A comparison between two evaluation algorithms for polynomial curves

J. Delgado, J. M. Peña
Departamento de Matemática Aplicada, Universidad de Zaragoza, 50009 Zaragoza, Spain

Abstract

In this paper we compare the de Casteljau algorithm with a more efficient algorithm associated to another shape preserving representation. We prove that the Bernstein basis is better conditioned than the basis associated to the more efficient algorithm. Error analysis results are presented. Numerical experiments are also included.

Keywords: Error analysis, Bernstein basis, de Casteljau algorithm, linear complexity, shape preservation.

1 Introduction

In computer-aided geometric design the usual representation of a polynomial curve is the so called Bernstein-Bezier form (see [9] and [14]). The Bernstein-Bezier form of a polynomial $p(t)$ of degree less than or equal to m on $[0,1]$ is given by

$$p(t) = \sum_{j=0}^{m} \alpha_j \, b_j^m(t), \quad t \in [0,1], \tag{1}$$

where $\alpha_j \in \mathbf{R}$, and

$$b_j^m(t) = \binom{m}{j} t^j (1-t)^{m-j}, \quad j = 0, \ldots, m, \tag{2}$$

is the corresponding Bernstein polynomial of degree m. The usual algorithm used to evaluate a polynomial in the form (1) is the de Casteljau algorithm.

The *collocation matrix* of $(u_0(t), \ldots, u_n(t))$ at $t_0 < \cdots < t_m$ in I is given by

$$M \begin{pmatrix} u_0, \ldots, u_n \\ t_0, \ldots, t_m \end{pmatrix} := (u_j(t_i))_{i=0,\ldots,m; j=0,\ldots,n}. \tag{3}$$

A matrix is *totally positive* if all its minors are nonnegative and a system of functions is totally positive when all its collocation matrices (3) are totally positive. In interactive design we want the shape of a parametrically defined curve to mimic the shape of its control polygon; thus we can predict or manipulate the shape of the curve by choosing or changing the control polygon suitably. In case of a normalized totally positive basis one knows that the curve imitates the shape of its control polygon, due to the variation diminishing properties of totally positive matrices (see [11], [2] and [16]). If $(u_0(t), \ldots, u_n(t))$ is a normalized totally positive basis then the curve γ inherits many shape properties of the control polygon. It is well known that the Bernstein basis is a normalized totally positive basis (see [1]).

We say that we cut a corner of a polygonal arc $C_0 \cdots C_n$ if we replace any vertex C_i by either

$$(1-\lambda)C_i + \lambda C_{i+1} \ (0 \le i < n)$$

or by

$$(1-\lambda)C_i + \lambda C_{i-1} \ (0 < i \le n)$$

for some $0 < \lambda < 1$. We say that an algorithm is a corner cutting algorithm if in each step of the algorithm we obtain a polygonal arc by cutting corners from another polygonal arc (see [6]). The de Casteljau algorithm is a corner cutting algorithm. It is well known that corner cutting algorithms present good stability properties because each step is formed by convex combinations (see [15]).

Although for polynomial curves the usual evaluation algorithm in CAGD is the de Casteljau algorithm, in some circumstances other evaluation algorithms can be useful. Horner algorithm is the usual evaluation algorithm for polynomials and has linear complexity but it is not a corner cutting algorithm and has many disadvantages in CAGD with respect to the de Casteljau algorithm. Recently, a linear complexity corner cutting algorithm used for the evaluation of curves and surfaces is the Wang-Ball algorithm (see [20], [19], [3], [12] and [18]). However, in [8] it was proved that the Wang-Ball basis is not normalized totally positive. In this paper we consider another representation with a corner cutting evaluation algorithm which simultaneously satisfies linear complexity and that it is associated to a normalized totally positive basis (see [4] and [5]).

Section 2 introduces the basic definitions and notations. In Section 3 we compare the theoretical results of the error analysis for the de Casteljau algorithm and for the algorithm corresponding to the basis of Section 2. In particular, we show that the backward error is lower for the second algorithm and, in contrast, we prove that the Bernstein basis is better conditioned. In Section 4 we include numerical experiments and the final conclusions.

