Spline curves in polar and Cartesian coordinates

Giulio Casciola and Serena Morigi
Department of Mathematics, University of Bologna, Italy

Abstract. A new class of spline curves in polar coordinates has been presented in [11] and independently considered in [5] . These are rational trigonometric curves in Cartesian coordinates and can be represented as NURBS. An alternative way to derive some useful tools for modelling splines in polar coordinates is provided. Moreover, an ad hoc algorithm of degree elevation for splines in polar coordinates is presented, and its efficiency and stability is proved.

§1. Introduction

Recently, in [11] a class of spline curves in polar coordinates was proposed. We refer to these curves as p-splines. They have proved to be a generalization of those considered in [10], which we call p-Bézier curves.

The p-splines were independently considered in [5], and called Focal splines. These classes of curves are interesting because they allow for modelling and interpolation of free forms in polar coordinates with the same facilities as Cartesian splines.

In [11], Sánchez-Reyes emphasizes the fact that the p-spline curves are piecewise rational Bézier in Cartesian coordinates but they are not rational splines. Actually, this last assertion is neither proved nor supported by any justification. In this paper we will provide the algorithm that leads to a representation of these curves as NURBS.

In addition to knot insertion, knot removal and subdivision, another known result from Sánchez-Reyes' papers is the possibility of making degree elevation from degree n to degree kn. In [2] and [4] we proposed different algorithms for degree elevation of p-Bézier curves. In this paper we will suggest how to use these results for p-splines, and we will provide some computational results.

Proceedings of Chamonix 1996
A. Le Méhauté, C. Rabut, and L. L. Schumaker (eds.), pp. 1–8.

ISBN 1-xxxxx-xxx-x.

These two results, together with their generalization to surfaces, have convinced us of the usefulness of extending our NURBS-based modelling system by supplying it with a modelling environment for p-spline curves and surfaces in polar, spherical, and mixed polar-Cartesian coordinates. This allows us to manage polar and spherical models as NURBS [3].

§2. P-spline curves

A p-spline curve $\underline{c}(t)$ of degree n is defined as

$$\underline{c}(t) = \begin{pmatrix} \rho(t) \\ \theta(t) \end{pmatrix} = \begin{pmatrix} \frac{1}{\sum_{i=0}^{K+n} \delta_i M_{i,n}(t)} \\ nt \end{pmatrix}$$

where $\theta(t)$ denotes the polar angle and $\rho(t)$ is the radius defined as the reciprocal of a trigonometric spline. Without loss of generality, we consider $t \in [-\Delta, \Delta]$. The functions $M_{i,n}(t)$ are normalized trigonometric B-splines [8] and are defined by the following recurrence relation

$$M_{i,n}(t) = \frac{\sin(t - t_i)}{\sin(t_{i+n} - t_i)} M_{i,n-1}(t) + \frac{\sin(t_{i+n+1} - t)}{\sin(t_{i+n+1} - t_{i+1})} M_{i+1,n-1}(t) \qquad (1)$$

$$M_{i,0}(t) = \begin{cases} 1, & \text{if} \quad t_i \le t < t_{i+1}, \\ 0, & \text{otherwise.} \end{cases}$$

on a non-decreasing knot sequence $\{t_i\}_{i=0}^{K+2n+1}$ satisfying the constraint $t_{i+n} - t_i < \pi$, $\forall i$. Note that each trigonometric spline piece is a trigonometric polynomial belonging to the space

$$T_n = \begin{cases} span\{1, \cos 2t, \sin 2t, \cos 4t, \sin 4t, ..., \cos nt, \sin nt\}, & \text{n even} \\ span\{\cos t, \sin t, \cos 3t, \sin 3t, ..., \cos nt, \sin nt\}, & \text{n odd} \end{cases}$$

The coefficients δ_i^{-1} and the Greville radial directions $\xi_i = \sum_{j=i+1}^{i+n} t_j$ define, in polar coordinates, the control points $\underline{d_i} = \left(\xi_i, \delta_i^{-1}\right)$ of the p-spline $\underline{c}(t)$. The knot constraint implies that, in polar coordinates, $\xi_i - \xi_{i-1} < \pi$ holds. Figure 1 illustrates an example of a p-spline curve and relative control polygon.

P-spline curves enjoy properties of local control, linear precision, convex hull, and variation diminishing inherited from splines in Cartesian coordinates. Moreover, p-splines of degree 2 are conic sections with foci at the origin of the coordinates.

