Inverse Circular Curves

Giulio Casciola [1)], Serena Morigi [1)]

Abstract. In this paper we show how interpreting rational Bézier curves and NURBS as inverse circular curves provides a convenient framework for geometric modelling with curves defined on circular arcs. Focal splines and p-splines are included as special cases.

§1. Introduction

Much of CAGD is based on curves and surfaces defined over intervals or planar domains, respectively. However, there are various applications where it is much more natural to choose the underlying domain to be some other curve or surface. The circle and sphere are of particular importance.

There has been a considerable amount of recent work dealing with curves defined on circular arcs. So-called polar (p-) Bézier curves were first studied in [12] and later in [2,3]. Analogous classes of splines were studied in [2,13] where they are called polar (p)-splines, and in [4,9], where they are called focal splines. Related classes of curves based on trigonometric splines were studied in [7], and certain dual focal splines were used for the design of cams in [10]. All of these curves were recognized as special types of rational Bézier curves or NURBS, respectively.

The aim of this paper is to show that more general classes of rational Bézier curves and NURBS can be used for design over circular domains. These curves and, as we will see, subclasses of them, provide a better alternative to p-Bézier and p-spline from the modelling point of view. Moreover, they present a natural generalization to patches on spherical triangles, that will be considered in a next paper.

Let P be a single-valued curve defined on the unit circular arc $A := \{v(t) : t \in I\}$ with center at the origin, in the form

$$P(t) := \frac{1}{p(t)} v(t), \tag{1.1}$$

where $p(t)$ is a positive scalar function defined over some interval I. Thus $P(t)$ is the point in $\mathbb{R}^2$ which lies at a distance $1/p(t)$ from the origin in the direction of

[1)] Department of Mathematics, University of Bologna, Italy, casciola@dm.unibo.it, morigi@dm.unibo.it. This work was supported by MURST, Cofin. 2000, by INdAM-GNIM special projects 2000, and by University of Bologna "Funds for selected research topics"

the vector $v(t)$. We will call inverse circular curves (ICC) the curve $P(t)$ of the form (1.1).

Alternatively, we can consider

$$P(t) := \frac{u(t)}{w(t)}, \qquad t \in I, \tag{1.2}$$

where $u(t)$ is a curve in $\mathbb{R}^2$ and $w(t)$ is a positive scalar function. Then assuming that

$$\theta(t) := \mathrm{Arg}(u(t))$$

is an increasing function of t, we can regard $v(t)$ as a projection mapping from $U := \{u(t) : t \in I\}$, that we will call projection domain, to the circular domain A by $v(t) := u(t)/\|u(t)\|$. Thus, if we define $p(t) := \frac{w(t)}{\|u(t)\|}$, then the ICC's turn out to be a particular class of the curves defined by (1.2). p-Bézier curves and p-splines are examples of ICC's.

We recall that curves on circular arcs of the form $vp(v)$ were studied in [1,7], where they are called circular Bernstein-Bézier curves. However, as noted there, they are not particularly useful for modelling since the natural associated control curves do not do a good job of predicting the shape of the curve. One can also consider the inverted curves $v/p(v)$, but they do not perform as well as the p-Bézier given in [12].

The paper is organized as follows. In Sect. 2 we recall some notation and basic facts about rational Bézier curves. In Sect. 3 we discuss a class of inverse circular curves defined by a de Casteljau type algorithm, and in Sect. 4 the corresponding class of circular Bernstein basis functions. The special case of p-Bézier is discussed in Sect. 5. Modelling and computational aspects of special ICC subclasses are shown in Sect. 6, and in Sect. 7 we discuss how ICC's can be smoothly joined together. Sect. 8 is devoted to a generalization of the results to NURBS.

§2. Rational Bézier Curves

In this section we recall some well-known facts about rational Bézier curves, see [5,6,11]. Let

$$B_i^n(t) := \binom{n}{i}(1-t)^{n-i}t^i, \qquad 0 \le i \le n,$$

be the classical Bernstein polynomials defined on $[0,1]$. Then a rational Bézier curve is a curve of the form

$$s(t) := \frac{u(t)}{w(t)} = \frac{\displaystyle\sum_{i=0}^{n} w_i b_i B_i^n(t)}{\displaystyle\sum_{i=0}^{n} w_i B_i^n(t)}, \qquad t \in [0,1],$$

where $\{b_i\}_{i=0}^n$ is a set of points in $\mathbb{R}^2$ (called the control points), and $\{w_i\}_{i=0}^n$ is a set of positive real numbers (called the weights).

A point on the curve s can be computed from the following de Casteljau algorithm, [6,11]:

Algorithm 2.1. *Let* $w_i^{(0)} := w_i$ *and* $u_i^{(0)} := w_i b_i$ *for* $i = 0, \ldots, n$, *and let* $t \in [0,1]$.

```
For k = 1 to n
  For i = 0 to n − k
    u_i^(k) := (1 − t) u_i^(k−1) + t u_{i+1}^(k−1)
    w_i^(k) := (1 − t) w_i^(k−1) + t w_{i+1}^(k−1)
Set u(t) := u_0^(n), w(t) := w_0^(n) and s(t) := u(t)/w(t)
```

It is well-known that rational Bézier curves have a convenient control structure defined by the polygon $\mathbb{P}$ which is formed by connecting the points b_i, b_{i+1} with straight lines for $i = 0, \ldots, n-1$. The curve s interpolates at its endpoints and assuming that the weights are positive, it lies in the convex hull of $\mathbb{P}$.

§3. A Class of Inverse Circular Curves

Let $\mathcal{U} := \{u_0, \ldots, u_n\}$ be a set of points in $\mathbb{R}^2$. Considering u_i as a vector, let $\theta_i := \mathrm{Arg}(u_i)$ be the angle between u_i and the x-axis measured in the counterclockwise direction. Suppose that $\theta_0 < \cdots < \theta_n$, and let $c_0, \ldots, c_n$ be positive real numbers. We define an associated ICC S on the unit circular arc $A := \langle \theta_0, \theta_n \rangle$ as follows:

Algorithm 3.1. *Let* $u_i^{(0)} := u_i$ *and* $c_i^{(0)} := c_i$ *for* $i = 0, \ldots, n$, *and let* $t \in [0,1]$.

```
For k = 1 to n
  For i = 0 to n − k
    u_i^(k) := (1 − t) u_i^(k−1) + t u_{i+1}^(k−1)
    c_i^(k) := (1 − t) ||u_i^(k−1)|| / ||u_i^(k)|| c_i^(k−1) + t ||u_{i+1}^(k−1)|| / ||u_i^(k)|| c_{i+1}^(k−1)
Set u(t) := u_0^(n), p(t) := c_0^(n), v(t) := u(t)/||u(t)||.
```

Then the corresponding inverse circular curve is given by $S(t) := \frac{v(t)}{p(t)}$.

