1}^{\ell}$ *of connection matrices, the algorithm sets up the universal spline* $\hat{F}$ *of* $\mathscr{S}(T, C)$ *as a piecewise Bézier curve in* $\mathbb{R}^{m+1}$.

• *Since we have adopted the convention of using* $(n+1)$*-fold end knots* $t_0 = \cdots = t_n$ *and* $t_{m+1} = \cdots = t_{n+m+1}$, *the first non-trivial segment of*

$\hat{F}$ is the n-th segment F_n over $[x_0, x_1] = [t_n, t_{n+1}]$ with Bézier points $\hat{b}_{n,0}, \dots, \hat{b}_{n,n}$. Motivated by the previous example, we set

$$\hat{b}_{n,k} := (0, \dots, 0, \underset{\substack{|\\ k}}{1}, 0, \dots, 0), \quad k = 0, \dots, n . \tag{23}$$

- *Suppose that the segments $\hat{F}_n, \hat{F}_{n+\mu_1}, \dots, \hat{F}_{n+\sum_{j=1}^{h-1}\mu_j}$ have already been constructed. Then the Bézier points $\hat{b}_{n+\sum_{j=1}^{h}\mu_j, 0}, \dots, \hat{b}_{n+\sum_{j=1}^{h}\mu_j, n}$ of $\hat{F}_{n+\sum_{j=1}^{h}\mu_j}$ over the next interval $[x_h, x_{h+1}]$ are defined as follows :*
  - *For $k = 0, \dots, n-\mu_h$ the points $\hat{b}_{n+\sum_{j=1}^{h}\mu_j, k}$ are defined in such a way that the derivatives $\hat{F}^{(k)}_{n+\sum_{j=1}^{h-1}\mu_j}(x_h)$ and $\hat{F}^{(k)}_{n+\sum_{j=1}^{h}\mu_j}(x_h)$ satisfy equation* (3) *for $1 \leqslant k \leqslant n-\mu_h$.*
  - *For $k = n-\mu_h+1, \dots, n$ the points $\hat{b}_{n+\sum_{j=1}^{h}\mu_{j,k}}$ are defined as*

$$\hat{b}_{n+\sum_{j=1}^{h}\mu_j, k} := (0, \dots, 0, \underset{\substack{|\\ \sum_{j=1}^{h}\mu_j + k}}{1}, 0, \dots, 0) . \quad \blacksquare \tag{24}$$

Two examples should clarify this approach :

EXAMPLE 3.3 (C^2-continuous cubics) : *We consider a cubic spline with breakpoints*

$$x_0 = 0, x_1 = 1, x_2 = 2, x_3 = 4, x_4 = 5, x_5 = 6 \tag{25}$$

of multiplicity

$$\mu_0 = 4, \mu_1 = \mu_2 = \mu_3 = \mu_4 = 1, \mu_5 = 4 \tag{26}$$

such that the corresponding knot vector T is given by

$$T = (t_i)_{i=0}^{11} = (0, 0, 0, 0, 1, 2, 4, 5, 6, 6, 6, 6) \tag{27}$$

with $n = 3, m = 7$. The $(n-\mu_j) \times (n-\mu_j)$-connection matrices C_j at x_j are given by

$$C_j = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad j = 1, 2, 3, 4 \tag{28}$$

i.e. the resulting spline is parametrically C^2-continuous. The Bézier points $\hat{b}_{3,0}$, $\hat{b}_{3,1}$, $\hat{b}_{3,2}$, $\hat{b}_{3,3}$, $\hat{b}_{4,1}$, $\hat{b}_{4,2}$, $\hat{b}_{4,3}$, $\hat{b}_{5,1}$, $\hat{b}_{5,2}$, $\hat{b}_{5,3}$, $\hat{b}_{6,1}$, $\hat{b}_{6,2}$, $\hat{b}_{6,3}$, $\hat{b}_{7,1}$, $\hat{b}_{7,2}$, $\hat{b}_{7,3}$ of the universal spline $\hat{F}$ in Bézier representation are given by the rows of the following table :

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 2 & 0 & 0 & 0 & 0 \\
0 & 1 & -4 & 4 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & -2 & 8 & -8 & 3 & 0 & 0 & 0 \\
0 & -12 & 44 & -40 & 9 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 6 & -22 & 20 & -9/2 & 3/2 & 0 & 0 \\
0 & 35/2 & -64 & 58 & -51/4 & 9/4 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & -35/2 & 64 & -58 & 51/4 & -9/4 & 2 & 0 \\
0 & -64 & 234 & -212 & 93/2 & -15/2 & -4 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} . \tag{29}$$

For the Bézier points $\hat{b}_{5,1}$ and $\hat{b}_{5,2}$, e.g., condition (3) *translates into*

$$\begin{aligned} \hat{b}_{5,1} &= 3 \cdot \hat{b}_{4,3} - 2 \cdot \hat{b}_{4,2} \\ &= (0, -2, 8, -8, 3, 0, 0, 0) \in \mathbb{R}^8 \end{aligned} \tag{30}$$

and

$$\begin{aligned} \hat{b}_{5,2} &= 9 \cdot \hat{b}_{4,3} - 12 \cdot \hat{b}_{4,2} + 4 \cdot \hat{b}_{4,1} \\ &= (0, -12, 44, -40, 9, 0, 0, 0) \in \mathbb{R}^8 \end{aligned} \tag{31}$$

(note that the length Δ_5 of the interval from t_5 to t_6 is given by $\Delta_5 = t_6 - t_5 = 2$), while the Bézier point $\hat{b}_{5,3}$ is unconstrained and is defined as

$$\hat{b}_{5,3} = (0, 0, 0, 0, 0, 1, 0, 0) \in \mathbb{R}^8 . \quad \blacksquare \tag{32}$$

EXAMPLE 3.4 (G^2-continuous cubic) : *Again we consider a cubic spline with breakpoints*

$$x_0 = 0, x_1 = 1, x_2 = 2, x_3 = 4, x_4 = 5, x_5 = 6 \tag{33}$$

of multiplicity

$$\mu_0 = 4, \mu_1 = \mu_2 = \mu_3 = \mu_4 = 1, \mu_5 = 4 \tag{34}$$

such that the corresponding knot vector T *is given by*

$$T = (t_i)_{i=0}^{11} = (0, 0, 0, 0, 1, 2, 4, 5, 6, 6, 6, 6) \tag{35}$$

with $n = 3$, $m = 7$. *The* $(n - \mu_j) \times (n - \mu_j)$*-connection matrices* C_j *at* x_j *are given by*

$$C_j = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad j = 1, 3, 4, \tag{36}$$

and

$$C_2 = \begin{pmatrix} 1 & 0 \\ 20 & 1 \end{pmatrix}, \tag{37}$$

i.e. these splines are geometrically G^2*-continuous. The Bézier points* $\hat{b}_{3,0}$, $\hat{b}_{3,1}$, $\hat{b}_{3,2}$, $\hat{b}_{3,3}$, $\hat{b}_{4,1}$, $\hat{b}_{4,2}$, $\hat{b}_{4,3}$, $\hat{b}_{5,1}$, $\hat{b}_{5,2}$, $\hat{b}_{5,3}$, $\hat{b}_{6,1}$, $\hat{b}_{6,2}$, $\hat{b}_{6,3}$, $\hat{b}_{7,1}$, $\hat{b}_{7,2}$, $\hat{b}_{7,3}$, *of the universal spline* $\hat{F}$ *in Bézier representation are given by the rows of the following table :*

