ARTICLE IN PRESS
COMAID:1489

Computer Aided Geometric Design ••• (••••) •••–•••

Contents lists available at ScienceDirect

Computer Aided Geometric Design

www.elsevier.com/locate/cagd

On a generalization of Bernstein polynomials and Bézier curves based on umbral calculus (II): de Casteljau algorithm ☆

Rudolf Winkel

University of Applied Sciences Bingen, Berlinstr. 109, D-55411 Bingen on the Rhine, Germany

ARTICLE INFO

Article history:
Received 24 August 2014
Received in revised form 27 April 2015
Accepted 28 April 2015
Available online xxxx

Keywords:
Umbral calculus
Bernstein polynomial
Bézier curve
de Casteljau algorithm

ABSTRACT

The investigation of the umbral calculus based generalization of Bernstein polynomials and Bézier curves is continued in this paper: First a generalization of the de Casteljau algorithm that uses umbral shift operators is described. Then it is shown that the quite involved umbral shifts can be replaced by a surprisingly simple recursion which in turn can be understood in geometrical terms as an extension of the de Casteljau interpolation scheme. Namely, instead of using only the control points of level $r-1$ to generate the points on level r as in the ordinary de Casteljau algorithm, one uses also points on level $r-2$ or more previous levels. Thus the unintuitive parameters in the algebraic definition of generalized Bernstein polynomials get geometric meaning. On this basis a new direct method for the design of Bézier curves is described that allows to adapt the control polygon as a whole by moving a point of the associated Bézier curve.

1. Introduction

In Winkel (2014) the properties of generalized Bernstein polynomials and generalized Bézier curves introduced earlier (Winkel, 2001) have been investigated. For the convenience of the reader we repeat here the definition of *generalized Bernstein polynomials* of order n for the sequence of real parameters $\bar{a} = (\bar{a}_1, \ldots, \bar{a}_n)$:

$$B_k^n(t;\bar{a}) = \frac{1}{\rho_n(\bar{a})}\binom{n}{k} p_k(t;\bar{a})p_{n-k}(1-t;\bar{a}), \quad 0 \le k \le n, \tag{1.1}$$

where for any non-negative integer n

$$p_n(t;\bar{a}) = \sum_{k=1}^{n} p_{n,k}t^k \tag{1.2}$$

with the *Bell polynomials* (Comtet, 1974; Roman, 1984)

$$p_{n,k}(\bar{a}) = \frac{1}{k!} \sum_{\substack{i_1+\cdots+i_k=n \\ i_1,\ldots,i_k>0}} \binom{n}{i_1,\ldots,i_k} \bar{a}_{i_1} \cdot \ldots \cdot \bar{a}_{i_k}\,. \tag{1.3}$$

☆ This paper has been recommended for acceptance by Ron Goldman.
E-mail address: winkel@fh-bingen.de.
URL: http://www.fh-bingen.de/lehrende/winkel-rudolf.html.

http://dx.doi.org/10.1016/j.cagd.2015.04.002

The division by $\rho_n(\bar{a}) = p_n(1;\bar{a})$ normalizes the generalized Bernstein polynomials and guarantees the partition of unity property by the generalized binomial formula

$$p_n(x+y;\bar{a}) = \sum_{k=0}^{n} \binom{n}{k} p_k(x;\bar{a}) p_{n-k}(y;\bar{a}) \,. \tag{1.4}$$

Subsequently we often write $p_n(t)$, $p_{n,k}$ and ρ_n instead of $p_n(t;\bar{a})$, $p_{n,k}(\bar{a})$ and $\rho_n(\bar{a})$ if the choice of $\bar{a}$ is clear from the context.

A parameter sequence $\bar{a}$ is *feasible*, if the normalization can be done, i.e., if $\rho_n \neq 0$. To simplify formulas we use subsequently often the *non-normalized generalized Bernstein polynomials*

$$\widetilde{B}^n_k(t;\bar{a}) = \rho_n \, B^n_k(t;\bar{a}) \,. \tag{1.5}$$

Without restriction of generality one can assume $\bar{a}_1 = 1$ in $\bar{a}$, since (by Winkel (2014, Thm. 5.4)) $(\bar{a}_1, \ldots, \bar{a}_n)$ and $(1, \frac{\bar{a}_2}{\bar{a}_1^2}, \ldots, \frac{\bar{a}_n}{\bar{a}_1^n})$ define the same generalized Bernstein polynomials of order n. For the ordinary Bernstein polynomials one has

$$B^n_k(t) = \binom{n}{k} t^k (1-t)^{n-k} = \widetilde{B}^n_k(t;(1,0,\ldots,0)) = B^n_k(t;(1,0,\ldots,0)) \,.$$

In Winkel (2014) it has been shown that the recursion formula for ordinary Bernstein polynomials

$$B^r_k(t) = (1-t) B^{r-1}_k(t) + t B^{r-1}_{k-1}(t)$$

can be generalized to

$$\widetilde{B}^r_k(x,y;\bar{a}) = \theta_{f,y} \widetilde{B}^{r-1}_k(x,y;\bar{a}) + \theta_{f,x} \widetilde{B}^{r-1}_{k-1}(x,y;\bar{a}) \tag{1.6}$$

for the *non-normalized bivariate generalized Bernstein polynomials* of order r with variables (x, y) and parameters $\bar{a}$:

$$\widetilde{B}^n_k(x,y;\bar{a}) = \binom{n}{k} p_k(x;\bar{a}) p_{n-k}(y;\bar{a}) \,. \tag{1.7}$$

Here θ_f is the *umbral shift operator* that generates the sequence of *associated polynomials* $p_n(x)$ according to

$$p_{n+1}(x) = \theta_f \; p_n(x) \;\; \text{for } n \in \mathbb{N}. \tag{1.8}$$

The subscripts x or y indicate that θ_f operates on x or y, respectively, and f corresponds to $\bar{a}$. We will give the complete definition of θ_f in Section 3.

The main purpose of the present paper is a detailed elaboration of the claim of Winkel (2014) that the generalized recursion (1.6) can be used to generalize the ordinary de Casteljau algorithm

$$\begin{aligned} &\mathbf{b}^0_k(t) = \mathbf{b}_k \quad (k = 0, \ldots, n) \\ &\mathbf{b}^r_k(t) = (1-t)\, \mathbf{b}^{r-1}_k(t) + t\, \mathbf{b}^{r-1}_{k+1}(t) \quad (r = 1 \ldots, n; k = 0, \ldots, n-r) \end{aligned}$$

that generates the Bézier curve

$$\mathbf{x}(t;\mathbf{b}) = \sum_{k=0}^{n} \mathbf{b}_k B^n_k(t) = \mathbf{b}^n_0(t) \quad (0 \le t \le 1)$$

by repeated linear interpolation out of the sequence $\mathbf{b} = (\mathbf{b}_0, \ldots, \mathbf{b}_n)$ of control points.

In Section 2 we investigate bivariate Bernstein type polynomials in variables (x, y), and in Section 3 the generalization of the de Casteljau algorithm with umbral shifts. In Section 4 the complicated umbral shifts are replaced by a surprisingly simple recursion that can be understood in geometrical terms as an extension of the de Casteljau interpolation scheme. Namely, instead of using only the control points of level $r-1$ to generate the points on level r as in the ordinary de Casteljau algorithm, one uses two or more or all previous levels down to the control points at level 0. Thus the unintuitive parameters in the algebraic definition of generalized Bernstein polynomials get geometric meaning that is illustrated extensively. In Section 5 a new direct method for the design of Bézier curves is described that uses the additional freedom of the parameters $\bar{a}_2, \bar{a}_3, \ldots$ to adapt the control polygon as a whole by moving a point of the associated Bézier curve.

Many references for the material used in this paper as well as an extensive and inspiring treatment of umbral calculus and CAGD can be found in Roman (1984), Farin (2001), Goldman (2003), respectively.

Please cite this article in press as: Winkel, R. On a generalization of Bernstein polynomials and Bézier curves based on umbral calculus (II): de Casteljau algorithm. Comput. Aided Geom. Des. (2015), http://dx.doi.org/10.1016/j.cagd.2015.04.002

2. Bivariate Bernstein type polynomials

The polynomials $\mathbb{R}_{\le n}[t] = \{p \in \mathbb{R}[t] \mid \deg p \le n\}$ of degree $\le n$ in t and the homogeneous polynomials $\mathbb{R}_n[x, y] = \{p \in \mathbb{R}[x, y] \mid \deg p = n\}$ of degree n in x and y form $(n+1)$-dimensional real vector spaces with respective canonical basis $\{t^k \mid k = 0, \dots, n\}$ and $\{x^k y^{n-k} \mid k = 0, \dots, n\}$. The mapping $t^k \mapsto x^k y^{n-k}$ is an isomorphism between $\mathbb{R}_{\le n}[t]$ and $\mathbb{R}_n[x, y]$. We define column vectors $\mathbf{t} = (1, t, \dots, t^n)^T$ and $\mathbf{h} = (x^0 y^n, \dots, x^n y^0)^T$.

The Bernstein polynomials $B_k^n(t) = \binom{n}{k} t^k (1-t)^{n-k}$ of order n, $0 \le k \le n$, and the bivariate homogeneous Bernstein polynomials $B_k^n(x, y) = \binom{n}{k} x^k y^{n-k}$ are alternative bases for $\mathbb{R}_{\le n}[t]$ and $\mathbb{R}_n[x, y]$, respectively. We define column vectors $\mathbf{B}(t) = (B_0^n(t), \dots, B_n^n(t))^T$ and $\mathbf{B}(x, y) = (B_0^n(x, y), \dots, B_n^n(x, y))^T$.

We call the substitution mapping $\Phi : \mathbb{R}_n[x, y] \to \mathbb{R}_{\le n}[t]$ with $\Phi(p(x, y)) = p(t, 1-t)$ the *canonical parametrization*. Clearly, Φ is a vector space isomorphism that maps the homogeneous monomials $x^k y^{n-k}$ to the basis $t^k (1-t)^{n-k}$ and the homogeneous Bernstein polynomials $B_k^n(x, y)$ to Bernstein polynomials $B_k^n(t)$.