It is clear that this algorithm produces the projection domain U

$$u(t) = \sum_{i=0}^{n} u_i B_i^n(t). \tag{3.1}$$

As $\theta(t) := \mathrm{Arg}(u(t))$ is monotone increasing for $t \in [0,1]$, then we can also regard S as a single-valued curve defined on the circular arc A by making use of the one–one correspondence $v(t) = \frac{u(t)}{\|u(t)\|}$ between points on U and points on A. Comparing

Algorithms 2.1 and 3.1, we see that the curve S is just a rational Bézier curve corresponding to setting $w_i = c_i\|u_i\|$ in Algorithm 2.1. As such, it inherits all of the usual properties of rational Bézier curves. Here we briefly discuss a few of these properties.

Using the identification $u_i = w_i b_i$, it is clear that the control polygon $\mathbb{P}$ associated with the curve S is obtained by connecting the control points

$$C_i := \frac{v_i}{c_i}, \qquad i = 0, \ldots, n,$$

where $v_i := u_i/\|u_i\|$ with straight lines. Trivially, the curve S lies in the convex hull of $\mathbb{P}$, and interpolates at its endpoints. Using the connection between the inverse circular curve S produced by Algorithm 3.1 and the rational Bézier curve s produced by Algorithm 2.1, we immediately get algorithms for degree-raising and subdividing ICC curves.

In the following we will show how Algorithm 3.1 can be seen as a de Casteljau like scheme. Given an arc $A := \langle \theta_0, \theta_n \rangle$ on the unit circle with endpoints v_0, v_n, then it is easy to see [1] that any v on A can be uniquely written in the form $v = \beta_1 v_0 + \beta_2 v_n$ with

$$\beta_1 = \frac{\sin(\theta_n - \theta)}{\sin(\theta_n - \theta_0)}, \quad \beta_2 = \frac{\sin(\theta - \theta_0)}{\sin(\theta_n - \theta_0)},$$

where $\theta_i := \mathrm{Arg}(v_i)$ for $i = 0, n$. The numbers β_1, β_2 are called the circular barycentric coordinates of v relative to A and, in general, $\beta_1 + \beta_2 > 1$.

The second statement in the inner loop, in Algorithm 3.1, can be rewritten as

$$c_i^{(k)} := \beta_{i,1}^{(k)} c_i^{(k-1)} + \beta_{i,2}^{(k)} c_{i+1}^{(k-1)},$$

where

$$\beta_{i,1}^{(k)} := (1-t)\frac{\|u_i^{(k-1)}\|}{\|u_i^{(k)}\|}, \qquad \beta_{i,2}^{(k)} := t\frac{\|u_{i+1}^{(k-1)}\|}{\|u_i^{(k)}\|}. \tag{3.2}$$

We note that

$$v_i^{(k)}(t) := \frac{u_i^{(k)}(t)}{\|u_i^{(k)}(t)\|} = (1-t)\frac{u_i^{(k-1)}}{\|u_i^{(k)}\|} + t\frac{u_{i+1}^{(k-1)}}{\|u_i^{(k)}\|} = \beta_{i,1}^{(k)} v_i^{(k-1)}(t) + \beta_{i,2}^{(k)} v_{i+1}^{(k-1)}(t) \tag{3.3}$$

for $i = 0, \ldots, n-k$ and $k = 1, \ldots, n$, where $v_i^{(0)} := v_i$. The relation (3.3) asserts that $\beta_{i,1}^{(k)}$ and $\beta_{i,2}^{(k)}$ are the circular barycentric coordinates of the unit vector $v_i^{(k)}$ in terms of the circular arc $\langle v_i^{(k-1)}, v_{i+1}^{(k-1)} \rangle$. More explicitly, if we define $\theta_i^{(k)} := \mathrm{Arg}(v_i^{(k)})$,

$$\beta_{i,1}^{(k)} = \frac{\sin(\theta_{i+1}^{(k-1)} - \theta(t))}{\sin(\theta_{i+1}^{(k-1)} - \theta_i^{(k-1)})}, \qquad \beta_{i,2}^{(k)} = \frac{\sin(\theta(t) - \theta_i^{(k-1)})}{\sin(\theta_{i+1}^{(k-1)} - \theta_i^{(k-1)})}. \tag{3.4},$$

where $\theta(t) := \theta_i^{(k-1)} + t(\theta_{i+1}^{(k-1)} - \theta_i^{(k-1)})$. Thus Algorithm 3.1 uses different circular baricentric coordinates for i-th arc at each step k. This allows us, as in the classical de Casteljau scheme, to obtain at each step k, values $c_i^{(k)}$, associated to vectors $v_i^{(k)}$ on the unit circular domain A.

§4. A Class of circular Bernstein basis functions

To better understand the nature of S, suppose we run Algorithm 3.1 with all coefficients equal to zero except for $c_i = 1$. Let $\mathcal{B}_i^n(t)$ be the corresponding value of $c_0^{(n)}$. Then

$$p(t) = \sum_{i=0}^{n} c_i \mathcal{B}_i^n(t),$$

where the $\mathcal{B}_0^n, \ldots, \mathcal{B}_n^n$ will be called circular Bernstein basis functions.

The next result shows that there is a close connection between the $\mathcal{B}_i^n(t)$ and the classical Bernstein polynomials.

Theorem 4.1. *For each $0 \leq i \leq n$,*

$$\mathcal{B}_i^n(t) = \frac{\|u_i\|}{\|u(t)\|} B_i^n(t), \tag{4.1}$$

where $u(t)$ is given by (3.1).

Proof: From definitions (1.2) and (1.1), and from the fact that $p(t) = \frac{w(t)}{\|u(t)\|} = \sum_{i=0}^{n} c_i \frac{\|u_i\|}{\|u(t)\|} B_i^n(t)$ the result follows. □

The next results shows that the linear precision property, on the unit circle, holds.

