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 the corresponding rational Bézier curves admit up to \( 1 + n! \) 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, Lupaș \( q \)-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 a positive real constant. Based on an operator that has been introduced by Lupaș [5], they introduce a system of rational basis functions that shares many properties with Bernstein polynomials. The Lupaș \( q \)-analogue of Bernstein 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 algo-
rithms 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 ratio-
nal 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 func-
tions 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

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

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

\[ \omega^n(t) = \sum_{i=0}^{n} w^n_i \beta^n_i(t). \]

with respect to the Bernstein polynomials \(\beta^n_i(t) = \binom{n}{i} t^i (1 - t)^{n-i}\) 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

\[ w^n_i = \frac{1}{\binom{n}{i}} \left( \sum_{K \cup L = \{1, \ldots, n\}} \prod_{k \in K} a_k \prod_{l \in L} b_l \right). \]
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

$$\rho^n_i(t) = \frac{w^n_i \beta^n_i(t)}{\omega^n(t)}.$$  \hspace{1cm} (5)

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 $(\lambda a_i, \lambda b_i)$, where $\lambda$ 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 be positive.

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

$$R^n = \text{span}\{\rho^n_i(t) \mid i = 0, \ldots, n\} = \{P(t)/\omega^n(t) \mid P(t) \in \Pi^n(t)\},$$ \hspace{1cm} (6)

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 (5) are defined for all integers $i$ by setting of $w^n_i = 0$ and $\rho^n_i(t) = 0$ whenever $i < 0$ or $i > n$.

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

**Proposition 1.**

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

(ii) **Partition of unity:** $\sum_{i=0}^{n} \rho^n_i(t) = 1$ almost everywhere\footnote{except for the roots of $\omega^n$}.

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

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

(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.

**Proposition 2.** The weights satisfy the recurrence formula

\[ w^n_i = a_n \frac{(n-i)}{n} w^{n-1}_i + b_n \frac{i}{n} w^{n-1}_{i-1}. \]  

(7)

**Proof.** The recurrence of the denominators

\[ \omega^n = \omega^{n-1} L_n \]  

(8)

implies the equation

\[ \sum_{i=0}^{n} w^n_i \beta^n_i(t) = \left[ \sum_{i=0}^{n-i} w^{n-1}_i \beta^{n-1}_i(t) \right] [a_n(1-t) + b_n t], \]  

(9)

from which we obtain

\[ w^n_i \beta^n_i(t) = a_n(1-t) w^{n-1}_i \beta^{n-1}_i(t) + b_n t w^{n-1}_{i-1} \beta^{n-1}_{i-1}(t). \]  

(10)

Dividing both sides by \( \beta^n_i(t) \) gives (7). \( \square \)

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

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

\[ \rho^n_i(t) = a_n(1-t) \rho^{n-1}_i(t) + b_n t \rho^{n-1}_{i-1}(t). \]  

(11)

**Proof.** Combining (8), (10) and (5) confirms (11). \( \square \)

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 \( \rho^n_i \) as a (non-constant) linear combination of \( \rho^{n-1}_i \) and \( \rho^{n-1}_{i-1} \). More precisely we have

\[ \rho^n_i(t) = \frac{n}{n-i} \frac{(1-t)}{L_n(t)} \frac{w^n_i}{w^{n-1}_i \rho^{n-1}_i(t)}, \]  

(12)

\[ \rho^n_i(t) = \frac{n}{i} \frac{t}{L_n(t)} \frac{w^n_i}{w^{n-1}_{i-1} \rho^{n-1}_{i-1}(t)} \]  

(13)

and any affine combination of (12) and (13) provides a valid formula.

As we shall see in the next section, Eq. (11) 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 (12) and (13), the recurrence (11) 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
\[ \rho^n_i(t) = a_{n+1}^{i+1} \frac{n + 1 - i}{n + 1} w^n_i \rho^{n+1}_i(t) + b_{n+1}^{i+1} \frac{i + 1}{n + 1} w^{n+1}_{i+1} \rho^{n+1}_{i+1}(t). \] (14)

Proof. Expressing \( \rho^{n+1}_{i+1}(t) \) using equation (13) and \( \rho^n_i(t) \) using equation (12) 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 \( \mathbb{R}^d \),
\[ c(t) = \sum_{i=0}^n P_i \rho^n_i(t). \] (15)