Similarly, the bivariate generalized Bernstein polynomials

$$B_k^n(x, y; \bar{a}) = \frac{1}{\rho_n} \binom{n}{k} p_k(x) p_{n-k}(y)$$

of order n, $0 \le k \le n$, are parametrized to generalized Bernstein polynomials $B_k^n(t; \bar{a})$. We define column vectors $\mathbf{B}(t; \bar{a}) = (B_0^n(t; \bar{a}), \dots, B_n^n(t; \bar{a}))^T$ and $\mathbf{B}(x, y; \bar{a}) = (B_0^n(x, y; \bar{a}), \dots, B_n^n(x, y; \bar{a}))^T$.

In Winkel (2014, Cor. 4.5) it has been shown that

$$\mathbf{B}(t; \bar{a}) = M(\bar{a})\, \mathbf{B}(t)\ ,$$

where

$$M(\bar{a}) = \frac{1}{\rho_n} \widetilde{M}(\bar{a}) = \frac{1}{\rho_n} \left(\widetilde{m}_{k,l}(\bar{a}) \right) \tag{2.1}$$

with

$$\widetilde{m}_{k,l}(\bar{a}) = \frac{\binom{n}{k}}{\binom{n}{l}} \sum_{j=0}^{k} \pi_{k,j}\, \pi_{n-k,n-k-l+j}$$
$$= \frac{\binom{n}{k}}{\binom{n}{l}} \sum_{i=0}^{n-k} \pi_{k,k+i+l-n}\, \pi_{n-k,i}$$

and

$$\pi_{n,k} = \sum_{\nu=0}^{k} \binom{n-\nu}{n-k} p_{n,\nu} \text{ with } p_{n,0} = \delta_{n,0} \text{ (using the Kronecker symbol)}.$$

Clearly, $\mathbf{B}(t; \bar{a})$ is a basis of $\mathbb{R}_{\le n}[t]$, if $M(\bar{a})$ is regular, which is the case for a generic choice of $\bar{a}$.

For the bivariate generalized Bernstein polynomials the analog result is

Theorem 2.1. *Let $\bar{a} = (\bar{a}_1, \dots, \bar{a}_n)$ be a feasible parameter sequence. Then the transformation matrix $M(x, y; \bar{a}) \in \mathbb{R}[x, y]^{(n+1)\times(n+1)}$ in*

$$\mathbf{B}(x, y; \bar{a}) = M(x, y; \bar{a})\, \mathbf{B}(x, y)$$

is given by

$$M(x, y; \bar{a}) = M(\widehat{a})$$

with

$$\widehat{a} = (\widehat{a}_1, \dots, \widehat{a}_n) \quad \text{and} \quad \widehat{a}_k = \frac{\bar{a}_k}{(x+y)^{k-1}}$$

or in non-normalized form by

$$\widetilde{M}(x, y; \bar{a}) = \rho_n M(x, y; \bar{a}) = \widetilde{M}(\widehat{a})\ .$$

Remark 2.2. The result that $M(x, y; \bar{a})$ is just $M(\bar{a})$ with $\bar{a}$ substituted with $\widehat{a}$ shows that the seemingly more general bivariate Bernstein polynomials $B_k^n(x, y; \bar{a})$ or $B_k^n(x, y)$ are in fact nothing more than the usual $B_k^n(t; \bar{a})$ or $B_k^n(t)$ in disguise. Nevertheless, it is sometimes much more convenient to operate with the bivariate form than with the usual, for example when considering the generalized de Casteljau algorithm with umbral shifts.

Please cite this article in press as: Winkel, R. On a generalization of Bernstein polynomials and Bézier curves based on umbral calculus (II): de Casteljau algorithm. Comput. Aided Geom. Des. (2015), http://dx.doi.org/10.1016/j.cagd.2015.04.002

For the proof of Theorem 2.1 we need the following lemma:

Lemma 2.3. *For fixed $n \in \mathbb{N}$ the basis of homogeneous monomials of degree n can be represented in terms of the homogeneous Bernstein basis of degree n by*

$$\mathbf{h} = T(x, y)\,\mathbf{B}(x, y) = \big(t_{kj}(x, y)\big)\ \mathbf{B}(x, y)$$

with

$$t_{kj}(x, y) = \frac{\binom{j}{k}}{\binom{n}{k}(x+y)^{n-k}}\ .$$

Proof. Since $t_{k,j}(x, y) \neq 0$ only if $k \le j \le n$, one computes

$$\begin{aligned}\sum_{j=0}^{n} t_{kj}(x, y) B_j^n(x, y) &= \sum_{j=k}^{n} \frac{\binom{j}{k}}{\binom{n}{k}(x+y)^{n-k}} \binom{n}{j} x^j y^{n-j} \\ &= \frac{1}{(x+y)^{n-k}} \sum_{j=k}^{n} \binom{n-k}{j-k} x^j y^{n-j} \\ &= \frac{1}{(x+y)^{n-k}} \sum_{j=0}^{n-k} \binom{n-k}{j} x^k x^j y^{(n-k)-j} = x^k\ . \qquad \square\end{aligned}$$

Proof of Theorem 2.1. Going through the steps of the proof of Winkel (2014, Thm. 4.4) one sees with the above lemma that the only new thing are the factors $(x+y)^{n-k}$ in the denominators. One gets:

$$\begin{aligned}\widetilde{m}_{k,l}(x, y; \bar{a}) &= \frac{\binom{n}{k}}{\binom{n}{l}} \sum_{j=0}^{k} \widehat{\pi}_{k,j}\ \widehat{\pi}_{n-k,n-k-l+j} \\ &= \frac{\binom{n}{k}}{\binom{n}{l}} \sum_{i=0}^{n-k} \widehat{\pi}_{k,k+i+l-n}\ \widehat{\pi}_{n-k,i}\end{aligned}$$

with

$$\widehat{\pi}_{n,k} = \sum_{\nu=0}^{k} \frac{\binom{n-\nu}{n-k}}{(x+y)^{n-\nu}}\, p_{n,\nu}\ .$$

Now formula (1.3) reveals that the summands $\bar{a}_{i_1} \cdot \ldots \cdot \bar{a}_{i_\nu}$ of $p_{n,\nu}$ all contain exactly ν factors whose indices sum up to n. Therefore

$$\frac{\bar{a}_{i_1} \cdot \ldots \cdot \bar{a}_{i_\nu}}{(x+y)^{n-\nu}} = \frac{\bar{a}_{i_1}}{(x+y)^{i_1-1}} \cdot \ldots \cdot \frac{\bar{a}_{i_\nu}}{(x+y)^{i_1-\nu}}$$

completes the proof. $\square$

Remark 2.4. The definition of the $\widehat{a}_k$ above is not unreasonable, if one observes that setting $\deg(\bar{a}_i) = i-1$ and $\deg(t^k) = k$ (as usual) gives $\deg(p_{n,k}(\bar{a})) = n-k$ such that $p_n(\bar{a})(t)$ becomes a homogeneous polynomial of degree n in the ring $\mathbb{R}[t, \bar{a}_1, \bar{a}_2, \ldots, \bar{a}_n]$. Thus the $\widehat{a}_k$'s have degree zero, since for all $k \in \mathbb{N}$

$$\deg(\widehat{a}_k) = \deg(\bar{a}_k) - \deg((x+y)^{k-1}) = 0\ .$$

3. Generalized de Casteljau algorithm with umbral shifts

The polynomials (1.2) used for the definition (1.1) of generalized Bernstein polynomials are systematically explored in umbral calculus (Roman, 1984). (The bars over the $\bar{a}_k$'s are used to conform with the notations used in umbral calculus.) We present here only the facts that are needed for the understanding of the umbral shift operators and the subsequent calculations in this paper.

Please cite this article in press as: Winkel, R. On a generalization of Bernstein polynomials and Bézier curves based on umbral calculus (II): de Casteljau algorithm. Comput. Aided Geom. Des. (2015), http://dx.doi.org/10.1016/j.cagd.2015.04.002

Let $\mathcal{P} = \mathbb{R}[[t]]$ denote the $\mathbb{R}$-algebra of formal power series in t with coefficients in $\mathbb{R}$ and let $\mathcal{P}^+ = \{f \in \mathbb{R}[[t]] \mid f(0) = 0, f'(0) \neq 0\}$ be the subalgebra of *delta series*. Then for any

$$f(t) = \sum_{n=1}^{\infty} \frac{a_n}{n!} t^n \in \mathcal{P}^+ \tag{3.1}$$

there exists a second delta series

$$\bar{f}(t) = \sum_{n=1}^{\infty} \frac{\bar{a}_n}{n!} t^n \in \mathcal{P}^+ \tag{3.2}$$

which is the compositional inverse of f, i.e., $(\bar{f} \circ f)(t) = (f \circ \bar{f})(t) = t$. Now $\bar{f}$ is the generating function for the *associated polynomials* $p_n(x)$ according to

$$e^{x\bar{f}(t)} = \sum_{n=0}^{\infty} \frac{p_n(x)}{n!} t^n \,. \tag{3.3}$$

Using the notation $p_n(t; \bar{a})$ in (1.2) instead of $p_n(x)$ as in (3.3) emphasizes that the associated polynomials depend on the parameters $\bar{a}$ and with the change from x to t we conform with the usual notation for the independent variable of Bernstein polynomials. The different meanings of x and t in different contexts in this paper should not be an obstacle.

