On de Casteljau-type algorithms for rational Bézier curves

Zbyněk Šír

Charles University in Prague, Czech Republic

Bert Jüttler

Johannes Kepler University of Linz, Austria

Abstract

We consider the space of rational functions of degree n with a common denominator. It is shown that – in addition to the standard rational de Casteljau algorithm – the corresponding rational Bézier curves admit up to $n!$ different de Casteljau-type algorithms, depending on the ordering of the elementary factors of the polynomial. Our observations generalize recent results of Han, Chu and Qiu [3], which cover the case of denominators of the form $\prod_{i=1}^{n}(1-t+q^{i-1}t)$ where q is a positive constant, to rational curves with general denominators.

Keywords: rational Bernstein functions, de Casteljau-type algorithm, rational Bézier curves, Lupaş q-analogue of Bernstein operator, degree elevation

1. Introduction

Rational Bézier and B-spline curves and surfaces are one of the the standard representations for free-form geometry in Computer Aided Design and Geometric Modeling [2, 4, 7]. The use of rational representations allows the exact description of conic sections and quadric surfaces (including spheres and cylinders), which are of fundamental interest for various applications.

Bernstein polynomials and B-splines form bases with optimal properties for the spaces of polynomials and spline functions of given degree (and knots in the case of spline function). The spaces of rational (spline) functions with a common denominator are spanned by basis functions with similar properties, which are constructed by collecting rational Bernstein functions or NURBS (Non-Uniform Rational B-splines) basis functions.

In a recent paper, Han, Chu and Qiu [3] consider rational functions with denominators of the form $\prod_{i=1}^{n}(1-t+q^{i-1}t)$ where q is positive a real constant. Based on an operator that has been introduced by Lupaş [5], they introduce a system of rational basis functions that shares many properties with Bernstein polynomials. The Lupaş q-analogue of Bernstein

Email addresses: `zbynek.sir@mff.cuni.cz` (Zbyněk Šír), `bert.juettler@jku.at` (Bert Jüttler)

operator consists essentially in replacing the usual binomials with their generalized version based on powers of a fixed real number q, see [3, 5] for details.

It is the scope of this short paper to show that similar results are available for a wider class of spaces of rational functions and to analyze the close relations to the standard approach in Computer Aided Geometric Design, which relies on the use of rational Bernstein functions. We shall see that this leads to several new de Casteljau-type evaluation algorithms for rational Bézier curves. While the immediate practical significance of these new algorithms is rather low, these observations might motivate more detailed investigations of nested spaces spanned by bases that are connected by simple recurrence relations to admit de Casteljau-type algorithms. This should extend some of the benefits of using Bernstein-Bézier representations, such as numerically stable evaluation algorithms, to more general spaces of functions. Such investigations would have a similar scope as the exploration of blossoming for non-polynomial functions [1].

More precisely, the present paper shows how the observations from [3] can be extended to spaces of rational functions with more general denominators, simply by using rational Bernstein functions. More precisely, we consider nested spaces of rational functions, obtained by successively multiplying the denominator with linear factors and raising the degree of the denominator. We derive recurrence formulas for the weights and basis functions of these spaces. Based on these recurrences it is observed that each ordering of the denominator factors provides a de Casteljau-type algorithm for curves expressed with respect to this basis.

2. Rational Bernstein functions

We consider an infinite sequence of linear factors

$$L_i(t) = a_i(1-t) + b_i t, \quad i \in \mathbb{Z}_+, \tag{1}$$

which are defined by the real coefficients a_i and b_i, $(a_i, b_i) \neq (0,0)$. If all coefficients are positive, then these factors do not possess roots in the interval $[0,1]$. Some of these factors may degenerate to constants. This is the case if the coefficients satisfy $a_i = b_i$.

