Evaluation and Properties of the Derivative of a NURBS Curve

Michael S. Floater

Abstract. In this paper we describe two natural ways to express the first derivative of a rational B-spline curve in terms of its control points and weights. These expressions generalise the established ones for non-rational curves and offer numerically stable methods for computing the derivative. Some useful properties are derived from one of them.

§1. Introduction

Many applications of B-spline curves require the evaluation of derivatives. Examples include Newton iteration in intersection algorithms, calculating extremal points, blending, and offsetting. It is sometimes desirable also to calculate bounds on the magnitude of at least the first derivative, see Filip, Magedson, & Markot [4] and on the direction of the first derivative (the hodograph), see Sederberg & Meyers [7].

There are two known formulas for the first derivative of a (non-rational) B-spline curve. For simplicity we can refer to these as the *hodograph formula*, equations (5–6), and the *tangent formula*, equation (8), respectively. The aim of this article is to extend these results to NURBS curves and to derive some useful properties from them. A good introduction to the application of NURBS is given by Piegl & Tiller [6].

Corresponding expressions are also well established for Bézier curves and they were recently extended to rational Bézier curves, see Floater [5]. Some of the results in this article use similar techniques to those used for rational Bézier curves. Others warrant a different approach, see Proposition 5 for example.

The reasons for the naming convention for the two formulas is (a) that the hodograph property follows from the first, and (b) the fact that the curve

Mathematical Methods in CAGD and Image Processing

Tom Lyche and L. L. Schumaker (eds.), pp. 1–15.

ISBN 0-12-000000-X.

is tangential to the vector joining the two penultimate points in the de Boor algorithm follows from the second. This second property is well known in the B-spline community. It can be demonstrated using an alternative approach. One can insert enough knots at the chosen point to subdivide the curve there. Since each subcurve is in Bézier form at that point, one obtains the tangent property. The advantage of the tangent formula itself and indeed of the hodograph formula are that they offer stable methods for computing the first derivative.

Another possible approach to evaluating a NURBS curve and its derivatives is through the well established fact that each of its segments can be converted to rational Bézier form by inserting the appropriate number of knots at either end. This means that we could find the derivative of a NURBS curve at a point by converting the curve locally to Bézier form and then applying formulas for the derivative of a Bézier curve. But this is unnecessary as well as inefficient and it is clearly superior to apply (14–15) or (18) directly.

The usual method of calculating the mth derivative $R^{(m)}(u)$ is to write the quotient R as $R = a/b$. Then $bR = a$ so that, by the Leibnitz formula, $b'R + bR' = a'$, $b''R + 2b'R' + bR'' = a''$, etc. Then R, R', R'' etc. can be computed in sequence. It is difficult, on the other hand, to envisage a general sequence of expressions of the form (14–15) for higher derivatives. In fact by considering a NURBS representation for a conic section (R has just one segment and degree 2) one can see that it is impossible to write R'' in the form

$$R'' = \sum_{i=2}^{L+n-1} \kappa(u)(P_i - 2P_{i-1} + P_{i-2}),$$

unlike non-rational B-splines. Unless R represents a parabola, the direction of R'' cannot be constant. In conclusion, the general Leibnitz method is probably the best in applications. The forms of equations (14–15) and (18) we derive in this paper are rather special to the first derivative.

A strong motivation behind the mathematics was to derive the hodograph property for NURBS curves. This property is well known for non-rational B-splines (it follows immediately from (5–6)) and has been used to great advantage in intersection algorithms by, for example, Sederberg & Meyers [7]. The property has been proved before for rational Bézier curves by Sederberg & Wang [8], and via a different approach by Floater [5].

The following example illustrates how useful the hodograph property is in intersection problems. Suppose that we are given a NURBS curve $R(u)$ in 2-D and we wish to find out whether it intersects itself. To try to obtain information about R we use the hodograph property, Theorem 6. Regarding the vectors $(P_i - P_{i-1})$ as points in 2-D, it is clear that the set $C \subset \mathbb{R}^2$ of all points X of the form

$$X = \sum_{i=1}^{L+n-1} \gamma_i (P_i - P_{i-1}),$$

where $\gamma_i \geq 0$, is an infinite cone whose apex is located at the origin. C is

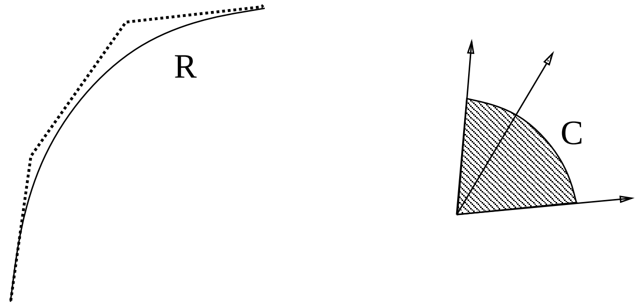

Figure 1. The cone angle is less than 180°.

known as the *direction cone* for R. Equation (14) shows that, for any u, $R'(u)$ lies inside the cone.

