Degree elevation for single-valued curves in polar coordinates

G.Casciola, M.Lacchini and S.Morigi *
Department of Mathematics, University of Bologna, Italy

Abstract

A new class of single-valued curves in polar coordinates obtained by a transformation of a subset of rational Bézier curves into Cartesian coordinates has recently been presented in (Sánchez-Reyes, 1990), and independently considered by P.de Casteljau, who called these curves focal Bézier. These curves are trigonometric polynomials that can be represented by a basis similar to the Bernstein polynomial basis. From their definition and expression in terms of the Fourier basis it is obvious that every curve of degree n can be expressed as a curve of degree kn, for any natural value k.

In this paper, two alternative formulae for degree elevation from degree n to kn are presented and relative proofs are given. A simple and efficient implementation is provided and its stability is numerically proved.

Keywords: Degree elevation; Polar curves; Cartesian rational curves; Reparametrization

1 Introduction

A class of single-valued curves in polar coordinates (which we refer to as p-Bézier curve), recently presented in (Sánchez-Reyes, 1990) and independently

*This research was supported by CNR-Italy, contract n.95.00730.CT01 in (P.de Casteljau, 1994), has been obtained by re-examining, in polar coordinates, the algorithm for evaluating a rational Bézier curve in Cartesian coordinates (which we call c-Bézier curve). This was made possible by an interpretation in polar coordinates of its well-known geometric meaning. It is interesting to note that the recursive algorithm for evaluating a rational Bézier curve defines, in polar coordinates, certain sinusoidal functions that form the direct analog, in the trigonometric field, of the Bernstein polynomials. Note that similar functions have been used in (Goodman and Lee,1984) and in (Lyche and Winter, 1979), but there trigonometric polynomials are defined using half angles. The main advantage of this class of curves in polar coordinates is that it provides a fast response to the Point Membership Classification problem (PMC). In practice, by a restriction of the c-Bézier curve set to those curves having corresponding p-Bézier curves it is possible to exploit the polar representation when a PMC problem has to be solved.

At the same time, this class of curves is interesting because it allows modeling and data best-fitting problems to be dealt with in polar coordinates. The generalisation that has been made for spline curves in (Sánchez-Reyes, 1992) and for single-valued surfaces in cylindrical and spherical coordinates in (Sánchez-Reyes, 1991) and (Sánchez-Reyes, 1994) is even more interesting. All these are ideal for modeling since they have the same properties as the c-Bézier curves.

Tools such as knot-insertion, subdivision, and knot-removal are automatically derived from the procedure followed to generate these curves, as pointed out in (Sánchez-Reyes, 1990). Another fundamental tool in B-spline based geometric design is degree elevation.

The possibility of carrying out degree elevation of every curve of degree n to kn, for any natural value k, is clear from their definition and representation in the Fourier basis, but there is no known method of achieving this, if not by interpolation. The interpolation technique takes $(kn+1)$ samples and sets a linear system of $kn+1$ equations, where the $(kn+1)$ unknowns are the degree-elevated curve coefficients.

This paper is organized as follows. Section 2 introduces some notations relating to p-Bézier curves, and sections 3 and 4 present two explicit formulae for the degree elevation of p-Bézier curves. In section 5 some details regarding the implementation are given, and relative computational and stability results are shown.

2 Single-valued curves in polar coordinates

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

$$\underline{c}(t) = \left\{ \begin{array}{l} \rho(t) = 1/\sum_{i=0}^{n} \delta_i^{-1} A_{i,n}(t) \\ \theta(t) = nt \end{array} \right.$$

where $\theta(t)$ denotes the polar angle and $\rho(t)$ is the radius, and without loss of generality, $t \in [-\Delta, \Delta]$, $2n\Delta < \pi$. The functions $A_{i,n}(t)$ are defined as follows:

$$A_{i,n}(t) = \frac{1}{\sin^n(2\Delta)} \binom{n}{i} \sin^{n-i}(\Delta - t) \sin^i(t + \Delta)$$

It is easy to show that functions $A_{i,n}(t)$, which we call the Bernstein basis trigonometric polynomials, span the linear space

$$T_n = \text{span}\left\{\sin^{n-i}(t)\cos^i(t)\right\} \quad i = 0, ..., n$$

of trigonometric polynomials of degree n, see (Goodman and Lee,1984).