For any positive integer n, we denote the product of the first n factors by

$$\omega^n(t) = L_1(t) \cdot \ldots \cdot L_n(t).$$

The product is a polynomial of degree at most n. It possesses a unique representation

$$\omega^n(t) = \sum_{i=0}^{n} w_i^n \beta_i^n(t)$$

with respect to the Bernstein polynomials $\beta_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$ of degree n. Following the usual approach in Computer Aided Geometric Design [2, 4, 7], the coefficients of this representation are called the *weights*. A simple computation confirms that

$$w_i^n = \frac{1}{\binom{n}{i}} \Big(\sum_{\substack{K \cup L = \{1,\ldots,n\} \\ |K|=(n-i), |L|=i}} \prod_{k \in K} a_k \prod_{l \in L} b_l \Big) . \tag{2}$$

Consequently, if all coefficients a_i and b_i are positive, then so are the weights.

The weights are used to define the *rational Bernstein functions*

$$\rho_i^n(t) = \frac{w_i^n \beta_i^n(t)}{\omega^n(t)}. \tag{3}$$

Note that these functions depend only on the ratios of $a_i : b_i$, i.e., they do not change if we replace (a_i, b_i) with $(\lambda_i a_i, \lambda_i b_i)$, where λ_i is a non-zero real number. Moreover if $a_i \cdot b_i \leq 0$ then $L_i(t)$ and consequently the basis functions will have a root within the interval $[0, 1]$. For these reasons, in applications the coefficients (a_i, b_i) will be positive.

If all weights are non-zero, then these functions span the space of rational functions of degree n with denominator $\omega^n(t)$,

$$R^n = \operatorname{span}\{\rho_i^n(t) \mid i = 0, \ldots, n\} = \{P(t)/\omega^n(t) \mid P(t) \in \Pi^n(t)\}, \tag{4}$$

where $\Pi^n(t)$ is the space of polynomials of degree n. These spaces are nested, i.e. $R^{n-1} \subset R^n$.

We extend these definitions to include the case $n = 0$ by defining

$$\omega^0(t) = \rho_0^0(t) = 1.$$

Consequently, R^0 is the linear space of constant functions. Moreover, the functions in (3) are defined for all integers i by setting of $w_i^n = 0$ and $\rho_i^n(t) = 0$ whenever $i < 0$ or $i > n$.

The rational Bernstein functions possess several useful properties, which are similar to the properties of the “Lupaş q-analogues of the Bernstein functions” [3]:

Proposition 1.

(i) Non-negativity: If all coefficients a_i, b_i are positive, then $\rho_i^n(t) \geq 0$ for $t \in [0, 1]$.

(ii) Partition of unity: $\sum_{i=0}^n \rho_i^n(t) = 1$ almost everywhere[1].

(iii) Endpoint interpolation: If all coefficients a_i, b_i are non-zero, then $\rho_i^n(0) = \delta_{i0}$ and $\rho_i^n(1) = \delta_{in}$.

(iv) Inverse property: $\rho_i^n(t) = \hat{\rho}_{n-i}^n(1-t)$, where $\hat{\rho}$ are the basis functions defined in an analogous way using the linear factors $\hat{L}_i(t) = b_i(1-t) + a_i t$.

(v) Reducibility: We obtain the classical polynomial Bernstein basis when $a_i = b_i = 1$.

The proofs of these observations follow directly from the definition of the rational Bernstein functions.

3. Recurrence relations

Before establishing recurrence relations, we need to analyze the weights in more detail.

[1] except for the roots of $\omega^n(t)$

Proposition 2. *The weights satisfy the recurrence formula*

$$w_i^n = a_n \frac{(n-i)}{n}\, w_i^{n-1} + b_n \frac{i}{n}\, w_{i-1}^{n-1}. \tag{5}$$

Proof. The recurrence of the denominators

$$\omega^n(t) = \omega^{n-1}(t)\, L_n(t) \tag{6}$$

implies the equation

$$\sum_{i=0}^{n} w_i^n \beta_i^n(t) = \left[\sum_{i=0}^{n-1} w_i^{n-1} \beta_i^{n-1}(t)\right] [a_n(1-t) + b_n t],$$

from which we obtain

$$w_i^n \beta_i^n(t) = a_n(1-t) w_i^{n-1} \beta_i^{n-1}(t) + b_n t w_{i-1}^{n-1} \beta_{i-1}^{n-1}(t). \tag{7}$$

Dividing both sides by $\beta_i^n(t)$ gives (5). $\square$

Based on these observations we derive a recurrence relation for the rational Bernstein functions.