2 Basic Definitions and Notations

In [4] J. Delgado and J. M. Peña introduced a new basis and proved that the new representation preserves monotonicity.

Definition 2.1. Let $(c_0^m(t),\ldots,c_m^m(t))$, $m\geq 2$, be the system of polynomials on $[0,1]$ defined by:

• If m is even c_i^m is given by

$$\begin{cases} (1-t)^m, & i=0\\ t(1-t)^{m-i}, & 1\leq i\leq \frac{m}{2}-1\\ 1-t^{\frac{m}{2}+1}-(1-t)^{\frac{m}{2}+1}, & i=\frac{m}{2}\\ t^i(1-t), & \frac{m}{2}+1\leq i\leq m-1\\ t^m & i=m. \end{cases}$$

• If m is odd, then $c_i^m(t)$ is defined as in the even case whenever the integer $i\notin\{\frac{m-1}{2},\frac{m+1}{2}\}$ and, otherwise, $c_i^m(t)$ is given by

$$\begin{cases} t(1-t)^{\frac{m+1}{2}}+\frac{1}{2}[1-t^{\frac{m+1}{2}}-(1-t)^{\frac{m+1}{2}}], & i=\frac{m-1}{2}\\ \frac{1}{2}[1-t^{\frac{m+1}{2}}-(1-t)^{\frac{m+1}{2}}]+t^{\frac{m+1}{2}}(1-t), & i=\frac{m+1}{2} \end{cases}$$

In addition, in [5] J. Delgado and J. M. Peña demonstrate that these bases are normalized totally positive (and so satisfy many shape preserving properties) and provide a corner cutting algorithm to evaluate curves of the form

$$\gamma(t)=\sum_{i=0}^{m}P_i\,c_i^m(t)$$

with linear time complexity.

Let us now introduce some standard notations in error analysis. Given $a\in\mathbf{R}$, the computed element in floating point arithmetic will be denoted by either $\mathrm{fl}(a)$ or by $\hat{a}$. As usual, to investigate the effect of rounding errors when working with floating point arithmetic we use the model

$$\mathrm{fl}(a\,\mathrm{op}\,b)=(a\,\mathrm{op}\,b)(1+\delta),\quad |\delta|\leq u, \tag{4}$$

although we also can use

$$\mathrm{fl}(a\,\mathrm{op}\,b)=\frac{(a\,\mathrm{op}\,b)}{(1+\varepsilon)},\quad |\varepsilon|\leq u, \tag{5}$$

with u the unit roundoff and *op* any of the elementary operations $+,-,\times,/$ (see pp. 44-45 of [13] for more details). Given $k\in\mathbf{N}_0$ such that $ku\ll 1$, let us define

$$\gamma_k:=\frac{ku}{1-ku}=ku+\mathcal{O}(u^2).$$

In order to perform error analysis of an algorithm, conditioning of the corresponding problem is also a very important aspect that must be taken into account. Given $f(t)=\sum_{i=0}^n c_iu_i(t)$, where $u=(u_0,\ldots,u_n)$ is a basis of the corresponding space,

$$S_u(f(t)):=\sum_{i=0}^{n}|c_i\,u_i(t)|, \tag{6}$$

is called a condition number for the evaluation of $f(t)$ with the basis u (see [10], [16] and [17]). Let us observe that $S_u(f(t))$ depends on the basis u, on the function f, and on the point t.

3 Error Analysis

The next two results have been obtained in [15] and [7], respectively, and show backward and forward error bounds of the de Casteljau algorithm and the algorithm corresponding to the basis of Section 2.