Moreover, it was also shown in (Lyche and Winter, 1979) that

$$T_n = \left\{ \begin{array}{ll} \text{span}\{1, \cos(2t), \sin(2t), \cos(4t), \sin(4t), .., \cos(nt), \sin(nt)\}, & n \text{ even} \\ \text{span}\{\cos(t), \sin(t), \cos(3t), \sin(3t), .., \cos(nt), \sin(nt)\}, & n \text{ odd.} \end{array} \right.$$

The coefficients δ_i and the Greville radial directions $\xi_i = -n\Delta + 2i\Delta$, $i = 0, .., n$ define the control points in polar coordinates $\underline{d_i} = (\delta_i, \xi_i)$ of $\underline{c}(t)$.

As an alternative to the geometric approach followed in (Sánchez-Reyes, 1990) to obtain p-Bézier curves from c-Bézier curves, we present an analytical proof by changing the coordinates.

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

$$\rho(\theta) = \frac{1}{\sum_{i=0}^{n} \delta_i^{-1} A_{i,n}(\theta/n)}$$

where $\theta \in [-n\Delta, n\Delta]$, then the correspondent curve in Cartesian coordinates will be given by

$$\rho(\theta)\begin{pmatrix}\cos\theta\\ \sin\theta\end{pmatrix}.$$

Since the following relationship between $A_{i,n}(\theta/n)$ functions and Bernstein polynomial functions $B_{i,n}(u)$, see (Sánchez-Reyes, 1994), holds:

$$A_{i,n}(\theta/n)=\left(\frac{\cos(\theta/n)}{\cos\Delta}\right)^n B_{i,n}(u)\qquad u\in[0,1] \tag{1}$$

where parameters u and θ are related by the equation:

$$u=\frac{1}{2}\left[1+\frac{\tan(\theta/n)}{\tan\Delta}\right] \tag{2}$$

and from (Goodman and Lee, 1984) it follows that

$$\mathit{cos}\ \theta=\sum_{i=0}^{n}\cos(\xi_i)A_{i,n}(\theta/n)\qquad \mathit{sin}\ \theta=\sum_{i=0}^{n}\sin(\xi_i)A_{i,n}(\theta/n) \tag{3}$$

then $\rho(\theta)$ in Cartesian coordinates will admit the following c-Bézier curve representation:

$$\underline{Q}(u)=\frac{\sum_{i=0}^{n}P_i w_i B_{i,n}(u)}{\sum_{i=0}^{n}w_i B_{i,n}(u)}\qquad u\in[0,1]$$

where weights $w_i=\delta_i^{-1}$ and control points $P_i=\delta_i\begin{pmatrix}\cos(\xi_i)\\ \sin(\xi_i)\end{pmatrix}$.

Vice versa every c-Bézier curve with a correspondent p-Bézier curve is characterized by the following properties or constraints:

1. control points P_i on Greville radial directions regularly spaced by a 2Δ angle,

2. $\|P_i\|_2=1/w_i$.

In the next sections we deal with the problem of degree elevation following two possible ways: in section 3, an explicit formula is identified to obtain the elevated degree $\underline{c}(t)$ via Cartesian coordinates following the ABC path shown in Fig.1. In section 4 an alternative formula is derived from the direct path D (see Fig.1).

$$\begin{tikzcd}
 & \text{c-Bézier} & & \text{p-Bézier} \\
\text{degree } n & \underline{Q}(u) \arrow[rr, Leftrightarrow, "A"] \arrow[d, Rightarrow, "B"'] & & \underline{c}(t) \arrow[d, Rightarrow, "D"] \\
\text{degree } kn & \underline{Q}(v) \arrow[rr, Leftrightarrow, "C"'] & & \underline{c}(s)
\end{tikzcd}$$

Figure 1: Degree elevation scheme

3 Degree elevation via Cartesian coordinates

Given the explicit relation between a p-Bézier curve and a c-Bézier curve, it seems natural to ask oneself whether it would be possible to derive a degree elevation algorithm for p-Bézier curves through the ABC path by means of reparametrization.

We have already seen that (1) allows you to pass from p-Bézier curves to c-Bézier curves where (1) is derived from (2). (Route A in Fig 1.)