Now if the curve is to intersect itself, it must turn through at least 180°. Therefore if the direction cone has an angle of less than 180° the curve cannot intersect itself, see Figure 1. If the angle is any greater, on the other hand, the direction cone does not tell us anything, see Figure 2. In the second case, however, we could subdivide the curve in the hope that each subcurve *does* have a cone angle of less than 180°. This idea can be the starting point for a recursive subdivision algorithm to find all self-intersection points:

Algorithm 1

1. If cone angle $< 180°$, then finish.
2. Else
   - 2.1. Subdivide the curve into subcurves R_1 and R_2.
   - 2.2. Find self-intersection points of R_1.
   - 2.3. Find self-intersection points of R_2.
   - 2.4. Find intersection points of R_1 and R_2.

This scheme assumes we have another algorithm for intersecting two NURBS curves R_1 and R_2. Once again we can use the direction cones C_1 and C_2 to advantage as part of a recursive subdivision/iteration algorithm. This time we have to consider both tangent directions for each curve. Set

$$\hat{C}_1 = \{X \in \mathbb{R}^2 : X \in C_1 \text{ or } -X \in C_1\},$$

and define $\hat{C}_2$ similarly, see Figure 3. The scheme also makes use of the bounding boxes B_1 and B_2 for R_1 and R_2.

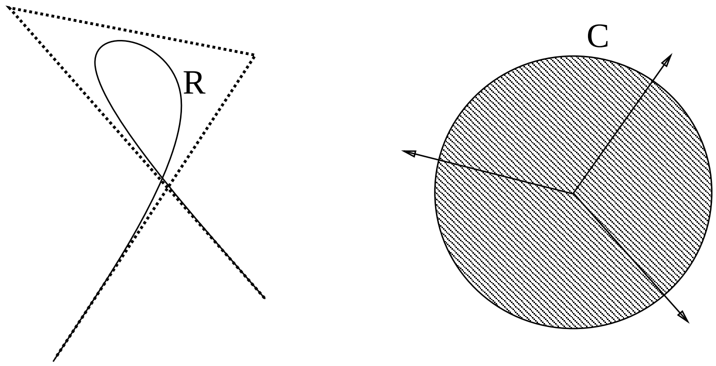

Figure 2. The cone angle is greater than 180°.

Algorithm 2

1. If $B_1 \cap B_2 = \emptyset$, then finish.
2. If $\hat{C}_1 \cap \hat{C}_2 = 0$, then there is at most one intersection.
   Stop subdividing and use Newton iteration.
3. Else
   3.1. Subdivide R_1 into subcurves R_{11} and R_{12}.
   3.2. Subdivide R_2 into subcurves R_{21} and R_{22}.
   3.3. Find $R_{11} \cap R_{21}$, $R_{12} \cap R_{21}$, $R_{11} \cap R_{22}$, and $R_{12} \cap R_{22}$.

In Section 2 we set out definitions and properties essential to the results in Section 3 where we present generalizations of the hodograph and tangent formulas to NURBS curves. The hodograph equation is then used to derive both the hodograph property in Section 4 and an upper bound on R' in Section 5.

§2. Definitions

To begin, we review some basic properties of non-uniform B-spline curves of both non-rational and rational type. In order to prove rigorous results we have to be specific and consistent about notation. We first of all describe some basic properties of ordinary B-splines and then go on to NURBS.

Following the notation of Farin [3], consider the B-spline curve containing L polynomial segments (some of which may be null). It is specified by the knot sequence $\{u_0 \le \cdots \le u_{L+2n-2}\}$ and the control points $P_0, \ldots, P_{L+n-1} \in \mathbb{R}^3$. The curve is defined by

$$S(u) = \sum_{i=0}^{L+n-1} N_{i,n}(u) P_i \tag{1}$$

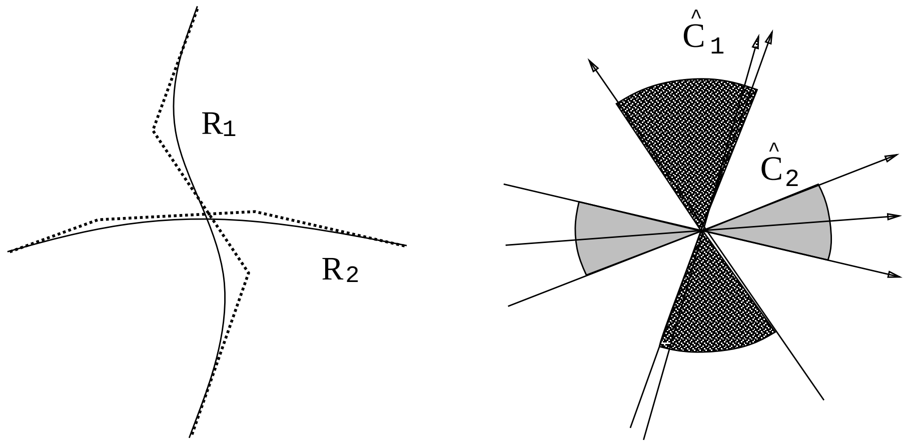

Figure 3. $\hat{C}_1$ and $\hat{C}_2$ intersect only at the origin.