It is shown in umbral calculus that the umbral shift operator θ_f generates the sequence of associated polynomials according to (1.8) if one defines (Roman, 1984, Cor. 3.6.6)

$$\theta_f\, p_n(x) = x\, \bar{f}'(f(t))\; p_n(x) = x\,[f'(t)]^{-1} p_n(x) \tag{3.4}$$

with t acting as linear differentiation operator on a polynomial $p \in \mathbb{R}[x]$ as

$$tp(x) = p'(x) \,.$$

With (1.8) it is not difficult to establish (see Winkel, 2014) the following *down recurrence* for the non-normalized bivariate generalized Bernstein polynomials

$$\widetilde{B}_0^0(x, y; \bar{a}) = 1 \text{ and for } r = 1, \ldots, n \text{ and } k = 0, \ldots, r:$$
$$\widetilde{B}_k^r(x, y; \bar{a}) = \theta_{f,y}\, \widetilde{B}_k^{r-1}(x, y; \bar{a}) + \theta_{f,x}\, \widetilde{B}_{k-1}^{r-1}(x, y; \bar{a}) \,, \tag{3.5}$$

where the operators $\theta_{f,x}$ and $\theta_{f,y}$ act on the variables x and y, respectively.

The canonical parametrization and the normalization by multiplication with ρ_r^{-1} then yields the recurrence for generalized Bernstein polynomials. Subsequently we often write θ_x and θ_y instead of $\theta_{f,x}$ and $\theta_{f,y}$ if f is clear from the context.

Example 3.1. For ordinary Bernstein polynomials with $\bar{a} = (1, 0, 0 \ldots)$ one computes $\bar{f}(t) = t = f(t)$ by (3.1)–(3.2), $p_n(x) = x^n$ by (3.3), $\rho_r = 1$ for all $r \in \mathbb{N}$, and with $\bar{f}'(f(t)) = [f'(t)]^{-1} = 1 = t^0$ (meaning no differentiation) one sees that $\theta_f = x$ is simply multiplication by x. Hence we have confirmed formula (3.2) as $p_{n+1}(x) = x\; p_n(x)$ in the ordinary Bernstein case. Canonical parametrization of (3.5) then gives the well-known recursion formula for Bernstein polynomials: $B_k^r(t) = (1-t)B_k^{r-1}(t) + tB_{k-1}^{r-1}(t)$.

Fig. 1 shows that the down recurrence (3.5) is a *triangular scheme* (Goldman, 2003) that generates the bottom level n of generalized Bernstein polynomials $\widetilde{B}_k^n(x, y; \bar{a})$ out of the single top polynomial $\widetilde{B}_0^0(x, y; \bar{a}) = 1$ at level 0 by increasing the degree r of the polynomials on every new level.

Theorem 3.2. *For fixed $f \in \mathcal{P}^+$, $n \in \mathbb{N}$ and $0 \le k \le n$ one has*

$$B_k^n(x, y; \bar{a}) = \frac{1}{\rho_n} \binom{n}{k} (\theta_{f,x})^k (\theta_{f,y})^{n-k}(1) \text{ with } \rho_n = p_n(x + y). \tag{3.6}$$

Proof. From the down recursion (3.5) one sees that $\widetilde{B}_k^n(x, y; \bar{a})$ is the sum of all differently ordered compositions of k times the operator θ_x and $n - k$ times the operator θ_y applied to 1. Alternatively, it is the sum over all paths that lead from the top of the triangular scheme to $\widetilde{B}_k^n$. Since there are $\binom{n}{k}$ such compositions or paths and since the operators θ_x and θ_y commute, the formula follows for the non-normalized case. But the normalizing factor ρ_n must be $p_n(x + y)$ according to the generalized binomial formula (1.4). □

Please cite this article in press as: Winkel, R. On a generalization of Bernstein polynomials and Bézier curves based on umbral calculus (II): de Casteljau algorithm. Comput. Aided Geom. Des. (2015), http://dx.doi.org/10.1016/j.cagd.2015.04.002

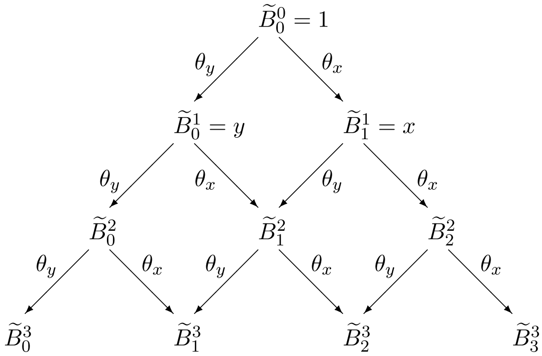

Fig. 1. Down recurrence for $n = 3$, arguments $(x, y; \bar{a})$ omitted.

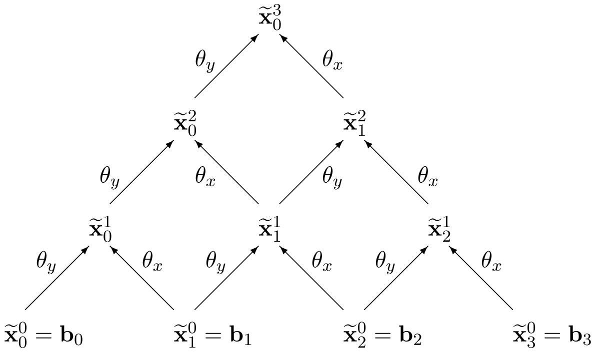

Fig. 2. Up recurrence for $n = 3$, arguments $(x, y; \bar{a}; \mathbf{b})$ omitted.

To the down recurrence there corresponds an *up recurrence* or *generalized de Casteljau algorithm* (see Fig. 2) that generates the generalized Bézier curve of degree n:

$$\mathbf{x}_0^n(t; \bar{a}; \mathbf{b}) = \sum_{k=0}^{n} \mathbf{b}_k B_k^n(t; \bar{a}) \quad (0 \le t \le 1) \tag{3.7}$$

or some point $\mathbf{x}_0^n(t_0; \bar{a}; \mathbf{b})$ with $t_0 \in [0, 1]$ on it from the sequence of control points $\mathbf{b} = (\mathbf{b}_0, \dots, \mathbf{b}_n)$. The control points can be viewed as constant generalized Bézier curves of order 0:

$$\mathbf{x}_k^0(t; \bar{a}; \mathbf{b}) = \mathbf{b}_k B_0^0(t; \bar{a}) = \mathbf{b}_k \quad (0 \le k \le n). \tag{3.8}$$

The bottom layer of this triangular scheme is formed by the $n + 1$ control points and every new layer is given by $n - r$ consecutive generalized Bézier curves of order r or $n - r$ points for fixed $t = t_0$ laying on these intermediary generalized Bézier curves

$$\mathbf{x}_k^r(t; \bar{a}; \mathbf{b}) = \sum_{j=0}^{r} \mathbf{b}_{k+j} B_k^r(t; \bar{a}) \quad (0 \le k \le n - r,\ 0 \le t \le 1). \tag{3.9}$$

Note that (3.7) and (3.8) are special case of (3.9) for $r = n$ and $r = 0$, respectively. The following theorem describes the up recurrence in terms of umbral shifts.

Theorem 3.3. *The* generalized de Casteljau algorithm with umbral shifts *results from normalization and canonical parametrization of the following up recurrence with* $\widetilde{\mathbf{x}}_k^r = \widetilde{\mathbf{x}}_k^r(x, y; \bar{a}; \mathbf{b})$

$$\begin{aligned} \widetilde{\mathbf{x}}_k^0 &= \mathbf{b}_k \quad \textit{for } k = 0, \dots, n, \\ \widetilde{\mathbf{x}}_k^r &= \theta_y \widetilde{\mathbf{x}}_k^{r-1} + \theta_x \widetilde{\mathbf{x}}_{k+1}^{r-1} \quad \textit{for } r = 1, \dots, n;\ k = 0, \dots, n - r \end{aligned} \tag{3.10}$$

where

$$\widetilde{\mathbf{x}}_k^r := \sum_{j=0}^{r} \mathbf{b}_{k+j} \widetilde{B}_j^r(x, y; \bar{a}) \quad (0 \le k \le n - r,\ 0 \le t \le 1). \tag{3.11}$$

Please cite this article in press as: Winkel, R. On a generalization of Bernstein polynomials and Bézier curves based on umbral calculus (II): de Casteljau algorithm. Comput. Aided Geom. Des. (2015), http://dx.doi.org/10.1016/j.cagd.2015.04.002

Proof. The initialization of the $\widetilde{\mathbf{x}}_k^0(x, y; \bar{a}; \mathbf{b})$ as control point $\mathbf{b}_k$ is consistent with (3.8) since $\rho_0 = 1$ for all $\bar{a}$. Assume that (3.10) is true for $r-1$. Then with some index shifting, the linearity of θ_f, and $\widetilde{B}_j^r(x, y; \bar{a}) = 0$ for $j < 0$ or $j > r$:

$$\begin{aligned}
\theta_y \widetilde{\mathbf{x}}_k^{r-1} + \theta_x \widetilde{\mathbf{x}}_{k+1}^{r-1} &\overset{(3.11)}{=} \theta_y \left(\sum_{j=0}^{r-1} \mathbf{b}_{k+j} \widetilde{B}_j^{r-1}(x, y; \bar{a}) \right) + \theta_x \left(\sum_{j=0}^{r-1} \mathbf{b}_{k+1+j} \widetilde{B}_j^{r-1}(x, y; \bar{a}) \right) \\
&= \theta_y \left(\sum_{j=k}^{k+r-1} \mathbf{b}_j \widetilde{B}_{j-k}^{r-1}(x, y; \bar{a}) \right) + \theta_x \left(\sum_{j=k+1}^{k+r} \mathbf{b}_j \widetilde{B}_{j-k-1}^{r-1}(x, y; \bar{a}) \right) \\
&= \theta_y \left(\sum_{j=k}^{k+r} \mathbf{b}_j \widetilde{B}_{j-k}^{r-1}(x, y; \bar{a}) \right) + \theta_x \left(\sum_{j=k}^{k+r} \mathbf{b}_j \widetilde{B}_{j-k-1}^{r-1}(x, y; \bar{a}) \right) \\
&= \sum_{j=0}^{k} \mathbf{b}_{k+j} \left(\theta_y \widetilde{B}_j^{r-1}(x, y; \bar{a}) + \theta_x \widetilde{B}_{j-1}^{r-1}(x, y; \bar{a}) \right) \\
&\overset{(3.5)}{=} \sum_{j=0}^{k} \mathbf{b}_{k+j} \widetilde{B}_j^r(x, y; \bar{a}) \overset{(3.11)}{=} \widetilde{\mathbf{x}}_k^r . \qquad \square
\end{aligned}$$

The appearance of derivations in the form of umbral shift operators in the above formulas seem to preclude a discrete generalized de Casteljau algorithm directly on the control points $\mathbf{b}_k^r$ of different orders $r = 1, \ldots, n$, but since all calculations are done on polynomials of degree less than n, purely algebraic calculations are possible. The next section will show that the elimination of differentiation yields a surprisingly simple and geometrically meaningful generalized de Casteljau algorithm.