Let $\underline{c}(s)$ be the p-Bézier curve of degree kn obtained by degree elevation from $\underline{c}(t)$; a direct analog of relation (1) obtained by:

$$v = \frac{1}{2}\left[1 + \frac{\tan(\theta/(kn))}{\tan(\Delta/k)}\right] \tag{4}$$

allows you to pass from $\underline{c}(s)$ to the correspondent c-Bézier curve $\underline{Q}(v)$ (Route C Fig. 1.).

In the following we attempt to make the relation existing between $\underline{Q}(u)$ and $\underline{Q}(v)$ explicit (Route B in Fig. 1.).

From (4) we obtain

$$\frac{\theta}{n} = k \ \arctan\left[(2v-1)A\right]$$

and substituting in (2)

$$u = \frac{1}{2}\left[1 + \frac{1}{B}\tan\left(k \ \arctan\left[(2v-1)A\right]\right)\right] \tag{5}$$

where $A = \tan(\Delta/k)$ and $B = \tan\Delta$.

Applying the formula

$$\tan(k\alpha) = \frac{\sum\limits_{\substack{j=0 \\ j\ odd}}^{k} \binom{k}{j}(-1)^{j div 2} \tan^j \alpha}{\sum\limits_{\substack{j=0 \\ j\ even}}^{k} \binom{k}{j}(-1)^{j div 2} \tan^j \alpha}$$

into (5), we obtain

$$u = \frac{1}{2}\left[1 + \frac{1}{B}\frac{\sum\limits_{\substack{j=0 \\ j\ odd}}^{k} \binom{k}{j}(-1)^{j div 2}(2v-1)^j A^j}{\sum\limits_{\substack{j=0 \\ j\ even}}^{k} \binom{k}{j}(-1)^{j div 2}(2v-1)^j A^j}\right] = \frac{p(v)}{q(v)} \qquad (6)$$

From this we deduce that $\underline{Q}(v)$ is none other than the reparametrization of $\underline{Q}(u)$ through the rational function (6) with numerator and denominator of degree k at most.

Theorem 1 *Let $\underline{Q}(u)$ be a c-Bézier curve, then the reparametrized curve $\underline{Q}(v)$ by the rational function (6) can be expressed in the following form:*

$$\underline{Q}(v) = \frac{\sum\limits_{j=0}^{kn} \overline{P_j w_j} B_{j,kn}(v)}{\sum\limits_{j=0}^{kn} \overline{w_j} B_{j,kn}(v)} \qquad v \in [0,1] \qquad (7)$$

where

$$\overline{w_j} = n!\binom{kn}{j}^{-1} \sum_{i=0}^{n} w_i \sum_{\Gamma_{ij}} \left(\prod_{l=0}^{k} \frac{(\beta_l - \alpha_l)^{i_l} \cdot \alpha_l^{j_l}}{i_l! j_l!} \right) \qquad (8)$$

and

$$\alpha_l = \sum_{j=0}^{l} \binom{l}{j}\binom{k}{j}^{-1} a_j, \qquad a_j = 2^{j-1} \sum_{i=j}^{k} (-1)^{i div 2 + i - j} \binom{i}{j}\binom{k}{i} B^{(i+1) mod 2} A^i \qquad (9)$$

$$\beta_l = \sum_{j=0}^{l} \binom{l}{j} \binom{k}{j}^{-1} b_j, \qquad b_j = 2^j B \sum_{\substack{i = j \\ i\ even}}^{k} (-1)^{i div 2 + i - j} \binom{i}{j} \binom{k}{i} A^i \tag{10}$$

$$\begin{aligned} \Gamma_{ij} \quad = \quad & \{(i_0, .., i_k, j_0, .., j_k)\,; i_0, .., i_k, j_0, .., j_k \geq 0 \;, \\ & i_1 + 2i_2 + .. + ki_k + j_1 + 2j_2 + .. + kj_k = j, \\ & i_0 + .. + i_k = n - i, \;\; j_0 + .. + j_k = i\} \end{aligned}$$

Analogously $\overline{P_j w_j}$ can be obtained substituting in (8) w_i with $P_i w_i$.

