Modelling of curves and surfaces in polar and Cartesian coordinates

G.Casciola and S.Morigi *

Department of Mathematics, University of Bologna, Italy

Abstract

A new class of spline curves in polar coordinates has been presented in (Sánchez-Reyes, 1992) and independently considered in (de Casteljau, 1994). These are rational trigonometric curves in Cartesian coordinates and can be represented as NURBS. From the relationship existing with the correspondent curves in Cartesian coordinates an alternative way to derive some useful tools for modelling splines in polar coordinates has been provided. In particular an ad hoc algorithm of degree elevation for splines in polar coordinates is presented. On the basis of these results we propose a modelling system for NURBS curves and surfaces supplied with a modelling environment for spline curves and surfaces in polar, spherical, and mixed polar-Cartesian coordinates.

1 Introduction

Recently, in (Sanchez-Reyes, 1992) 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 (Sanchez-Reyes, 1990), which we call p-Bézier curves.

Soon afterwards, classes of spline surfaces in cylindrical, spherical and tubular coordinates were proposed in (Sanchez-Reyes, 1991, 1994a, 1994b).

*This research was supported by CNR-Italy, Contract n.95.00730.CT01

The p-splines were independently considered in (de Casteljau, 1994), which he called Focal splines. These classes of curves and surfaces are interesting because they allow for modelling and interpolation of 2D and 3D free forms in polar coordinates with the same facilities as Cartesian splines. Another important aspect to be considered is the possibility of providing a rapid response to the Point Membership Classification problem.

In all of the cited papers Sanchez-Reyes emphasizes the fact that the p-spline curves and surfaces are piecewise rational Bézier in Cartesian coordinates and they are not rational splines. Actually, this last assertion is neither proved nor supported by any justification. In this paper we will not only show that a p-spline curve or surface is a NURBS, but we will also provide the algorithm that leads to a representation of these curves and surfaces as NURBS.

In addition to knot insertion, knot removal and subdivision, another known result from Sanchez-Reyes' papers is the possibility of making degree elevation for p-splines from degree n to degree kn, although an explanation of how to achieve this has not been given. In (Casciola et al., 1995) an algorithm for degree elevation for p-Bézier curves has been proposed. In this paper we will suggest how to use it for p-splines, together with tests on efficiency and numerical stability.

These two results, in addition to others with less theoretical consequences but essential for practical purposes, 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 in order to manage polar and spherical models in the best way.

This environment makes use both of original tools and of tools already developed for NURBS, and it must be considered as an added potentiality to generate and model NURBS curves and surfaces. In particular, at the moment, the polar, spherical and mixed environment allows: the modelling of p-spline curves by interpolation and control-point modification; the modelling of tensor-product p-spline surfaces by interpolation of a mesh of points, by interpolation of ϕ-curves or θ-curves (skinning), by revolution of a ϕ curve, or by swinging; the possibility to model product surfaces (see Koparkar and Mudur, 1984) in mixed coordinates. As is already known , the latter includes sweep and swung surfaces.

2 P-spline curves and surfaces

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. Without loss of generality, we consider $t \in [-\Delta, \Delta]$. The functions $M_{i,n}(t)$ satisfy the following recurrence relation:

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

and

$$M_{i,0}(t) = \begin{cases} 1 & \text{if} \quad \mathrm{t_i \leq t < t_{i+1}} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

on a non-decreasing knot sequence $\{t_i\}_{i=0}^{K+2n+1}$ where

$$-\Delta = t_0 = \cdots = t_n \qquad t_{K+n+1} = \cdots = t_{K+2n+1} = \Delta$$

and the constraint $t_{i+n} - t_i < \pi$, $\forall i$, is satisfied .

It is easy to recognize that the functions $M_{i,n}(t)$ are the normalized trigonometric B-splines (see Lyche and Winter, 1979). Considering a support reduced to a single interval, these are identical with Bernstein basis trigonometric polynomials and span the space

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

with $t \in [0, \pi)$.

It is known (see Koch et al., 1995) that the control curve of the trigonometric spline $\sum_{i=0}^{K+n} \delta_i M_{i,n}(t)$ is given by

$$G(t) = g_i(t) \;\; i = 1, \cdots, K+n \qquad t \in \left[t^*_{i-1}, t^*_i\right], \qquad t^*_i = \frac{1}{n} \sum_{j=i+1}^{i+n} t_j$$

where

$$g_i(t) = \frac{\sin n(t - t^*_{i-1})\delta_i + \sin n(t^*_i - t)\delta_{i-1}}{\sin n(t^*_i - t^*_{i-1})}$$

interpolates the points $\left(t^*_{i-1}, \delta_{i-1}\right)$, (t^*_i, δ_i).