4. Generalized de Casteljau algorithm without umbral shifts

Let f and its compositional inverse $\bar{f}$ be defined with coefficients a_n and $\bar{a}_n$, respectively, as in (3.1)–(3.2). The formula (3.4) for the umbral shift operator shows that when parameters $\bar{a}_n$ for the generalized Bernstein polynomials are given, then we need to compute the a_n from them. This can be achieved by one of the formulas for Lagrange inversion, for which the most suitable in our context (Comtet, 1974, 5.8 Thm. E [8f]) is:

$$\begin{aligned}
a_1 &= 1 = \bar{a}_1 \\
a_n &= \sum_{k=1}^{n-1} (-n)_k p_{n-1,k}\left(\frac{\bar{a}_2}{2}, \frac{\bar{a}_3}{3}, \ldots, \frac{\bar{a}_n}{n}\right) \\
&\quad \text{for } n \geq 2 \text{ with } (-n)_k := (-n)(-n-1) \cdot \ldots \cdot (-n-k+1) = (-1)^k \frac{(n+k-1)!}{(n-1)!} , \qquad (4.1)
\end{aligned}$$

because we may have computed the Bell polynomials $p_{n-1,k}(\bar{a}_1, \ldots, \bar{a}_{n-1})$ already. Setting

$$[f'(t)]^{-1} = \sum_{n=0}^{\infty} \frac{d_n}{n!} t^n \qquad (4.2)$$

gives $d_0 = 1$ and the recursion

$$d_n = -\sum_{k=1}^{n} \binom{n}{k} a_{k+1} d_{n-k} \quad \text{for } n \in \mathbb{N} \qquad (4.3)$$

by the convolution of coefficients in

$$1 = f'(t)[f'(t)]^{-1} = \left(\sum_{n \geq 0} \frac{a_{n+1}}{n!} t^n \right) \left(\sum_{n \geq 0} \frac{d_n}{n!} t^n \right) .$$

The first few coefficients are

$$\begin{aligned}
d_1 &= \bar{a}_2 \\
d_2 &= -\bar{a}_2^2 + \bar{a}_3
\end{aligned}$$

Please cite this article in press as: Winkel, R. On a generalization of Bernstein polynomials and Bézier curves based on umbral calculus (II): de Casteljau algorithm. Comput. Aided Geom. Des. (2015), http://dx.doi.org/10.1016/j.cagd.2015.04.002

$$d_3 = 3\bar{a}_2^3 - 4\bar{a}_2\bar{a}_3 + \bar{a}_4$$
$$d_4 = -15\bar{a}_2^4 + 25\bar{a}_3\bar{a}_2^2 - 7\bar{a}_4\bar{a}_2 - 4\bar{a}_3^2 + \bar{a}_5$$
$$d_5 = 105\bar{a}_2^5 - 210\bar{a}_3\bar{a}_2^3 + 60\bar{a}_4\bar{a}_2^2 + 70\bar{a}_2\bar{a}_3^2 - 11\bar{a}_5\bar{a}_2 - 15\bar{a}_3\bar{a}_4 + \bar{a}_6\,.$$

Then

$$\theta_f = x\,[f'(t)]^{-1} = x\sum_{n\geq 0}\frac{d_n}{n!}t^n = x + x\sum_{n\geq 1}\frac{d_n}{n!}t^n =: x + xD_f\,. \tag{4.4}$$

Note, that in D_f one needs only the first m summands if θ_f is applied to a polynomial of degree m.

Remark 4.1. From the definition (1.1)–(1.3) of generalized Bernstein polynomials it is clear that the polynomials of order r in (3.10) can depend only on parameters $\bar{a}_1, \ldots, \bar{a}_r$ and that the parameters $\bar{a}_{r+1}, \ldots, \bar{a}_n$ do not appear up to level r. The same conclusion can be drawn also from (3.6) or the up recursion (3.10), because for the generation of a level r polynomial or curve one needs D_f only up to d_{r-1} and from (4.1) and (4.4) one concludes that a_n and d_{r-1} depend exactly on $\bar{a}_2, \ldots, \bar{a}_r$.

Remark 4.2. Using the representation $\theta_f = x + xD_f$ in formula (3.6) for generalized Bernstein polynomials one can compute a formula that splits every $\widetilde{B}_k^n(x, y; \bar{a})$ into a sum of the ordinary homogeneous Bernstein polynomial $B_k^n(x, y)$ and a rather complicated rest:

$$\begin{aligned}\widetilde{B}_k^n(x, y; \bar{a}) &= \binom{n}{k}(x + xD_{f,x})^k(y + yD_{f,y})^{n-k}(1)\\ &= \binom{n}{k}x^k y^{n-k} + \binom{n}{k}\left[x^k\,\bar{D}_{f,y}^{n-k} + y^{n-k}\,\bar{D}_{f,x}^k + \bar{D}_{f,x}^{n-k}\,\bar{D}_{f,x}^k\right](1)\end{aligned}$$

with

$$\bar{D}_{f,x}^k = \sum_{i=1}^{k}\binom{k}{i}x^{k-i}(xD_{f,x})^i$$
$$\bar{D}_{f,y}^{n-k} = \sum_{j=1}^{n-k}\binom{n-k}{j}y^{n-k-j}(yD_{f,y})^j\,.$$

We know now how to compute the umbral shift operator θ_f as a finite linear differential operator from given $\bar{a}$. Next we take advantage of the fact, that we have to apply it only on very special polynomials that are connected in a seemingly tangled way to the coefficients of the operator.

Theorem 4.3. *For fixed parameter sequence $\bar{a}$ and $n \in \mathbb{N}$ the derivatives of the associated polynomials are*

$$\partial_x p_n(x) = \sum_{k=0}^{n-1}\binom{n}{k}\bar{a}_{n-k}\,p_k(x)\,, \tag{4.5}$$

and for the bivariate and parametrized generalized Bernstein polynomials

$$\partial_x\widetilde{B}_k^n(x, y; \bar{a}) = \sum_{i=1}^{k}\binom{n}{i}\bar{a}_i\,\widetilde{B}_{k-i}^{n-i}(x, y; \bar{a}) \tag{4.6}$$

$$\partial_y\widetilde{B}_k^n(x, y; \bar{a}) = \sum_{j=1}^{n-k}\binom{n}{j}\bar{a}_j\,\widetilde{B}_k^{n-j}(x, y; \bar{a})\,, \tag{4.7}$$

$$d_t\widetilde{B}_k^n(t; \bar{a}) = \sum_{i=1}^{k}\binom{n}{i}\bar{a}_i\,\widetilde{B}_{k-i}^{n-i}(t; \bar{a}) - \sum_{j=1}^{n-k}\binom{n}{j}\bar{a}_j\,\widetilde{B}_k^{n-j}(t; \bar{a})\,. \tag{4.8}$$

Proof. Using Thm. 2.4.9 of Roman (1984) and the umbral calculus definition of the action of a functional t^k applied to a monomial x^n

$$\langle t^k \mid x^n\rangle = n!\delta_{n,k}$$

Please cite this article in press as: Winkel, R. On a generalization of Bernstein polynomials and Bézier curves based on umbral calculus (II): de Casteljau algorithm. Comput. Aided Geom. Des. (2015), http://dx.doi.org/10.1016/j.cagd.2015.04.002

one computes

$$\begin{aligned} p_n'(x) &= \sum_{k=0}^{n-1} \binom{n}{k} \langle t | p_{n-k}(x) \rangle \, p_k(x) \\ &\overset{(1.2)}{=} \sum_{k=0}^{n-1} \binom{n}{k} p_{n-k,1} \, p_k(x) \\ &\overset{(1.3)}{=} \sum_{k=0}^{n-1} \binom{n}{k} \bar{a}_{n-k} \, p_k(x) \, . \end{aligned}$$

Then

$$\begin{aligned} \binom{n}{k} p_k'(x) p_{n-k}(y) &= \binom{n}{k} \sum_{i=0}^{k-1} \binom{k}{i} \bar{a}_{k-i} \, p_i(x) p_{n-k}(y) \\ &= \sum_{i=0}^{k-1} \bar{a}_{k-i} \binom{n}{k} \binom{k}{i} p_i(x) p_{n-k}(y) \\ &= \sum_{i=0}^{k-1} \bar{a}_{k-i} \binom{n}{k-i} \binom{n-k+i}{i} p_i(x) p_{n-k}(y) \\ &= \sum_{i=0}^{k-1} \binom{n}{k-i} \bar{a}_{k-i} \, \widetilde{B}_i^{n-k+i}(x, y; \bar{a}) \, . \end{aligned}$$

Instead of summing over i from 0 to $k-1$ one can sum over $k-i$ from k to 1. This gives (4.6). Using the symmetry $\widetilde{B}_k^n(x, y; \bar{a}) = \widetilde{B}_{n-k}^n(y, x; \bar{a})$ one computes

$$\begin{aligned} \partial_y \widetilde{B}_k^n(x, y; \bar{a}) &= \partial_y \widetilde{B}_{n-k}^n(y, x; \bar{a}) \\ &\overset{(4.6)}{=} \sum_{j=1}^{n-k} \binom{n}{j} \bar{a}_j \, \widetilde{B}_{n-k-j}^{n-j}(y, x; \bar{a}) \\ &= \sum_{j=1}^{n-k} \binom{n}{j} \bar{a}_j \, \widetilde{B}_k^{n-j}(x, y; \bar{a}) \, . \end{aligned}$$

For the last formula one observes that $d_t \widetilde{B}_k^n(t; \bar{a})$ is nothing but the canonical parametrization of $\partial_x \widetilde{B}_k^n(x, y; \bar{a}) - \partial_y \widetilde{B}_k^n(x, y; \bar{a})$. □

The next theorem shows that also l-fold differentiation yields surprisingly simple formulas.