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 2 & 0 & 0 & 0 & 0 \\
0 & 1 & -4 & 4 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & -2 & 8 & -8 & 3 & 0 & 0 & 0 \\
0 & -52 & 204 & -200 & 49 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 26 & -102 & 100 & -49/2 & 3/2 & 0 & 0 \\
0 & 155/2 & -304 & 298 & -291/4 & 9/4 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & -155/2 & 304 & -298 & 291/4 & -9/4 & 2 & 0 \\
0 & -284 & 1114 & -1092 & 533/2 & -15/2 & 4 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}. \tag{38}$$

In this example the Bézier point $\hat{b}_{5,1}$ *is again given as*

$$\begin{aligned} \hat{b}_{5,1} &= 3 \cdot \hat{b}_{4,3} - 2 \cdot \hat{b}_{4,2} \\ &= (0, -2, 8, -8, 3, 0, 0, 0) \in \mathbb{R}^8 \end{aligned} \tag{39}$$

while $\hat{b}_{5,2}$ is defined as

$$\begin{aligned} \hat{b}_{5,2} &= 49 \,.\, \hat{b}_{4,3} - 52 \,.\, \hat{b}_{4,2} + 4 \,.\, \hat{b}_{4,1} \\ &= (0, -52, 204, -200, 49, 0, 0, 0) \in \mathbb{R}^8 . \end{aligned} \tag{40}$$

The Bézier point $\hat{b}_{5,3}$ is again unconstrained and is set to

$$\hat{b}_{5,3} = (0, 0, 0, 0, 0, 1, 0, 0) \in \mathbb{R}^8 . \quad \blacksquare \tag{41}$$

Algorithm 4.2 constructs a universal spline $\hat{F}$ in Bézier form. There are other ways to construct universal splines. It is easily shown that F is universal for $\mathcal{S}(T, C)$ iff dim Aff $(F) = m$. Moreover, any two universal splines $\hat{F}_1$ and $\hat{F}_2$ are equivalent in the following sense : there exists a unique affine map

$$\Phi_{1,2} : \text{Aff } (\hat{F}_1) \to \text{Aff } (\hat{F}_2) \tag{42}$$

that is $1 - 1$ and onto and that maps the curve $\hat{F}_1$ onto the curve $\hat{F}_2$.

It should be clear from the preceding discussion that properties of the members of a spline space $\mathcal{S}(T, C)$ that are invariant under affine transformations can be detected simply by looking at the universal spline $\hat{F}$ for $\mathcal{S}(T, C)$. Hence a spline $F \in \mathcal{S}(T, C)$ will e.g. satisfy the convex hull and/or variation diminishing property iff this property is satisfied by the universal spline $\hat{F}$ for $\mathcal{S}(T, C)$. This is rather straightforward.

4. A GEOMETRIC CONSTRUCTION FOR SPLINE CONTROL POINTS

As mentioned in Section 2, the total positivity of the connection matrices implies the existence of B-spline-like basis functions $N_i^n(u)$ such that every spline curve $F \in \mathcal{S}(T, C)$ has a unique representation

$$F(u) = \sum_{i=0}^{m} N_i^n(u) \,.\, d_i \,. \tag{43}$$

where the coefficients $d_i \in \mathbb{R}^d$ are the control points. In this section we use universal splines to construct these control points of a geometrically continuous spline curve by intersecting osculating flats. This construction does not work for arbitrary spline curves since arbitrary spline curves may be degenerate. However, this construction is always guaranteed to work for universal splines. Since any spline $F \in \mathcal{S}(T, C)$ is the image of a universal spline $\hat{F}$ for $\mathcal{S}(T, C)$ under a unique affine map Φ, this construction can be used to construct the control points of a given geometrically continuous spline curve of any degree.

We start with the definition of osculating flats : let F be a differentiable curve in $\mathbb{R}^d$. The first k derivative vectors $F'(u)$, $F''(u)$, ..., $F^{(k)}(u)$ span a linear subspace $T_u^k F$ of $\mathbb{R}^d$. Its translate

$$\mathrm{Osc}_k F(u) := F(u) + T_u^k F \tag{44}$$

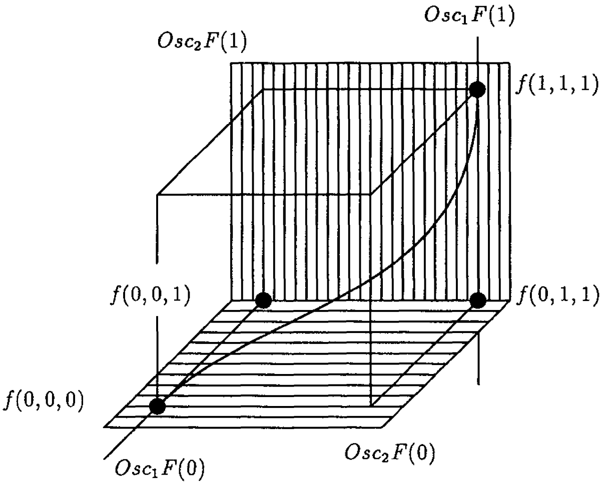

Figure 2. — Osculating flats and Bézier points for a cubic Bézier curve.

is an affine subspace of $\mathbb{R}^d$ and is called the k-th *osculating flat* of F at u. Similarly, if F is differentiable from the left or right, then the first k left, respectively right, derivatives span a linear subspace $T_u^k F_-$, respectively $T_u^k F_+$, and its translate

$$\mathrm{Osc}_k F_-(u) := F(u) + T_u^k F_- , \tag{45}$$

respectively

$$\mathrm{Osc}_k F_+(u) := F(u) + T_u^k F_+ , \tag{46}$$

is again an affine subspace of $\mathbb{R}^d$.

