Optimality of Bernstein Representations for Computational Purposes*

J. Delgado
Departamento de Matemática Aplicada/IUMA, Universidad de Zaragoza, Spain
jorgedel@unizar.es

J. M. Peña†
Departamento de Matemática Aplicada/IUMA, Universidad de Zaragoza, Spain
jmpena@unizar.es

Abstract

In this article, optimal properties of the Bernstein basis of polynomials are revisited. In particular, these include optimal shape preserving properties and optimal stability for the evaluation in computer aided geometric design, minimal conditioning of its collocation matrices and fastest convergence rates of the corresponding iteration approximation property. Recent advances on stable evaluation algorithms for this basis will be also presented and discussed.

Keywords: Bernstein basis, optimal properties, de Casteljau algorithm, error analysis
AMS subject classifications: 65G50, 65D17

1 Introduction

The importance of the Bernstein basis of the space of polynomials on a compact interval in Approximation Theory has been well known for many decades. In Computer Aided Geometric Design (CAGD) this basis has played a key role since the beginning of this subject. It is the standard basis for the design of polynomial curves, and its multivariate extensions are also the standard bases for the design of polynomial surfaces. In fact, in Section 2 we recall some relevant properties in CAGD for which this basis is optimal.

The usual algorithm to evaluate a polynomial represented in the Bernstein basis is the de Casteljau algorithm. In Section 3 we present and illustrate results and techniques on the error analysis when using this evaluation algorithm.

*Submitted: July 1, 2011; Revised: Oct. 23, 2011; Second revision: Feb. 2, 2012; Accepted: Feb. 2, 2012.

†J. Delgado and J.M. Peña were partially supported by MTM2009-07315 Spanish Research Grant and by Gobierno de Aragón.

2 Optimal Properties of the Bernstein Basis

This section presents several optimal properties of the Bernstein basis. Let us start by introducing some notations.

A matrix is *totally positive* (TP) if all its minors are nonnegative. A nonnegative matrix is *stochastic* if the elements of each row sum up to 1. We denote by P_n the space of polynomials of degree not greater than n.

Let $\mathcal{U}$ be a vector space of real functions defined on an interval $I \subset \mathbf{R}$ and $(u_0(t), \ldots, u_n(t))$ $(t \in I)$ a basis of $\mathcal{U}$. The *collocation matrix* of $(u_0(t), \ldots, u_n(t))$ at $t_0 < \cdots < t_m$ in I is given by

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

A system of functions is TP when all its collocation matrices are TP. In Computer Aided Geometric Design (CAGD) the functions $u_0, \ldots, u_n$ also satisfy $\sum_{i=0}^n u_i(t) = 1$ $\forall\, t \in I$ (i.e. the system $(u_0, \ldots, u_n)$ is *normalized*), and a normalized TP system is denoted by NTP. In fact, shape preserving representations are associated with NTP bases (see [2]).

The following result was proved in Theorem 4.3 of [2].

Theorem 2.1 *Let $(b_0^n, \ldots, b_n^n)$ be the Bernstein basis. A basis $(v_0, \ldots, v_n)$ of P_n is normalized totally positive if and only if there exists a stochastic totally positive matrix K such that*

$$(v_0, \ldots, v_n) = (b_0^n, \ldots, b_n^n)K.$$

The previous result shows that all NTP bases of the space P_n can be obtained by multiplying the Bernstein basis by TP stochastic matrices. A nonsingular TP stochastic matrix can be factorized in terms of a product of bidiagonal stochastic matrices (see Theorem 2.6 of Chapter 4 of [11]). This biadiagonal decomposition can be interpreted geometrically as a corner cutting algorithm (see p. 66 of [11]), which in turn leads to the following interpretation in CAGD of Theorem 2.1: the Bernstein basis has *optimal shape preserving properties* among all bases of P_n. In [2], the unique NTP basis of a space satisfying the property satisfied by the Bernstein basis in Theorem 2.1 was called the normalized B-basis of the space. This concept generalizes Bernstein bases to more general spaces of functions with NTP bases.

Given a basis $u = (u_0, \ldots, u_n)$ of a real vector space $\mathcal{U}$ of functions defined on a subset I of $\mathbf{R}$ and a function $f \in \mathcal{U}$, we can write $f(t) = \sum_{i=0}^n c_i u_i(t)$ for all $t \in I$, where $c_i \in \mathbf{R}$ for all $i = 0, \ldots, n$. The stability of the basis $u = (u_0, \ldots, u_n)$ with respect to the evaluation at a point is measured by the function $C_u : \mathcal{U} \times I \to \mathbf{R}_+$ given by

$$C_u(f, t) := \sum_{i=0}^n |c_i u_i(t)|.$$

The following optimality result is a consequence of Theorem 3 of [7]. It shows that there does not exist a nonnegative basis of polynomials better conditioned than the Bernstein basis for any polynomial and at any point.