Proposition 3. *The rational Bernstein functions satisfy the recurrence formula*

$$\rho_i^n(t) = \frac{a_n(1-t)}{L_n(t)}\, \rho_i^{n-1}(t) + \frac{b_n t}{L_n(t)}\, \rho_{i-1}^{n-1}(t). \tag{8}$$

Proof. Combining (6), (7) and (3) confirms (8). $\square$

This recurrence will be used in the next section to derive a de Casteljau-type algorithm for evaluating rational Bézier curves.

Note that there are infinitely many formulas expressing ρ_i^n as a (non-constant) linear combination of ρ_i^{n-1} and ρ_{i-1}^{n-1}. More precisely we have

$$\rho_i^n(t) = \frac{n}{n-i}\, \frac{(1-t)}{L_n(t)}\, \frac{w_i^n}{w_i^{n-1}}\, \rho_i^{n-1}(t), \tag{9}$$

$$\rho_i^n(t) = \frac{n}{i}\, \frac{t}{L_n(t)}\, \frac{w_i^n}{w_{i-1}^{n-1}}\, \rho_{i-1}^{n-1}(t) \tag{10}$$

and any affine combination of (9) and (10) provides a valid formula.

As we shall see in the next section, Eq. (8) can be used to derive a de Casteljau-type algorithm, since the coefficients on the right-hand side are independent of i. Among all affine combinations of (9) and (10), the recurrence (8) is the only one with this property.

Another formula expresses each rational Bernstein function of degree n in terms of two functions of degree $n+1$, thereby confirming the nested nature of the spaces R^n.

Proposition 4. *The rational Bernstein functions satisfy*

$$\rho_i^n(t) = a_{n+1}\frac{n+1-i}{n+1}\,\frac{w_i^n}{w_i^{n+1}}\,\rho_i^{n+1}(t) + b_{n+1}\frac{i+1}{n+1}\,\frac{w_i^n}{w_{i+1}^{n+1}}\,\rho_{i+1}^{n+1}(t). \tag{11}$$

Proof. Expressing $\rho_{i+1}^{n+1}(t)$ using equation (10) and $\rho_i^{n+1}(t)$ using equation (9) leads to the formula. □

This result allows to formulate an algorithm for *degree elevation*. Due to the linear independence of the rational Bernstein functions, there exists only one formula of this kind.

4. De Casteljau-type algorithms

Given the control points $P_0, \ldots, P_n \in \mathbb{R}^d$ for some dimension d, we define a rational Bézier curve in $\mathbb{R}^d$,

$$c(t) = \sum_{i=0}^{n} P_i \rho_i^n(t).$$

If the weights are positive, then a rational Bézier curve possesses the convex hull property and shape-preserving properties, i.e., it is variation diminishing and convexity preserving [8, p. 112]. This includes the case of a rational Bézier curve with weights defined by linear factors (1) with positive constants a_i and b_i. In particular, this also covers curves defined by the "Lupaş q-analogues of the Bernstein functions" as observed – independently of the shape-preserving properties of rational Bézier curves – in [3].

Definition 5. For any given value of $t \in [0,1]$, the *de Casteljau-type algorithm* defines recursively the points

$$\begin{aligned} P_i^0(t) &= P_i, \text{ for } i = 0, \ldots n; \\ P_i^j(t) &= \frac{a_j(1-t)}{L_j}\,P_i^{j-1}(t) + \frac{b_j t}{L_j}\,P_{i+1}^{j-1}(t), \text{ for } j = 1, \ldots, n \text{ and } i = 0, \ldots, n-j. \end{aligned} \tag{12}$$

Proposition 6. *The points defined in the de Casteljau-type algorithm satisfy*

$$P_i^j(t) = \sum_{k=0}^{j} P_{i+k}\rho_k^j(t). \tag{13}$$

In particular we have $P_0^n(t) = c(t)$.