Theorem 4.2. *Given a set of points v_i on a circular arc, then a point $v(t) = (\cos\theta(t), \sin\theta(t))^T$ can be represented as*

$$v(t) = \sum_{i=0}^{n} v_i \mathcal{B}_i^n(t).$$

Proof: *The result follows from Theorem 4.1 or from (3.3).* □

We conclude this section with one example of basis functions $\mathcal{B}_i^n$ for a simple case of $n = 2$, by using relation (4.1) or by the explicit form given in Appendix. Notice that these basis are, in general, irrational functions.

Example 4.3. *Let A be the arc associated with a projection domain U defined by $u_0 := (1,0)^T$,$u_1 := (1/2, 1/2)^T$ and $u_2 := (0,1)^T$, and let $n = 2$. Then the associated basis functions are*

$$\mathcal{B}_0^2(v(t)) = \beta_{01}^{(1)} \beta_{01}^{(2)} = \frac{(1-t)^2}{(2t^2 - 2t + 1)^{1/2}}$$

$$\mathcal{B}_1^2(v(t)) = \beta_{02}^{(1)} \beta_{01}^{(2)} + \beta_{11}^{(1)} \beta_{02}^{(2)} = \frac{(1-t)t\sqrt{2}}{(2t^2 - 2t + 1)^{1/2}},$$

$$\mathcal{B}_2^2(v(t)) = \beta_{12}^{(1)} \beta_{02}^{(2)} = \frac{t^2}{(2t^2 - 2t + 1)^{1/2}}$$

Fig. 1 shows the basis functions $\mathcal{B}_i^2$ (solid lines) and the classical Bernstein basis polynomials B_i^2 (dashed lines) for $i = 0, 1, 2$.

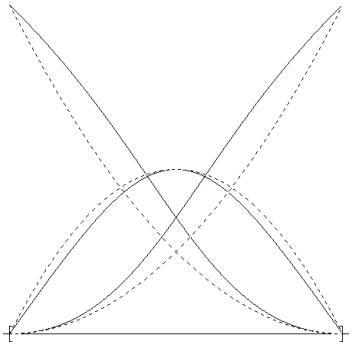

Fig. 1. $\{\mathcal{B}_i\}_{i=0}^2$ (solid lines) and $\{B_i^2\}_{i=0}^2$ (dashed lines).

§5. Polar (p-)Bézier Curves

In this section we show that p-Bézier curves of degree n are just ICC's corresponding to vectors u_i which are chosen to be equally spaced on a circular arc A. Suppose the endpoints of the circular arc are the unit vectors u_0 and u_n with arguments $\theta_0 < \theta_n$. To describe p-Bézier curves, suppose

$$h = (\theta_n - \theta_0)/n, \tag{5.1}$$

and let

$$b_1(t) := \frac{\sin(\frac{\theta_n - \theta(t)}{n})}{\sin(h)}, \qquad b_2(t) := \frac{\sin(\frac{\theta(t) - \theta_0}{n})}{\sin(h)}, \tag{5.2}$$

where

$$\theta(t) := \theta_0 + t(\theta_n - \theta_0).$$

Let

$$\bar{\mathcal{B}}_i^n(t) := \frac{n!}{i!\,(n-i)!}\, b_1(t)^{n-i}\, b_2(t)^i. \tag{5.3}$$

The $\bar{\mathcal{B}}_0^n, \ldots, \bar{\mathcal{B}}_n^n$ are fan-transformed versions of certain basis functions defined in [1].

Now given coefficients $c_0, \ldots, c_n$, the corresponding p- Bézier curve [12] is defined by

$$S(t) = \frac{v(t)}{p(t)}, \qquad 0 \le t \le 1,$$

where

$$p(t) := \sum_{i=0}^{n} c_i \bar{\mathcal{B}}_i^n(t), \tag{5.4}$$

and $v(t)$ is the unit vector with $\mathrm{Arg}(v(t)) = \theta(t)$. Clearly, the p-Bézier curve S can be evaluated by the following version of the de Casteljau algorithm:

Algorithm 5.1. *Suppose* $c_i^{(0)} := c_i$ *for* $i = 0, \ldots, n$. *Given* $t \in [0,1]$, *let* $v(t)$ *be the unit vector with* $\theta(t) = \mathrm{Arg}(v(t))$, *and let* b_1, b_2 *be as in (5.2).*

```
For k = 1 to n
  For i = 0 to n − k
    c_i^(k) := b_1 c_i^(k−1) + b_2 c_{i+1}^(k−1)
```

Set $p(t) := c_0^{(n)}$ and $S(t) = \frac{v(t)}{p(t)}$

We can now identify a p-Bézier curve as an ICC curve.

Theorem 5.2. *Suppose* $u_0, \ldots, u_n$ *are unit vectors with*

$$\theta_i := \mathrm{Arg}(u_i) = \theta_0 + ih, \qquad i = 0, \ldots, n,$$

where h *is given in (5.1). Then the p-Bézier curve* S *produced by Algorithm 5.1 is the same curve as the ICC* S *produced by Algorithm 3.1.*

Proof: By (3.3), it is easy to see by induction that

$$\theta_{i+1}^{(k)} - \theta_i^{(k)} = h, \qquad i = 0, \ldots, n-k-1, k = 1, \ldots, n$$

which implies by (3.4) that

$$\beta_{i,1}^{(k)} = b_1(t), \qquad \beta_{i,2}^{(k)} = b_2(t),$$

for all $i = 0, \ldots, n-k$ and all $k = 1, \ldots, n$. It follows that the value produced by Algorithm 3.1 is the same produces by Algorithm 5.1 as asserted. □

As functions of θ, it follows from (5.2) and (5.3) that the $\bar{\mathcal{B}}_0^n, \ldots, \bar{\mathcal{B}}_n^n$ and thus also the function p defined in (5.4) is a trigonometric polynomial in the space (see [2],[8])

$$\mathcal{T}_n := \begin{cases} \text{span } \{1, \cos(\frac{2\theta}{n}), \sin(\frac{2\theta}{n}), \ldots, \cos(\theta), \sin(\theta)\}, & n \text{ even}, \\ \text{span } \{\cos(\frac{\theta}{n}), \sin(\frac{\theta}{n}), \cos(\frac{3\theta}{n}), \sin(\frac{3\theta}{n}), \ldots, \cos(\theta), \sin(\theta)\}, & n \text{ odd}. \end{cases}$$

§6. Modelling with ICC

In this section we describe the use of ICC for a circular modelling environment comparing flexibility and performance with the p-Bézier approach.