Theorem 4.4. *For fixed $\bar{a}$, $n \in \mathbb{N}$ and $l \in \mathbb{N}_0$ the l-th derivatives for the bivariate generalized Bernstein polynomials are*

$$\partial_x^l \widetilde{B}_k^n(x, y; \bar{a}) = l! \sum_{j=l}^{k} \binom{n}{j} p_{j,l}(\bar{a}) \, \widetilde{B}_{k-j}^{n-j}(x, y; \bar{a}) \tag{4.9}$$

$$\partial_y^l \widetilde{B}_k^n(x, y; \bar{a}) = l! \sum_{j=l}^{n-k} \binom{n}{j} p_{j,l}(\bar{a}) \, \widetilde{B}_k^{n-j}(x, y; \bar{a}) \, . \tag{4.10}$$

Proof. First of all, the formulas are correct for $l = 0$ (no differentiation), because $p_{j,0}(\bar{a}) = \delta_{j,0}$, and also for $l = 1$, because one gets the formulas (4.6)–(4.7) of the previous theorem from $p_{j,1}(\bar{a}) = \bar{a}_j$. For $l = 2$ one computes with the abbreviation $\widetilde{B}_k^n = \widetilde{B}_k^n(x, y; \bar{a})$

$$\partial_x^2 \widetilde{B}_k^n \overset{(4.6)}{=} \sum_{i=1}^{k} \bar{a}_i \binom{n}{i} \partial_x \widetilde{B}_{k-i}^{n-i}$$

$$\overset{(4.6)}{=} \sum_{i=1}^{k} \bar{a}_i \binom{n}{i} \sum_{i'=1}^{k-i} \bar{a}_{i'} \binom{n-i}{i'} \widetilde{B}_{k-i-i'}^{n-i-i'}$$
$$= \sum_{i=1}^{k} \sum_{i'=1}^{k-i} \bar{a}_i \bar{a}_{i'} \binom{n}{i} \binom{n-i}{i'} \widetilde{B}_{k-i-i'}^{n-i-i'}$$

and in general with the abbreviation $j = i_1 + i_2 + \cdots + i_l$

$$\begin{aligned}
\partial_x^l \widetilde{B}_k^n &= \sum_{i-1=1}^{k} \sum_{i_2=1}^{k-i_1} \cdots \sum_{i_l=1}^{k-i_1-i_2-\cdots-i_{l-1}} \bar{a}_{i_1} \bar{a}_{i_2} \ldots \bar{a}_{i_l} \binom{n}{i_1} \binom{n-i_1}{i_2} \cdots \binom{n-i_1-i_2-\cdots-i_{l-1}}{i_l} \widetilde{B}_{k-i_1-i_2-\cdots-i_l}^{n-i_1-i_2-\cdots-i_l} \\
&= \sum_{i-1=1}^{k} \sum_{i_2=1}^{k-i_1} \cdots \sum_{i_l=1}^{k-i_1-i_2-\cdots-i_{l-1}} \frac{n!}{i_1! i_2! \ldots i_l! (n-j)!} \bar{a}_{i_1} \bar{a}_{i_2} \ldots \bar{a}_{i_l} \widetilde{B}_{k-j}^{n-j} \\
&= \sum_{j=l}^{k} \sum_{\substack{i_1+\cdots+i_l=j \\ i_1,\ldots,i_l>0}} \binom{n}{j} \binom{j}{i_1, i_2, \ldots, 1_l} \bar{a}_{i_1} \bar{a}_{i_2} \ldots \bar{a}_{i_l} \widetilde{B}_{k-j}^{n-j} \\
&\overset{(1.3)}{=} \sum_{j=l}^{k} \binom{n}{j} l! \, p_{j,l}(\bar{a}) \widetilde{B}_{k-j}^{n-j} .
\end{aligned}$$

Then

$$\begin{aligned}
\partial_y^l \widetilde{B}_k^n(x, y; \bar{a}) &= \partial_y^l \widetilde{B}_{n-k}^n(y, x; \bar{a}) \\
&\overset{(4.9)}{=} l! \sum_{j=l}^{n-k} \binom{n}{j} p_{j,l}(\bar{a}) \widetilde{B}_{n-k-j}^{n-j}(y, x; \bar{a}) \\
&= l! \sum_{j=l}^{n-k} \binom{n}{j} p_{j,l}(\bar{a}) \widetilde{B}_k^{n-j}(x, y; \bar{a}) . \qquad \square
\end{aligned}$$

Going back to the triangular scheme for the down recurrence (Fig. 1) one observes that partial differentiation by x gives terms laying on the diagonal starting at $\widetilde{B}_k^n$ and going up to the left whereas partial differentiation by y gives terms on the diagonal to the upper right.

For the main theorem about the generalized de Casteljau algorithm below we need a formula that seems not to be mentioned in the umbral calculus literature.

Lemma 4.5. *For fixed $\bar{a}$ and $j \in \mathbb{N}_0$ one has the identity*

$$\sum_{i=0}^{j} d_i \, p_{j,i}(\bar{a}) = \bar{a}_{j+1} . \tag{4.11}$$

Proof. From

$$\sum_{j=0}^{\infty} \frac{\bar{a}_{j+1}}{j!} t^j \overset{(3.2)}{=} \bar{f}'(t) = \bar{f}'(f(\bar{f}(t))) = [f'(\bar{f}(t))]^{-1} \overset{(4.2)}{=} \sum_{i=0}^{\infty} \frac{d_i}{i!} \bar{f}(t)^i$$

one concludes with the formula for conjugate representation for associated sequences (Roman, 1984, Thm. 2.4.4):

$$p_{j,i}(\bar{a}) = \frac{\text{coefficient of } t^j \text{ in } \bar{f}(t)^i}{j!}$$

and $\bar{f}(t)^i = t^i + \mathcal{O}(t^{i+1})$ that

$$\sum_{i=0}^{\infty} \frac{d_i}{i!} \bar{f}(t)^i = \sum_{j=0}^{\infty} \frac{1}{j!} \left(\sum_{i=0}^{j} d_i \, p_{j,i}(\bar{a}) \right) t^j .$$

Comparison of coefficients gives (4.11). $\square$

Please cite this article in press as: Winkel, R. On a generalization of Bernstein polynomials and Bézier curves based on umbral calculus (II): de Casteljau algorithm. Comput. Aided Geom. Des. (2015), http://dx.doi.org/10.1016/j.cagd.2015.04.002

Now the laborious metamorphosis of the generalized de Casteljau algorithm that transforms the caterpillar with umbral shifts into the butterfly without umbral shifts can be completed with the next theorem. We emphasize the difference between functions and their differentiation in Theorem 3.3 and the elementary algebraic operations on points in Theorem 4.6 with the suggestive notations $\mathbf{x}_k^r$ and $\mathbf{b}_k^r$ although the functions become points upon evaluation and the points become functions if x, y, t are taken as variables on a suitable range.

Theorem 4.6. *Given a sequence $\mathbf{b} = (\mathbf{b}_0, \ldots, \mathbf{b}_n)$ of control points in $\mathbb{R}^N$ and a (feasible) parameter sequence $\bar{a}$. Then the* generalized de Casteljau algorithm *for $\widetilde{\mathbf{b}}_k^r = \widetilde{\mathbf{b}}_k^r(x, y; \bar{a}; \mathbf{b})$ is given by*

$$\widetilde{\mathbf{b}}_k^0 = \mathbf{b}_k \text{ for } k = 0, \ldots, n,$$

$$\widetilde{\mathbf{b}}_k^r = y\,\widetilde{\mathbf{b}}_k^{r-1} + x\,\widetilde{\mathbf{b}}_{k+1}^{r-1} + \sum_{i=2}^{r} \binom{r-1}{i-1} \bar{a}_i \left(y\,\widetilde{\mathbf{b}}_k^{r-i} + x\,\widetilde{\mathbf{b}}_{k+i}^{r-i} \right) \quad \text{for } r = 1, \ldots, n \text{ and } k = 0, \ldots, n-r. \tag{4.12}$$

With similar initialization of the recursion and ranges for the indices the normalized recursion for $\mathbf{b}_k^r = \mathbf{b}_k^r(x, y; \bar{a}; \mathbf{b})$ is given by

$$\mathbf{b}_k^r = \frac{1}{\rho_n} \sum_{i=1}^{r} \binom{r-1}{i-1} \rho_{r-i}\, \bar{a}_i \left(y\,\mathbf{b}_k^{r-i} + x\,\mathbf{b}_{k+i}^{r-i} \right) , \tag{4.13}$$

and the parametrized recursion for $\mathbf{b}_k^r = \mathbf{b}_k^r(t; \bar{a}; \mathbf{b})$ by

$$\mathbf{b}_k^r = \frac{1}{\rho_n} \sum_{i=1}^{r} \binom{r-1}{i-1} \rho_{r-i}\, \bar{a}_i \left((1-t)\,\mathbf{b}_k^{r-i} + t\,\mathbf{b}_{k+i}^{r-i} \right) . \tag{4.14}$$

Proof. It is enough to prove the first recursion (4.12).