for $u_{n-1} \leq u \leq u_{L+n-1}$, where $N_{i,n}$ is the ith basis function of degree n. $N_{i,n}$ is only nonzero in the interval $[u_{i-1}, u_{i+n}]$. Note that, strictly speaking, one ought to include two more knots u_{-1} and u_{L+2n-1} to justify the last statement but these knots have no influence over the curve (for $u_{n-1} \leq u \leq u_{L+n-1}$) and are therefore omitted. The basis functions can be defined recursively by the Mansfield, de Boor, Cox recursion, see de Boor [2]:

$$N_{i,k}(u) = \left(\frac{u - u_{i-1}}{u_{i+k-1} - u_{i-1}} \right) N_{i,k-1}(u) + \left(\frac{u_{i+k} - u}{u_{i+k} - u_i} \right) N_{i+1,k-1}(u) \tag{2}$$

for $k \in \{1, \ldots, n\}$, and

$$N_{i,0} = \begin{cases} 1, & \text{if } u_{i-1} \leq u < u_i; \\ 0, & \text{otherwise.} \end{cases}$$

The B-spline basis functions are generalisations of the Bernstein ones; by putting $L = 1$, $u_0 = \cdots = u_{n-1} = 0$, and $u_n = \cdots = u_{2n-1} = 1$ we find $N_{i,n} = B_{i,n}$, the ith Bernstein polynomial of degree n.

The curve $S(u)$ itself can be defined recursively by the de Boor algorithm, due to de Boor [1]. Suppose that $u_I \leq u < u_{I+1}$. By considering the support of the basis functions one can deduce that the only control points which influence the curve in this parameter interval are $P_{I-n+1}, \ldots, P_{I+1}$. Let $P_{i,0}(u) = P_i$ for $i = I - n + 1, \ldots, I + 1$, and for $u_I \leq u < u_{I+1}$. Then define the functions $\alpha_{i,k}(u)$ and $P_{i,k}(u)$ as

$$\alpha_{i,k} = (u - u_{i-1})/(u_{i+n-k} - u_{i-1}) \tag{3}$$

and

$$P_{i,k} = (1 - \alpha_{i,k})P_{i-1,k-1} + \alpha_{i,k}P_{i,k-1} \tag{4}$$

for $k = 1, \ldots, n$ and $i = I - n + k + 1, \ldots, I + 1$. Then it can be shown that $S(u) = P_{I+1,n}$, see Farin [3]. In the special case $u = u_I$ it is only necessary to apply $n - r$ steps of the algorithm, where r is the multiplicity of the knot u_I and then $S(u) = P_{I+1,n-r}$.

In order to make comparisons for the results in Section 3, we state here the hodograph and tangent formulas for the first derivative of a B-spline.

The hodograph formula, given in Farin [3], is the following description of the derivative of $S(u)$:

$$S'(u) = \sum_{i=1}^{L+n-1} \lambda_i(u)(P_i - P_{i-1}), \tag{5}$$

where

$$\lambda_i(u) = nN_{i,n}(u)/(u_{n+i-1} - u_{i-1}). \tag{6}$$

In order to demonstrate this, one needs the following relationship between the derivative of a B-spline basis function and lower degree basis functions, see de Boor [1]:

$$N'_{i,k}(u) = \left(\frac{k}{u_{i+k-1} - u_{i-1}}\right) N_{i,k-1}(u) - \left(\frac{k}{u_{i+k} - u_i}\right) N_{i+1,k-1}(u) \tag{7}$$

for $k \in \{1, \ldots, n\}$. The hodograph formula (5–6) then follows directly from differentiating (1), applying (7), and making a change of subscripts.

The tangent formula for B-splines is as follows. For the sake of completeness we prove this here.

Proposition 1. *Given the B-spline curve of degree n defined by equation (1), the derivative of $S(u)$ is*

$$S'(u) = n\left(\frac{P_{I+1,n-1}(u) - P_{I,n-1}(u)}{u_{I+1} - u_I}\right) \tag{8}$$

where the $P_{i,n-1}$ terms are defined in (3–4).