§3. NURBS representation of p-splines

It is known [11] that the class of p-spline restricted to a single segment (p-Bézier curves) represents a subclass of rational Bézier curves in Cartesian coordinates. Therefore, we can state that p-splines represent a subclass of NURBS.

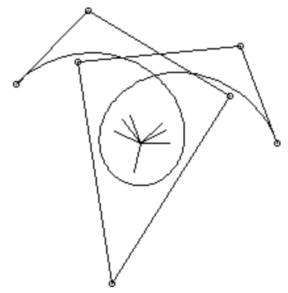


Fig. 1. p-spline curve of degree $n = 3$ together with its control polygon - $\{t_i\} = \{0, 0, 0, 0, 0.75, 1.5, 2.25, 3, 3, 3, 3\}$, $\underline{d_i} = \{(0, 0.3), (0.75, 0.3), (2.25, 0.22), (4.5, 0.3), (6.75, 0.22), (8.25, 0.3), (9, 0.3)\}$.

A first approach to obtain a NURBS representation of a p-spline curve has been suggested in [11]. Given a p-spline curve over an arbitrary knot sequence, this can be converted by subdivision into a piecewise curve whose individual pieces are p-Bézier curves, so that every p-Bézier curve can be represented in terms of rational Bézier curves in Cartesian coordinates.

In the alternative approach proposed here, a non-piecewise Bézier representation of a p-spline $\underline{c}(t)$ as a NURBS curve $\underline{q}(v)$ will be provided.

Let $\underline{c}(t)$ be the p-spline represented as a scalar function

$$\rho(\frac{\theta}{n}) = \frac{1}{\sum_{i=0}^{K+n} \delta_i M_{i,n}(\frac{\theta}{n})} \tag{2}$$

where $\theta \in [-n\Delta, n\Delta]$. Then the correspondent curve of (2) in Cartesian coordinates will be obtained by a simple change of coordinates:

$$\rho(\frac{\theta}{n}) \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \tag{3}$$

Applying the identities [6]

$$\cos\theta = \sum_{i=0}^{K+n} \cos\xi_i M_{i,n}(\tfrac{\theta}{n}) \qquad \sin\theta = \sum_{i=0}^{K+n} \sin\xi_i M_{i,n}(\tfrac{\theta}{n})$$

relation (3) assumes the following trigonometric rational form:

$$\frac{\sum_{i=0}^{K+n} \binom{\cos\xi_i}{\sin\xi_i} M_{i,n}(\frac{\theta}{n})}{\sum_{i=0}^{K+n} \delta_i M_{i,n}(\frac{\theta}{n})} \tag{4}$$

In [7] the important transformation $\gamma_n : P_n -> T_n$; $(\gamma_n f)(x) = \cos^n x \cdot f(\tan x)$, was provided; more precisely, if $p \in P_n$ on $[\tan\alpha, \tan\beta]$, then $\gamma_n p \in T_n$ on $[\alpha, \beta]$ when $-\frac{\pi}{2} < \alpha < \beta < \frac{\pi}{2}$. From this assertion it follows that a polynomial B-spline is proportional to a trigonometric B-spline. In particular, the following important relation can easily be proved:

$$M_{i,n}(t) = \frac{\cos^n t}{\prod_{j=i+1}^{i+n} \cos t_j} N_{i,n}(\varphi(t)) \tag{5}$$

$$\varphi(t) = \tan t \qquad -\frac{\pi}{2} < t < \frac{\pi}{2}$$

where the $N_{i,n}$ are the polynomial B-spline functions defined on the knot sequence $\{\varphi(t_i)\}$. In virtue of (5), relation (4) becomes

$$\underline{q}(v) = \frac{\sum_{i=0}^{K+n} \begin{pmatrix} \cos\xi_i \\ \sin\xi_i \end{pmatrix} \left(\prod_{j=i+1}^{i+n} \cos t_j \right)^{-1} N_{i,n}(v)}{\sum_{i=0}^{K+n} \left(\prod_{j=i+1}^{i+n} \cos t_j \right)^{-1} \delta_i N_{i,n}(v)}$$

where

$$v = \varphi(t) = \frac{1}{2}\left[1 + \frac{\tan t}{\tan \Delta}\right] \tag{6}$$

Thus, we can conclude that a p-spline in Cartesian coordinates has the following NURBS representation:

$$\underline{q}(v) = \frac{\sum_{i=0}^{K+n} P_i w_i N_{i,n}(v)}{\sum_{i=0}^{K+n} w_i N_{i,n}(v)} \qquad v \in [0,1]\,, \tag{7}$$

with weights $w_i = \delta_i / (\prod_{j=i+1}^{i+n} \cos t_j)$ and control points $P_i = \delta_i^{-1} \binom{\cos\xi_i}{\sin\xi_i}$; the $N_{i,n}(v)$ functions are defined over a knot sequence $\{v_i\}$ obtained applying relation (6) to the knots t_i. Note that the P_i are given by the transformation in Cartesian coordinates of the p-spline control points $\underline{d_i}$.