Theorem 2.2 *Let $b = (b_0^n, \ldots, b_n^n)$ be the Bernstein basis. Then there does not exist (up to reordering and positive scaling) another basis $u = (u_0, \ldots, u_n)$ of nonnegative functions in P_n such that $C_u(p, t) \le C_b(p, t)$ for all $t \in [0, 1]$ and $p \in P_n$.*

NTP bases also satisfy the property known as the *progressive iteration approximation* property (see [3]), which plays an important role in the approximation of interpolating curves in CAGD. Let us briefly describe this property. Let us consider a sequence of points $(P_i)_{i=0}^n$ such that the ith point is assigned to a parameter value t_i for $i = 0, 1, \ldots, n$, and a basis $(u_0, \ldots, u_n)$. First we construct a starting curve $\gamma^0(t) = \sum_{i=0}^n P_i^0 u_i(t)$ with $P_i^0 = P_i$ for all $i \in \{0, 1, \ldots, n\}$. Then, computing the adjusting vector $\Delta_i^0 = P_i - \gamma^0(t_i)$ we can take $P_i^1 = P_i^0 + \Delta_i^0$, for $i = 0, 1, \ldots, n$, and construct a new curve as $\gamma^1(t) = \sum_{i=0}^n P_i^1 u_i(t)$. Iterating this process we can get a sequence of curves $\{\gamma^k\}_{k=0}^\infty$. The progressive iteration approximation property holds when this curve sequence converges to a curve interpolating the given initial sequence of points (for more details in the iterative process associated to this property, see [3]). Taking into account that the Bernstein basis is the normalized B-basis of the space P_n (see Example 4.1 of Chapter 4 of [11]), we can deduce from Theorem 4 of [3] the optimal convergence speed of the Bernstein basis.

Theorem 2.3 *The Bernstein basis provides a progressive iterative approximation with the fastest convergence rates among all NTP bases of P_n.*

One key tool used in the proof of the previous result was also fundamental for the following application about the optimal conditioning of the collocation matrices of the Bernstein basis. This fact focuses on the minimal eigenvalues of these matrices and is recalled in the following result. Let us recall that TP matrices have all their eigenvalues nonnegative (cf. Corollary 6.6.7 of [1]). The following result follows from the proof of Theorem 4 of [3].

Theorem 2.4 *The minimal eigenvalue of a Bernstein collocation matrix is always greater than or equal to the minimal eigenvalue of the corresponding collocation matrix of another NTP basis of P_n.*

Given a nonsingular matrix A, let us consider the condition number $\kappa_\infty(A) := \|A\|_\infty \|A^{-1}\|_\infty$. The following result corresponds to Theorem 2.1 (i) of [4] and shows that the collocation matrices of the Bernstein basis are the best conditioned among all the corresponding collocation matrices of NTP bases of the space P_n on $[0, 1]$.

Theorem 2.5 *Let $(b_0^n, \ldots, b_n^n)$ be the Bernstein basis, let $(v_0, \ldots, v_n)$ be another NTP basis of P_n on $[0, 1]$, let $0 \le t_0 < t_1 < \cdots < t_n \le 1$ and $V := M\begin{pmatrix} v_0,\ldots,v_n \\ t_0,\ldots,t_n \end{pmatrix}$ and $B := M\begin{pmatrix} b_0^n,\ldots,b_n^n \\ t_0,\ldots,t_n \end{pmatrix}$. Then*

$$\kappa_\infty(B) \le \kappa_\infty(V).$$

Given a matrix $A = (a_{ij})_{1 \le i,j \le n}$, we denote by $|A|$ the matrix whose (i, j)-entry is $|a_{ij}|$. Let us recall that the Skeel condition number of a nonsingular matrix A, $\mathrm{Cond}(A)$, was introduced by Skeel and measures effects of perturbations of the data in linear systems $Af = c$. It is defined as

$$\mathrm{Cond}(A) := \| \, |A^{-1}| \, |A| \, \|_\infty.$$

The following result was presented in Theorem 2.1 (ii) of [4], and is similar to Theorem 2.5, but using the Skeel condition number of the transposes of the collocation matrices.

Theorem 2.6 *Let $(b_0^n, \ldots, b_n^n)$ be the Bernstein basis, let $(v_0, \ldots, v_n)$ be another TP basis of P_n on $[0,1]$, let $0 \leq t_0 < t_1 < \cdots < t_n \leq 1$ and $V := M\begin{pmatrix} v_0,\ldots,v_n \\ t_0,\ldots,t_n \end{pmatrix}$ and $B := M\begin{pmatrix} b_0^n,\ldots,b_n^N \\ t_0,\ldots,t_n \end{pmatrix}$. Then*

$$\mathrm{Cond}(B^T) \leq \mathrm{Cond}(V^T).$$

Finally, let us mention that many computations with collocation matrices of the Bernstein basis can be performed with high relative accuracy (see [10, 8]). The next section deals with the problem of evaluating a polynomial represented in the Bernstein basis.