Modelling in a circular modelling environment means to design a single-valued curve that approximates a control polygon defined by $n+1$ control points $(v_i, \frac{1}{c_i})$, where v_i lie on a given circular arc A, and $\frac{1}{c_i}$ represents the distance from the origin in the direction v_i. An interactive modelling tool has to guarantee the single-valued request and provide good shape approximation.

For example, the p-Bézier framework is a good circular modelling environment, as described in [12], where, given a circular domain A and the number $n+1$ of control points, the directions v_i are fixed to be equally spaced on A and c_i are free scalar parameters for modelling. As suggested in [12], the most efficient way to evaluate these curves turns out to be by evaluating their Bézier rational representation, that requires given a domain point $\bar{v} \in A$, the evaluation of $\bar{t}$ by means of an explicit and expensive trigonometric relation [12].

Our ICC proposal is a circular modelling environment extremely flexible because we can exploit the modelling parameters and tools inherited from rational Bézier curves, and we are not limited to equally spaced v_i on A. However, the evaluation of a generic ICC is cumbersome due to the fact that given a domain point $\bar{v} \in A$, we need to determine the corresponding parameter $\bar{t}$ by intersection.

In any case the evaluation of an ICC can be performed by its rational representation and not using algorithm 3.1, that we have proposed only as a theoretical tool.

In this section we present two ICC subclasses that lead to a simplification in the ICC evaluation while keeping good modelling properties in order to get a class of curves more powerful and efficient than p-Bézier curves. These subclasses are characterized by special projection domains U which automatically guarantee that $\mathrm{Arg}(u(t))$ is an increasing function of t, thus obtaining single-valued curves.

6.1. ICC on linear projection domain

Suppose we choose two arbitrary vectors $u_0, u_n \in \mathbb{R}^2$ with $\mathrm{Arg}(u_0) < \mathrm{Arg}(u_n)$, then we define the u_i equally spaced in the straight segment defined by u_0, u_n, such that U is the linear projection domain defined as $u(t) = (1-t)u_0 + tu_n = \sum_{i=0}^{n} u_i B_i^n(t)$.

A more advantageous way, with respect to Algorithm 2.1 or 3.1, to evaluate an ICC derives directly from its representation in the form $S(t) = \frac{u(t)}{w(t)} = u(t)/\sum_{i=0}^{n} \|u_i\| c_i B_i^n(t)$. In fact, given $\bar{v} = v(\bar{t})$ on A, the corresponding value $S(\bar{t})$ can be evaluated following the steps:

1. Compute $\bar{t}$ as intersection between $u(t)$ and $\{\alpha\bar{v} : \alpha \geq 0\}$;
2. Compute $u(\bar{t})$;
3. Apply de Casteljau algorithm to compute the scalar function $w(t)$;
4. Set $S(\bar{t}) := \frac{u(\bar{t})}{\|u_0^n\| c_0^n}$.

In this case of linear projection domain U, step 1 trivially reduces to an explicit and simple formula, while step 2 is given by $u(\bar{t}) = u_0 + \bar{t}(u_n - u_0)$. Thus the ICC evaluation algorithm has a total cost of $\frac{1}{2}n(n+1)+6$ multiplications/divisions, and $\frac{1}{2}n(n+1)+6$ additions/subtractions.

This choice of projection domain is particularly easy to work with in a circular modelling environment. Fig. 2 shows several curves corresponding to linear projection domains. Note that in all figures presented the ICC curves are drawn with solid lines while their control polygons are drawn with dashed lines and the control points are marked with open disks. The u_i are marked with black dots and the circular domain, together with black arrows denoting the v_i, is also illustrated.

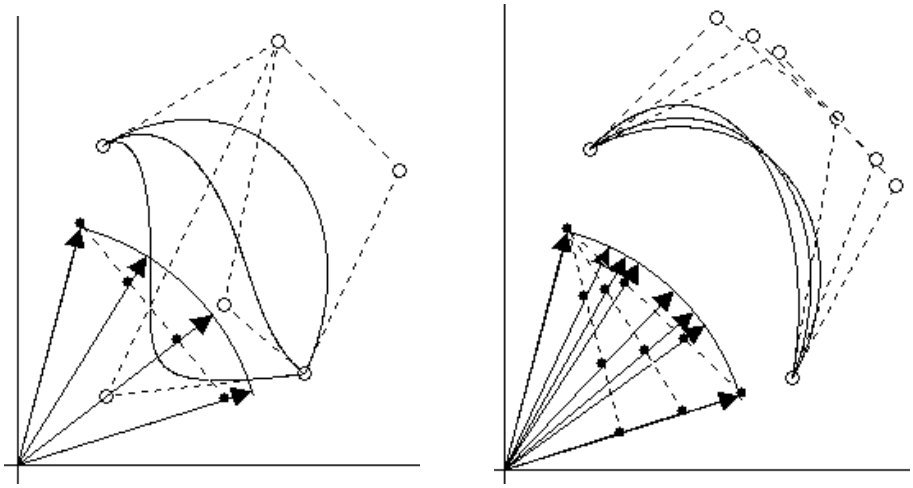

Fig. 2. ICC with linear projection domain U, $n = 3$; (left) the pulling effect of a control point; (right) the change of $\|u_0\|$ with the values $\{1.0, 0.75, 0.5\}$.

Fig. 2 on the left shows the effect of adjusting the values of the c_i while holding the v_i fixed. This moves the control points and pulls the curve along with them. Note that, in this case, only the modulus of each control point can be modified, while in a general ICC, also the control point positions can be changed, but constrained to keep the increasing angular order.

Fig. 2 on the right shows the effect of changing the module of the vector u_0 while holding the c_i fixed. This leads to alter the position of the v_i and consequently of the control points.