If F is a polynomial its osculating flats can easily be represented in terms of its Bézier points : in fact, the k-th osculating flat of F at 0, respectively at

1, is simply the affine space spanned by its first, respectively last, $k+1$ Bézier points, i.e.

$$\operatorname{Osc}_k F(0) = \left\{ \sum_{i=0}^{k} \alpha_i b_i \,\middle|\, \sum_i \alpha_i = 1 \right\} \tag{47}$$

and

$$\operatorname{Osc}_k F(1) = \left\{ \sum_{i=n-k}^{n} \alpha_i b_i \,\middle|\, \sum_i \alpha_i = 1 \right\}. \tag{48}$$

It has been observed e.g. in [55] that the Bézier points of a non-degenerate polynomial $\hat{F}$ as in Example 3.1 can be constructed by intersecting osculating flats (see *fig.* 2). In fact, in this situation the k-th Bézier point b_k is given as

$$b_k = \operatorname{Osc}_k \hat{F}(0) \cap \operatorname{Osc}_{n-k} \hat{F}(1). \tag{49}$$

More generally : let

$$(\underbrace{u_1, \ldots, u_1}_{\mu_1}, \ldots, \underbrace{u_h, \ldots, u_h}_{\mu_h}) \quad \text{with} \quad \mu_1 + \cdots + \mu_h = n \tag{50}$$

be a sequence of real numbers. Then for a non-degenerate polynomial $\hat{F}$ as in Example 4.1 the expression

$$\hat{f}(\underbrace{u_1, \ldots, u_1}_{\mu_1}, \ldots, \underbrace{u_h, \ldots, u_h}_{\mu_h}) := \bigcap_{i=1}^{h} \operatorname{Osc}_{n-\mu_i} \hat{F}(u_i) \tag{51}$$

is always well defined. It follows immediately from this definition that the map $\hat{f}$ is symmetric, and that $\hat{f}$ satisfies $\hat{f}(u, \ldots, u) = \hat{F}(u)$. In addition, it can be shown that $\hat{f}$ is affine in every argument. Therefore $\hat{f}$ is the *polar form* or *blossom* of $\hat{F}$ [55].

Unfortunately, these definitions break down if the polynomial F is degenerate in the sense that its Bézier points are affinely dependent. In fact, the above construction even fails for a simple degree 3 polynomial in the plane, since all its osculating planes are equal and hence do not intersect properly. Therefore this method of intersecting osculating flats is rejected in [55] for the study of splines.

It turns out, however, that the above construction will always work for universal splines. In fact, universal splines have been set up in exactly such a way as to guarantee that osculating k-flats intersect properly. We are therefore able to construct the control points of a universal spline $\hat{F}$ simply by intersecting osculating flats. If the Bézier representation of

$\hat{F}$ is known this amounts to nothing more than solving a system of linear equations where the coefficients are given by the Bézier points. Details of our Maple implementation are given in Section 7. The main results are summarized in the following theorem :

THEOREM 4.1 : *Let $\hat{F}$ be a universal spline of degree n for a spline space $\mathcal{S}(T, C)$. Then the following holds :*

- *Consider a subsequence* $(t^*_{i+1}, \ldots, t^*_{i+n}) = (\underbrace{u_1, \ldots, u_1}_{\mu_1}, \ldots, \underbrace{u_h, \ldots, u_h}_{\mu_h})$ *of an arbitrary knot vector refinement T^* of T and define*

$$O_{n-\mu_j} \hat{F}(u_j) = \begin{cases} \mathrm{Osc}_{n-\mu_1} \hat{F}_+(u_1) & if \quad j = 1 \\ \mathrm{Osc}_{n-\mu_j} \hat{F}(u_j) & if \quad 2 \leqslant j \leqslant h-1 \\ \mathrm{Osc}_{n-\mu_h} \hat{F}_+(u_h) & if \quad j = h\,. \end{cases} \tag{52}$$

Then the expression

$$\hat{f}(\underbrace{u_1, \ldots, u_1}_{\mu_1}, \ldots, \underbrace{u_h, \ldots, u_h}_{\mu_h}) :=$$

$$:= \bigcap_{j=1}^{h} O_{n-\mu_j} \hat{F}(u_j) \quad with \quad \mu_1 + \cdots + \mu_h = n \tag{53}$$

is well-defined.

- *The resulting map $\hat{f}$ is symmetric in its n arguments and satisfies*

$$\hat{f}(u, \ldots, u) = \hat{F}(u)\,. \tag{54}$$

- *If $\hat{F}$ is parametrically $C^{n-\mu_i}$-continuous at the knots t_i then $\hat{f}$ is multiaffine* [56], [60], *and hence is the polar form of $\hat{F}$. For arbitrary connection matrices $\hat{f}$ is multirational : More precisely : Given a subsequence $(t^*_{i+1}, \ldots, t^*_{i+n-1})$ of an arbitrary knot vector refinement T^* of T with $t^*_k < t^*_{k+1}$, the points*

$$\hat{f}(t^*_{i+1}, \ldots, t^*_k, u, t^*_{k+1}, \ldots, t^*_{i+n-1}) \tag{55}$$

are collinear, and the expression $\hat{f}(t^*_{i+1}, \ldots, t^*_k, u, t^*_{k+1}, \ldots, t^*_{i+n-1})$ is rational in u for $u \in [t^*_k, t^*_{k+1}]$.

- *The spline control points $\hat{d}_0, \ldots, \hat{d}_m$ of $\hat{F}$ are given by*

$$\hat{d}_i = \hat{f}(t_{i+1}, \ldots, t_{i+n})\,. \tag{56}$$

They are affinely independent and form an affine frame for the m-dimensional affine space $\mathrm{Aff}(\hat{F})$ *as defined by* (18). ■

A full proof of Theorem 4.1 is given in [62]. Instead of repeating this rather technical proof here we will illustrate the workings of Theorem 4.1 by looking at the two concrete examples of the preceding section :

EXAMPLE 4.2 : *We start with Example* 3.3. *The B-spline control points are given as follows :*

$$\begin{aligned}
\hat{d}_0 &= \hat{f}(0,0,0) = \mathrm{Osc}_0\,\hat{F}(0)\\
\hat{d}_1 &= \hat{f}(0,0,1) = \mathrm{Osc}_1\,\hat{F}(0) \cap \mathrm{Osc}_2\,\hat{F}(1)\\
\hat{d}_2 &= \hat{f}(0,1,2) = \mathrm{Osc}_2\,\hat{F}(0) \cap \mathrm{Osc}_2\,\hat{F}(1) \cap \mathrm{Osc}_2\,\hat{F}(2)\\
&\vdots\\
\hat{d}_7 &= \hat{f}(6,6,6) = \mathrm{Osc}_0\,\hat{F}(6)\,.
\end{aligned}$$