Proposition 3.1. Let $b^m=(b_0^m,\ldots,b_m^m)$ be the Bernstein basis defined on $[0,1]$ and let us consider the de Casteljau algorithm. Let $p(t)=\sum_{k=0}^m f_k\,b_k^m(t)$ ($f_k\in\mathbf{R}$ $\forall\,k$) be a polynomial of degree less than or equal to m and let us suppose that $2mu<1$, where u is the roundoff unit. Then the value $\widehat{p}(t)=fl(p(t))$ calculated with the de Casteljau algorithm satisfies

$$\widehat{p}(t)=\sum_{k=0}^{m}\overline{f}_k\,b_k^m(t),\quad\text{where}\quad\frac{|\overline{f}_k-f_k|}{|f_k|}\leq\gamma_{2m} \tag{7}$$

and

$$|p(t)-\widehat{p}(t)|\leq\gamma_{2m}\sum_{k=0}^{m}|f_k|b_k^m(t)=\gamma_{2m}\,S_{b^m}(p(t)).$$

Proposition 3.2. Let us consider the basis $c^m=(c_0^m,\ldots,c_m^m)$ and the corresponding corner cutting evaluation algorithm associated. Let us suppose that $2mu<1$, where u is the roundoff unit. Then, if $p(t)=\sum_{k=0}^m f_k\,c_i^m(t)$ ($f_k\in\mathbf{R}\;\forall\,k$), the value $\widehat{p}(t)=fl(p(t))$ calculated with the corresponding algorithm satisfies

$$\widehat{p}(t)=\sum_{k=0}^{m}\overline{f}_k\,c_k^m(t),\quad\text{with}\quad\frac{|f_k-\overline{f}_k|}{|f_k|}\leq\gamma_{i_k} \tag{8}$$

and

$$|p(t)-\widehat{p}(t)|\leq\sum_{i=0}^{m}|f_k|\,\gamma_{i_k}\,c_k^m(t)\leq\gamma_{2m}\,S_{c^m}(p(t)),$$

where

$$\begin{aligned} &i_0=2m,\quad i_m=2m,\\ &i_k=2m+2-2k,\quad 1\leq k\leq\left\lfloor\frac{m}{2}\right\rfloor-1,\\ &i_k=2k+2,\quad \left\lfloor\frac{m+1}{2}\right\rfloor+1\leq k\leq m-1, \end{aligned}$$