6.2. ICC on quadratic projection domain

The case where $u(t)$ is a quadratic projection domain is important because it allows the construction of conic sections. To obtain a quadratic projection domain U, we can begin with any three vectors $\bar{u}_0, \bar{u}_1, \bar{u}_2$, with $\mathrm{Arg}(\bar{u}_0) < \mathrm{Arg}(\bar{u}_1) < \mathrm{Arg}(\bar{u}_2)$, and create the curve

$$u(t) := \sum_{i=0}^{2} \bar{u}_i B_i^2(t). \tag{6.1}$$

In order to define $u(t)$ as in (3.1) we can degree elevate $u(t)$ in (6.1) thus obtaining the vectors $u_0, \ldots, u_n$.

Concerning the evaluation of this ICC subclass we refer to the procedure given in subsection 6.1 where steps 1 and 2 are suitable modified by using (6.1). This algorithm has a total cost of $\frac{1}{2}n(n+1)+11$ multiplications/divisions, a square root and $\frac{1}{2}n(n+1)+12$ additions/subtractions.

Figure 3 shows the representation of a quarter circle with center at the origin and radius 2 as an ICC with a quadratic projection domain U. Here $u_0 = (1,0)^T$, $u_1 = (1/\sqrt{2}, 1/\sqrt{2})^T$, and $u_2 = (0,1)^T$.

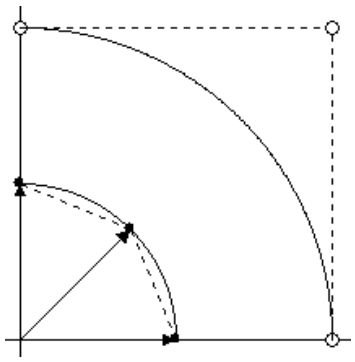

Fig. 3. ICC with quadratic projection domain representing a quarter of circle of radius 2.

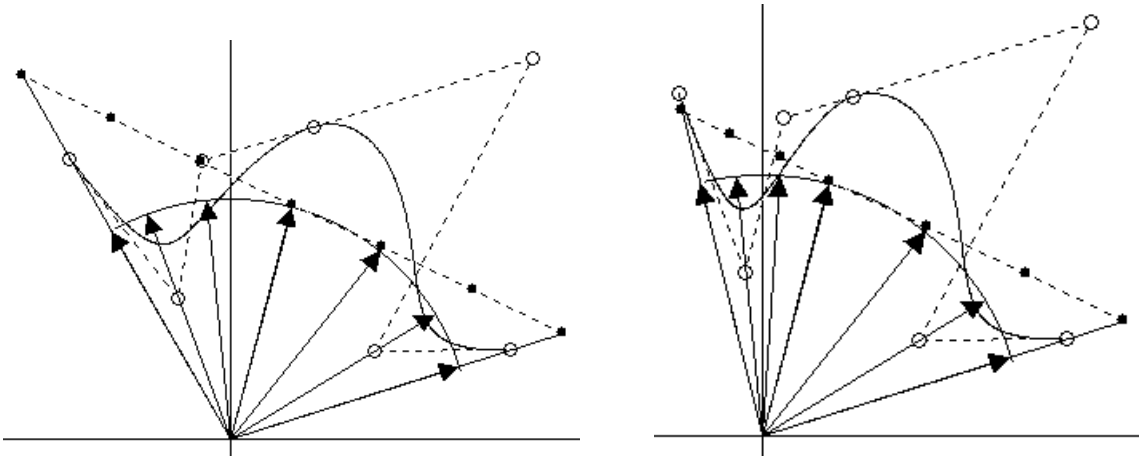

Fig. 4. S and $\tilde{S}$ joining with C^1 (left) and G^1 (right) parametric continuity.

The two proposed subclasses result more computational efficient than the p-Bézier proposal and more flexible for a circular modelling environment. These advantages will be more significant in the circular spline setting (see section 8).

§7. Joining ICC's Smoothly

In this section we briefly explore the question of how to join two inverse circular curves smoothly. Suppose S and $\tilde{S}$ are the inverse circular curves defined for $t \in [0, \Delta]$ and $\widetilde{t} \in [0, \widetilde{\Delta}]$ corresponding to $\{u_i\}_{i=0}^n$, $\{c_i\}_{i=0}^n$ and $\{\widetilde{u}_i\}_{i=0}^n$, $\{\widetilde{c}_i\}_{i=0}^n$, respectively. We denote the corresponding control points of S and $\tilde{S}$ by $C_i := v_i/c_i$ and $\widetilde{C}_i := \widetilde{v}_i/\widetilde{c}_i$, for $i = 0, \ldots, n$.

Clearly, S and $\tilde{S}$ join with C^0 continuity at $\widetilde{C}_0$ if and only if

$$\widetilde{C}_0 = C_n$$

that is

$$\widetilde{c}_0 = c_n \quad and \quad \|\widetilde{u}_0\| = \alpha \|u_n\|, \tag{7.1}$$

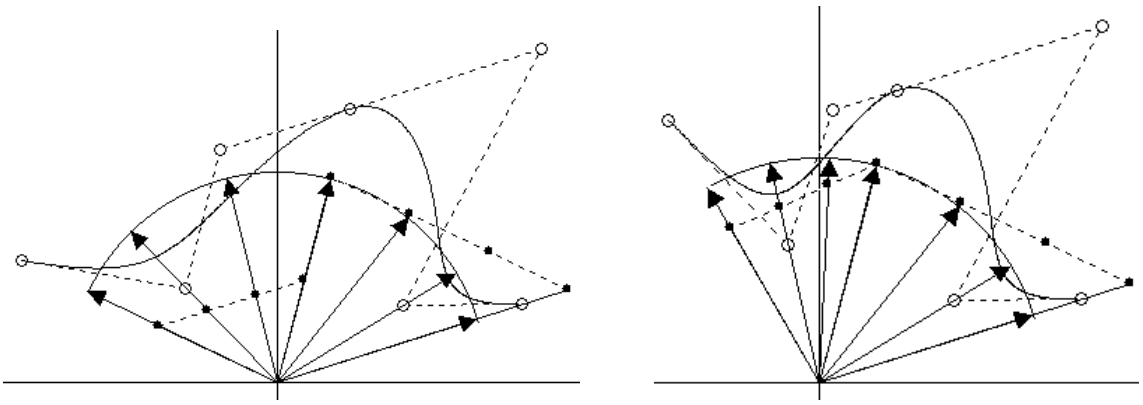

Fig. 5. S and $\tilde{S}$ joining with C^1 (left) and G^1 (right) rational continuity.