Proof. We proceed by mathematical induction. For $j = 0$ we get (13) by the convention $\rho_0^0(t) \equiv 1$. For the induction step we obtain

$$\begin{aligned} P_i^j(t) &= \frac{a_j(1-t)}{L_j} P_i^{j-1}(t) + \frac{b_j t}{L_j} P_{i+1}^{j-1}(t) \\ &= \frac{a_j(1-t)}{L_j} \left(\sum_{k=0}^{j-1} P_{i+k} \rho_k^{j-1}(t) \right) + \frac{b_j t}{L_j} \left(\sum_{k=0}^{j-1} P_{i+k+1} \rho_k^{j-1}(t) \right) \\ &= \sum_{k=0}^{j} P_{i+k} \left(\frac{a_j(1-t)}{L_j} \rho_k^{j-1}(t) + \frac{b_j t}{L_j} \rho_{k-1}^{j-1}(t) \right) = \sum_{k=0}^{j} P_{i+k} \rho_k^j(t), \end{aligned}$$

where the last equality follows from (8). □

The maximum number of different de Casteljau-type algorithms of this form is $n!$ (This is a factorial, not an exclamation mark!). Indeed, if all linear factors are different, then their permutations define the different algorithms. Note that all these de Casteljau-type algorithms are different from the standard rational de Casteljau algorithm, see Example 9.

For each step (12) of these algorithms, the ratio used to generate the new point from the two existing ones is the same for all i. This is different from the standard de Casteljau algorithm (see Figure 4), where a different ratio is used in each linear combination.

Our approach can be extended to quadratic elementary factors of the denominator as follows. Consider linear factors with complex coefficients. If two consecutive linear factors are complex conjugate, then their product is real and the composition of the corresponding two steps in the de Casteljau-type algorithms gives linear combinations with real coefficients. This leads to a de Casteljau-type algorithm also for quadratic elementary factors, since these can be split into two adjacent complex conjugate linear factors. Consequently, we can extend this approach to rational curves with any denominator. We demonstrate this approach by applying it to one example, see Example 10.

Finally we note that the rational de Casteljau-type algorithm described in Definition 5 provides a geometric interpretation for the influence of the constants a_i, b_i to the shape of the curve, cf. Figure 1. Without loss of generality, we consider the constants a_n, b_n associated with the last linear factor, as we can always reorder the linear factors. In the final step we generate a blend curve between the two curves $P_0^{n-1}(t)$ and $P_1^{n-1}(t)$. Both curves are rational Bézier curves with the same weights (determined by the first $n-1$ linear factors) but with different control polygons. The control points of the first and second curve are $P_0, \ldots, P_{n-1}$ and $P_1, \ldots, P_n$, respectively. The ratio a_n/b_n determines the influence of both curves to the final result. If it is equal to one, then the final curve is simply a linear blending curve between the two curves. For larger or smaller values, it remains closer to the first or to the second curve, respectively. A similar but less intuitive interpretation (as a blend between three curves) can be given in the case of two complex conjugate linear factors.

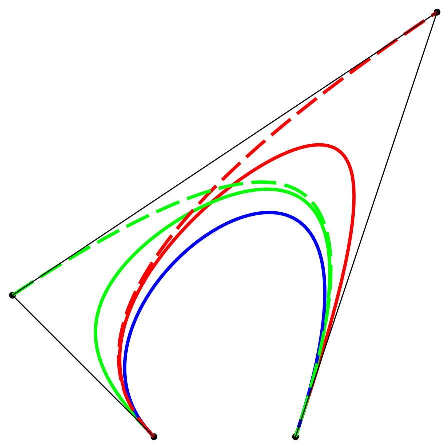

Figure 1: Geometric interpretation of the ratio a_n/b_n for a cubic rational Bézier curve. The last step of the de Casteljau-type algorithm generates a blend between the two quadratic rational Bézier curves (shown as dashed lines), which are defined by the first three and by the last three control points. Depending on the ratio a_n/b_n, the cubic curve (shown as solid line with different colors) follows the first (dashed red) or the second (dashed green) quadratic curve more closely. The picture shows the curves obtained for the ratios 20 (red), 1 (blue) and 1/20 (green).

5. Examples

We present several examples that illustrate the findings of this paper.