Note again that intersecting osculating flats is nothing more than solving a linear system of equations where the coefficients are given by the Bézier points. We illustrate this procedure by explicitly computing the control points $\hat{d}_2 = \hat{f}(0,1,2)$. *Since* $\ldots, 0, 1, 2, \ldots$ *is a subsequence of the original knot vector* T, *the control point* $\hat{d}_2$ *satisfies*

$$\hat{d}_2 = \hat{f}(0,1,2) = \mathrm{Osc}_2\,\hat{F}(0) \cap \mathrm{Osc}_2\,\hat{F}(2)\,, \tag{57}$$

and (47), (48) *yield*

$$\mathrm{Osc}_2\,\hat{F}(0) = \mathrm{span}\,\left\{\hat{b}_{3,0}, \hat{b}_{3,1}, \hat{b}_{3,2}\right\} \tag{58}$$

and

$$\mathrm{Osc}_2\,\hat{F}(2) = \mathrm{span}\,\left\{\hat{b}_{4,1}, \hat{b}_{4,2}, \hat{b}_{4,3}\right\}\,. \tag{59}$$

Therefore $\hat{d}_2$ *satisfies both*

$$\hat{d}_2 = \hat{b}_{3,0}\cdot r_0 + \hat{b}_{3,1}\cdot r_1 + \hat{b}_{3,2}\cdot r_2,\ r_0 + r_1 + r_2 = 1 \tag{60}$$

and

$$\hat{d}_2 = \hat{b}_{4,1}\cdot s_1 + \hat{b}_{4,2}\cdot s_2 + \hat{b}_{4,3}\cdot s_3,\ s_1 + s_2 + s_3 = 1\,. \tag{61}$$

Using the results of Table (29) *together with the condition*

$$r_0 + r_1 + r_2 = 1 \tag{62}$$

this yields the following system of equations :

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 1 & 4 & 0 \\ 0 & 0 & 0 & -2 & -4 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} r_0 \\ r_1 \\ r_2 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} . \tag{63}$$

Solving for r_0, r_1, r_2, s_1, s_2, s_3 *we obtain*

$$r_0 = 0, \quad r_1 = -1, \quad r_2 = 2 \tag{64}$$

and

$$s_1 = 2, \quad s_2 = -1, \quad s_3 = 0 , \tag{65}$$

and $\hat{d}_2$ *is given as*

$$\hat{d}_2 = 0 \cdot b_{3,0} - b_{3,1} + 2 \cdot b_{3,2} \tag{66}$$

$$= (0, -1, 2, 0, 0, 0, 0, 0) \in \mathbb{R}^8 . \tag{67}$$

The results below have been obtained using the linalg *package of* Maple [18]. *The control points* $\hat{d}_0, \ldots, \hat{d}_7$ *are given by the rows of the following table*

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & -10 & 8 & 0 & 0 & 0 & 0 \\ 0 & -17 & 62 & -56 & 12 & 0 & 0 & 0 \\ 0 & 29 & -106 & 96 & -21 & 3 & 0 & 0 \\ 0 & -64 & 234 & -212 & 93/2 & -15/2 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} . \tag{68}$$

Note that it is obvious from this table that the B-spline control points $\hat{d}_0, \ldots, \hat{d}_7$ *of* $\hat{F}$ *are in fact affinely independent.* ■

EXAMPLE 4.3 : *Next we consider the* β*-spline of Example 3.4. As suggested by Theorem* 4.1 *the table of* β*-spline control points below shows the same pattern as the corresponding table in the previous example. Again, the* β*-spline control points* $\hat{d}_0, \ldots, \hat{d}_7$ *are given by the rows of the following table :*

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 29/23 & -110/23 & 104/23 & 0 & 0 & 0 & 0 \\ 0 & -77 & 302 & -296 & 72 & 0 & 0 & 0 \\ 0 & 129 & -506 & 496 & -121 & 3 & 0 & 0 \\ 0 & -284 & 1114 & -1092 & 533/2 & -15/2 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} . \tag{69}$$

■

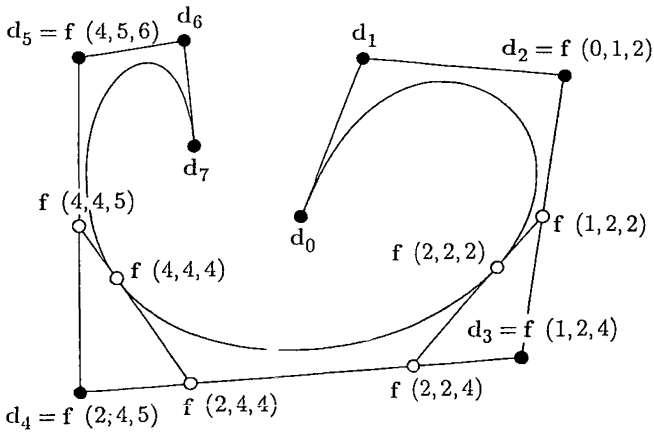

Figure 3. — An affine image F of the universal spline $\hat{F}$ of Example 4.3. Note particularly the spline control points $d_0, \ldots, d_7$ and the Bézier points $b_{5,0} = f(2, 2, 2)$, $b_{5,1} = f(2, 2, 4)$, $b_{5,2} = f(2, 4, 4)$ and $b_{5,3} = f(4, 4, 4)$.

We conclude this section by pointing out that the computations above are *invariant under affine maps* : Given an arbitrary spline $F = \Phi(\hat{F})$ we can therefore compute its spline control points $d_0, \ldots, d_m$ by simply applying the affine map Φ to the control points $\hat{d}_0, \ldots, \hat{d}_m$ of the universal spline $\hat{F}$, i.e.

$$d_i = \Phi(\hat{d}_i) , \quad i = 0, \ldots, m . \tag{70}$$

5. COMPUTING THE BÉZIER POINTS

A suitable method for rendering β-splines is to convert from the β-spline representation to the representation as a piecewise Bézier curve : Once the

Bézier polygons have been constructed, each curve segment can be drawn using Bézier curve algorithms. For cubic β-splines such a construction was first given in [11] (see also [6], [23]). This construction has subsequently been generalized to degree 4 and 5 in [12], [13], [28], [29], [50], [52]. [41] gives a geometric construction for Frenet-frame-continuity of arbitrary degree. However, as pointed out in [6] and [23], « an algorithm for geometrically constructing the Bézier polygons of a G^k β-spline for arbitrary degree and arbitrary shape parameters is currently unknown. »

In this section we develop such an algorithm and give a geometric construction for the Bézier points of a geometrically continuous spline curve from the given control points. The algorithm works for geometrically continuous splines of arbitrary degree with arbitrary shape parameters.