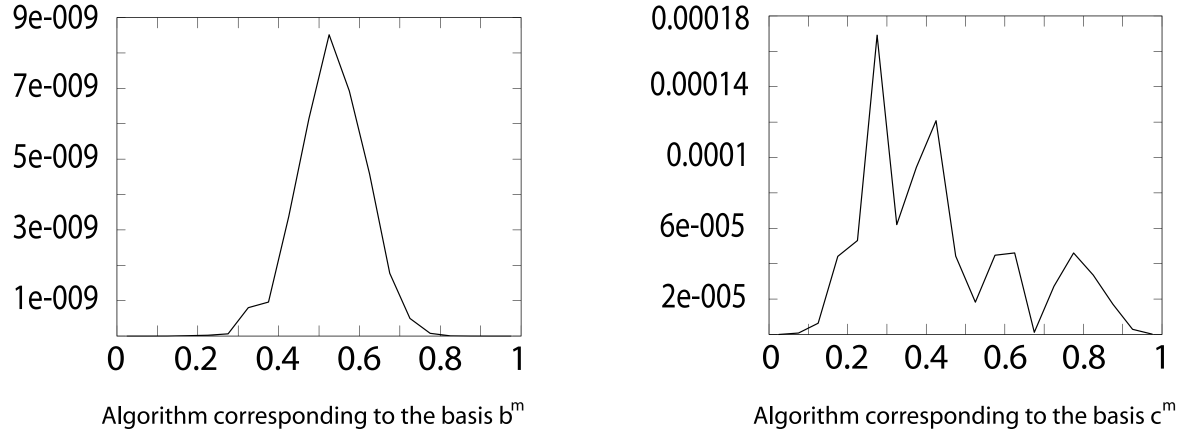

Figure 1: Evaluation of the polynomial p_1

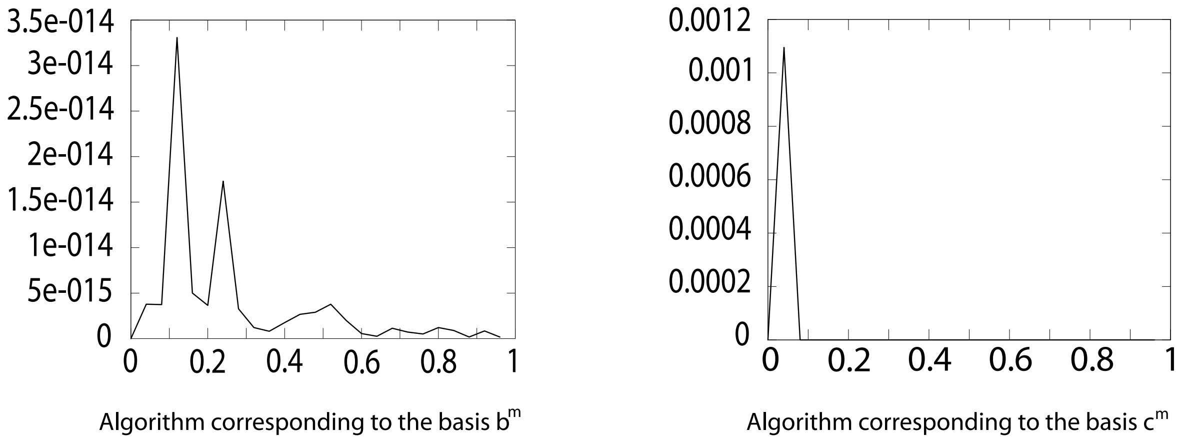

Figure 2: Evaluation of the polynomial p_2

and, in addition, if m is odd $i_{\frac{m-1}{2}} = i_{\frac{m+1}{2}} = m+3$ and, if m is even $i_{\frac{m}{2}} = m+2$.

The previous results shows that the backward error bound (8) corresponding to the basis c^m is slightly smaller than the backward error bound (7) corresponding to the Bernstein basis b^m. However, the following result proves that the Bernstein basis b^m is always better conditioned than the basis c^m.

Theorem 3.3. Let $b^m = (b_0^m, \ldots, b_m^m)$ and $c^m = (c_0^m, \ldots, c_m^m)$ be the bases defined by formula (2) and Definition 2.1, respectively. Then $S_{b^m}(p(t)) \le S_{c^m}(p(t))$ for all polynomial p of degree at most m and for all $t \in [0,1]$.
Proof. Let A_m be the matrix of change of basis such that $c^m = b^m A_m$. By Theorems 5 and 6 of [5] the basis c^m is totally positive. Then, by Theorem 4.3 of [1] the matrix A_m is totally positive and, in particular, nonnegative. Hence the result follows from Lemma 2.1 of [17] (see also Proposition 1 of [10]).

4 Numerical Tests

The numerical experiments performed in this section have been carried out in double precision floating point arithmetic and we have compared the computed evaluations with the exact evaluations obtained with Mathematica. We have performed two kinds of numerical experiments comparing relative errors when evaluating polynomials through the de Casteljau algorithm (using the Bernstein basis b^m) and the algorithm presented in [5] (using the basis c^m). On one hand, we considered random polynomials of degree 10 and 20, and we observed that both algorithms behave very well.

On the other hand, we considered two polynomials which were originally studied by Wilkinson (see [21] and

[22], where he showed ill-conditioning properties of the roots):

$$p_1(x) := \prod_{k=1}^{20} \left(x - \frac{k}{20} \right), \quad p_2(x) := \prod_{k=1}^{20} \left(x - \frac{2}{2^k} \right).$$

We can observe in Figures 1 and 2 that the de Casteljau algorithm presents a better behaviour close to the roots of the corresponding polynomial. The explanation of this fact in spite of the less backward error of the algorithm associated to the basis c^m (see Propositions 3.1 and 3.2) comes from the better conditioning of Bernstein basis (see Theorem 3.3). However, the polynomials that usually appear in CAD do not present the pathological behaviour of Wilkinson polynomials and have a low degree. So, we can benefit from the greater efficiency of the algorithm associated to the basis c^m when we use polynomials without problematic stability properties.