We note that the t^*_is satisfy $n(t^*_i - t^*_{i-1}) < \pi$ in view of the knot constraints; these guarantee that $g_i(t)$ can be defined everywhere and can interpolate $\left(t^*_{i-1}, \delta_{i-1}\right)$ and (t^*_i, δ_i).

The control curve of $\underline{c}(t)$ is defined as

$$\underline{F}(t) = \begin{pmatrix} \frac{1}{g_i(t)} \\ nt \end{pmatrix} \qquad i = 1, \cdots, K + n$$

where

$$\begin{aligned} \frac{1}{g_i(t)} &= \frac{\sin n(t^*_i - t^*_{i-1})}{\sin n(t - t^*_{i-1})\delta_i + \sin n(t^*_i - t)\delta_{i-1}} = \\ &= \frac{\sin(\xi_i - \xi_{i-1})}{\sin(\theta - \xi_{i-1})\delta_i + \sin(\xi_i - \theta)\delta_{i-1}} \end{aligned}$$

having set $\xi_i = nt^*_i$. Every i-th piece of the control curve $\underline{F}(t)$ is, in polar coordinates, a straight segment which interpolates the points $\left(\xi_{i-1}, \delta^{-1}_{i-1}\right)$, $\left(\xi_i, \delta^{-1}_i\right)$.

Thus, the $\underline{F}(t)$ is a control polygon, and the coefficients δ^{-1}_i and the Greville radial directions ξ_i define, in polar coordinates, the control points $\underline{d_i} = \left(\xi_i, \delta^{-1}_i\right)$ 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 ($\underline{F}(t)$), and the associated trigonometric spline function, together with its control curve ($G(t)$).

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.

A tensor product p-spline surface $\underline{s}(u, v)$, in spherical coordinates, of degree n in u and m in v, is defined as follows:

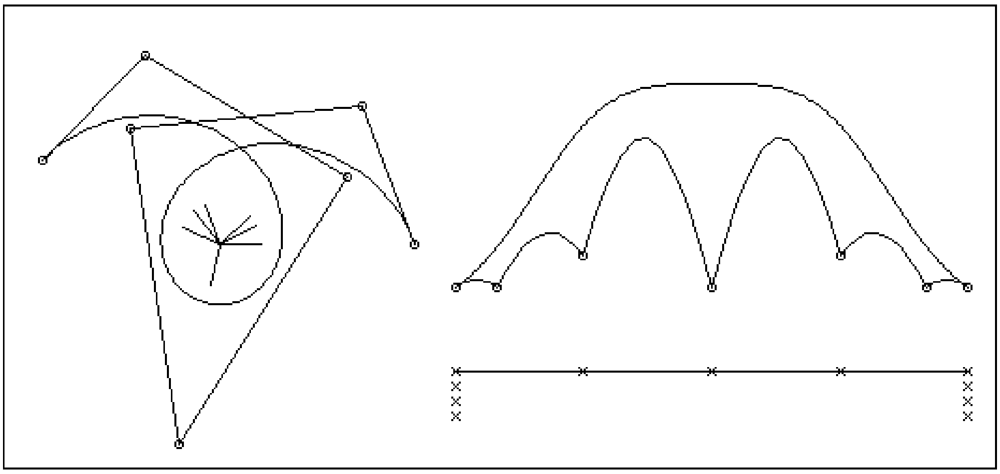

Figure 1: Example of p-spline curve of degree $n = 3$ together with its control polygon (left) and associated trigonometric spline with its control curve (right) - $\{t_i\} = \{0,0,0,0,0.75,1.5,2.25,3,3,3,3\}$, $\{\left(\xi_i, \delta_i^{-1}\right)\} = \{(0,0.3),(0.75,0.3),(2.25,0.22),(4.5,0.3),(6.75,0.22),(8.25,0.3),(9,0.3)\}$

$$\underline{s}(u,v) = \begin{pmatrix} \rho(u,v) \\ \theta(u) \\ \phi(v) \end{pmatrix} = \begin{pmatrix} \frac{1}{\sum_{i=0}^{K+n}\sum_{j=0}^{H+m}\delta_{ij}M_{i,n}(u)M_{j,m}(v)} \\ nu \\ mv \end{pmatrix}$$

where $[u,v] \in [a,b] \times [c,d]$.

A product surface of $\rho(\theta)$ and $\underline{d}(u)$ with scaling factor $f(u)$ in polar-Cartesian mixed coordinates is given by

$$\underline{r}(u,\theta) = \begin{pmatrix} f(u)\cos\theta\rho(\theta) + d_1(u) \\ f(u)\sin\theta\rho(\theta) + d_2(u) \\ d_3(u) \end{pmatrix}$$

with

$$f(u) = \sum_{i=0}^{H+m} f_i R_{i,m}(u)$$

$$\underline{d}(u) = \sum_{i=0}^{H+m} D_i R_{i,m}(u)$$

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

where $f(u)$ and $\underline{d}(u)$ are NURBS, with the $R_{i,m}$ rational B-spline functions, and $\rho(\theta)$ is a p-spline curve.