Example 7. For the special choice $a_i = a$, $b_i = b$ we obtain $w_i^n = a^{n-i}b^i$. In this case, the rational basis functions are the Bernstein polynomials composed with a rational reparametrization of degree 1 that maps the boundaries of the interval $[0, 1]$ onto itself. More precisely we get

$$\rho_i^n(t) = \beta_i^n\left(\frac{bt}{a(1-t)+bt}\right). \qquad \diamondsuit$$

Example 8. For the special choice $a_i = 1$, $b_i = q^{i-1}$, where q is a positive real number we get the "Lupaş q-analogues of the Bernstein functions", which were considered earlier in [3]. The authors of that paper observed that the weights admit a particularly nice closed-form representation in this case. $\diamondsuit$

Example 9. Consider three linear factors

$$L_1(t) = 3(1-t) + t, \quad L_2(t) = 6(1-t) + 5t, \quad L_3(t) = 1(1-t) + 3t.$$

We obtain the weights

$$\begin{aligned} &w_0^1 = 3, w_1^1 = 1 \\ &w_0^2 = 18, w_1^2 = \frac{21}{2}, w_2^2 = 5 \\ &w_0^3 = 18, w_1^3 = 25, w_2^3 = \frac{68}{3}, w_3^3 = 15. \end{aligned}$$

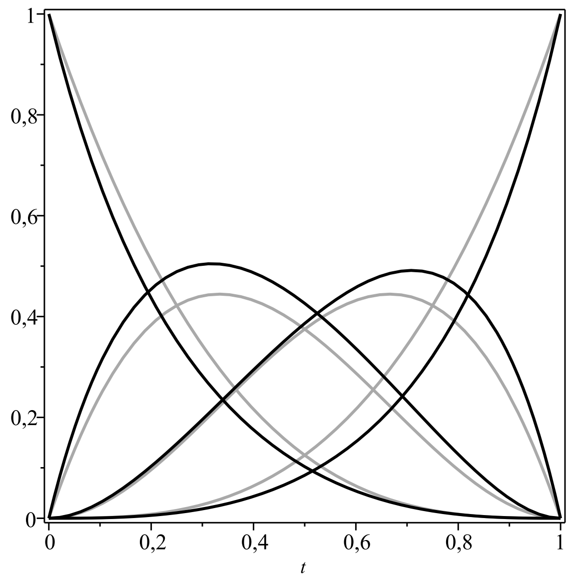

Figure 2: The rational Bernstein functions of degree three from Example 9 (black) compared to the Bernstein polynomials (gray).

The corresponding cubic rational basis functions $\rho_i^3(t)$ are displayed in Figure 2. We consider a curve with control points

$$P_0 = [0,0], P_1 = [-1,1], P_2 = [2,3], P_3 = [1,0].$$

The de Casteljau-type algorithm for $t = 1/2$ generates the points P_i^j

$j\backslash i$	0	1	2	3
0	$[0,0]$	$[-1,1]$	$[2,3]$	$[1,0]$
1	$[-\frac{1}{4},\frac{1}{4}]$	$[-\frac{1}{4},\frac{3}{2}]$	$[\frac{7}{4},\frac{9}{4}]$	
2	$[-\frac{1}{4},\frac{9}{11}]$	$[\frac{29}{44},\frac{81}{44}]$		
3	$[\frac{19}{44},\frac{279}{176}]$,			

which are displayed on Figure 3, top left. The five additional permutations of the factors L_1, L_2 and L_3 lead to five further de Casteljau-type algorithms that generate the same curve point.