Acknowledgements

This work was partially supported by the Spanish Research Grant BFM2003-03515.

References

[1] J. M. Carnicer, and J. M. Pena, "Shape preserving representations and optimality of the Bernstein basis", Advances in Computational Mathematics, 1, pp. 173-196, 1993.

[2] J. M. Carnicer, and J. M. Pena, "Totally positive bases and optimallity of B-splines", Computer Aided Geometric Design, 11, pp. 635-656, 1994.

[3] N. Dejdumrong, and H. N. Phien, "Efficient algorithms for Bezier curves", Computer Aided Geometric Design, 17, pp. 247-250, 2000.

[4] J. Delgado, and J. M. Peña, "Monotonicity preservation of some polynomial and rational representations", In Information Visualisation, IEEE Computer Society, Los Alamitos (CA), 2002 pp. 57-62, 2003.

[5] J. Delgado, and J. M. Peña, "A shape preserving representation with an evaluation algorithm of linear complexity", Computer Aided Geometric Design, 20, pp. 1-10, 2003.

[6] J. Delgado, and J. M. Peña, "Corner cutting systems", Computer Aided Geometric Design, 22, pp. 81-97, 2005.

[7] J. Delgado, "Algoritmos de evaluación eficientes para el diseño de curvas y superficies", PhD thesis, Universidad de Zaragoza, 2005.

[8] J. Delgado, and J. M. Peña, "On the generalized Ball bases", To appear in Advances in Computational Mathematics.

[9] G. Farin, "Curves and Surfaces for Computer Aided Geometric Design" (4th edition), San Diego, Academic Press, 1996.

[10] R. T. Farouki, and T. N. T. Goodman, "On the optimal stability of Bernstein basis", Math. Comp., 65, pp. 1553-1566, 1996.

[11] T. N. T. Goodman, "Shape preserving representations" In T. Lyche, and L. L. Schumaker, eds., Mathematical Methods in CAGD, Academic Press, Boston, pp. 333-357, 1989.

[12] W. Guojin, and C. Min, "New algorithms for evaluating parametric surface", Progress in Natural Science, 11, pp. 142-148, 2001.

[13] N. J. Higham, "Accuracy and Stability of Numerical Algorithms", SIAM, Philadelphia, 1996.

[14] J. Hoscheck, and D. Lasser, "Fundamentals of Computer Aided Geometric Design", Wellesley, AKPeters, 1993.

[15] E. Mainar, and J. M. Peña, "Error analysis or corner cutting algorithms", Numerical Algorithms, 22, pp. 41-52, 1999.

[16] J. M. Peña, "Shape preserving representations in Computer Aided-Geometric Design", Nova Science Publishers, Inc, 1999.

[17] J. M. Peña, "On the optimal stability of bases of univariate functions", Numerische Mathematik, 91, pp. 305-318, 2002.

[18] H. Shi-Min, W. Guojin and S. Jiaguang, "A Type of Triangular Ball Surface and Its Properties", J. of Comput. Sci. & Technol., 13, pp.63-72, 1998.

[19] H. Shi-Min, W. Guo-Zhao, and J. Tong-Guang, "Properties of two types of generalized Ball curves", Computer-Aided Design, 28, pp. 125-133, 1996.

[20] G. J. Wang, "Ball curve of high degree and its geometric properties", Appl. Math.: A journal of Chinese Universities, 2, pp. 126-140, 1987.

[21] J. H. Wilkinson, "The evaluation of the zeros of ill-conditioned polynomials", Parts I and II, Numer. Math., 1, pp. 150-166 and pp. 167-180, 1959.

[22] J. H. Wilkinson, "Rounding Errors in Algebraic Processes", Notes on Applied Science, 32 (Her Majesty's Stationery Office, London), 1963.