For example, the NURBS representation of the p-spline curve illustrated in Figure 1 has knot vector $\{v_i\} = \{0, 0, 0, 0, 0.4\bar{6}, 0.5, 0.5\bar{3}, 1, 1, 1, 1\}$, and weights $\{w_i\} = \{1, 0.09668, 0.00933,\ 0.00066, 0.00933, 0.09668, 1\}$.

If $2\Delta \geq \pi$, in order to satisfy the applicability conditions of relation (5), it will be necessary to subdivide the p-spline curve into piecewise p-splines defined on intervals whose size is less than π.

§4. Tools for p-splines

Of the many tools that play an important role in a spline-based modelling system, we report knot insertion, subdivision, knot removal, and degree elevation. Algorithms for knot insertion, subdivision, and knot removal for p-splines can be obtained from analogous algorithms for trigonometric splines, as can be easily deduced from the definition of $\underline{c}(t)$.

Alternative algorithms can be obtained using relation (5) and the analogous algorithms for polynomial splines. For example, the knot insertion algorithm for p-splines may be schematized through the following steps:

Let $t_\ell < \hat{t} \leq t_{\ell+1}$ be the knot to be inserted.

1. Compute $c_i = \delta_i / (\prod_{j=i+1}^{i+n} \cos t_j) \qquad i = \ell - n, \cdots, \ell$;

2. Insert knot $\varphi(\hat{t})$ by means of the polynomial spline algorithm on c_i coefficients to achieve $\hat{c}_i$ over the $\{\hat{v}_i\}$ knot partition;
3. Compute $\hat{\delta}_i = (\prod_{j=i+1}^{i+n} \cos \hat{t}_j)/\hat{c}_i \qquad i = \ell - n, \cdots, \ell + 1.$

   In fact, applying (5) to $\underline{c}(t)$, we have

$$\frac{1}{\sum_{i=0}^{K+n} \delta_i M_{i,n}(t)} = \frac{1}{cos^n t \ \sum_{i=0}^{K+n} c_i N_{i,n}(v)} \tag{8}$$

executing knot insertion for polynomial splines, and applying relation (5) once again, we obtain

$$= \frac{1}{cos^n t \ \sum_{i=0}^{K+n+1} \hat{c}_i \hat{N}_{i,n}(v)} = \frac{1}{\sum_{i=0}^{K+n+1} \hat{\delta}_i \hat{M}_{i,n}(t)}$$

with c_i, $\hat{c}_i$ and $\hat{\delta}_i$ as indicated in 1.,2. and 3.

Analogously, relation (5) can be used in order to evaluate the p-spline $\underline{c}(t)$, referring the evaluation of a trigonometric spline to a polynomial spline. It should be noted that these tips can improve the efficiency of a p-spline-based modelling system.

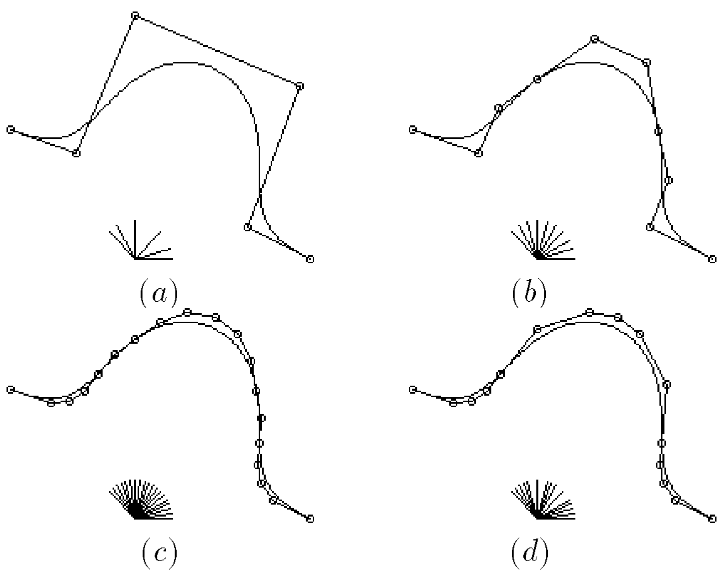


Fig. 2. Degree elevation steps; (a) original cubic p-spline curve , (b) subdivision in 3 p-Bézier curves, (c) degree elevation of each p-Bézier curve , (d) degree-elevated curve after the knot removal step.