Proof: We prove, by mathematical induction, the more general statement

$$P'_{i,k} = \frac{k(P_{i,k-1} - P_{i-1,k-1})}{u_{i+n-k} - u_{i-1}} = k\alpha'_{i,k}(P_{i,k-1} - P_{i-1,k-1}). \tag{9}$$

When $k = 1$ the equation can be seen to hold by differentiating equation (4). Now we show that (9) holds for any $k \in \{2, \ldots, n\}$ provided it holds when k is replaced by $k - 1$.

Differentiating (4) with respect to u gives

$$P'_{i,k} = -\alpha'_{i,k}P_{i-1,k-1} + (1 - \alpha_{i,k})P'_{i-1,k-1} + \alpha'_{i,k}P_{i,k-1} + \alpha_{i,k}P'_{i,k-1}.$$

Now by the induction step we can expand $P'_{i-1,k-1}$ and $P'_{i,k-1}$ so that

$$\begin{aligned} P'_{i,k} =& \alpha'_{i,k}(P_{i,k-1} - P_{i-1,k-1}) \\ & + (k-1)(1-\alpha_{i,k})\alpha'_{i-1,k-1}(P_{i-1,k-2} - P_{i-2,k-2}) \\ & + (k-1)\alpha_{i,k}\alpha'_{i,k-1}(P_{i,k-2} - P_{i-1,k-2}). \end{aligned}$$

The second two terms are now rearranged by applying the following identities:

$$\alpha_{i,k}\alpha'_{i,k-1} = \alpha'_{i,k}\alpha_{i,k-1},$$

$$(1-\alpha_{i,k})\alpha'_{i-1,k-1} = \alpha'_{i,k}(1-\alpha_{i,k-1}),$$

$$(1-\alpha_{i,k})\alpha'_{i-1,k-1} - \alpha_{i,k}\alpha'_{i,k-1} = \alpha'_{i,k}(1-\alpha_{i,k-1}) - \alpha'_{i,k}\alpha_{i-1,k-1}.$$

These are verified in a straightforward manner from (3). The rearrangement yields

$$\begin{aligned} P'_{i,k} =& \alpha'_{i,k}(P_{i,k-1} - P_{i-1,k-1}) \\ & + (k-1)\{\alpha'_{i,k}(1-\alpha_{i,k-1})P_{i-1,k-2} + \alpha'_{i,k}\alpha_{i,k-1}P_{i,k-2} \\ & - \alpha'_{i,k}(1-\alpha_{i-1,k-1})P_{i-2,k-2} - \alpha'_{i,k}\alpha_{i-1,k-1}P_{i-1,k-2}\} \\ =& \alpha'_{i,k}(P_{i,k-1} - P_{i-1,k-1}) + (k-1)\alpha'_{i,k}(P_{i,k-1} - P_{i-1,k-1}) \end{aligned}$$

and we are finished. ■

Now we can define the NURBS curve as a generalisation of the the previously defined B-spline. The NURBS curve is specified from precisely the same data except that there are, in addition, (positive) weights $w_0, \ldots, w_{L+n-1} \in \mathbb{R}$ associated with the control points. The curve is defined to be

$$R(u) = \frac{\sum_{i=0}^{L+n-1} N_{i,n}(u) w_i P_i}{\sum_{i=0}^{L+n-1} N_{i,n}(u) w_i}. \tag{10}$$

Note that by setting all the weights to one we get back to (1).

There is an analogous rational version of the de Boor algorithm, see Farin [3]:

$$\alpha_{i,k} = (u - u_{i-1})/(u_{i+n-k} - u_{i-1}) \tag{11}$$

and

$$w_{i,k} = (1-\alpha_{i,k})w_{i-1,k-1} + \alpha_{i,k}w_{i,k-1} \tag{12}$$

and

$$w_{i,k}P_{i,k} = (1-\alpha_{i,k})w_{i-1,k-1}P_{i-1,k-1} + \alpha_{i,k}w_{i,k-1}P_{i,k-1} \tag{13}$$

with the same subscripts as before. One then finds $R(u) = P_{I+1,n}$. Again, setting all weights to one reproduces the non-rational scheme (3–4).

§3. The derivative of a rational B-spline curve

Throughout this article we will only consider one parameter value u at a time so there is no ambiguity in dropping the u. We will also assume that R is differentiable at u. The only situation where this might not be so is when u is a knot value whose multiplicity is n. First we generalize the hodograph formula to NURBS curves.