where α is a free non zero parameter (see Fig.5, where $\alpha = 0.5$ on the left, and $\alpha = 1$ on the right). Moreover, by the well-known results on rational Bézier curves [5,6], S and $\tilde{S}$ join with G^1 continuity at $\widetilde{C}_0$ (i.e., their tangents at $\widetilde{C}_0$ point have the same direction) if and only if $C_{n-1}, C_n, \widetilde{C}_0, \widetilde{C}_1$ lie on a common line. This is simply given modifying $\widetilde{c}_1$ in the following manner:

$$\widetilde{c}_1 := \frac{\sin(\theta_n - \theta_{n-1})}{\sin(\widetilde{\theta}_1 - \theta_{n-1})c_n + \sin(\theta_n - \widetilde{\theta}_1)c_{n-1}} \tag{7.2}$$

where $\widetilde{\theta}_i = \mathrm{Arg}(\widetilde{u}_i)$ $(\theta_i = \mathrm{Arg}(u_i))$.

They join with C^1 continuity at $\widetilde{C}_0$ (i.e., they have the same tangent at $\widetilde{C}_0$) if and only if in addition to (7.1), we have

$$\frac{1}{\widetilde{\Delta}}(\widetilde{C}_1 - \widetilde{C}_0)\frac{\widetilde{c}_1\|\widetilde{u}_1\|}{\widetilde{c}_0\|\widetilde{u}_0\|} = \frac{1}{\Delta}(C_n - C_{n-1})\frac{c_{n-1}\|u_{n-1}\|}{c_n\|u_n\|}. \tag{7.3}$$

In practice, mantaining the given v_1 direction, we compute $\widetilde{c}_1$ by (7.2), thus obtaining a modified $\widetilde{C}_1$ and we choose $\|u_1\|$ by relation (7.3) so that the two tangents at $\widetilde{C}_0$ have also the same modulo.

The above formula comprises both the sufficient continuity conditions on the components of the rational curve represented in homogeneous coordinates (parametric continuity), and the necessary and sufficient conditions on the components of the rational curve itself that lead to the so-called rational continuity, see [5,6].

Fig. 4 and 5 show examples of two inverse circular curves S and $\tilde{S}$ with $n = 3$ joined together with C^1 (left) and G^1 (right) continuity in the linear projection domain case. In this case the C^1 parametric continuity requires that U and $\widetilde{U}$ are aligned (see Fig. 4), while the C^1 rational continuity allows linear projection domains which are not aligned with each other (see Fig. 5). As a consequence of this, using a sequence of such projection domains, we can define inverse circular curves on circular arcs of arbitrary length, where each segment of the curve is defined over a circular arc of length less than π.

§8. A class of ICC spline

We first recall some well-known facts about NURBS curves, see [5,6,11]. Given positive integers m and n, suppose $t_0 = \cdots = t_m \le t_{m+1} \le \cdots \le t_{n+1} = \cdots = t_{n+m+1}$ where $t_{i+m+1} > t_i$ for all i. Let $N_0^m, \ldots, N_n^m$ be the corresponding normalized B-splines of degree m (order $m+1$, see [14]). Then a non-uniform rational B-spline (NURBS) is a curve of the form

$$s(t) := \frac{u(t)}{w(t)} = \frac{\displaystyle\sum_{i=0}^{n} w_i b_i N_i^m(t)}{\displaystyle\sum_{i=0}^{n} w_i N_i^m(t)}, \qquad t \in [t_m, t_{n+1}],$$

where $\{b_i\}_{i=0}^n$ is a set of points in $\mathbb{R}^2$ (called the control points), and $\{w_i\}_{i=0}^n$ is a set of positive real numbers (called the weights). If we connect b_i and b_{i+1} with a straight line for each $i = 0, \ldots, n-1$, the resulting curve is called the control polygon.

It is well known that $s(t_m) = b_0$, $s(t_{n+1}) = b_n$, and that the curve s is tangent to the control polygon at the points b_0 and b_n. It is also known that if the weights are all nonnegative, then the curve s lies in the convex hull of the control polygon. The following algorithm [6,11] can be used to compute NURBS:

Algorithm 8.1. *Let* $w_i^{(0)} := w_i$ *and* $u_i^{(0)} := w_i b_i$ *for* $i = 0, \ldots, n$, *and let* $t \in [t_\ell, t_{\ell+1})$ *for some* $m \le \ell \le n$.

`For` $k = 1$ `to` m
  `For` $i = \ell - m$ `to` $\ell - k$

$$\alpha_i^{(k)} := \frac{(t - t_{i+k})}{(t_{i+m+1} - t_{i+k})}$$

$$u_i^{(k)} := (1 - \alpha_i^{(k)}) u_i^{(k-1)} + \alpha_i^{(k)} u_{i+1}^{(k-1)}$$

$$w_i^{(k)} := (1 - \alpha_i^{(k)}) w_i^{(k-1)} + \alpha_i^{(k)} w_{i+1}^{(k-1)}$$

`Set` $u(t) := u_{\ell-m}^{(m)}$, $w(t) := w_{\ell-m}^{(m)}$, `and` $s(t) := \frac{u(t)}{w(t)}$

To get an associated class of inverse circular curves, we can follow the construction of Sect. 3. Suppose $\mathcal{U} := \{u_0, \ldots, u_n\}$ is a set of vectors in $\mathbb{R}^2$, and let $\theta_i := \mathrm{Arg}(u_i)$ be the angle between u_i and the x-axis measured in the counterclockwise direction. Suppose that $\theta_0 < \cdots < \theta_n$, and let $c_0, \ldots, c_n$ be positive real numbers.
We define an associated ICC spline S by the following algorithm:

Algorithm 8.2. *Let* $u_i^{(0)} := u_i$ *and* $c_i^{(0)} := c_i$ *for* $i = 0, \ldots, n$, *and let* $t \in [t_\ell, t_{\ell+1})$ *for some* $m \le \ell \le n$.