Proof

Represent the numerator and denominator of (6) in the Bernstein polynomial basis as follows:

$$u = \frac{\sum\limits_{j=0}^{k} a_j v^j}{\sum\limits_{j=0}^{k} b_j v^j} = \frac{\sum\limits_{l=0}^{k} \alpha_l B_{l,k}(v)}{\sum\limits_{l=0}^{k} \beta_l B_{l,k}(v)} \tag{11}$$

with a_j, b_j, α_l, and β_l given as in (9) and (10).

Consider the $\underline{Q}(u)$ denominator and apply reparametrization (6) expressed in the form (11),

$$\sum_{i=0}^{n} w_i B_{i,n}(u) = \sum_{i=0}^{n} w_i \binom{n}{i} \left[\frac{\sum\limits_{l=0}^{k} (\beta_l - \alpha_l) B_{l,k}(v)}{\sum\limits_{l=0}^{k} \beta_l B_{l,k}(v)} \right]^{n-i} \left[\frac{\sum\limits_{l=0}^{k} \alpha_l B_{l,k}(v)}{\sum\limits_{l=0}^{k} \beta_l B_{l,k}(v)} \right]^{i} \tag{12}$$

Using Leibniz's formula for the numerator of (12), we obtain

$$\sum_{i=0}^{n} w_i \binom{n}{i} \sum_{\substack{i_0 + .. + i_k = n - i, \\ j_0 + .. + j_k = i \\ i_0, .., i_k, j_0, .., j_k \geq 0}} (n-i)! i! \left(\prod_{l=0}^{k} \frac{(\beta_l - \alpha_l)^{i_l} \cdot \alpha_l^{j_l}}{i_l! j_l!} \right) (1-v)^{kn-(I+J)} v^{I+J}$$

where

$$I = i_1 + 2i_2 + ... + ki_k \qquad \text{and} \qquad J = j_1 + 2j_2 + ... + kj_k$$

Since $0 \leq I+J \leq kn$, from the index exchange $j = I+J$, the j-th term in (8) is immediately deduced.

Proceeding in a similar manner for $\overline{P_j w_j}$, the result (7) follows. □

In order to obtain $\underline{c}(s)$ starting from $\underline{Q}(v)$, note that the constraint 1. of section 2 follows from the assumptions made, while, in general, to satisfy constraint 2. it is necessary to scale the weights $\overline{w_j}$. The scaling factor S can be computed as:

$$S = \frac{w_0}{\overline{w_0}}$$

It should be noted that for the denominator $q(v)$ in (6) it holds $q(v) = q(1-v)$, so that in (11) we have $\beta_l = \beta_{k-l}$. By dividing coefficients α_l and β_l in (11) by β_0 we obtain now $\alpha_0 = 0$ and $\alpha_k = 1$ because the rational function (6) maps $[0,1]$ in $[0,1]$. From (8) it results that $\overline{w_0} = w_0$ ($\overline{w_{kn}} = w_n$) and therefore it becomes not necessary to scale the weights $\overline{w_j}$.

4 Degree elevation via polar coordinates

An alternative formulation for the degree elevation itself is derived from a more careful analysis of the explanation that allows us to obtain the degree elevation in polar coordinates.

Suppose n is even. The curve of degree n can be written in terms of the Fourier basis of arguments $2it$ or $2i\theta/n$, $i = 0, ..., n/2$, but also of arguments $\frac{2ik\theta}{kn}$ or $2iks$, with $ki = 0, ..., kn/2$, and $s \in \left[-\frac{\Delta}{k}, \frac{\Delta}{k}\right]$ that correspond to the arguments of a curve of degree kn defined in $\left[-\frac{\Delta}{k}, \frac{\Delta}{k}\right]$.

Assume, for example $n = 2$; the curve can be represented 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 $kn = 4$, $\{1, \cos(2s), \sin(2s), \cos(4s), \sin(4s)\}$.

Analogously for odd n.

Following the example above but using the Bernstein polynomial trigonometric basis a formula for degree elevation is derived which can be more widely expressed as follows.