Proposition 2. *The derivative of the NURBS curve R defined by equation (10) is*

$$R'(u) = \sum_{i=1}^{L+n-1} \lambda_i(u)(P_i - P_{i-1}), \tag{14}$$

where

$$\lambda_i(u) = \frac{1}{w^2_{I+1,n}(u)} \sum_{j=0}^{i-1} \sum_{k=i}^{L+n-1} (N'_{k,n}(u)N_{j,n}(u) - N_{k,n}(u)N'_{j,n}(u))w_j w_k. \tag{15}$$

Proof: Suppose that our parameter value u lies in the knot interval $[u_I, u_{I+1})$. The only basis functions at u which are nonzero are $N_{I-n+1}, \ldots, N_{I+1}$ so we can rewrite (1) as

$$R(u) = \sum_{i=I-n+1}^{I+1} \mu_i(u) w_i P_i, \qquad \mu_i(u) = \frac{N_{i,n}(u)}{\sum_{k=I-n+1}^{I+1} N_{k,n}(u)w_k} = \frac{N_{i,n}(u)}{w_{I+1,n}(u)}.$$

Then

$$\begin{aligned} \mu'_i =& \frac{1}{w^2_{I+1,n}} \left(\sum_{k=I-n+1}^{I+1} N_{k,n} w_k N'_{i,n} - \sum_{k=I-n+1}^{I+1} N'_{k,n} w_k N_{i,n} \right) \\ =& \frac{1}{w^2_{I+1,n}} \sum_{k=I-n+1}^{I+1} (N'_{i,n} N_{k,n} - N_{i,n} N'_{k,n}) w_k. \end{aligned}$$

Now, at any u, R' is in the form

$$R' = \sum_{i=I-n+1}^{I+1} a_i P_i,$$

where $a_i = \mu'_i(u) w_i$ and we wish to rewrite it in the form

$$R' = \sum_{i=I-n+2}^{I+1} \lambda_i (P_i - P_{i-1}). \tag{16}$$

By successively eliminating the control point with the lowest subscript, we find

$$
\begin{aligned}
R' =& a_{I-n+1}P_{I-n+1} + a_{I-n+2}P_{I-n+2} + \cdots + a_{I+1}P_{I+1} \\
=& - a_{I-n+1}(P_{I-n+2} - P_{I-n+2}) + (a_{I-n+1} + a_{I-n+2})P_{I-n+2} \\
& + \cdots + a_{I+1}P_{I+1} \\
=& - a_{I-n+1}(P_{I-n+2} - P_{I-n+2}) - (a_{I-n+1} + a_{I-n+2})(P_{I-n+3} - P_{I-n+2}) \\
& + (a_{I-n+1} + a_{I-n+2} + a_{I-n+3})P_{I-n+3} + \cdots + a_{I+1}P_{I+1} \\
=& \cdots \\
=& - \sum_{i=I-n+1}^{I+1} \sum_{j=I-n+1}^{i-1} a_j(P_1 - P_{i-1}) + \sum_{j=I-n+1}^{I+1} a_j P_{I+1}.
\end{aligned}
$$

Now the coefficient of the P_{I+1} term is the sum of all the a_j, and this is equivalent to

$$
\frac{1}{w^2_{I+1,n}(u)} \sum_{j=I-n+1}^{I+1} \sum_{k=I-n+1}^{I+1} (N'_{j,n}(u)N_{k,n}(u) - N_{j,n}(u)N'_{k,n}(u))w_j w_k
$$

which, by symmetry, is zero. Therefore setting

$$
\lambda_i = - \sum_{j=I-n+1}^{i-1} a_j,
$$

we can express R' in the form of (16) and then

$$
\begin{aligned}
\lambda_i(u) =& \frac{-1}{w^2_{I+1,n}(u)} \sum_{j=I-n+1}^{i-1} \sum_{k=I-n+1}^{I+1} (N'_{j,n}(u)N_{k,n}(u) - N_{j,n}(u)N'_{k,n}(u))w_i w_k \\
=& \frac{1}{w^2_{I+1,n}(u)} \sum_{j=I-n+1}^{i-1} \sum_{k=I-n+1}^{I+1} (N'_{k,n}(u)N_{j,n}(u) - N_{k,n}(u)N'_{j,n}(u))w_i w_k.
\end{aligned}
$$

Again by symmetry,

$$
\sum_{j=I-n+1}^{i-1} \sum_{k=I-n+1}^{i-1} (N'_{k,n}(u)N_{j,n}(u) - N_{k,n}(u)N'_{j,n}(u))w_i w_k = 0,
$$

so these terms can be removed from the expression and we arrive at

$$
\lambda_i(u) = \frac{1}{w^2_{I+1,n}(u)} \sum_{j=I-n+1}^{i-1} \sum_{k=i}^{I+1} (N'_{k,n}(u)N_{j,n}(u) - N_{k,n}(u)N'_{j,n}(u))w_i w_k. \quad (17)
$$

Lastly, due to the fact that all basis functions are zero in $[u_I, u_{I+1})$ except for $N_{I-n+1}, \ldots, N_{I+1}$, we can extend the subscript ranges so that this expression becomes valid for any parameter $u \in [u_{n-1}, u_{L+n-1}]$. Therefore

$$R'(u) = \sum_{i=1}^{L+n-1} \lambda_i(u)(P_i - P_{i-1}),$$

where

$$\lambda_i(u) = \frac{1}{w_{I+1,n}^2(u)} \sum_{j=0}^{i-1} \sum_{k=i}^{L+n-1} (N'_{k,n}(u)N_{j,n}(u) - N_{k,n}(u)N'_{j,n}(u))w_j w_k.$$

This completes the proof. ■

In the next section, we show that every $\lambda_i(u)$ is nonnegative for all u. Now we generalize the tangent formula to NURBS curves.