In the next section we will show how all the curves and surfaces considered in our spline-based modelling system can be represented as NURBS.

3 NURBS representation of p-splines

It is known (Sanches-Reyes, 1992) that the class of p-spline restricted to a single segment (p-Bézier curves or Focal Bèzier) represents a subclass of rational Bèzier curves in Cartesian coordinates. Therefore, we can state that p-splines represent a subclass of NURBS.

A first approach to obtain a NURBS representation of a p-spline curve has been suggested in (Sanchez-Reyes 1992). 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 Bèzier curves in Cartesian coordinates.

In the alternative approach proposed here, a non-piecewise Bèzier 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

$$\rho(\frac{\theta}{n}) = \frac{1}{\sum_{i=0}^{K+n} \delta_i M_{i,n}(\frac{\theta}{n})} \tag{3}$$

where $\theta \in [-n\Delta, n\Delta]$.

Then the correspondent curve of (3) in Cartesian coordinates will be obtained by a simple change of coordinates:

$$\rho(\frac{\theta}{n}) \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \tag{4}$$

Applying the identities

$$\begin{array}{c} \cos\theta = \sum_{i=0}^{K+n} \cos\xi_i M_{i,n}(\frac{\theta}{n}) \\ \sin\theta = \sum_{i=0}^{K+n} \sin\xi_i M_{i,n}(\frac{\theta}{n}) \end{array}$$

(see Goodman and Lee, 1984), relation (4) assumes the following trigonometric rational form:

$$\frac{\sum_{i=0}^{K+n} \binom{\cos \xi_i}{\sin \xi_i} M_{i,n}(\frac{\theta}{n})}{\sum_{i=0}^{K+n} \delta_i M_{i,n}(\frac{\theta}{n})} \tag{5}$$

In (Koch, 1988) the important transformation $\gamma_n : P_n -> T_n$; $(\gamma_n f)(x) = \cos^n x\ f(\tan x)$, 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:

$$M_{i,n}(t) = \frac{\cos^n t}{\prod_{j=i+1}^{i+n} \cos t_j} N_{i,n}(\varphi(t)) \tag{6}$$

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

where the $N_{i,n}$ are the polynomial B-spline functions defined on the knot sequence $\{\varphi(t_i)\}$.

In virtue of (6), relation (5) becomes

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

where

$$v = \varphi(t) = \frac{1}{2} \left[1 + \frac{\tan t}{\tan \Delta} \right] \tag{7}$$

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

$$\underline{q}(v) = \frac{\sum_{i=0}^{K+n} P_i w_i N_{i,n}(v)}{\sum_{i=0}^{K+n} w_i N_{i,n}(v)} \qquad v \in [0,1]\,, \tag{8}$$

with weights

$$w_i = \frac{\delta_i}{\prod_{j=i+1}^{i+n} \cos t_j} \tag{9}$$

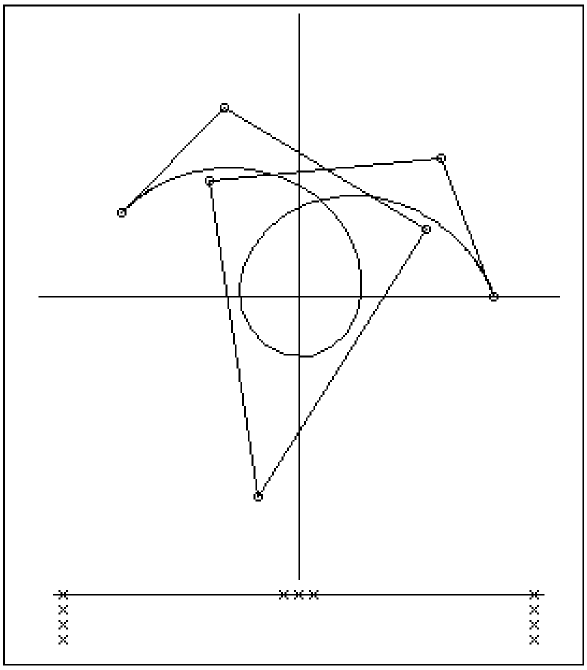

Figure 2: NURBS representation of the p-spline curve shown in Fig.1. $n = 3$, $\{v_i\} = \{0, 0, 0, 0, 0.4\bar{6}, 0.5, 0.5\bar{3}, 1, 1, 1, 1\}$, $\{w_i\} = \{1, 0.09668, 0.00933, 0.00066, 0.00933, 0.09668, 1\}$.

and control points $P_i = \delta_i^{-1} \begin{pmatrix} \cos \xi_i \\ \sin \xi_i \end{pmatrix}$; the $N_{i,n}(v)$ functions are defined over a knot sequence $\{v_i\}$ obtained applying relation (7) to the knots t_i. Note that the P_i are given by the transformation in Cartesian coordinates of the p-spline control points (ξ_i, δ_i^{-1}).