Theorem 2 *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*, then*

$$p(t) = \sum_{r=0}^{kn} \overline{c_r} A_{r,kn}(s), \qquad s = t/k, \qquad s \in \left[-\frac{\Delta}{k}, \frac{\Delta}{k}\right] \tag{13}$$

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

$$\gamma_j = \prod_{\substack{d=1 \\ d\ odd}}^{D} \frac{\left[(-1)^{ddiv2}\binom{k}{d}\right]^{i_d + j_d}}{i_d! j_d!} \cdot \tan^{I+J}\left(\frac{2\Delta}{k}\right)$$

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

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

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

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

Proof

Given the $p(t)$ of degree n, change parameter $t = ks$:

$$p(t) = \frac{1}{\sin^n(2\Delta)} \sum_{j=0}^{n} c_j \binom{n}{j} \sin^{n-j} k(\frac{\Delta}{k} - s) \sin^j k(s + \frac{\Delta}{k})$$

from the expansion of $\sin(k\alpha)$ to the power of $\sin\alpha$ and $\cos\alpha$

$$\sin(k\alpha) = \sum_{\substack{i=1 \\ i\ odd}}^{D} (-1)^{idiv2}\binom{k}{i} \sin^i\alpha \cos^{k-i}\alpha$$

and applying Leibniz 's formula, we deduce:

$$p(t) = \frac{n!}{\sin^n(2\Delta)} \sum_{j=0}^{n} c_j \binom{n}{j} \Big\{ \sum_{\Gamma_j} (n-j)!j! \left(\prod_{\substack{d=1 \\ d\ odd}}^{D} \frac{\left[(-1)^{ddiv2} \binom{k}{d}\right]^{i_d+j_d}}{i_d! j_d!} \right) \cdot$$

$$\cdot \sin^I(\frac{\Delta}{k} - s) \cos^{k(n-j)-I}(\frac{\Delta}{k} - s) \sin^J(s + \frac{\Delta}{k}) \cos^{kj-J}(s + \frac{\Delta}{k}) \Big\} \qquad (14)$$

By the angle summation formulae, applying (3) in case $n = 1$ and developing the binomial, we have

$$\cos^{kj-J}(s + \frac{\Delta}{k}) = \left[\cos s \cos \frac{\Delta}{k} - \sin s \sin \frac{\Delta}{k}\right]^{kj-J}$$

$$= \frac{1}{\sin^{kj-J}(\frac{2\Delta}{k})} \left[\sin(\frac{\Delta}{k} - s) + \sin(s + \frac{\Delta}{k}) \cos(\frac{2\Delta}{k})\right]^{kj-J}$$

$$= \frac{1}{\sin^{kj-J}(\frac{2\Delta}{k})} \sum_{l=0}^{kj-J} \binom{kj-J}{l} \left[\sin(s + \frac{\Delta}{k}) \cos(\frac{2\Delta}{k})\right]^{l} \sin^{kj-J-l}(\frac{\Delta}{k} - s)$$

and, in a similar manner,

$$\cos^{k(n-j)-I}(\frac{\Delta}{k} - s) = \frac{1}{\sin^{k(n-j)-I}(\frac{2\Delta}{k})} \cdot$$

$$\cdot \sum_{h=0}^{k(n-j)-I} \binom{k(n-j)-I}{h} \sin^h(s + \frac{\Delta}{k}) \left[\sin(\frac{\Delta}{k} - s) \cos(\frac{2\Delta}{k})\right]^{k(n-j)-I-h}$$

Substituting into (14)

$$p(t) = \frac{n!}{\sin^n(2\Delta)} \sum_{j=0}^{n} c_j \left\{ \sum_{\Gamma_j} \left(\prod_{\substack{d=1 \\ d\ odd}}^{D} \frac{\left[(-1)^{ddiv2} \binom{k}{d}\right]^{i_d+j_d}}{i_d! j_d!} \right) \sin^{I+J}(\frac{2\Delta}{k}) \cdot \right.$$

$$\cdot \sum_{h=0}^{k(n-j)-I} \sum_{l=0}^{kj-J} \frac{\binom{k(n-j)-I}{h}\binom{kj-J}{l}}{\binom{kn}{l+h+J}} \cos^{k(n-j)-I-h+l}(\frac{2\Delta}{k}) A_{l+h+J,kn}(s) \Bigg\}$$

Setting $r = l+h+J$ and since $0 \le r \le kn$, summing all the contributions for each r, (13) is deduced. $\square$