Proposition 3. *The derivative of the NURBS curve R defined by equation (10) is*

$$R'(u) = n\frac{w_{I,n-1}(u)w_{I+1,n-1}(u)}{w_{I+1,n}^2(u)} \left(\frac{P_{I+1,n-1}(u) - P_{I,n-1}(u)}{u_{I+1} - u_I}\right), \qquad (18)$$

where the $P_{i,k}$ $w_{i,k}$ terms are defined in (11–13).

Proof: Let $R(u) = a(u)/b(u)$. Then

$$R'(u) = (b(u)a'(u) - b'(u)a(u))/b^2(u). \qquad (19)$$

Observe that the numerator $a(u)$ and denominator $b(u)$ of $R(u)$ are themselves non-rational B-spline curves of dimensions 3 and 1 respectively. The control points of $b(u)$ are precisely the weights w_i of $R(u)$. The weight functions $w_{i,k}(u)$ computed from the de Boor scheme (3–4) (with $P_{i,k}$ replaced by $w_{i,k}$) are identical to those computed from the rational scheme (11–13). Therefore, applying Proposition 1 means that

$$b' = n(w_{I+1,n-1} - w_{I,n-1})/(u_{I+1} - u_I).$$

Now consider $a(u)$. Its ith control point is the product $w_i P_i$ which we can rename Q_i say. From the de Boor algorithm (3–4), we find

$$Q_{i,k} = (1 - \alpha_{i,k})Q_{i-1,k-1} + \alpha_{i,k}Q_{i,k-1}.$$

Then Proposition 1 implies

$$a' = n(Q_{I+1,n-1} - Q_{I,n-1})/(u_{I+1} - u_I).$$

But by considering (13) one can see that at each step in the algorithm

$$Q_{i,k}(u) = w_{i,k}(u)P_{i,k}(u),$$

and so

$$b' = n(w_{I+1,n-1}P_{I+1,n-1} - w_{I,n-1}P_{I,n-1})/(u_{I+1} - u_I).$$

The substitution of these identities into (19) implies

$$\begin{aligned}
&(u_{I+1} - u_I)w_{I+1,n}^2 R' \\
&\quad= n\{((1-\alpha_{I+1,n})w_{I,n-1} + \alpha_{I+1,n}w_{I+1,n-1}) \\
&\qquad\qquad (w_{I+1,n-1}P_{I+1,n-1} - w_{I,n-1}P_{I,n-1}) \\
&\qquad - (w_{I+1,n-1} - w_{I,n-1}) \\
&\qquad\qquad ((1-\alpha_{I+1,n})w_{I,n-1}P_{I,n-1} + \alpha_{I+1,n}w_{I+1,n-1}P_{I+1,n-1})\} \\
&\quad= n\{w_{I+1,n-1}w_{I,n-1}P_{I+1,n-1} - w_{I,n-1}w_{I+1,n-1}P_{I,n-1}\}
\end{aligned}$$

as claimed. ■

We could, in fact, obtain a more general expression. Using exactly the same proof but with different subscripts (see the proof of Proposition 1), it follows that

$$P'_{i,k}(u) = k\frac{w_{i-1,k-1}w_{i,k-1}}{w_{i,k}^2}\left(\frac{P_{i,k-1} - P_{i-1,k-1}}{u_{i-n+k} - u_{i-1}}\right).$$

§4. The hodograph property

In this section we apply Proposition 2 (the hodograph formula) to show that NURBS curves have the hodograph property. A Bézier or B-spline curve $K(u)$ with control points $P_0, \ldots, P_m$ is said to have the hodograph property if the derivative of the curve can be expressed in the form

$$K'(u) = \sum_{i=1}^{m} \lambda_i(u)(P_i - P_{i-1}),$$

where $\lambda_i \geq 0$ for any i or u.

Since it has already been shown that R' is in this form it is a matter of showing $\lambda_i(u) \geq 0$. In order to do this we present two intermediate results (Lemma 4 and Proposition 5). Note the similarity of (20) to (2).