In Figure 2 the NURBS representation of the p-spline curve illustrated in Figure 1 is shown.

If $2\Delta \geq \pi$, in order to satisfy the applicability conditions of relation (6), it will be necessary to subdivide the p-spline curve into piecewise p-splines defined on intervals whose size is less than π.

Also, it should be noted that if the p-spline is defined on a generic interval $[a, b]$, the translation, non-translation, at the origin of this interval involves differently parametrized curves in Cartesian coordinates that can be converted into each other by means of a suitable linear rational reparametrization.

The existence of the corresponding Cartesian rational spline $\underline{q}(v)$ of a given p-spline $\underline{c}(t)$, allows us to assure that the piecewise rational Bézier curve, obtained following the approach suggested by Sanchez-Reyes, can be transformed into $\underline{q}(v)$ by a suitable linear rational reparametrization of each piece, followed by consecutive knot removal until the partition $\{v_i\}$ is achieved.

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

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 $\hat{\delta}_i = \frac{\prod_{j=i+1}^{i+n} \cos \hat{t}_j}{\hat{c}_i} \qquad i = \ell - n, \cdots, \ell + 1.$

In fact, applying (6) to $\underline{c}(t)$, we have

$$\frac{1}{\sum_{i=0}^{K+n} \delta_i M_{i,n}(t)} = \frac{1}{cos^n t \ \sum_{i=0}^{K+n} c_i N_{i,n}(v)} \tag{10}$$

executing knot insertion for polynomial splines, (10) becomes

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

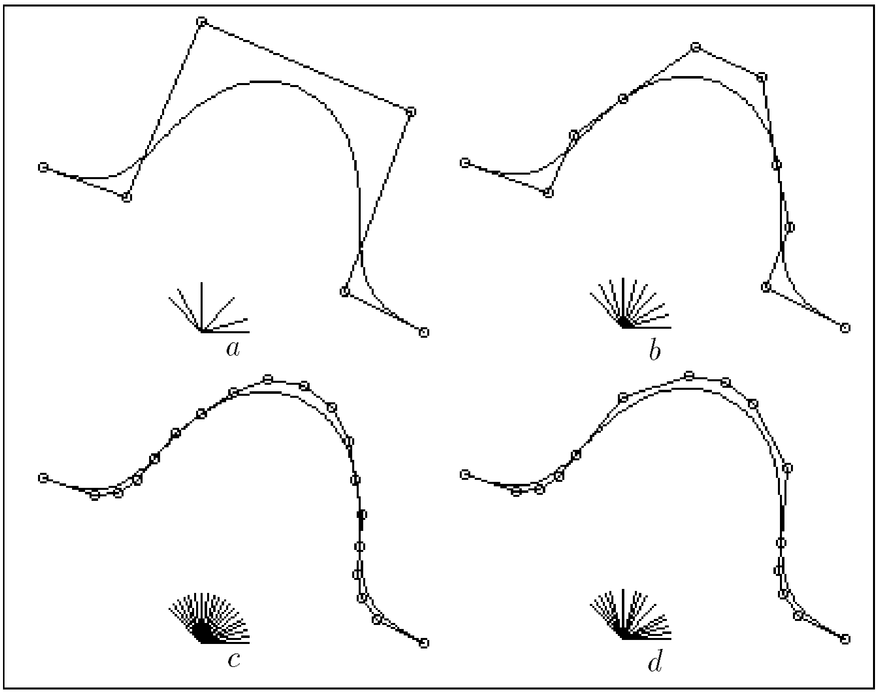

Figure 3: Degree elevation steps; (a) original cubic p-spline curve , (b) subdivision in 3 p-Bézier curves, (c) degree elevation of each p-Bézier curve , (d) degree-elevated curve after the knot removal step; the degree is raised to 6.

and applying relation (6) once again, we obtain

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

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

Analogously, relation (6) 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.

Unlike the above-considered tools, the algorithm for the degree elevation of a p-spline is not achievable either from the trigonometric spline degree elevation algorithm considered in (see Alfeld et al., 1995), 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.

Assume, for example, $n = 2$; the p-spline can be represented, for each segment, in the Fourier basis $\{1, \cos 2t, \sin 2t\}$, for $t \in [-\Delta, \Delta]$, and also $\{1, \cos 4s, \sin 4s\}$ $s \in \left[-\frac{\Delta}{2}, \frac{\Delta}{2}\right]$. Therefore, it can be expressed by the Fourier basis of elevated degree $kn = 4$, $\{1, \cos 2s, \sin 2s, \cos 4s, \sin 4s\}$.