`For` $k = 1$ `to` m

$\texttt{For}\ \ i = \ell - m\ \ \texttt{to}\ \ \ell - k$

$$\alpha_i^{(k)} := \frac{(t - t_{i+k})}{(t_{i+m+1} - t_{i+k})}$$

$$u_i^{(k)} := (1 - \alpha_i^{(k)}) u_i^{(k-1)} + \alpha_i^{(k)} u_{i+1}^{(k-1)}$$

$$c_i^{(k)} := (1 - \alpha_i^{(k)}) \frac{\|u_i^{(k-1)}\|}{\|u_i^{(k)}\|} c_i^{(k-1)} + \alpha_i^{(k)} \frac{\|u_{i+1}^{(k-1)}\|}{\|u_i^{(k)}\|} c_{i+1}^{(k-1)}$$

$\texttt{Set}\ \ u(t) := u_{\ell-m}^{(m)},\ \ p(t) := c_{\ell-m}^{(m)},\ \ v(t) := \frac{u(t)}{\|u(t)\|}.$

Then the associated ICC is given by $S(t) := \frac{v(t)}{p(t)}$

It is clear that this algorithm produces the projection domain U

$$u(t) = \sum_{i=0}^{n} u_i N_i^m(t).$$

Comparing Algorithms 8.1 and 8.2, we see that the curve S is just a NURBS curve s corresponding to setting $w_i = c_i \|u_i\|$ in Algorithm 8.1.

For each $0 \le i \le n$, let $\mathcal{N}_i^m$ be the function which is defined by Algorithm 8.2 using the coefficient vector with all zero components except for $c_i = 1$. Then

$$p(t) = \sum_{i=0}^{n} c_i \mathcal{N}_i^m(t).$$

Following the proof of 4.1, it is not hard to see that

$$\mathcal{N}_i^m(t) = \frac{\|u_i\|}{\|u(t)\|} N_i^m(t). \tag{8.1}$$

Since these ICC's are just special cases of NURBS useful in a circular environment, it is clear that they inherit all the advantages of NURBS. In particular, we can define a control polygon $\mathbb{P}$ by connecting the control points $C_i := v_i/c_i$ and $C_{i+1} = v_{i+1}/c_{i+1}$ with straight lines for $i = 0, \ldots, n-1$. Then the curve S lies in the convex hull of $\mathbb{P}$, and interpolates it at the endpoints. Moreover, the well-known algorithms for degree raising, knot insertion, and subdivision can be applied.

As $\theta(t) := \mathrm{Arg}(u(t))$ is monotone increasing for $t \in [0, 1]$, then we can also regard S as single-valued curve defined on the circular arc $A := \langle \theta_0, \theta_n \rangle$ by making use of the one–one correspondence $v(t) = \frac{u(t)}{\|u(t)\|}$ between points on U and points on A.

As in the rational Bézier curve case, in practice we propose to use the special ICC spline subclasses with linear or quadratic projection domain U.

Fig. 6 shows two examples of ICC splines. In both cases $m = 3$ and $n = 11$, and the B-splines are defined using the knot sequence $\{t_i\}_{i=0}^{15}$ with $t_0 = t_1 = t_2 = t_3 = 0$

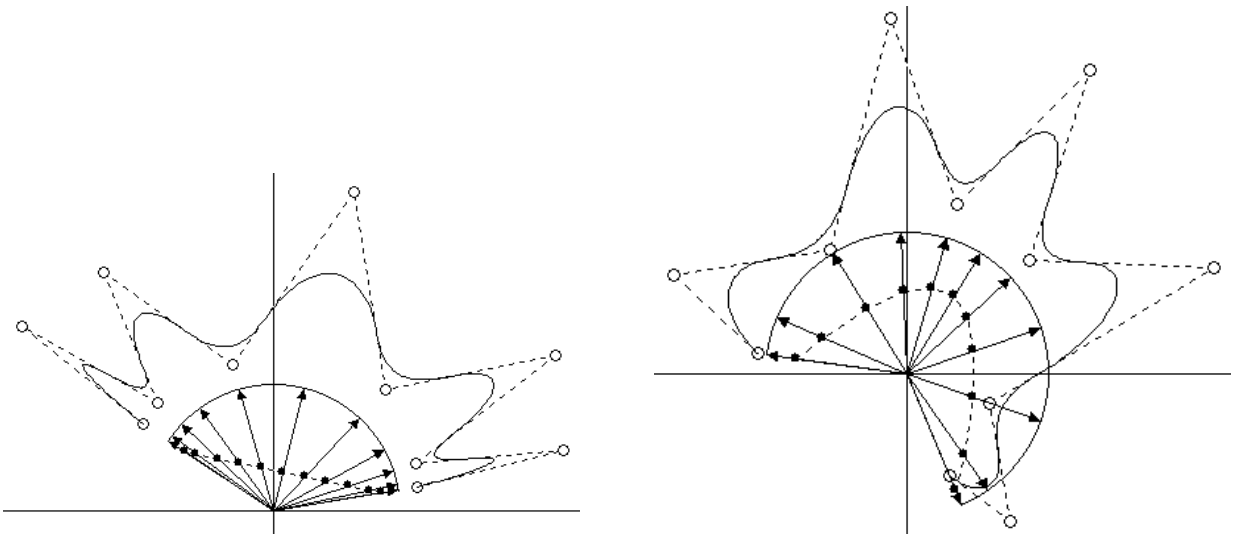

Fig. 6. Two ICC splines of degree 2.

and $0 < t_4 < \cdots < t_{11} < 1$ and $t_{12} = t_{13} = t_{14} = t_{15} = 1$. In the figure on the left U is a linear projection domain, while on the right it is a quadratic projection domain.

Analogously to what has been asserted by Theorem 5.2, we can establish the following connection between p-splines and the inverse circular curves described by Algorithm 8.2.

Theorem 8.3. *An ICC spline* $S(t) = \frac{v(t)}{p(t)}$, *where* $p(t) = \sum_{i=0}^{n} c_i \mathcal{N}_i^m(t)$, *with knots* $\{t_i\}$ *is a p-spline curve* $S(\tau) = \frac{v(\tau)}{p(\tau)}$, *where* $p(\tau) = \sum_{i=0}^{n} c_i \bar{\mathcal{N}}_i^m(\tau)$, *and knots* $\{\tau_i\}$, *if* $t = \tan(\tau)$, *the projection domain* U *for the ICC is defined by vectors* u_i *with* $\|u_i\| = \frac{1}{\prod_{j=i+1}^{i+m} \cos \tau_j}$, *and* $v_i = \frac{u_i}{\|u_i\|} = (\cos \xi_i, \sin \xi_i)$, *where* $\xi_i = \sum_{j=i+1}^{i+m} \tan^{-1}(t_j)$.