In fact, most of this algorithm has already been developed in the preceding section : Recall from Theorem 4.1 that the spline control points $\hat{d}_i$, $i = 0, \dots, m$ of a universal spline $\hat{F}$ form an affine frame for the m-dimensional affine space Aff $(\hat{F})$. It is therefore possible to represent the Bézier points of a universal spline $\hat{F}$ as affine combinations of the control points. The barycentric coordinates in these affine combinations are obtained by applying to the Bézier points of $\hat{F}$ a simple coordinate transformation M that sends the control points $\hat{d}_i$, $i = 0, \dots, m$ to the unit vectors

$$e_i = (0, \dots, 0, \underset{\substack{| \\ i}}{1}, 0, \dots, 0) \in \mathbb{R}^{m+1}, \quad i = 0, \dots, m, \tag{71}$$

i.e. M is defined by the system of equations

$$M \cdot \hat{d}_i = e_i, \quad i = 0, \dots, m. \tag{72}$$

This leads to the following algorithm :

ALGORITHM 5.1 (Bézier points from control points) : *Set up the universal spline $\hat{F}$ in Bézier form according to Algorithm* 3.2.

- *Compute the spline control points*

$$\hat{d}_i = \hat{f}(t_{i+1}, \dots, t_{i+n}) \tag{73}$$

according to Theorem 4.1.

- *Set up the inverse M^{-1} of the $(m+1) \times (m+1)$-transformation matrix M by taking $\hat{d}_i$ as i-th column of M^{-1}, for $i = 0, \dots, m$.*
- *Compute M as the inverse of M^{-1}.*

• *Apply M to the Bézier points $\hat{b}_{i,k}$, $i = n, \dots, m$, $k = 0, \dots, n$. The resulting coefficients*

$$(\alpha_{i,k}^{0}, \dots, \alpha_{i,k}^{m})^{t} = M \cdot \hat{b}_{i,k} \tag{74}$$

are the barycentric coordinates of

$$\hat{b}_{i,k} = \sum_{j=0}^{m} \alpha_{i,k}^{j} \cdot \hat{d}_j \tag{75}$$

w.r.t. the control points $\hat{d}_0, \dots, \hat{d}_m$. ■

Note that the barycentric coordinates $\alpha_{i,k}^{j}$ are nothing else but the discrete β-splines which correspond to the conversion to Bézier representation by multiple knot insertion. Again, we illustrate the workings of Algorithm 5.1 by means of the following two examples :

EXAMPLE 5.2 : *We start with Example* 3.3. *The* 8×8-*matrix* M^{-1} *is obtained by taking the control points $\hat{d}_i$, $i = 0, \dots, 7$ (= rows in the table of Example* 4.2) *as columns of* M^{-1} *which gives*

$$M^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 3 & -17 & 29 & -64 & 0 \\ 0 & 0 & 2 & -10 & 62 & -106 & 234 & 0 \\ 0 & 0 & 0 & 8 & -56 & 96 & -212 & 0 \\ 0 & 0 & 0 & 0 & 12 & -21 & 93/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 & -15/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{76}$$

Inverting M^{-1} *we get the transformation matrix*

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1/2 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 5/8 & 1/3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/8 & 7/12 & 1/12 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/12 & 7/12 & 1/8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/3 & 5/8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{77}$$

Applying M to the Bézier points $\hat{b}_{i,k}$ (= rows in the table of Example 3.3) *we get the barycentric coordinates of the Bézier points w.r.t. the B-spline control* *points $\hat{d}_0, \ldots, \hat{d}_7$. These barycentric coordinates of $\hat{b}_{3,0}, \ldots, \hat{b}_{7,3}$ are given by the rows of the following table :*

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/4 & 5/8 & 1/8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3/4 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 7/12 & 1/12 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3/4 & 1/4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/4 & 3/4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/12 & 7/12 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/4 & 3/4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/8 & 5/8 & 1/4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{78}$$

The Bézier point $\hat{b}_{5,0}$ e.g. *is given as*

$$\hat{b}_{5,0} = 1/3 \,.\, \hat{d}_2 + 7/12 \,.\, \hat{d}_3 + 1/12 \,.\, \hat{d}_4 \,.$$

Inspection of the above table again verifies that the Bézier polygon is obtained out of the control polygon by a corner cutting process. In particular, this implies the convex hull and variation diminishing property. This is true in general for geometrically continuous spline curves with nonsingular totally positive connection matrices [27], [37]. ■

EXAMPLE 5.3 : *Next we consider the β-spline of Example* 3.4. *The transformation matrix M is given by*

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1/2 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 55/104 & 1/13 & 0 & 0 & 0 \\ 0 & 0 & 0 & 23/104 & 851/936 & 23/216 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/72 & 121/216 & 1/8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/3 & 5/8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{79}$$

and the barycentric coordinates of the Bézier points $\hat{b}_{3,0}, \ldots, \hat{b}_{7,3}$ w.r.t. the β-spline control points $\hat{d}_0, \ldots, \hat{d}_7$ are given by the rows of the following table :

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/4 & 55/104 & 23/104 & 0 & 0 & 0 & 0 \\ 0 & 0 & 29/52 & 23/52 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3/26 & 23/26 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/13 & 851/936 & 1/72 & 0 & 0 & 0 \\ 0 & 0 & 0 & 23/24 & 1/24 & 0 & 0 & 0 \\ 0 & 0 & 0 & 23/72 & 49/72 & 0 & 0 & 0 \\ 0 & 0 & 0 & 23/216 & 121/216 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/4 & 3/4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/8 & 5/8 & 1/4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{80}$$

A comparison of the tables in Example 5.2 *and Example* 5.3 *also illustrates the well-known effect of varying the shape parameter* β_2 *at* x_2 *from* 0 *to* 20 : *First of all, the shape of the curve is only altered* locally, i.e. *Bézier points far out are left unchanged. Secondly, when raising the tension parameter* β_2 *at* x_2, *the nearby Bézier points are moved towards* $\hat{d}_3$ *so that the joint* $F(t_5)$ *is attracted to the control point* $\hat{d}_3$. ■

Note that the output of Algorithm 5.1 is invariant under affine maps. Therefore the given results hold not only for universal splines but for any spline F in $\mathcal{S}(T, C)$. Thus Algorithm 5.1 really provides a complete solution to the above stated problem of constructing the Bézier polygons of a geometrically continuous spline curve of arbitrary degree and arbitrary shape parameters from the spline control polygon.