Unlike the above-considered tools, the algorithm for the degree elevation of a p-spline is not achievable either from the degree elevation algorithm for a trigonometric polynomial considered in [1], or from the degree elevation algorithm for polynomial splines. In fact, the application of such algorithms does not modify the parametric interval size, as results from the definition of $\underline{c}(t)$. From this the need emerges for an ad hoc algorithm to determine the degree elevated p-spline curve.

4.1. Degree elevation for p-splines

From the expression of a p-spline in terms of the Fourier basis, one can deduce that this subset of curves is closed with respect to degree elevation from degree n to degree kn, for any natural value k [10].

Following the idea in [9] for polynomial splines, we propose a degree elevation technique for p-splines that consists in the following steps:

1. decompose the p-spline into piecewise p-Bézier curves (subdivision);
2. apply degree elevation to each p-Bézier curve;
3. remove unnecessary knots until the continuity of the original curve is guaranteed (knot removal).

In order to realize step 2., the following result is exploited [4].

Degree elevation formula for p-Bézier curves

Let $p(t) = \sum_{j=0}^{n} c_j A_{j,n}(t),\ t \in [-\Delta, \Delta]$ be a generic trigonometric polynomial of degree n in the Bernstein trigonometric basis, then

$$p(s) = \sum_{r=0}^{kn} \overline{c}_r A_{r,kn}(s), \qquad s = t/k, \qquad s \in \left[-\frac{\Delta}{k}, \frac{\Delta}{k}\right]$$

where

$$\overline{c}_r = \frac{d_r}{\binom{kn}{r} \sin^n(2\Delta)}$$

and d_r are the components of the vector

$$\mathbf{d} = \sum_{j=0}^{n} \binom{n}{j} c_j \mathbf{a}^{n-j} \otimes \mathbf{b}^j.$$

with

$$b_j = \binom{k}{j} \sin(2j\frac{\Delta}{k}), \qquad a_j = b_{k-j}, \qquad j = 0, \ldots, k.$$

The symbol $\otimes$ means convolution between two vectors, and $\mathbf{b}^j$ means convolution $\mathbf{b} \otimes \mathbf{b} \otimes \cdots \otimes \mathbf{b}$, $(j-1)$ times.

Our implementation of the above formula provides a preprocessing phase that performs the vectors $\mathbf{a}$ and $\mathbf{b}$, and the binomial coefficients.

Note that step 2. uses an optimized version of this algorithm, in fact it requires only one preprocessing phase for the degree elevation of all the p-Bézier curves, and this also contributes to the efficiency of the degree elevation algorithm for p-splines.

In addition, the algorithm results to be numerically stable as the p-Bèzier degree elevation algorithm is intrinsically stable [4].

In Figure 2, the three main algorithm steps are tested on an initial p-spline curve of degree $n = 3$ with 2 single interior knots, to obtain a p-spline of degree $kn = 6$ with 2 interior knots, both having a multeplicity of 4.

Computational results

The algorithm has been implemented in Pascal (BORLAND 7.0), carried out in double precision (15-16 significant figures), and tested on a Pentium 90 PC.

The test curves considered, without loss of generality, have been chosen with $\theta \in [0, \pi]$, the coefficients $\delta_i^{-1} = 1$, $i = 0, \cdots, n$, and randomly distributed knots.