Following the idea in (Piegl and Tiller, 1994) for polynomial splines, we provide 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 (Casciola et al., 1995).

Degree elevation formula for p-Bézier curves

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

$$p(t) = \sum_{r=0}^{kn} \overline{c_r} A_{r,kn}(s), \qquad s = t/k, \qquad s \in \left[-\frac{\Delta}{k}, \frac{\Delta}{k}\right] \tag{11}$$

where

$$\overline{c_r} = \frac{n!}{\sin^n(2\Delta)} \binom{kn}{r}^{-1} \sum_{j=0}^{n} c_j \sum_{\Gamma_j} \gamma_j \eta_{j,r}$$

and

$$\gamma_j = \prod_{\substack{d=1 \\ d\ odd}}^{D} \frac{\left[(-1)^{d div 2}\binom{k}{d}\right]^{i_d+j_d}}{i_d! j_d!} \cdot \tan^{I+J}\left(\frac{2\Delta}{k}\right)$$

$$\eta_{j,r} = \sum_{h=max(0,r-kj)}^{min(k(n-j)-I,r-J)} \binom{k(n-j)-I}{h}\binom{kj-J}{r-J-h} \cos^{k(n-j)+r-2h}\left(\frac{2\Delta}{k}\right)$$

$D =$ max odd less than or equal to k,

$$\begin{aligned} \Gamma_j \quad = \quad & \{(i_1, i_3, .., i_D, j_1, j_3, .., j_D) ; i_1, i_3, .., i_D, j_1, j_3, .., j_D \geq 0 \ , \\ & i_1 + i_3 + .. + i_D = n - j, \ \ j_1 + j_3 + .. + j_D = j\} \end{aligned}$$

$$I = i_1 + 3i_3 + ... + Di_D \qquad \text{and} \qquad J = j_1 + 3j_3 + ... + Dj_D$$

Although the formula may appear complex and its implementation may not be immediately comprehensible, in (Casciola et al., 1995) useful details and tips to produce an efficient and stable algorithm are provided.

At first sight, it is easy to realize that the critical point of the formula consists in the cycle in Γ_j ; this is due both to the difficulty in determining the ordered sequences $(i_1, i_3, .., i_D, j_1, j_3, .., j_D)$, and to their large quantity that makes the cycle expensive.

For this purpose, it should be pointed out that only ordered sequences of $(k+1)div2$ items $i_1, i_3, .., i_D$ satisfying $i_1+i_3+..+i_D = n-j$ with $j = 0, \cdots, n$ needed to be determined, because the $j_1, j_3, .., j_D$ items are the same. The ordered sequences in Γ_j are obtained by making suitable combinations of the $i_1, i_3, .., i_D$ items. Moreover, we can observe that the number of ordered sequences in Γ_j for $k = 2$ is reduced to one, and, for small k and n , the number of ordered sequences remains limited.

Considering that, generally, in a spline modelling system, the most frequently used maximum degrees are 2,3 or 4 , it is easy to see how this formula can produce good results.

The proposed algorithm provides a preprocessing phase that performs the entities that recur many times in the given formula; in particular, the coefficients γ_j are precomputed. Note that, being $\gamma_j = \gamma_{n-j}$, the number of

$k \setminus n$	1	2	3	4
2	0.00	2.379	1.876	1.498
3	0.00	1.427	1.161	0.921
4	0.00	1.019	0.853	0.675
5	0.00	0.793	0.668	0.533
6	0.00	0.649	0.552	0.440
7	0.00	0.549	0.468	0.375
8	0.00	0.476	0.408	0.327

Table 1: Convergence of control points to the curve; each entry has to be multiplied by 10^{-3}.

these coefficients actually computed is reduced to half of the cardinality of Γ_j.

Only one preprocessing phase is required 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 Figure 3, 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.

Tests were performed on the following aspects:

1. numerical stability of the algorithm;

evaluated by means of a convergency test on the control points of the degree elevated curve to the curve itself, and by means of a maximum deviation test between the original p-spline and the associated degree-elevated p-spline;

$k \setminus n$	1	2	3	4
2	0.017	0.033	0.037	0.107
3	0.033	0.073	0.180	0.400
4	0.036	0.106	0.267	0.627
5	0.053	0.220	0.767	2.270

$(1a)$

$k \setminus n$	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

$(1b)$

$k \setminus n$	1	2	3	4
2	0.036	0.073	0.147	0.253
3	0.073	0.187	0.340	0.807
4	0.090	0.220	0.587	1.250
5	0.107	0.440	1.500	4.510

$(2a)$

$k \setminus n$	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

$(2b)$

$k \setminus n$	1	2	3	4
2	0.040	0.147	0.220	0.400
3	0.113	0.293	0.620	1.250
4	0.113	0.327	0.880	1.944
5	0.147	0.653	2.310	7.500

$(3a)$

$k \setminus n$	1	2	3	4
2	0.293	0.880	2.010	4.033
3	0.587	2.270	6.190	12.80
4	1.030	4.830	13.70	29.76
5	1.650	8.720	25.90	54.20

$(3b)$

Table 2: Execution time ($10^{-2} sec$) results of degree elevation ;(a) our implementation, (b) interpolation technique; (1), (2) and (3) respectively with 1, 3 and 5 internal knots.