Lemma 4. *For any $i \in \{I-n+2, \ldots, I+1\}$ and $k \in \{1, \ldots, n\}$, the derivative of $N_{i,k}$ satisfies the identity*

$$N'_{i,k} = \frac{k}{k-1}\left\{\left(\frac{u - u_{i-1}}{u_{i+k-1} - u_{i-1}}\right)N'_{i,k-1} + \left(\frac{u_{i+k} - u}{u_{i+k} - u_i}\right)N'_{i+1,k-1}\right\}. \quad (20)$$

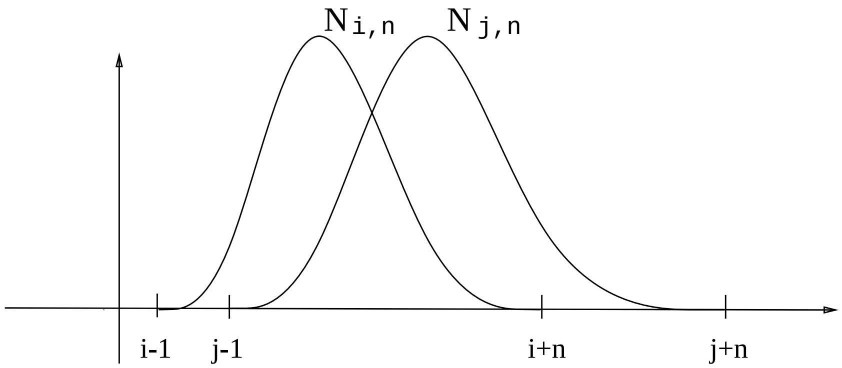

Figure 4. $N'_{j,n}(u)N_{i,n}(u) - N_{j,n}(u)N'_{i,n}(u) \geq 0$ for all u.

Proof: First, we write the left-hand side of (20) in terms of $N_{i,k-1}$ and $N_{i+1,k-1}$ using equation (7). We then go a step further by expanding each of these two terms using (2) and the left-hand side of (20) becomes

$$aN_{i,k-2} + bN_{i+1,k-2} + cN_{i+2,k-2},$$

where a, b, c are functions of u and the knots.

The right-hand side of (20) can also be expanded using (7) applied to each of the terms $N'_{i,k-1}$ and $N'_{i+1,k-1}$. The right-hand side of (20) then becomes

$$AN_{i,k-2} + BN_{i+1,k-2} + CN_{i+2,k-2},$$

where A, B, C are also functions of u and the knots.

It is now a straightforward algebraic exercise to show that $a = A$, $b = B$, and $c = C$. ■

By combining the identities (2) and (20) we now obtain the following result. See Figure 4.

Proposition 5. *Suppose that $N_{i,n}$ and $N_{j,n}$ are any two B-spline basis functions of degree n over any knot sequence and $j > i$. Then*

$$D_{i,j,n}(u) = N'_{j,n}(u)N_{i,n}(u) - N_{j,n}(u)N'_{i,n}(u) \geq 0 \tag{21}$$

for every $u \in \mathbb{R}$ for which $N_{i,n}$ and $N_{j,n}$ are differentiable.

Proof: It follows by induction on the degree n. Recall that the support of $N_{i,n}$ is $[u_{i-1}, u_{i+n}]$ and the support of $N_{j,n}$ is $[u_{j-1}, u_{j+n}]$. Suppose $n = 1$. Then the only non-trivial case occurs when $j = i + 1$ and $u_i < u < u_j$. In that case

$$
\begin{aligned}
N'_{j,1}(u)N_{i,1}(u)-&N_{j,1}(u)N'_{i,1}(u) \\
&=N'_{i+1,1}(u)N_{i,1}(u)-N_{i+1,1}(u)N'_{i,1}(u) \\
&=\left(\frac{1}{u_{i+1}-u_i}\right)\left(\frac{u_{i+1}-u}{u_{i+1}-u_i}\right)-\left(\frac{-1}{u_{i+1}-u_i}\right)\left(\frac{u-u_i}{u_{i+1}-u_i}\right) \\
&=\frac{1}{u_{i+1}-u_i}\geq 0.
\end{aligned}
$$

Now we show that (21) holds assuming it holds when n is replaced by $n-1$ and i and j are replaced by any other integers. We may assume that $j-1<i+n$, otherwise the supports of the two basis functions do not coincide. Now suppose $j-1<u<i+n$. If one rewrites (2) applied to $N_{i,n}$ and $N_{j,n}$ as

$$N_{i,n}=a_1N_{i,n-1}+a_2N_{i+1,n-1}$$

and

$$N_{j,n}=b_1N_{j,n-1}+b_2N_{j+1,n-1}$$

then, from (20),

$$N'_{i,n}=\frac{n}{n-1}(a_1N'_{i,n-1}+a_2N'_{i+1,n-1})$$

and

$$N'_{j,n}=\frac{n}{n-1}(b_1N'_{j,n-1}+b_2N'_{j+1,n-1}),$$

where $a_1,a_2,b_1,b_2\geq 0$. By substituting these expressions into $D_{i,j,n}$ we find that $D_{i,j,n}$ is equivalent to

$$\frac{n}{n-1}(a_1b_1D_{i,j,n-1}+a_1b_2D_{i+1,j,n-1}+a_2b_1D_{i,j+1,n-1}+a_2b_2D_{i+1,j+1,n-1}).$$

But $j>i$ implies $j+1>i+1$, $j+1>i$ and $j\geq i+1$. Thus all terms in this expression are non-negative by the induction step except possibly for $D_{i+1,j,n-1}$ when $j=i+1$ but then $D_{i+1,j,n-1}=D_{j,j,n-1}=0$ anyway. Therefore $D_{i,j,n}\geq 0$ as required. ■

Note that if the two basis functions above happen to be Bézier basis functions, then the proposition can be proved more directly, see Floater [5].