$k \backslash n$	1	2	3	4
2	0.024	0.025	0.055	0.110
3	0.025	0.055	0.110	0.189
4	0.029	0.055	0.134	0.250
5	0.030	0.085	0.135	0.305

$(1a)$

$k \backslash n$	1	2	3	4
2	0.109	0.293	0.733	1.470
3	0.187	0.733	2.010	4.067
4	0.293	1.470	4.216	9.133
5	0.473	2.600	7.910	17.53

$(1b)$

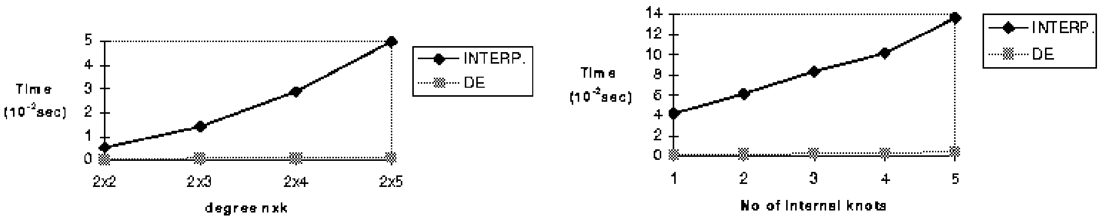
$k \backslash n$	1	2	3	4
2	0.024	0.055	0.135	0.299
3	0.025	0.109	0.190	0.384
4	0.029	0.110	0.250	0.495
5	0.030	0.135	0.329	0.580

$(2a)$

$k \backslash n$	1	2	3	4
2	0.180	0.547	1.313	2.640
3	0.326	1.420	3.880	8.127
4	0.620	2.900	8.453	18.57
5	0.993	5.380	16.07	36.50

$(2b)$

Table 1: Execution time ($10^{-2}sec$) results of degree elevation.



(a) (b)

Fig. 3. Degree elevation results.

In order to evaluate performance, the algorithm for the degree elevation of p-splines was compared with the interpolation technique, the only means at our disposal for degree-elevating a p-spline. Table 1 reports a comparison of execution times required by our algorithm (a) and by the interpolation technique (b) for 1 and 3 interior knots.

The graphs in Figure 3 provide a clearer understanding of these results. The graph (a) shows timings in the second column in Table (2a) and (2b); the graph (b) illustrates the execution times as functions of the number of internal knots, while the degree kn remains unchanged at value 4×3. Note that the execution times of our algorithm result to be widely winner.

All of the tests considered used p-spline curves with single interior knots. It is clear that our algorithm performs better when the knot multeplicity is increased.

Acknowledgments. Supported by CNR-Italy, Contract n.95.00730.CT01.

References

1. Alfeld P.,Neamtu M., Schumaker L.L., Circular Bernstein-Bézier Polynomials, *Mathematical Methods for Curves and Surfaces*, Morten Dæhlen, Tom Lyche, Larry L. Schumaker (eds.), Vanderbilt University Press, Nashville & London, 1995.
2. Casciola G., Lacchini M. and Morigi S. , Degree elevation for single-valued curves in polar coordinates, Dept. of Math. , Bologna, Italy, **13**, 1996.
3. Casciola G. and Morigi S. , Modelling of curves and surfaces in polar and Cartesian coordinates, Dept. of Math., Bologna, Italy, **12**, 1996.
4. Casciola G., Morigi S., Sánchez-Reyes J., Degree elevation for p-Bézier curves, submitted to Comput. Aided Geom. Design, October 1996.
5. de Casteljau P., Splines Focales, *Curves and Surfaces in Geometric Design*, P.-J. Laurent, A. Le Méhauté, and L. L. Schumaker (eds.), A. K. Peters, Wellesley MA, 1994, 91–103.
6. Goodman T.N.T. and Lee S.L., B-Splines on the circle and trigonometric B-Splines, in: Singh S.P. et al.eds., *Approximation theory and spline function* , Reidel D. Publishing Company, 1984, 297–325.
7. Koch P.E., Multivariate Trigonometric B-splines, J. Approx. Theory **54** 1988, 162–168.
8. Lyche T. and Winther R., A Stable Recurrence Relation for Trigonometric B-Splines, J. Approx. Theory **25** 1979, 266–279.
9. Piegl L. and Tiller W., Software-engineering approach to degree elevation of B-Spline curves, Computer-Aided Design **26** 1994, 17–28.
10. Sánchez-Reyes J., Single-valued curves in polar coordinates, Computer-Aided Design **22** 1990, 19–26.
11. Sánchez-Reyes J., Single-valued spline curves in polar coordinates, Computer Aided Design **24** 1992, 307–315.

Giulio Casciola and Serena Morigi
Dept. of Math., University of Bologna
P.zza di Porta S.Donato, 5
40127 BOLOGNA - ITALY
`casciola@dm.unibo.it`
Dept. of Math webb page `http://dm.unibo.it`