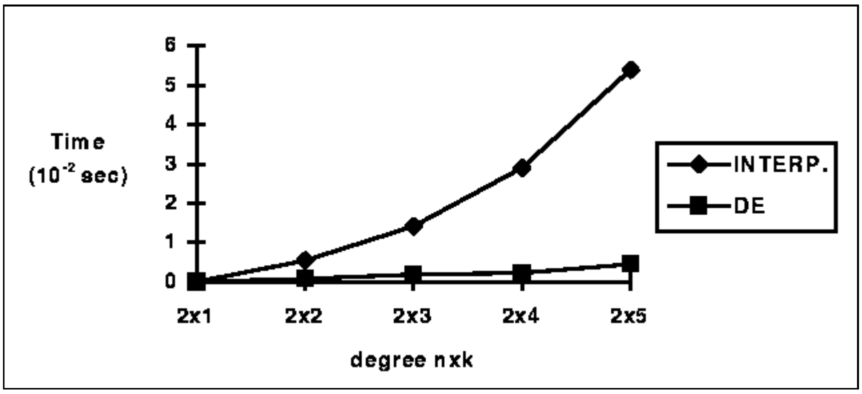

Figure 4: Degree elevation results shown in the second column in Table 2a

2. execution time of the algorithm;

evaluated as a function of the start degree n, the increment factor k, and the number L of interior knots.

Table 1 summarizes convergency results of control points to the curve, for p-spline test curves of degrees $n = 1, 2, 3, 4$ with 2 interior knots and increasing k. The values reported were evaluated by computing

$$\max_{i=0,\cdots,K+kn+L(kn-n)} |\underline{c}(\xi_i) - \delta_i^{-1}| \tag{12}$$

Numerical stability was assessed by

$$MAXERR := \|\underline{c}(t) - \underline{c}(s)\|_{\infty D} \tag{13}$$

on a uniformly-spaced set of points, where $\underline{c}(s)$ denotes the degree-elevated curve, resulting in

$$10^{-16} \leq MAXERR \leq 10^{-15} \tag{14}$$

This reveals the accuracy of our 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 2 reports a comparison of execution times required by our algorithm and by the interpolation technique for 1,3 and 5 interior knots, with $n < 5$ and $k < 6$.

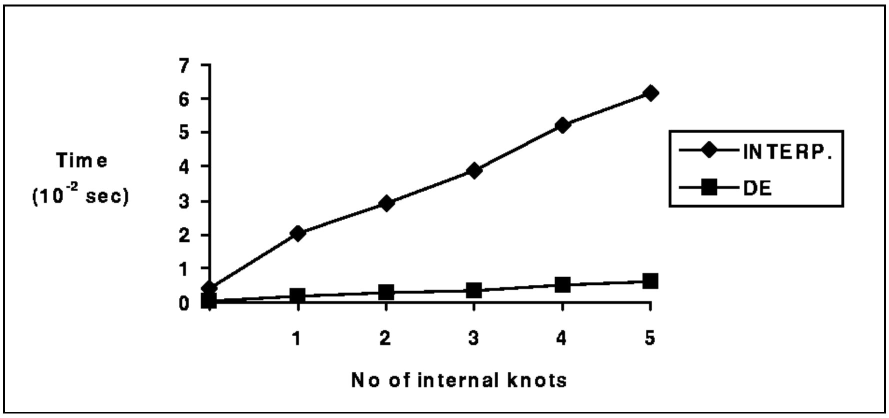

Figure 5: Degree elevation results as a function of the number of interior knots

The results emphasize the performance of our algorithm when compared with the interpolation technique; Figures 4 and 5 provide a clearer understanding of these results.

In Figure 4 timings relating to an initial curve of degree $n = 2$ with 3 interior knots are reported for increasing k values.

While the execution times of our algorithm were slightly worse than the linear growth rate, the interpolation has an exponential growth rate.

The tests in Figure 5 illustrate the execution times as functions of the number of internal knots, while the degree kn remains unchanged at value 9.

From Figure 5 we can observe that the performance of our algorithm for p-splines takes full advantage of the preprocessing phase, making the growth rate less than the linear one.