Proof: The proof is simply reduced to verify that

$$\bar{\mathcal{N}}_i^m(\tau) = \mathcal{N}_i^m(t). \tag{8.2}$$

Using the following relation given in [3]

$$\bar{\mathcal{N}}_i^m(\tau) = \frac{\cos(\tau)^m}{\prod_{j=i+1}^{i+m} \cos \tau_j} N_i^m(t)$$

where $\tau \in [-\pi/2, \pi/2]$ and $t \in [0, 1]$, and relation (8.1), we have

$$\bar{\mathcal{N}}_i^m(\tau) = \frac{\cos(\tau)^m}{\prod_{j=i+1}^{i+m} \cos \tau_j} \frac{\|u(t)\|}{\|u_i\|} \mathcal{N}_i^m(t). \tag{8.3}$$

Now,

$$\|u(t)\| = \|\sum_{i=0}^{n} u_i N_i^m(t)\| = \|\sum_{i=0}^{n} u_i \frac{\prod_{j=i+1}^{i+m} \cos \tau_j}{\cos(\tau)^m} \bar{\mathcal{N}}_i^m(\tau)\|$$

and applying the linear precision property $v = \sum_{i=0}^{n} v_i \bar{\mathcal{N}}_i^m(\tau)$(*see*[3]) and relation given for $\|u_i\|$,

$$\|u(t)\| = \frac{1}{\cos(\tau)^m}. \tag{8.4}$$

Replacing (8.4) and relation given for $\|u_i\|$ in (8.3) we have (8.2). □

The advantage that we have got computationally in the ICC case with respect to the p-Bézier is even more magnified in the ICC-spline case. In fact, by a computational point of view, the ICC spline with linear or quadratic projection domain can be evaluated using procedure suggested in subsection 6.1, where de Casteljau scheme in step 3. is replaced by the de Boor scheme, while p-splines require an expensive conversion to NURBS (see [2]) and the NURBS curve evaluation itself that is certainly more expensive.

§9. Appendix

An explicit representation of the basis function $\mathcal{B}_i^n$ can be derived from the algorithm 3.1. Each of the basis functions $\mathcal{B}_i^n$ consists of a sum of products of the form

$$\prod_{k=1}^{n} \beta_{i_k,m_k}^{(k)},$$

with $0 \le i_k \le i$ and $m_k \in \{1,2\}$. The number of such terms in the sum is equal to the number of paths in the de Casteljau diagram leading from the coefficient $c_i^{(0)}$ to the value $\mathcal{B}_i^n(t) = c_n^{(0)}$. To make this more precise, let $\mathcal{I}_i$ be the set of distinct n-vectors obtained by permuting the components of $(1,\ldots,1,2,\ldots,2)$, where 1 appears $n-i$ times and 2 appears i times.

Theorem 9.1. *For all* $0 \le i \le n$,

$$\mathcal{B}_i^n(t) = \sum_{(m_1,\ldots,m_n)\in\mathcal{I}_i} \prod_{k=1}^{n} \beta_{i_k,m_k}^{(k)}, \tag{9.1}$$

where

$$i_k := i + \sum_{\nu=1}^{k} (1 - m_\nu).$$

Proof: Each vector $(m_1,\ldots,m_n)$ describes a distinct path from $c_i^{(0)}$ to $c_0^{(n)}$. The value $m_k = 1$ corresponds to moving down and to the right in the de Casteljau array. This corresponds to multiplying by a $\beta_{i_k,1}^{(k)}$ factor. $m_k = 2$ corresponds to moving down and to the left, and corresponds to multiplying by a $\beta_{i_k,2}^{(k)}$ factor. □

Acknowledgments. We would like to thank Larry L. Schumaker for many productive discussions and his valuable suggestions.

References

1. Alfeld, P., M. Neamtu, and L. L. Schumaker, Circular Bernstein-Bézier polynomials, in *Curves and Surfaces with Applications in CAGD,* P.J. Laurent, A. Le Méhauté, and L. L. Schumaker (eds), A.K.Peters, Wellesley, 1994, 11–20.
2. Casciola, G. and S. Morigi, Spline curves in polar and Cartesian coordinates, in *Curves and Surfaces with Applications in CAGD,* P.J. Laurent, A. Le Méhauté, and L. L. Schumaker (eds), A.K.Peters, Wellesley, 1994, 61–68.
3. Casciola, G., S. Morigi, and J. Sanchez-Reyes, Degree elevation for p-Bézier curves, Comput. Aided Geom. Design **15** (1998), 313–322.
4. de Casteljau, P., Splines focales, in *Curves and Surfaces with Applications in CAGD,* P.J. Laurent, A. Le Méhauté, and L. L. Schumaker (eds), A.K.Peters, Wellesley, 1994, 91–103.
5. Farin, G., *Curves and Surfaces for Computer-Aided Geometric Design. A Practical Guide,* Academic Press, San Diego, 1997.
6. Hoschek, J. and D. Lasser, *Fundamentals of Computer Aided Geometric Design,* A. K. Peters, Boston, 1993.
7. Koch, P. E., M. Neamtu, T. Lyche, and L. L. Schumaker, Control curves and knot insertion for trigonometric splines, Advances in Comp. Math. **3** (1995), 405–424.
8. Morigi, S. and M. Neamtu, Some results for a class of generalized polynomials, Adv. Comput. Math. **12** (2000), 133–149.
9. Neamtu, M., H. Pottmann, and L. L. Schumaker, Dual focal splines and rational curves with rational offsets, Math. Eng. Ind. **7** (1999), 139–154.
10. Neamtu, M., H. Pottmann, and L. L. Schumaker, Designing NURBS cam profiles using trigonometric splines, ASME J. Mech. Design **120** (1998), 175–180.
11. Piegl, L. and W. Tiller, *The NURBS book,* Springer-Verlag, Berlin, 1997.
12. Sánchez-Reyes, J., Single-valued curves in polar coordinates, Computer-Aided Design **22** (1990), 19–26.
13. Sánchez-Reyes, J., Single-valued spline curves in polar coordinates, Computer-Aided Design **24** (1992), 307–315.
14. Schumaker, L. L., *Spline Functions: Basic Theory,* Wiley, New York, 1981.