Note that these algorithms do not provide the tangent property of the classical de Casteljau algorithm in general, i.e., the line connecting the last two points is generally not tangent to the curve. Similarly, these algorithms do not have a subdivision property in general and cannot be used to split the curve. In contrast, algorithms obtained using the blossoming approach (cf. [6]) possess these properties. ♢

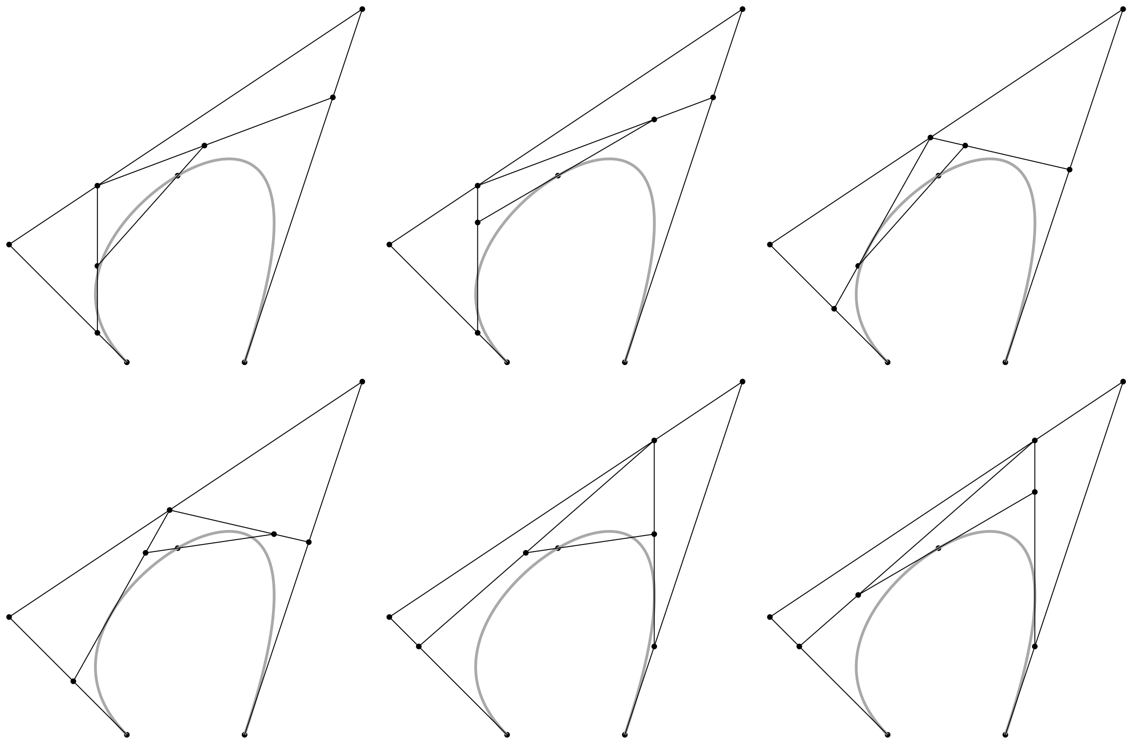

Figure 3: Six different de Casteljau-type algorithms for value $t = 1/2$.

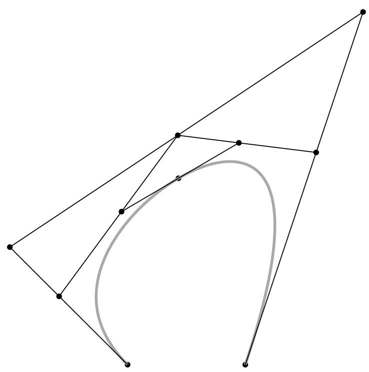

Figure 4: The standard rational de Casteljau algorithm for value $t = 1/2$.