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

5 Modelling examples

In this section we show some examples of surfaces obtained using our NURBS-based modelling system. This system is supplied with a modelling environment for p-spline curves and surfaces in polar, spherical and mixed polar-Cartesian coordinates.

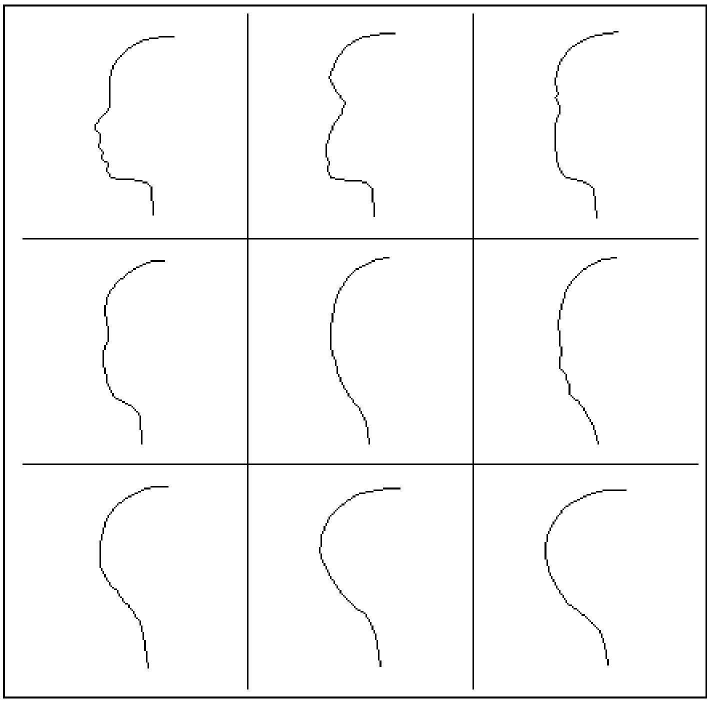

Figure 6: p-spline profile curves in polar coordinates for a half face

In the first example we have modelled a face in polar-spherical coordinates by the skinning technique.

In Figure 6 the profile curves for half face are shown. These p-spline curves have different degree 2 and 3, and different knot partition. They are obtained by interpolation in polar coordinates. Before applying the skinning technique in spherical coordinates, in order to obtain a tensor product p-spline surface, all the profile curves have been maked compatible by using our degree elevation algorithm and knot insertion technique. In Figure 7 the resulting modelling face is shown.

The second example, in Figure 8, shows a tensor product p-spline surface modelled by changing its control points. Modelling this kind of surfaces is complex in Cartesian coordinates but becomes easy in spherical coordinates because is sufficient to modify the coefficients according to a specific pattern.

The last example shows a swinging surface as a particular case of a product surface in mixed polar-Cartesian coordinates.

Figure 9 shows the p-spline trajectory curve modelled in polar coordinates

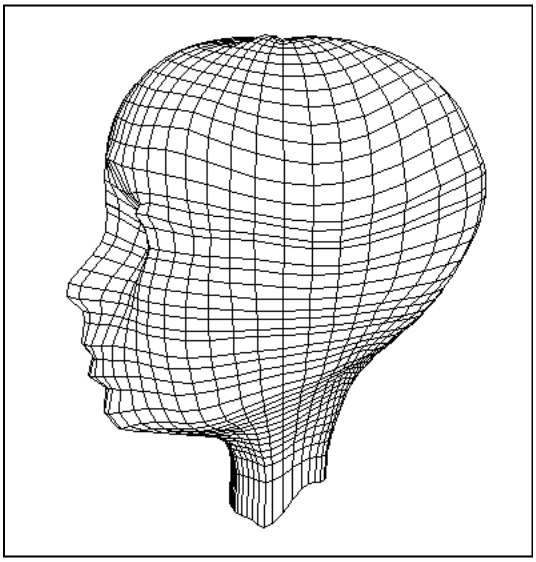

Figure 7: p-spline surface in spherical coordinates obtained by skinning

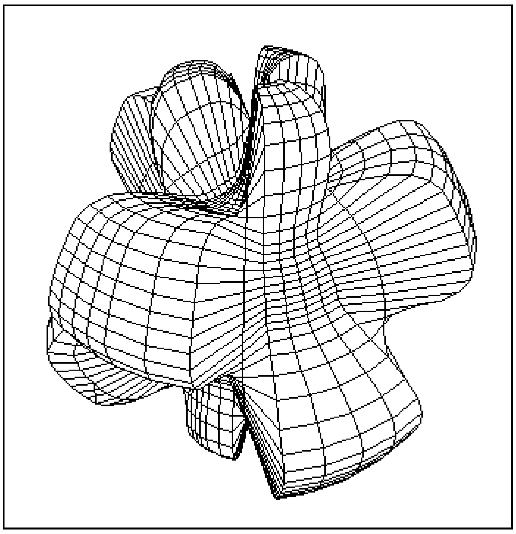

Figure 8: p-spline surface in spherical coordinates obtained by modelling the control grid