Note that for $k = 2$ the expression for the coefficients in (13) can be simplified as follows:

$$\overline{c_r} = n!\binom{2n}{r}^{-1} \sum_{j=0}^{n} \frac{c_j}{j!(n-j)!} \sum_{h=max(0,r-2j)}^{min(n-j,r-j)} \binom{n-j}{h}\binom{j}{r-j-h} \cos^{r-2j-2h}(\Delta) \quad (15)$$

Example. Let $\underline{c}(t)$ be a p-Bézier curve of degree $n = 2$ with $\delta_0^{-1} = \delta_2^{-1} = 1$, $\delta_1^{-1} = cos(2\Delta)$ and $t \in [-\Delta, \Delta]$, representing an arc of unit circle; applying (15) we obtain $\underline{c}(s)$ of degree $kn = 4$, $s \in \left[-\frac{\Delta}{2}, \frac{\Delta}{2}\right]$ and coefficients:

$$\begin{aligned}
\overline{\delta_0^{-1}} &= \delta_0^{-1} = 1 \\
\overline{\delta_1^{-1}} &= \tfrac{1}{2cos\Delta}(\delta_0^{-1} + \delta_1^{-1}) = cos\Delta \\
\overline{\delta_2^{-1}} &= \tfrac{1}{6}\left[\tfrac{\delta_0^{-1}}{cos^2\Delta} + 2\delta_1^{-1}(1 + cos^2\Delta) + \tfrac{\delta_2^{-1}}{cos^2\Delta}\right] = \tfrac{1}{3}[1 + 2cos^2\Delta] \\
\overline{\delta_3^{-1}} &= \tfrac{1}{2cos\Delta}(\delta_1^{-1} + \delta_2^{-1}) = cos\Delta \\
\overline{\delta_4^{-1}} &= \delta_2^{-1} = 1
\end{aligned}$$

Note that these coefficients are the same obtained in (Sanchez-Reyes, 1994), but in a different manner.

5 Implementation and numerical results

Although the formulae (8) and (13) may appear complex and their implementation may not be immediately comprehensible, this section will demonstrate how, using simple strategies, it is possible to produce an algorithm easily and efficiently. Formula (13) will be compared to the interpolation technique in terms of performance and analysed according to its numerical stability.

The main computational bottleneck in formulae (8) and (13) is the cycle in Γ_{ij} and Γ_j respectively. This is partly due to the difficulty in generating these index sequences, and, above all, to the high cost of the cycle itself.

The problem, in general, requires the determination of all the ordered sequences of m items, each item referred to as x_i, and each having a value between 0 and VAL. It is well known that the total number of these m-tuples is given by $(VAL+1)^m$. As we are only interested in those m-tuples that satisfy the following constraints

$$x_1 + x_2 + \cdots x_m = \ell, \qquad \ell = 0, \cdots, VAL \tag{16}$$

the total number of these is reduced to:

$$\sum_{\ell=0}^{VAL} NS(m,\ell)$$

where

$$NS(m,\ell) = \sum_{i=0}^{\ell} NS(m-1,i)$$

with

$$NS(1,i) = 1$$

Our implementation produces permutations of $m-1$ items setting the m-th item to the value required to reach ℓ.

Furthermore, permutations of items x_i, whose sum is greater than ℓ, are not generated.

The algorithm used to produce the m-tuples is the following:

```
procedure perm(d,m,end,sum)
begin
for j=0 to end
    s[d]:=j
    ssum:=sum+j
    if (d < m-1) then perm(d+1,$\ell$-ssum,ssum)
    else
        s[m]:=$\ell$-ssum
        for i=1 to m
            sequence[counttot+count][i]:=s[i]
        count:=count+1
end

procedure seq_generation(m,VAL,sequence,numseq)
begin
counttot:=0
for $\ell$=0 to VAL
    count:=0
    if (m=1) then
        sequence[counttot+count][1]:=$\ell$
        count:=count+1
    else
        perm(1,m,$\ell$,0)
    countot:=counttot+count
    numseq[$\ell$]:=count
end
```

where the "sequence" array contains the x_i items, while "numseq" stores the number of m-tuples satisfying (16) for a certain ℓ.