Theorem 6. *NURBS curves have the hodograph property.*

Proof: Proposition 2 demonstrates that R can be written as a linear combination of the direction vectors P_i-P_{i-1}, i.e. in the form of equation (14). It remains to show that each coefficient $\lambda_i(u)$ given by equation (15) is non-negative for all u. But this is now elementary because we know the weights are all positive and Proposition 5 shows that whenever $k>j$,

$$N'_{k,n}(u)N_{j,n}(u)-N_{k,n}(u)N'_{j,n}(u)\geq 0$$

for all u. ■

§5. Bounds on the derivative

As a second application of Proposition 2, we compute an upper bound on the first derivative of a NURBS curve in terms of the control points, knots, and weights from the hodograph formula.

Proposition 7. *The first derivative of the NURBS curve defined by (10) is bounded as follows.*

$$\|R'(u)\| \leq n\frac{W^2}{w^2} \max_{i=1,\ldots,L+n-1} \left\{ \frac{\|P_i - P_{i-1}\|}{u_{i+n-1} - u_{i-1}} \right\}, \tag{22}$$

where

$$w = \min_{i=0,\ldots,L+n-1} \{w_i\} \qquad \textit{and} \qquad W = \max_{i=0,\ldots,L+n-1} \{w_i\}.$$

Proof: First we bound $\lambda_i(u)$ in (15). Suppose $u_I \leq u < u_{I+1}$. Note that because the function $w_{I+1,n}$ is constructed from linear interpolations of the weights, it too is bounded below by w, i.e. $w_{I+1,n}(u) \geq w$ for any u. Then, from (17),

$$\begin{aligned}|\lambda_i| \leq & \frac{W^2}{w^2} \sum_{j=I-n+1}^{i-1} \sum_{k=i}^{I+1} (N'_{k,n} N_{j,n} - N_{k,n} N'_{j,n}) \\ = & \frac{W^2}{w^2} \sum_{j=I-n+1}^{i-1} \sum_{k=I-n+1}^{I+1} (N'_{k,n} N_{j,n} - N_{k,n} N'_{j,n}).\end{aligned}$$

Now it is a well known result from spline theory that

$$\sum_{k=I-n+1}^{I+1} N_{k,n} = 1.$$

Further, by using equation (7),

$$\sum_{j=I-n+1}^{i-1} N'_{j,n} = -n\frac{N_{i,n-1}}{u_{i+n-1} - u_{i-1}}$$

and, since $N_{I+2,n-1} = 0$ for $u_I \leq u < u_{I+1}$,

$$\sum_{k=I-n+1}^{I+1} N'_{k,n} = 0.$$

Therefore

$$|\lambda_i| \leq n\frac{W^2}{w^2} \frac{N_{i,n-1}}{u_{i+n-1} - u_{i-1}}.$$

Since this bound on λ_i is independent of the particular parameter range chosen, we substitute it into (14) and obtain

$$|R'(u)| \leq n\frac{W^2}{w^2} \max_{i=1,\ldots,L+n-1} \left\{ \frac{\|P_i - P_{i-1}\|}{u_{i+n-1} - u_{i-1}} \right\}$$

as required. ∎

References

1. de Boor, C., On calculating with B-splines, J. Approx. Theory **6** (1972), 50–62.
2. de Boor, C., *A Practical Guide to Splines*, Springer-Verlag, New York, 1978.
3. Farin, G., *Curves and surfaces for computer aided geometric design*, Academic Press, San Diego, 1988.
4. Filip, D., R. Magedson, and R. Markot, Surface algorithms using bounds on derivatives, Computer-Aided Geom. Design **3** (1986), 295–311.
5. Floater, M. S., Derivatives of rational Bézier curves, Computer-Aided Geom. Design, to appear.
6. Piegl, L., and W. Tiller, Curve and surface constructions using rational B-splines, Computer-Aided Design **19** (1987), 485–498.
7. Sederberg, T. W., and R. J. Meyers, Loop detection in surface patch intersections, Computer-Aided Geom. Design **5** (1988), 161–171.
8. Sederberg, T. W., and X. Wang, Rational hodographs, Computer-Aided Geom. Design **4** (1988), 333-335.