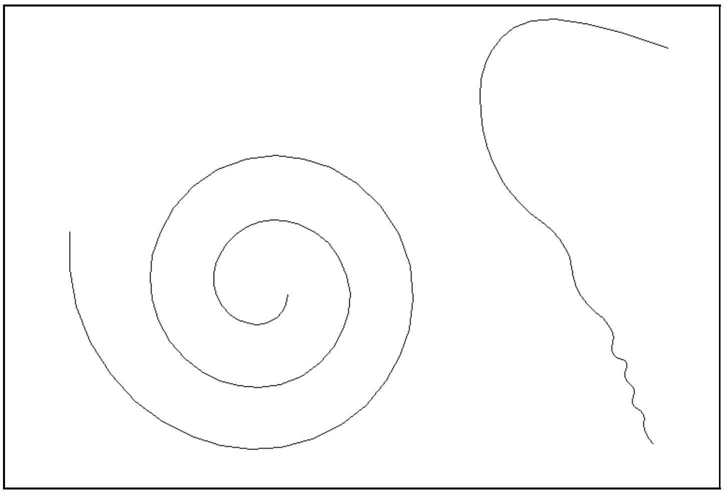

Figure 9: left: p-spline trajectory; right: NURBS profile

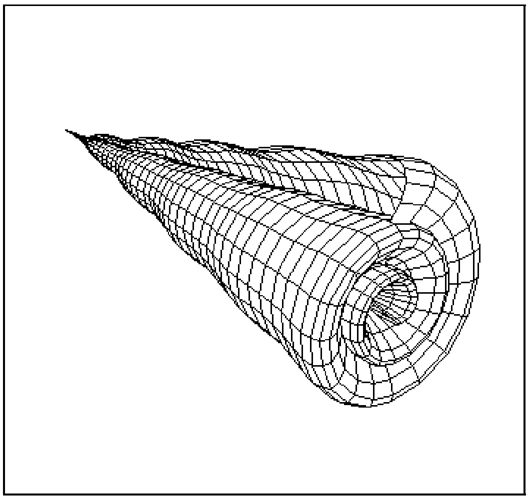

Figure 10: swinging surface in mixed polar-Cartesian coordinates

and the NURBS profile curve in Cartesian coordinates. The resulting surface is shown in Figure 10.

Once the p-spline surface has been modelled, it is possible to represent it as a NURBS surface, generalizing the result in section 3. This turns out to be important both for having the model in a standard NURBS form, and to exploit for p-splines the note techniques for NURBS as, for example, the rendering algorithms.

References

Alfeld P.,Neamtu M., Schumaker L.L. (1995), Circular Bernstein-Bezier Polynomials, in: Daehlen M., Lyche T., Schumaker L.L. eds., Mathematical Methods in CAGD.

Casciola G., Lacchini M. and Morigi S. (1995), Degree elevation for single-valued curves in polar coordinates, submitted to CAGD.

de Casteljau P. (1994), Splines Focales, in: Laurent P.J. et al.eds., *Curves and Surfaces in Geometric Design* , Peters A.K. Wellesley, 91-103.

Goodman T.N.T. and Lee S.L. (1984), 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, 297-325.

Koch P.E. (1988), Multivariate Trigonometric B-splines, Journal of Approximation Theory, 54, 162-168.

Koch P.E., Lyche T., Neamtu M. and Schumaker L. (1995), Control curves and knot insertion for trigonometric splines, Advances in Computational Mathematics, 3, 405-424.

Koparkar P.A. and Mudur S.P. (1984), Computational Techniques for Processing Parametric Surfaces, Computer Vision, Graphics, and Image Processing, 28, 303-322.

Lyche T. and Winther R. (1979), A Stable Recurrence Relation for Trigonometric B-Splines, Journal of Approximation theory, 25, 266-279.

Piegl L. and Tiller W. (1994), Software-engineering approach to degree elevation of B-Spline curves, Computer Aided Design, 26, 17-28.

Sanchez-Reyes J. (1990), Single-valued curves in polar coordinates, Computer Aided Design, 22, 19-26.

Sanchez-Reyes J. (1991), Single-valued surfaces in cylindrical coordinates, Computer Aided Design, 23, 561-568.

Sanchez-Reyes J. (1992), Single-valued spline curves in polar coordinates, Computer Aided Design, 24, 307-315.

Sanchez-Reyes J. (1994a), Single-valued surfaces in spherical coordinates, Computer Aided Geometric Design, 11, 491-517.

Sanchez-Reyes J. (1994b), Single-valued tubular patches , Computer Aided Geometric Design, 11, 565-592.