As we can see from formulae (8) and (13) the number m of items for any sequence is $(k+1)$ and $(k+1)div2$ respectively, while the value of VAL is n for both. Therefore, the cost of cycle Γ_{ij} in (8) is greater than the cost of cycle Γ_j in (13).

As a result, the two formulae differ significantly in cost, which has also been pointed out in their implementation phase.

Therefore, from this point onwards, only formula (13) will be considered.

In order to be usable in practice our implementation is a compromise between space consumption and performance time.

The algorithm provides for a preprocessing phase that performs the entities recurring many times in formula (13). In particular, coefficients γ_j are preprocessed. Observing that $\gamma_j = \gamma_{n-j}$, the number of coefficients actually computed is reduced to

$$\sum_{i=0}^{ndiv2} NS((k+1)div2, i)NS((k+1)div2, n-i)$$

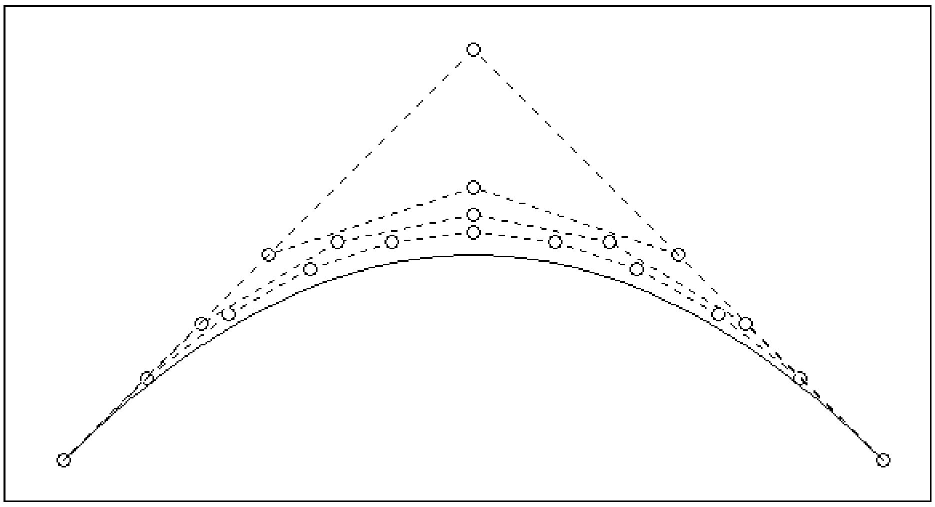
Figure 2: Control polygons of degree kn curves with $n = 2$ and $k = 1, 2, 3, 5$

Computation of coefficients $\overline{c_r}$, by means of a direct implementation of (13), follows the preprocessing phase.

Note that the limits for j in (13) can be restricted for any $\overline{c_r}$ to the range $max(0, r - kn + n), min(r, n)$, because the η_{jr} vanish when j is not part of the range.

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 $\theta \in [0, \pi]$ and the coefficients $\delta_i = 1$, $i = 0, \cdots, n$. One of these curves is shown in Fig 2.

The three tables presented in this section report computation and stability results regarding running with $kn \leq 64$. This limitation is due to reasons of practical applicability and to reducing memory storage requirements. The memory size required by our algorithm for $kn \leq 64$ is given by $\simeq 67Kbytes$.

A comparison of computational costs is summarized in Tables 1 and 2, which report the execution times required in order to solve a degree elevation problem from degree n to degree kn by means of (13) and interpolation respectively.

As indicated by the results in Table 1, our implementation of (13) ob-

$k\backslash n$	2	3	4	5	6	7	8	16	32
2	0.03	0.03	0.04	0.04	0.04	0.07	0.07	0.36	1.79
3	0.03	0.07	0.18	0.32	0.56	0.92	1.46	19.5	–
4	0.04	0.11	0.29	0.55	1.02	1.76	2.92	47.9	–
5	0.11	0.37	1.13	2.81	6.48	13.5	26.2	–	–
6	0.14	0.51	1.65	4.28	10.1	21.7	43.0	–	–
7	0.25	1.24	4.94	15.7	44.2	110	254	–	–
8	0.29	1.28	6.29	20.5	59.1	150	351	–	–
16	2.51	27.1	207	–	–	–	–	–	–
32	24.8	–	–	–	–	–	–	–	–