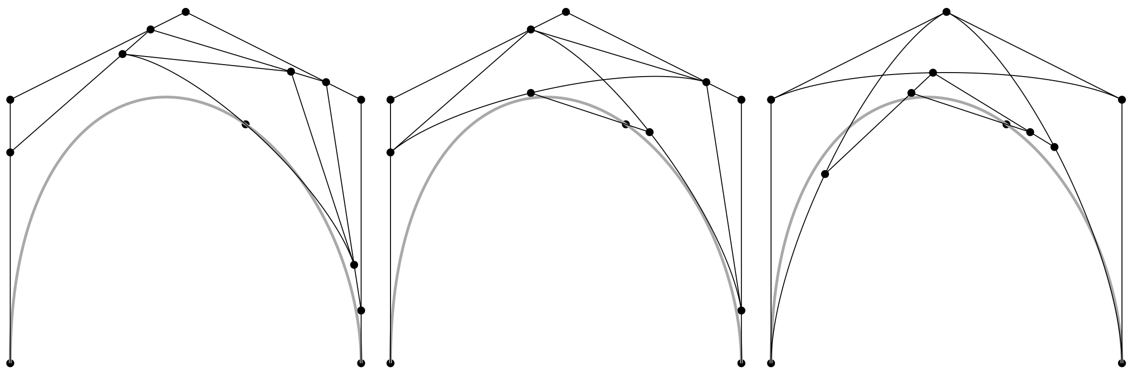

Figure 5: Three different de Casteljau-type algorithms for value $t = 2/3$ for a rational curve with two complex conjugate linear factors in the denominator. Some of the intermediate control points are located on quadratic rational curves, since we had to merge two steps of the algorithm.

Example 10. Consider four linear factors

$$L_1(t) = L_2(t) = (1-t) + 2t, \quad L_3(t) = \overline{L_4(t)} = -(1+3i)(1-t) + (1-i)t.$$

The factors L_1 and L_2 are real and identical, and the factors L_3 and L_4 are complex conjugate. The corresponding quartic rational basis functions are thus real with the last row of weights being

$$w_0^4 = 10, w_1^4 = 11, w_2^4 = \frac{29}{3}, w_3^4 = 6, w_4^4 = 8.$$

We consider a curve with the control points

$$P_0 = [-1, 0], P_1 = [-1, 3/2], P_2 = [0, 2], P_3 = [1, 3/2], P_4 = [1, 0].$$

We can still apply the de Casteljau-type algorithms as in the previous example. After the step corresponding to one complex factor we get complex control points, which become real again after the step corresponding to the complex conjugate. We display in Figure 5 the three de Casteljau-type algorithms which are obtained by performing the two complex conjugate steps immediately one after the other.

More precisely, if the two linear factors $L_{j+1}(t)$ and $L_{j+2}(t)$ are complex conjugate, then the points $P_i^{j+2}(t)$ lies on a conic section (a quadratic rational curve) with the control points $P_i^j(t)$, $P_{i+1}^j(t)$, $P_{i+2}^j(t)$ and the weights determined by the denominator $L_{j+1}(t) \cdot L_{j+2}(t)$. These quadratic curves are plotted in Figure 5 at the corresponding places in the de Casteljau-type algorithm. $\Diamond$

6. Conclusion

We have analyzed nested spaces of rational functions, obtained by successively multiplying the denominator with linear factors. We were able to determine the recurrence formulas for weights and basis functions of these spaces. Each ordering of the denominator factors provides a de Casteljau-type algorithm for curves expressed with respect to these rational basis.

The algorithms can be extended in a straightforward way to the case of tensor-product patches. Indeed, in this case each variable is handled separately. Future research could be devoted to triangular rational patches with denominators that have only linear elementary factors, and to rational spline curves and surfaces.

References

[1] M. Brilleaud and M.-L. Mazure. Design with L-splines. *Numer. Algorithms*, 65(1):91–124, 2014.

[2] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design.* Morgan Kaufmann, 2001.

[3] L.-W. Han, Y. Chu, and Z.-Y. Qiu. Generalized Bézier curves and surfaces based on Lupaş-analogue of Bernstein operator. *Journal of Computational and Applied Mathematics*, 261:352–363, 2014.

[4] J. Hoschek and D. Lasser. *Fundamentals of Computer Aided Geometric Design.* AK Peters, Wellesley (Mass.), 1996.

[5] A. Lupaş. A q-analogue of the Bernstein operator. *Sem. Numer. Statist. Calc.*, 9:85–92, 1987.

[6] M.-L. Mazure. Extended Chebyshev spaces in rationality. *BIT Numerical Mathematics*, 53(4):1013–1045, 2013.

[7] L. Piegl and W. Tiller. *The NURBS book*. Springer, 2013.

[8] H.J. Wolters. Rational techniques. In G. Farin, J. Hoschek, and M.-S. Kim, editors, *Handbook of Computer Aided Geometric Design*, chapter 5, pages 111–140. Elsevier, 2002.