$$\begin{aligned}
\widetilde{B}_j^r &\overset{(1.6)}{=} \theta_y \widetilde{B}_j^{r-1} + \theta_x \widetilde{B}_{j-1}^{r-1} \\
&\overset{(1.7),(4.2)}{=} y \left(\sum_{l=0}^{r-1-j} \frac{d_l}{l!} \partial_y{}^l \widetilde{B}_j^{r-1} \right) + x \left(\sum_{l=0}^{j-1} \frac{d_l}{l!} \partial_x{}^l \widetilde{B}_{j-1}^{r-1} \right) \\
&\overset{(4.9),(4.10)}{=} y \left(\sum_{l=0}^{r-j-1} d_l \sum_{i=l}^{r-1-j} \binom{r-1}{i} p_{i,l} \widetilde{B}_j^{r-1-i} \right) + x \left(\sum_{l=0}^{j-1} d_l \sum_{i=l}^{j-1} \binom{r-1}{i} p_{i,l} \widetilde{B}_{j-1-i}^{r-1-i} \right) \\
&= y \left(\sum_{i=0}^{r-1-j} \binom{r-1}{i} \left[\sum_{l=0}^{i} d_l p_{i,l} \right] \widetilde{B}_j^{r-1-i} \right) + x \left(\sum_{i=0}^{j-1} \binom{r-1}{i} \left[\sum_{l=0}^{i} d_l p_{i,l} \right] \widetilde{B}_{j-1-i}^{r-1-i} \right) \\
&\overset{(4.11)}{=} y \left(\sum_{i=0}^{r-1-j} \binom{r-1}{i} \bar{a}_{i+1} \widetilde{B}_j^{r-1-i} \right) + x \left(\sum_{i=0}^{j-1} \binom{r-1}{i} \bar{a}_{i+1} \widetilde{B}_{j-1-i}^{r-1-i} \right) .
\end{aligned}$$

This gives with $\widetilde{\mathbf{x}}_k^r$ replaced by $\widetilde{\mathbf{b}}_k^r$:

$$\begin{aligned}
\widetilde{\mathbf{b}}_k^r &\overset{(3.11)}{=} \sum_{j=0}^{r} \mathbf{b}_{k+j} \widetilde{B}_j^r \\
&= y \left(\sum_{j=0}^{r} \mathbf{b}_{k+j} \sum_{i=0}^{r-1-j} \binom{r-1}{i} \bar{a}_{i+1} \widetilde{B}_j^{r-1-i} \right) + x \left(\sum_{j=0}^{r} \mathbf{b}_{k+j} \sum_{i=0}^{j-1} \binom{r-1}{i} \bar{a}_{i+1} \widetilde{B}_{j-1-i}^{r-1-i} \right) \\
&= y \left(\sum_{i=0}^{r-1} \binom{r-1}{i} \bar{a}_{i+1} \sum_{j=0}^{r-1-i} \mathbf{b}_{k+j} \widetilde{B}_j^{r-1-i} \right) + x \left(\sum_{i=0}^{r-1} \binom{r-1}{i} \bar{a}_{i+1} \sum_{j=i+1}^{r} \mathbf{b}_{k+j} \widetilde{B}_{j-1-i}^{r-1-i} \right) \\
&\overset{(3.11)}{=} \sum_{i=0}^{r-1} \binom{r-1}{i} \bar{a}_{i+1} \left(y\,\widetilde{\mathbf{b}}_k^{r-1-i} + x \sum_{j=0}^{r-1-i} \mathbf{b}_{k+j+i+1} \widetilde{B}_j^{r-1-i} \right) \\
&\overset{(3.11)}{=} \sum_{i=0}^{r-1} \binom{r-1}{i} \bar{a}_{i+1} \left(y\,\widetilde{\mathbf{b}}_k^{r-1-i} + x\,\widetilde{\mathbf{b}}_{k+i+1}^{r-1-i} \right) .
\end{aligned}$$

With a final index shift and a splitting of the sum one arrives at (4.12). □

Please cite this article in press as: Winkel, R. On a generalization of Bernstein polynomials and Bézier curves based on umbral calculus (II): de Casteljau algorithm. Comput. Aided Geom. Des. (2015), http://dx.doi.org/10.1016/j.cagd.2015.04.002

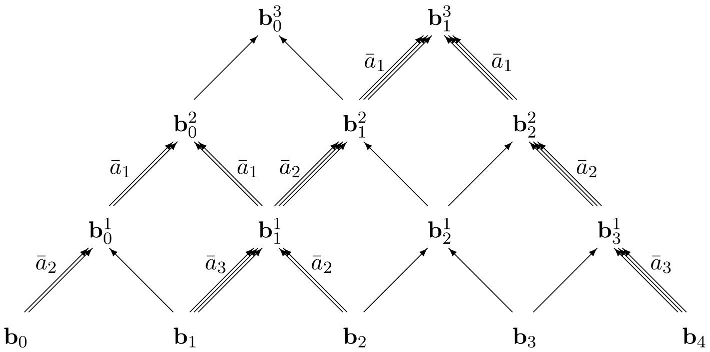

Fig. 3. Generalized de Casteljau algorithm up to level $r = 3$. According to (4.12)–(4.14) the point $\mathbf{b}_0^2$ depends on $\mathbf{b}_0^1$, $\mathbf{b}_1^1$ via $\bar{a}_1$, and on $\mathbf{b}_0$, $\mathbf{b}_2$ via $\bar{a}_2$ (double arrows). The point $\mathbf{b}_1^3$ depends on $\mathbf{b}_1^2$, $\mathbf{b}_2^2$ via $\bar{a}_1$, on $\mathbf{b}_1^1$, $\mathbf{b}_3^1$ via $\bar{a}_2$, and on $\mathbf{b}_1$, $\mathbf{b}_4$ via $\bar{a}_3$ (triple arrows).

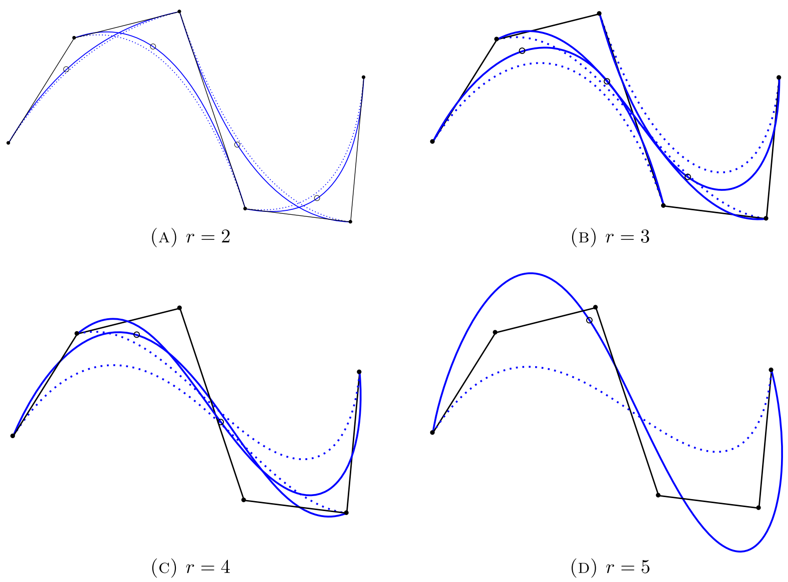

Fig. 4. (Intermediary) generalized Bézier curves $\mathbf{x}_k^r$ for $\bar{a} = (1, -.1, 0, 0, 0)$ with points $\mathbf{b}_k^r$ for $t = .4$, ordinary Bézier curves dotted.

Hence we have discovered a geometrically meaningful interpretation of the parameters $\bar{a}_k$ that was not at all obvious from the original algebraic–combinatorial definition:

For $\bar{a} = (\bar{a}_1, 0, \ldots, 0)$ with $\bar{a}_1 = 1$ one gets the usual de Casteljau algorithm that generates the intermediary control points on level r by linear interpolation of two adjacent points from the previous level $r - 1$. A parameter $\bar{a}_2 \neq 0$ adds an $\bar{a}_2$-weighted linear interpolation of two points separated by one point from level $r - 2$. An $\bar{a}_3 \neq 0$ adds an $\bar{a}_3$-weighted linear interpolation of two points separated by two points from level $r - 3$, and so on.

Fig. 3 illustrates that the points that have an influence on some $\mathbf{b}_j^r$ lay on the left and right diagonal going down from $\mathbf{b}_j^r$.

In principle, the dependence of points on some level on more than just the previous level points can be achieved in many ways — but usually not in a way that doesn't sacrifice important properties such as affine invariance. Probably the only possible way to retain all the properties listed in Winkel (2014) is to use generalized Bézier polynomials as introduced in Winkel (2001). With formula (4.14) it is now easy to deduce an additional property that was hard to see from the algebraic definition (1.1)–(1.3): Though the generalized Bernstein polynomials do not in general respect the convex hull property — they include for example as special case the Lagrange polynomials with equidistant nodes —, they have the property of linear precision by an easy induction using (4.14). In other words, if the basic control points $\mathbf{b}_j^0$ lay on a straight line, then all $\mathbf{b}_j^r$ for all levels r lay on this line.