Definition 5. For any given value of \( t \in [0, 1] \), the de Casteljau-type algorithm defines recursively the points
\[ P^0_i = P_i, \text{ for } i = 0, \ldots, n; \]
\[ P^j_i = \frac{a_j(1 - t)}{L_j} P^{j-1}_i + \frac{b_j t}{L_j} P^{j-1}_{i+1}, \text{ for } j = 1, \ldots, n \text{ and } i = 0, \ldots, (n - j). \] (17)

Proposition 6. The points defined in the de Casteljau-type algorithm satisfy
\[ P^j_i = \sum_{k=0}^j P_{i+k} \rho^j_k(t). \] (18)

In particular we have \( P^n_0 = c(t) \).

Proof. We proceed by mathematical induction. For \( j = 0 \) we get (18) by the convention \( \rho^0_0(t) \equiv 1 \). For the induction step we get
\[ P^j_i = \frac{a_j(1 - t)}{L_j} P^{j-1}_i + \frac{b_j t}{L_j} P^{j-1}_{i+1} = \]
\[ = \frac{a_j(1 - t)}{L_j} \left( \sum_{k=0}^{j-1} P_{i+k} \rho^{j-1}_k(t) \right) + \frac{b_j t}{L_j} \left( \sum_{k=0}^{j-1} P_{i+k+1} \rho^{j-1}_k(t) \right) = \]
\[ = \sum_{k=0}^j P_{i+k} \left( \frac{a_j(1 - t)}{L_j} \rho^{j-1}_k(t) + \frac{b_j t}{L_j} \rho^{j-1}_{k-1}(t) \right) = \sum_{k=0}^j P_{i+k} \rho^j_k(t), \]
where the last equality follows from (11).
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 (17) 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 3), 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 conjugate complex, 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 conjugate complex linear factors. Consequently, we can extend this approach to rational curves with any denominator. We demonstrate this approach on one example, see Example 10.

5. Examples

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

**Example 7.** For the special choice \(a_i = a\), \(b_i = b\) we obtain \(w^n_i = a^{n-i}b^i\). 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

\[
\rho^n_i(t) = \beta^n_i \left( \frac{bt}{a(1-t) + bt} \right). \tag{22}
\]

**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.

**Example 9.** Consider three linear factors

\[
L_1(t) = 3(1-t) + t, \quad L_2(t) = 6(1-t) + 5t, \quad L_3(t) = 1(1-t) + 3t. \tag{23}
\]

We obtain the weights

\[
w_0^1 = 6, w_1^1 = \frac{2}{3} \tag{24}
\]

\[
w_0^2 = 36, w_1^2 = 14, w_2^2 = \frac{10}{3} \tag{25}
\]

\[
w_0^3 = 36, w_1^3 = 50, w_2^3 = \frac{136}{3}, w_3^3 = 30. \tag{26}
\]
The corresponding cubic rational basis functions $\rho^3_i(t)$ are displayed in Figure 1. We consider a curve with control points

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

(27)

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

<table>
<thead>
<tr>
<th>$j$ \ $i$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>[0, 0]</td>
<td>[-1, 1]</td>
<td>[2, 3]</td>
<td>[1, 0]</td>
</tr>
<tr>
<td>1</td>
<td>$[-\frac{1}{4}, \frac{1}{4}]$</td>
<td>$[-\frac{1}{4}, \frac{3}{4}]$</td>
<td>$[\frac{7}{12}, \frac{1}{4}]$</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$[-\frac{1}{4}, \frac{9}{11}]$</td>
<td>$[\frac{20}{44}, \frac{81}{44}]$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$[\frac{19}{44}, \frac{279}{176}]$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(28)

which are displayed on Figure 2, 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 2: Six different de Casteljau-type algorithms for value $t = 1/2$.

Figure 3: The standard rational de Casteljau algorithm for value $t = 1/2$. 
Example 10. Consider four linear factors

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

Factors \( L_1 \) and \( L_2 \) are real and identical, factors \( L_3 \) and \( L_4 \) are complex conjugated. The corresponding quartic rational basis functions are thus real. We consider a curve with 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 run the de Casteljau-type algorithms like in the previous example. After the step corresponding to one complex factor we get complex control points which however become again real after the conjugated step. We displayed on Figure 4 the three de Casteljau-type algorithms corresponding where the two conjugated steps are done immediately one after the other. In this case the produced points lie on conics with corresponding control points.

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