Table 1: Execution time (in 10^{-2} sec) by our implementation of (13)

$k\backslash n$	2	3	4	5	6	7	8	16	32
2	0.04	0.07	0.15	0.25	0.37	0.51	0.70	4.39^*	32.0^*
3	0.07	0.18	0.37	0.59	0.92	1.35	1.91	13.1^*	–
4	0.15	0.37	0.66	1.16	1.88	2.81	4.03^*	32.6^*	–
5	0.25	0.59	1.14	2.09	3.36	5.20^*	7.58^*	–	–
6	0.33	0.88	1.83	3.36	5.53^*	8.61^*	12.6^*	–	–
7	0.48	1.28	2.75	5.09^*	8.57^*	13.3^*	19.6^*	–	–
8	0.66	1.83	3.96^*	7.43^*	12.5^*	19.6^*	28.8^*	–	–
16	3.92^*	12.4^*	32.0^*	–	–	–	–	–	–
32	28.6^*	–	–	–	–	–	–	–	–

Table 2: Execution time (in 10^{-2} sec) by interpolation technique

$k\backslash n$	2	3	4	5	6	7	8	16	32
1	38.0	22.6	19.1	14.7	12.8	10.7	9.59	4.81	2.41
2	12.4	10.0	8.14	6.77	5.78	5.04	4.46	2.32	1.18
3	7.42	6.20	5.17	4.32	3.74	3.27	2.91	1.53	–
4	5.30	4.56	3.79	3.20	2.76	2.42	2.16	1.14	–
5	4.12	3.57	2.99	2.53	2.19	1.92	1.71	–	–
6	3.37	2.95	2.47	2.10	1.81	1.59	1.42	–	–
7	2.85	2.50	2.10	1.79	1.55	1.36	1.21	–	–
8	2.47	2.18	1.83	1.56	1.35	1.19	1.06	–	–
16	1.19	1.07	0.90	–	–	–	–	–	–
32	0.59	–	–	–	–	–	–	–	–

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

tained a better performance, when compared with interpolation, for any n and small k, and for any k and small n.

Experimental tests were performed in order to analyse the numerical stability of our implementation. For this purpose, a comparison of the degree elevated curve with the original curve was made by computing

$$MAXERR := \|\underline{c}(t) - \underline{c}(s)\|_{\infty D} \tag{17}$$

on a uniformly spaced set of points.

With our algorithm,

$$10^{-16} \leq MAXERR \leq 10^{-15} \tag{18}$$

is obtained for any $kn \leq 64$, whereas, by applying interpolation,

$$10^{-15} \leq MAXERR \leq 10^{-09} \tag{19}$$

results for the tests in Table 2 without *.

Note that the tests marked by * are not reliable, as will be seen below.

Another stability test considered the convergence of the sequence of values δ_i to the curve. This was evaluated in Table 3 using

$$max_{i=0,\cdots,kn}|\underline{c}(\xi_i) - \delta_i| \tag{20}$$

A convergent behaviour can be observed by reading Table 3 in columns. Cases $n = 2$ and $k = 1, 2, 3, 5$ are illustrated in Fig 2.

Regarding the interpolation technique, tests marked with * in Table 2 emphasize a divergent behaviour of the sequence δ_i to the curve. This is due to an increase in the condition number of the matrix associated with the linear system.

6 Concluding remarks

- We can observe that by decomposing $k = k_1 \cdot k_2 \cdots k_m$ into m prime factors and performing m steps, the performance generally improves. In particular, as we can see from Table I, when k is a power of 2, the repeated applications of the algorithm for $k = 2$ significantly reduce the execution time.

- An algorithm for the degree elevation of single-valued spline curves in polar coordinates has been achieved through the following steps: (a) decompose the spline curve in polar coordinates into piecewise p-Bézier curves (knot-insertion); (b) apply degree elevation to each p-Bézier curve, and (c) remove unnecessary knots (knot-removal).

  Note that step (b) is optimized because the degree elevation for all p-Bézier curves requires only one preprocessing stage.

  In the Cartesian case, in (Piegl and Tiller, 1994), it is shown that this method is very competitive with existing direct algorithms for the degree elevation of B-spline curves.

References

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.

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