Figs. 4 and 5 show for two different control polygons how the intermediary and final generalized Bézier curves look like in comparison with the ordinary intermediary and final Bézier curves. (The points on the curves for $t = .4$ are not computed by evaluation but by (4.14)!) In Fig. 4 the parameter sequence uses a small negative $\bar{a}_2$, whereas the higher parameters are zero. This draws the (intermediary) generalized Bernstein polynomials closer to the control polygon. In Fig. 5 the parameter sequence for the interpolation of the control polygon is used as determined in Winkel (2014). The control polygon starts

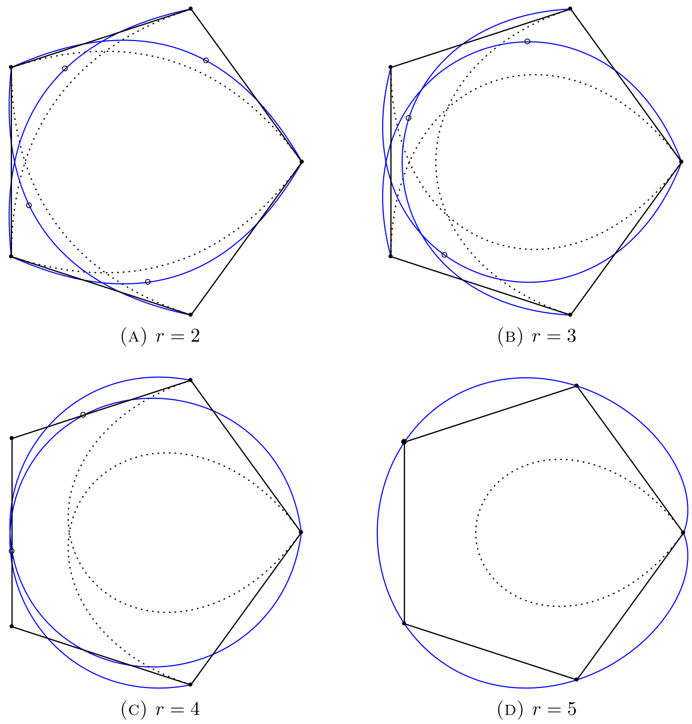

Fig. 5. (Intermediary) generalized Bézier curves $\mathbf{x}_k^r$ for $\bar{a} = (1, -1/5, 2/25, -6/125, 24/625)$ with points $\mathbf{b}_k^r$ for $t = .4$, ordinary Bézier curves dotted.

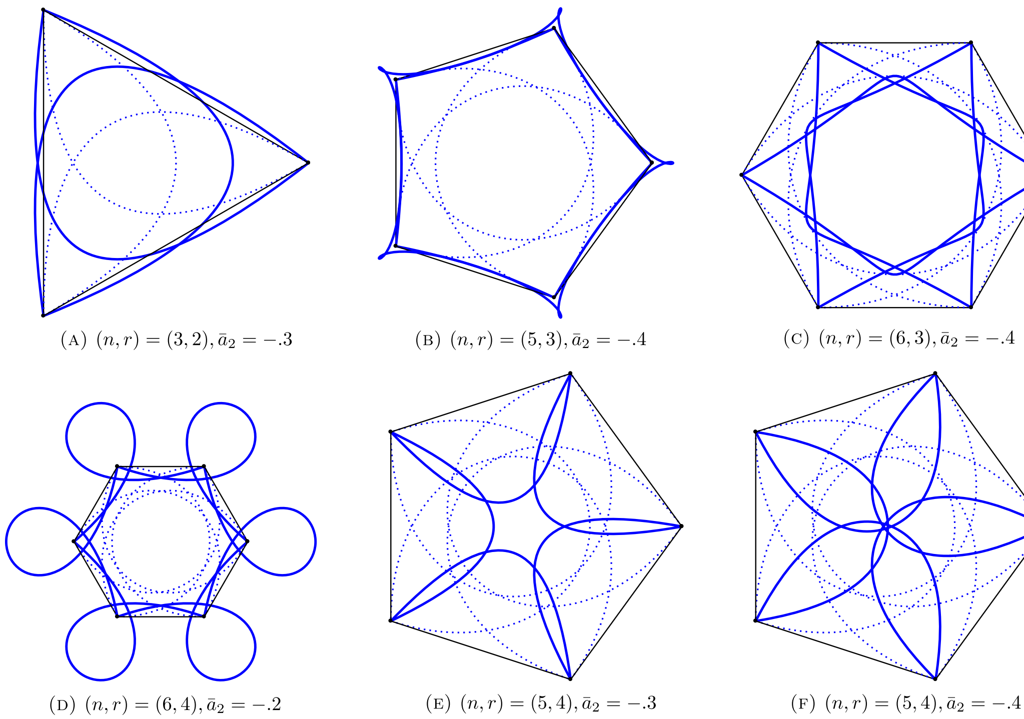

Fig. 6. Intermediary generalized Bézier curves $\mathbf{x}_k^r$ for $\bar{a} = (1, \bar{a}_2, 0, 0, 0)$, ordinary Bézier curves dotted.

and ends at the point most to the right. Note that the intermediary curves are not interpolating, because the parameter sequences for interpolation are different for different numbers of control points.

Fig. 6 shows more intermediary Bézier curves to give an impression how versatile these curves are (and because they look nice). I each case the end of the control polygon overlaps suitably with the beginning to achieve perfect symmetry.

Please cite this article in press as: Winkel, R. On a generalization of Bernstein polynomials and Bézier curves based on umbral calculus (II): de Casteljau algorithm. Comput. Aided Geom. Des. (2015), http://dx.doi.org/10.1016/j.cagd.2015.04.002

Finally in this section we note that from the proof of Theorem 4.6 one gets immediately a down recurrence as (3.5) for bivariate generalized Bernstein polynomials but now without umbral shifts.

Corollary 4.7. *Given a (feasible) parameter sequence* $\bar{a}$*. Then for* $\widetilde{B}^r_k = \widetilde{B}^r_k(x, y; \bar{a})$ *is given by*

$\widetilde{B}^0_0 = 1$ *and for* $r = 1, \dots, n$ *and* $k = 0, \dots, r$:

$$\widetilde{B}^r_k = y \sum_{i=1}^{r-k} \binom{r-1}{i-1} \bar{a}_i \widetilde{B}^{r-i}_k + x \sum_{i=1}^{k} \binom{r-1}{i-1} \bar{a}_i \widetilde{B}^{r-i}_{k-i} . \tag{4.15}$$

With similar initialization of the recursion and ranges for the indices the normalized recursion for $B^r_k = B^r_k(x, y; \bar{a})$ *is given by*

$$B^r_k = \frac{y}{\rho_r} \sum_{i=1}^{r-k} \binom{r-1}{i-1} \rho_{r-i}\, \bar{a}_i B^{r-i}_k + \frac{x}{\rho_r} \sum_{i=1}^{k} \binom{r-1}{i-1} \rho_{r-i}\, \bar{a}_i B^{r-i}_{k-i} . \tag{4.16}$$

5. A new direct method for the design of Bézier curves

Usually the designer of a Bézier curve sets up in an intuitive way a control polygon and inspects the resulting Bézier curve. If the curve is not as desired, then the control points are moved interactively one by one, until the result is satisfying. In the past there have also been invented a number of *direct methods* for curve design, that allow to change the whole Bézier curve by picking a point on the curve and changing its geometric constraints, e.g., its position in ambient space, the tangent direction and magnitude, or the magnitude of the curvature at the point (see for example Bartels and Beatty, 1989; Fowler and Bartels, 1993; Gleicher, 1992). The purpose of the present section is to indicate, how the additional freedom gained by the parameters $\bar{a}_2, \bar{a}_3, \dots$ can be used to extend the known direct methods. Note that by the generalized de Casteljau algorithm of Theorem 4.6 these parameters do not describe local differential–geometric properties, but global properties of the generalized Bézier curves, e.g., an $\bar{a}_2 > 0$ and an $\bar{a}_2 < 0$ leads to a global decrease resp. increase (cf. Fig. 4) of curvature compared to the ordinary Bézier curves. We hope that other researchers can apply this approach to their particular problem.

The new direct method is described and illustrated first from the perspective of the designer, then we discuss the mathematics behind the working steps. (Note, that also a weighting of control points can be done in the usual way in the generalized setup.)

5.1. Working steps — the designer's perspective

(1) At some stage in the design process a control polygon $CP(\mathbf{b})$ for a sequence of control points $\mathbf{b} = (\mathbf{b}_0, \dots \mathbf{b}_n)$ is displayed together with its associated Bézier curve $\mathbf{x}(t; \mathbf{b})$. In addition, a point $S_1 := \mathbf{x}(t_1; \mathbf{b})$ for some $t_1 \in (0, 1)$ is displayed. We call this point the *first shaping point* (Fig. 7(A)). The designer can move S_1 on $\mathbf{x}(t; \mathbf{b})$. As we will see below, it will be favorable to choose a place for S_1 where the Bézier curve substantially deviates from the control polygon. Based on this criterion the software can also propose an initial position of S_1.
(2) The designer can move the shaping point S_1 to a new position $S_{1,new}$ aside $\mathbf{x}(t; \mathbf{b})$. Simultaneously a new Bézier curve $\mathbf{x}_{new}$ through the moved $S_{1,new}$ is displayed. The original curve $\mathbf{x}(t; \mathbf{b})$ can still remain visible for comparison (Fig. 7(B)).
(3) A soon as the new Bézier curve $\mathbf{x}_{new}$ is confirmed by the designer, a new control polygon $CP(\mathbf{b}')$ for a new sequence of control points $\mathbf{b}'$ is displayed such that $\mathbf{x}_{new} = \mathbf{x}(t; \mathbf{b}')$ (Fig. 7(C)).
(4) The design process can be continued again with steps 1–3 as above or with adjustments of single control points in the usual way.

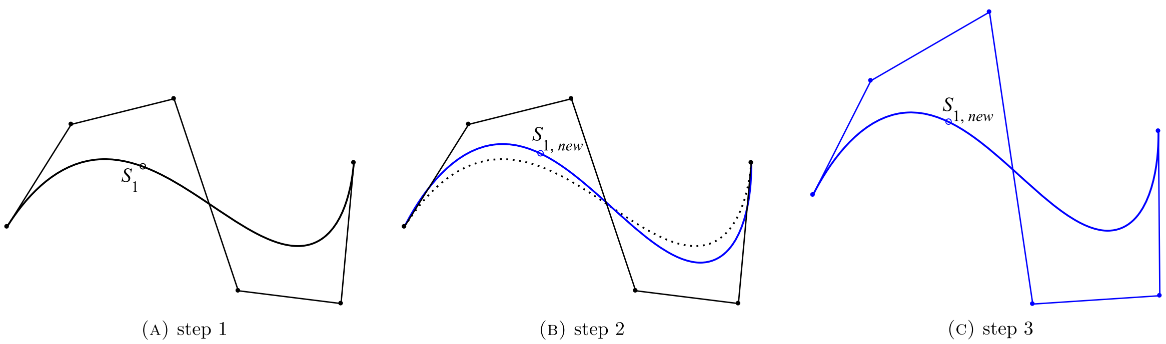

Fig. 7. Illustration of the three steps of the proposed design method.