As a corollary, Algorithm 5.1 also provides an explicit representation of the locally supported basis functions $N_i^n(u)$, $i = 0, \dots, m$ with

$$F(u) = \sum_{i=0}^{m} N_i^n(u) \, . \, d_i \, . \tag{81}$$

In fact, the Bézier ordinates of $N_i^n(u)$ are given by the i-th column of Table (78) and Table (80), respectively :

EXAMPLE 5.4 (Locally supported basis functions) : *The normalized B-spline* $N_4^3(u)$ *of Example* 3.3 *has the Bézier ordinates*

$$\begin{aligned} b_{5,0} &= 1/12, & b_{5,1} &= 1/4, & b_{5,2} &= 3/4, & b_{5,3} &= 7/12 \\ b_{6,0} &= 7/12, & b_{6,1} &= 1/2, & b_{6,2} &= 1/4, & b_{6,3} &= 1/8 \end{aligned} \tag{82}$$

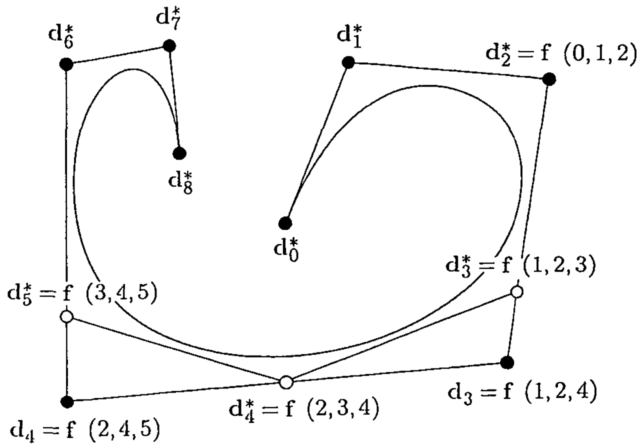

Figure 4. — Inserting the new knot $u = 3$.

while the normalized β-spline $N_4^3(u)$ of Example 3.4 *has the Bézier ordinates*

$$\begin{aligned} b_{5,0} &= 1/72\ , & b_{5,1} &= 1/24, & b_{5,2} &= 49/72, & b_{5,3} &= 121/216 \\ b_{6,0} &= 121/216, & b_{6,1} &= 1/2\ , & b_{6,2} &= 1/4\ , & b_{6,3} &= 1/8\ . \end{aligned} \tag{83}$$

6. KNOT INSERTION

Knot insertion algorithms for cubic β-splines have been given in [11], [24], [25], [48], [49]. [11] uses a geometric construction while [48] and [49] use the theory of discrete β-splines. [13], [29], and [47] extend these results to quartics and quintics. However, a knot insertion algorithm for geometrically continuous spline curves of arbitrary degree has previously been unknown. We will now use the results of the previous sections to give such a knot insertion algorithm for geometrically continuous spline curves of arbitrary degree and arbitrary shape parameters.

In fact, using our results of the preceding sections, knot insertion becomes almost trivial. Suppose that a new knot u with $t_\ell \leq u < t_{\ell+1}$ is to be inserted in the knot vector of a universal spline $\hat{F}$ of arbitrary degree n. Theorem 4.1 tells us that the new control points $\hat{d}_i^*$ are given by intersecting osculating flats at the old knots t_i with certain osculating flats at the new knot u. All we have to do therefore is to determine the osculating flats $\mathrm{Osc}_k\, F_\ell(u)$ at the new knot u. But according to (47), $\mathrm{Osc}_k\, F_\ell(u)$ is given by the Bézier points of F_ℓ w.r.t. the interval $[u, t_{\ell+1}]$. These Bézier points can be obtained from the Bézier points of F_ℓ w.r.t. $[t_\ell, t_{\ell+1}]$ by simple de Casteljau subdivision. This leads to the following algorithm :

ALGORITHM 6.1 (Knot insertion) : *Suppose that the Bézier points of the universal spline* $\hat{F}$ *are expressed in barycentric coordinates w.r.t. the spline control points* $\hat{d}_0, \ldots, \hat{d}_m$ *according to Algorithm* 5.1. *In order to insert a new knot* u *with* $t_\ell \leq u < t_{\ell+1}$ *do the following* :

- *Use de Casteljau subdivision to subdivide* F_ℓ *at* u *and to compute the Bézier points* $b^*_{\ell,0}, \ldots, b^*_{\ell,n}$ *of* F_ℓ *w.r.t.* $[t_\ell, u]$ *and the Bézier points* $b^*_{\ell+1,0}, \ldots, b^*_{\ell+1,n}$ *of* F_ℓ *w.r.t.* $[u, t_{\ell+1}]$.
- *Compute the new control points* d^*_i *over the refined knot vector*

$$T^* = (t_0, \ldots, t_\ell, u, t_{\ell+1}, \ldots, t_{n+m+1}) \tag{84}$$

according to Theorem 4.1 *as*

$$\hat{d}^*_i = \hat{f}(t^*_{i+1}, \ldots, t^*_{i+n}) . \quad \blacksquare \tag{85}$$

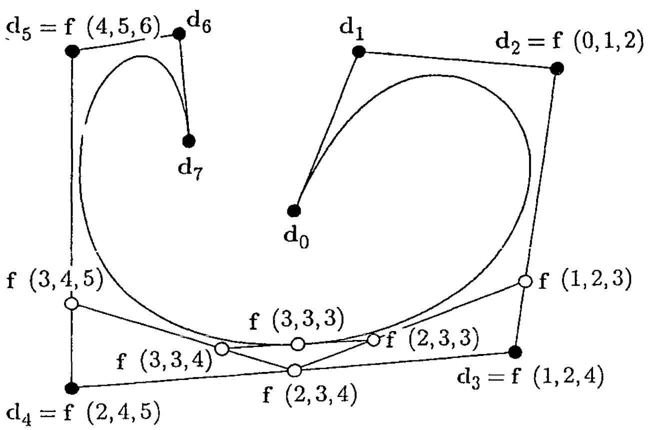

Figure 5. — Multiple knot insertion and evaluation at $u = 3$.

Theorem 4.1 implies that for $\ell - n + 1 \leq i \leq \ell$ the new control points $\hat{d}^*_i$ are of the form

$$\hat{d}^*_i = a_i(u) \cdot d_i + (1 - a_i(u)) \cdot d_{i-1} \tag{86}$$

where $a_i(u)$ is a rational function in u. Note that for $n = 3$ the knot insertion algorithm in [11] is a special case of Algorithm 6.1. If $\hat{F}$ is parametrically C^k-continuous, the functions $a_i(u)$ become affine in u, and Algorithm 6.1 reduces to the insertion algorithm [9] for B-splines.

Similar to our results in the previous sections, the output of Algorithm 6.1 is again affinely invariant. Therefore the results of Algorithm 6.1 not only hold for universal splines, but carry over to any spline F in $\mathcal{S}(T, C)$. In other words: Algorithm 6.1 is in fact a knot insertion algorithm for any given geometrically continuous spline curve of arbitrary degree and arbitrary shape parameters.

We conclude this section by mentioning that multiple knot insertion yields an evaluation algorithm for geometrically continuous spline curves of arbitrary degree. This follows from the fact that successive knot insertion will eventually compute the expression $f(u, ..., u)$ which is equal to the function value $F(u)$. In the case of parametrically C^k-continuous splines the resulting evaluation algorithm reduces to the well-known de Boor algorithm for the evaluation of a B-spline curve. For geometrically continuous spline curves of arbitrary degree this algorithm is new.

7. MAPLE IMPLEMENTATION

Our algorithms for computing with geometrically continuous spline curves of arbitrary degree have been implemented using Maple, a symbolic computation system designed and implemented at the University of Waterloo. In this section we briefly discuss the data structures and the general setup of our program. We focus on our algorithm for computing the Bézier points of a geometrically continuous spline curve from the given control points.

The Bézier points $b_{i,0}, ..., b_{i,n}$ of the i-th segment $F_i = F|_{[t_i, t_{i+1}]}$ depend linearly on the control points $d_{i-n}, \dots, d_i$, i.e. we have

$$\begin{pmatrix} b_{i,0} \\ \vdots \\ b_{i,n} \end{pmatrix} = A_i \cdot \begin{pmatrix} d_{i-n} \\ \vdots \\ d_i \end{pmatrix} \tag{87}$$

where the $(n+1) \times (n+1)$-matrix A_i depends on the knots $t_{i-n+1}, ..., t_{i+n}$, on their multiplicities, and on the connection matrices at these knots. In order to compute A_i our program starts out by setting up the universal spline $\hat{F}_{[i-n+1, i+n]}$ over the interval $[t_{i-n+1}, t_{i+n}]$ in Bézier representation. The computation of this universal spline only depends on the knot multiplicities in the subsequence

$$(t_{i-n+1}, ..., t_i, t_{i+1}, ..., t_{i+n}) . \tag{88}$$

For $n = 3$ e.g. the possible knot configurations of the left knots $t_{i-2}, ..., t_i$ are given by the sequences

$$(t_{i-2} = t_{i-1} = t_i),\ (t_{i-2} < t_{i-1} = t_i),\ (t_{i-2} = t_{i-1} < t_i), \text{ and}$$
$$(t_{i-2} < t_{i-1} < t_i) , \tag{89}$$

i.e. there are $2^{n-1} = 4$ different configurations. Combining the possible configurations to the left and to the right we see that there are $2^{2n-2} = (2^{n-1})^2$ different knot configurations totally. These different configurations are indexed by the knot multiplicities. For $n = 3$ e.g. the index (2, 1 ; 3) corresponds to the knot sequence

$$(t_{i-2} = t_{i-1} < t_i < t_{i+1} = t_{i+2} = t_{i+3}) . \tag{90}$$

Our program sets up the universal spline $\hat{F}_{[i-n+1, i+n]}$ separately for each configuration. Using Algorithm 5.1 the program then computes an explicit expression for the matrix A_i that only depends on the sequence of knot multiplicities and the entries of the connection matrices. These results are then stored in a look-up table that is indexed by the possible sequences of knot multiplicities as explained above. Using the conversion routines from Maple to C that are provided by Maple 4.4, and using *awk*, these look-up tables are then converted to C^{++} code. Since evaluation of the matrices A_i in the look-up tables only requires multiplications and divisions, the evaluation of these matrices is then fast enough to be used for interactive applications written in C^{++}.

8. CONCLUSION

This paper has introduced the concept of universal splines and shown how universal splines lead to new algorithms and techniques for computing with geometrically continuous spline curves of arbitrary degree. Using a projectively invariant formulation of the continuity conditions between adjacent segments it is possible to extend the concept of universal splines from affine to projective invariance. Preliminary investigation suggests that it will hence be possible to generalize the constructions of this paper from polynomial to rational splines. Other objects of study include triangular patch surfaces and algebraic curves. It will be interesting to determine whether the techniques of this paper can be further extended to handle these geometric objects as well.

REFERENCES

[1] B. A. BARSKY, The Beta-spline : a local representation based on shape parameters and fundamental geometric measures, PhD Dissertation, Univ. of Utah, Salt Lake City, USA, 1981.

[2] B. A. BARSKY and J. C. BEATTY, Local control of bias and tension in Beta-splines, *ACM Trans. Graph.* 2, 109-134, 1983.

[3] B. A. BARSKY, *Computer Graphics and Geometric Modelling Using Beta-splines,* Springer, 1988.

[4] B. A. BARSKY, Introducing the rational Beta-spline, Proc. 3rd Int. Conf. Eng. Graphics Descr. Geometry, Vienna, 1988.

[5] B. A. BARSKY and T. D. DEROSE, Geometric continuity of parametric curves : Three equivalent characterizations, *IEEE Comput. Graph. Appl.* 9(5), 60-68, 1989.

[6] B. A. BARSKY and T. D. DEROSE, Geometric continuity of parametric curves : Constructions of geometrically continuous splines, *IEEE Comput. Graph. Appl.* 60-68, 1990.

[7] R. H. BARTELS and J. C. BEATTY, Beta-splines with a difference, Technical Report CS-83-40, Dept. of Computer Science, Univ. of Waterloo, 1983.

[8] R. H. BARTELS, J. C. BEATTY and B. A. BARSKY, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling,* Morgan Kaufmann Publishers, 1987.

[9] W. BOEHM, Inserting new knots into a *B*-spline curve, *Comput. Aided Design,* 12, 50-62, 1980.

[10] W. BOEHM, G. FARIN and J. KAHMANN, A survey of curve and surface methods in CAGD, *Comput. Aided Geom. Design* 1, 1-60, 1984.

[11] W. BOEHM, Curvature continuous curves and surfaces, *Comput. Aided Geom. Design* 2, 313-323, 1985.

[12] W. BOEHM, Smooth curves and surfaces, in : Farin, G. (ed.), *Geometric Modeling, Algorithms and New Trends,* SIAM, 1987.

[13] W. BOEHM, Rational geometric splines, *Comput. Aided Geom. Design* 4, 67-77, 1987.

[14] C. DE BOOR, On calculating with *B*-splines, J. Approx. Theory 6, 50-62, 1972.

[15] C. DE BOOR, *A Practical Guide to Splines,* Springer, New York, 1978.

[16] P. DE CASTELJAU, *Formes à pôles,* Hermes, Paris, 1985.

[17] P. DE CASTELJAU, *Shape Mathematics and CAD,* Kogan Page Ltd, London, 1986.

[18] B. W. CHAR *et al., Maple Reference Manual,* 5th ed., Watcom Publ. Ltd, Waterloo, 1988.

[19] E. COHEN, T. LYCHE and R. F. RIESENFELD, Discrete *B*-splines and subdivision techniques in computer aided geometric design and computer graphics, *Comput. Graph. Image Process.* 14, 87-111, 1980.

[20] E. COHEN, A new local basis for designing with tensioned splines, *ACM Trans. Graph.* 6(2), 81-122, 1987.

[21] H. S. M. COXETER, *Introduction to Geometry,* Wiley, New York, 1961.

[22] T. D. DEROSE, Geometric continuity : a parametrization independent measure of continuity for computer aided geometric design, PhD Dissertation, UC Berkeley, Berkeley, U.S.A., 1985.

[23] T. D. DEROSE and B. A. BARSKY, Geometric continuity, shape parameters, and geometric constructions for Catmull-Rom splines, *ACM Trans. Graph.* 7, 1-41, 1988.