Please cite this article in press as: Winkel, R. On a generalization of Bernstein polynomials and Bézier curves based on umbral calculus (II): de Casteljau algorithm. Comput. Aided Geom. Des. (2015), http://dx.doi.org/10.1016/j.cagd.2015.04.002

It is also possible to display a second shaping point $S_2 := \mathbf{x}(t_2; \mathbf{b})$ with $t_2 \in (0,1)$, $t_2 \neq t_1$, that allows finer adjustments and more flexibility. Actually, there are up to $n-1$ shaping points for a Bézier curve of degree n possible, but one or at most two should be sufficient in practice.

5.2. *Working steps – the mathematician's perspective*

(1) $\mathbf{x}(t; \mathbf{b})$ is computed from $CP(\mathbf{b})$ in the usual way, e.g., by subdivision, and S_1 is displayed for some predefined t_1, e.g., $t_1 = .4$, or in a more sophisticated fashion, e.g., by inspection of the bending of $CP(\mathbf{b})$.

(2) Let $\mathbf{x}(\bar{a}_2) := \mathbf{x}(t_1; \bar{a}; \mathbf{b})$ for $\bar{a} = (1, \bar{a}_2, 0, \dots, 0)$ be the *curve of variation of* $\bar{a}_2$ *at* t_1 for short the $\bar{a}_2$*-curve* (see Fig. 8). In other words: $\mathbf{x}(\bar{a}_2)$ is computed with the formula for the generalized Bézier curve

$$\mathbf{x}(t; \bar{a}; \mathbf{b}) = \sum_{k=0}^{n} \mathbf{b}_k B_k^n(t; \bar{a}) \quad (0 \le t \le 1) \tag{5.1}$$

but with variable $\bar{a}_2$ and fixed t_1. A suitable domain for $\bar{a}_2$ could be the interval $[r_{\max}(n) + \epsilon, 2]$, where $r_{\max}(n)$ is the largest zero of

$$\rho_n(\bar{a}_2) := p_n(1; (1, a_2, 0, \dots, 0)) = 1 + \binom{n}{2}\bar{a}_2 + \mathcal{O}(\bar{a}_2^2)\,. \tag{5.2}$$

Why is $r_{\max}(n)$ of interest and why is (5.2) true? First of all, note that the zeros of $p_n(1; \bar{a})$ as polynomial in $\bar{a}_2$ are the values of $\bar{a}_2$ where the associated generalized Bernstein polynomials are not defined, or in geometric terms, where the generalized Bernstein polynomials blow up and flip over by passing through infinity. To determine $r_{\max}(n)$ one observes that by (1.3) one has $p_{n,n-1}(\bar{a}) = \bar{a}_1^n$ and

$$p_{n,n-1}(\bar{a}) = \frac{1}{(n-1)!} \cdot (n-1) \cdot \frac{n!}{2!} \bar{a}_1^{n-2} \bar{a}_2 = \binom{n}{2} \bar{a}_1^{n-2} \bar{a}_2\,.$$

Evaluation of $p_n(t; (1, a_2, 0, \dots, 0))$ at $t = 1$ in (1.2) then gives (5.2). Since all terms of $p_n(1; \bar{a})$ are positive, $\rho_n(\bar{a}_2)$ has negative roots only, the maximum one being equal ($n = 2, 3$) or smaller ($n \ge 4$) than

$$\frac{-1}{\binom{n}{2}} = -\frac{2}{n(n-1)}\,.$$

This gives the lower boundary $-.1$ for $\bar{a}_2$ used in Fig. 8. A slightly smaller choice like $-.12$ continues the $\bar{a}_2$-curves vastly beyond the convex hull of the control polygon in the direction of infinity, because for $p_5(1; (1, a_2, 0, 0, 0)) = 15\bar{a}_2^2 + 10\bar{a}_2 + 1$ one has $r_{\max}(n) \approx -.1225$. On the other hand, using 3, 4, 5, or greater numbers as the upper boundary for $\bar{a}_2$ does not lead to a visible continuation of the $\bar{a}_2$-curves. Note further that the $\bar{a}_2$-curves are straight only if the control polygon is highly symmetric as in Fig. 8(A), and that the $\bar{a}_2$-curves need not be perpendicular to the Bézier curve $\mathbf{x}(t; \mathbf{b})$ (Fig. 8(B)).

For the design process in step 2 it would be comfortable, if $S_{1,new}$ would not be restricted to the $\bar{a}_2$-curve at a fixed t_1, but could be moved freely between different $\bar{a}_2$-curves. A remark of caution is necessary here, because in general different $\bar{a}_2$-curves intersect. So the topic of $\bar{a}_2$-curve and the appropriate change between different $\bar{a}_2$-curves needs further investigation.

The Bézier curve $\mathbf{x}_{new}$ that accompanies $S_{1,new}$ on the design display is of course the generalized Bézier curve $\mathbf{x}(t; \bar{a}; \mathbf{b})$ (5.1) for the respective choice of $\bar{a}_2$. As discussed in Winkel (2014) the curve $\mathbf{x}(t; \bar{a}; \mathbf{b})$ can be computed efficiently as ordinary Bézier curve $\mathbf{x}(t; \mathbf{b}')$ for a transformed control polygon $\mathbf{b}' = \mathbf{b}(\bar{a})$: let $C(\mathbf{b})$ and $C(\mathbf{b}')$ be the matrices with control points $\mathbf{b}_0, \dots \mathbf{b}_n$ and $\mathbf{b}'_0, \dots \mathbf{b}'_n$, respectively, as column vectors. Then $C(\mathbf{b}') = C(\mathbf{b}(\bar{a})) = C(\mathbf{b})\ M(\bar{a})$ with the transformation matrix (2.5).

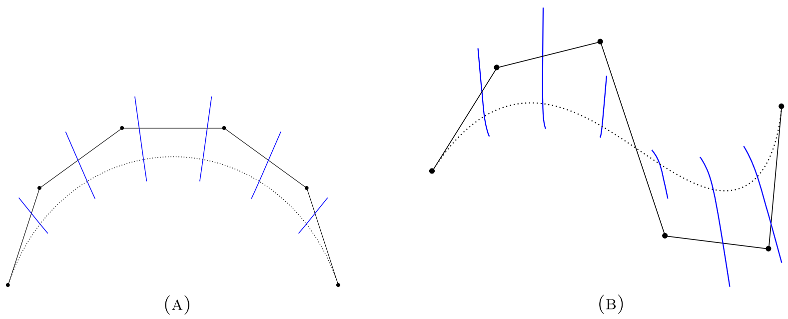

Fig. 8. a_2-Curves with $-.1 < \bar{a}_2 < 2$ for six equidistant $t \in [0, 1]$, ordinary Bézier curves dotted.

In general the use of further shaping points $S_2, S_3, \dots, S_{n-1}$ relies similarly on the respective variation of the parameters $\bar{a}_3, \bar{a}_4, \dots, \bar{a}_n$.

(3) The new control polygon $CP(\mathbf{b}')$ is simply the display of the last transformed control polygon computed in step 2 for $\mathbf{x}_{new} = \mathbf{x}(t; \bar{a}; \mathbf{b}) = \mathbf{x}(t; \mathbf{b}')$ in the background.

Remark 5.1. Instead of the parameter $\bar{a}_2$ one can also use the *responsiveness* c as described in Winkel (2014, Thm. 2.1), in particular, if one wants to have exact interpolation of control points as design option.

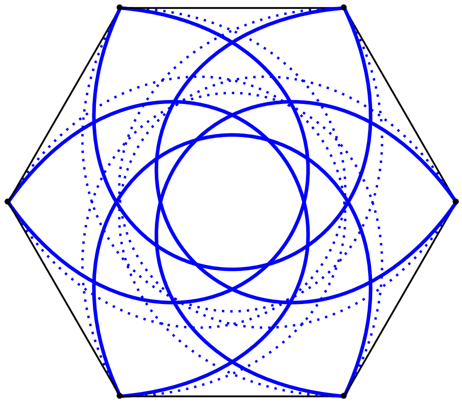

Fig. 9. $\mathbf{x}_k^5$ for $n = 6$ and $\bar{a} = (1, -.3, 0, 0, 0, 0)$.

References

Bartels, R., Beatty, J., 1989. A technique for the direct manipulation of spline curves. Graph. Interface 89, 33–39.
Comtet, L., 1974. Advanced Combinatorics. Reidel, Dordrecht.
Farin, G., 2001. Curves and Surfaces for Computer Aided Geometric Design. A Practical Guide, 5th ed. Academic Press, Boston.
Fowler, B., Bartels, R., 1993. Constraint-based curve manipulation. IEEE Comput. Graph. Appl. 13, 43–49.
Gleicher, M., 1992. Integrating constraints and direct manipulation. In: Proceedings of the 1992 Symposium on Interactive 3D Graphics. ACM Press, Cambridge, Massachusetts, United States, pp. 171–174.
Goldman, R., 2003. Pyramid Algorithms. A Dynamic Programming Approach to Curves and Surfaces for Geometric Modelling. Morgan Kaufmann Publishers, San Francisco.
Roman, S., 1984. The Umbral Calculus. Academic Press, Boston.
Winkel, R., 2001. Generalized Bernstein polynomials and Bézier curves: an application of umbral calculus to computer aided geometric design. Adv. Appl. Math. 27, 51–81.
Winkel, R., 2014. On a generalization of Bernstein polynomials and Bézier curves based on umbral calculus. Comput. Aided Geom. Des. 31, 227–244.