[24] P. DIERCKX and B. TYTGAT, Inserting new knots into Beta-spline curves, in :

Lyche, T and Schumaker, L L (eds), *Mathematical Methods in Computer Aided Geometric Design* 195-206, Academic Press, 1989

[25] P DIERCKX and B TYTGAT, Generating the Bezier points of a β-spline curve, *Comput Aided Geom Design* 6, 279-291, 1989

[26] N DYN, A EDELMANN and C A MICCHELLI, A locally supported basis function for the representation of geometrically continuous curves, Analysis 7, 313-341, 1987

[27] N DYN and C A MICCHELLI, Piecewise polynomial spaces and geometric continuity of curves, IBM Res Rep Mathematical Sciences Dept , IBM T J Watson Research Center, Yorktown Heights, N Y , 1985

[28] M ECK and D LASSER, B-spline-Bezier representation of geometric spline curves, Preprint 1254, FB Mathematik, TH Darmstadt, 1989

[29] M ECK, Algorithms for geometric spline curves, Preprint 1309, FB Mathematik, TH Darmstadt, 1990

[30] G E FARIN, Visually C^2-cubic splines, *Comput Aided Design* 14, 137-139, 1982

[31] G E FARIN, Some remarks on V^2-splines, *Comput Aided Geom Design* 2, 325-328, 1985

[32] G E FARIN, *Curves and Surfaces for Computer Aided Geometric Design,* Academic Press, 1988

[33] G GEISE, Uber beruhrende Kegelschnitte einer ebenen Kurve, *Z Angew Math Mech* 42(7/8), 297-304, 1962

[34] R N GOLDMAN and C A MICCHELLI, Algebraic aspects of geometric continuity, in Lyche, T and Schumaker, L L (eds), *Mathematical Methods in Computer Aided Geometric Design,* 313-332, Academic Press, 1989

[35] R N GOLDMAN and B A BARSKY, On Beta-continuous functions and their application to the construction of geometrically continuous curves and surfaces, in Lyche, T and Schumaker, L L (eds), *Mathematical Methods in Computer Aided Geometric Design,* 299-312, Academic Press, 1989

[36] R N GOLDMAN, Blossoming and knot insertion algorithms for B-spline curves, to appear in *Comput Aided Geom Design*

[37] T N T GOODMAN, Properties of Beta-splines, *J Approx Theory* 44, 132-153, 1985

[38] T N T GOODMAN and K UNSWORTH, Generation of Beta-spline curves using a recurrence relation, in Earnshaw, R (ed), *Fundamental Algorithms for Computer Graphics,* 325-357, Springer, 1985

[39] T N T GOODMAN and C A MICCHELLI, Corner cutting algorithms for the Bezier representation of free from curves, IBM Research Report RC 12139, IBM T J Watson Research Center, Yorktown Heights, N Y , 1986

[40] T N T GOODMAN and K UNSWORTH, Manipulating shape and producing geometric continuity in Beta-spline curves, *IEEE Comput Graph Appl* 6(2), 50-56, 1986

[41] T N T GOODMAN, Constructing piecewise rational curves with Frenet frame continuity, to appear in *Comput Aided Geom Design*

[42] J. GREGORY, Geometric continuity, in : Lyche, T. and Schumaker, L. L. (eds.), *Mathematical Methods in Computer Aided Geom. Design,* Academic Press, 1989.

[43] H. HAGEN, Geometric spline curves, *Comput. Aided Geom. Design* 2, 223-227, 1985.

[44] M. E. HOHMEYER and B. A. BARSKY, Rational Continuity : Parametric, Geometric, and Frenet Frame Continuity of Rational Curves, *ACM Trans. Graph.* 8(4), 1989.

[45] J. HOSCHEK and D. LASSER, *Grundlagen der geometrischen Datenverarbeitung,* Teubner, 1989.

[46] B. JOE, Rational Beta-spline curves and surfaces and discrete Beta-splines, Technical Report TR 87-04, Dept. of Computing Science, Univ. of Alberta, 1987.

[47] B. JOE, Quartic Beta-splines, Technical Report TR 87-11, Dept. of Computing Science, Univ. of Alberta, 1987.

[48] B. JOE, Discrete Beta-splines, Computer Graphics 21(4) (Proc. SIGGRAPH'87), 137-144, 1987.

[49] B. JOE, Multiple-knot and rational cubic β-splines, *ACM Trans. Graph.* 8(2), 100-120, 1989.

[50] D. LASSER and M. ECK, Bézier representation of geometric spline curves, Technical Report NPS-53-88-004, Naval Postgraduate School, Monterey, 1988.

[51] G. M. NIELSON, Some piecewise polynomial alternatives to splines under tension, in : Barnhill, R. E. and Riesenfeld, R. F. (eds.), *Computer Aided Geometric Design,* Academic Press, 1974.

[52] H. POTTMANN, Curves and tensor product surfaces with third order geometric continuity, Proc. 3rd Int. Conf. Eng. Graphics Descr. Geometry, Vienna, 1988.

[53] H. POTTMANN, Projectively invariant classes of geometric continuity, *Comput. Aided Geom. Design* 6, 307-322, 1989.

[54] H. PRAUTZSCH, A round trip to *B*-splines via de Casteljau, *ACM Trans. Graph.* 8(3), 243-254, 1989.

[55] L. RAMSHAW, BLOSSOMING : A connect-the-dots approach to splines, Digital Systems Research Center, Palo Alto, 1987.

[56] L. RAMSHAW, Béziers and *B*-splines as multiaffine maps, in : *Theoretical Foundations of Computer Graphics and CAD,* 757-776, Springer, 1988.

[57] L. RAMSHAW, Blossoms are polar forms, *Comput. Aided Geom. Design* 6, 323-358, 1989.

[58] L. L. SCHUMAKER, *Spline Functions : Basic Theory,* John Wiley & Sons, New York, 1981.

[59] H.-P. SEIDEL, Knot insertion from a blossoming point of view, *Comput. Aided Geom. Design* 5, 81-86, 1988.

[60] H.-P. SEIDEL, A new multiaffine approach to *B*-splines, *Comput. Aided Geom. Design* 6, 23-32, 1989.

[61] H.-P. SEIDEL, Polynome, Splines und symmetrische rekursive Algorithmen im Computer Aided Geometric Design, Habilitationsschrift, Tübingen, 1989.

[62] H.-P. SEIDEL, Geometric Constructions and Knot Insertion for Geometrically Continuous Spline Curves of Arbitrary Degree, Research Report CS-90-24, Department of Computer Science, University of Waterloo, Waterloo, 1990.

[63] M. C. STONE and T. D. DEROSE, A geometric characterization of parametric cubic curves, *ACM Trans. Graph.* 8, 147-163, 1989.