3 Evaluation of Bernstein Representations

In this section, we consider error analysis when evaluating polynomials represented in the Bernstein basis. Corner cutting algorithms form the main family of algorithms in CAGD. An algorithm is said to be a corner cutting algorithm if each of its steps consists of a convex combination. An example of corner cutting algorithm is the de Casteljau algorithm. This is the usual algorithm for evaluating an nth-degree Bernstein polynomial

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

where

$$b_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i},$$

at a point $t \in [0,1]$ (see [6]). Let us recall this algorithm.

Algorithm 1 *De Casteljau algorithm* for the evaluation of a polynomial p at a point t

---

Require: $t \in [0,1]$, $n \geq 0$ and $(c_i)_{i=0}^n$
Ensure: $p(t) = \sum_{i=0}^n c_i b_i^n(t)$
  for $i = 0$ to n **do**
    $c_i^0 = c_i$
  end for
  for $r = 1$ to n **do**
    for $i = 0$ to $n - r$ **do**
      $c_i^r = (1-t)c_i^{r-1} + tc_{i+1}^{r-1}$
    end for
  end for
  $p(t) = c_0^n$

As we can observe, the de Casteljau algorithm is a corner cutting algorithm. In [9] Mainar and Peña carried out a backward and forward error analysis of corner cutting algorithms. In particular, the forward error analysis of corner cutting algorithms presented in Corollary 3.2 of [9] applied to the particular case of the de Casteljau algorithm and the Bernstein basis gives rise to the following result.

Theorem 3.1 *Let $p(t) = \sum_{i=0}^{n} c_i b_i^n(t)$ be a polynomial of nth-degree represented in the Bernstein basis $b = (b_0^n, \ldots, b_n^n)$. If $2nu < 1$, where u is the unit roundoff, then the value $\widehat{p}(t) = fl(p(t))$ computed with the de Casteljau algorithm satisfies*

$$|\widehat{p}(t) - p(t)| \leq \gamma_{2n} \sum_{i=0}^{n} |c_i| b_i^n(t) = \gamma_{2n} C_b(p, t),$$

where $\gamma_{2n} := \frac{2nu}{1-2nu}$.

The previous result presents an upper bound of the absolute error of the approximation $\widehat{p}(t)$ to $p(t)$ provided by the de Casteljau algorithm. An estimation of this upper bound can be obtained by evaluating the polynomial $q(t) = \sum_{i=0}^{n} |c_i| b_i^n(t)$ with the de Casteljau algorithm and multiplying the obtained value by the approximation

$$\gamma_{2n} = \frac{2nu}{1 - 2nu} = 2nu + \mathcal{O}(u^2) \approx 2nu.$$

Example 3.2 *Let us consider the 20th-degree polynomial given by*

$$p(t) = \prod_{k=1}^{20} \left(t - \frac{k}{20} \right). \tag{1}$$

The 20 roots of this polynomial are uniformly distributed in the interval $[1/20, 1]$. This polynomial was firstly considered by Wilkinson in [12, 13], where its ill-conditioning was showed.

First we have computed in exact arithmetic the coefficients $c_0, c_1, \ldots, c_{20}$ such that $p(t) = \sum_{i=0}^{20} c_i b_i^{20}(t)$ and the exact values of the polynomial at the points of the following mesh:

$$M = \left\{ \frac{1}{100} + \frac{i}{29} \frac{98}{100}, \quad i = 0, 1 \ldots, 29 \right\}. \tag{2}$$

Then we have evaluated the polynomial by Algorithm 1 in floating point arithmetic at the points in M with double precision. In addition, we have computed the absolute errors corresponding to the obtained approximations. Finally, we have calculated an estimation of the upper bounds of the absolute errors by evaluating the polynomial $\sum_{i=0}^{n} |c_i| b_i^n(t)$ with Algorithm 1 and multiplying the approximation obtained by $2nu$. These data can be seen in Figure 1. We can observe in this figure that the forward error analysis provides quite realistic and sharp bounds for the absolute error. In addition, the shape of the curve corresponding to the error bounds mimics the shape of the curve corresponding to the absolute errors, that is, as the absolute error increases, the corresponding forward bound also increases, showing a good behaviour.

The previous example assumes that the exact coefficients of the Wilkinson polynomial have been computed exactly. It should be noted however that the exact computation of these coefficients in floating point arithmetic is a very ill-conditioned problem.

The upper bound of the absolute error when evaluating $p(t)$ provided in Theorem 3.1 is computed independently of the approximation $\widehat{p}(t)$ given by the de Casteljau algorithm. But, using data calculated while performing the de Casteljau algorithm, more realistic upper bounds of the absolute error can be obtained. Error analyses leading to these kinds of bounds are called running error analyses. In Section 4 of [9] a running error analysis of corner cutting algorithms was carried out. This error

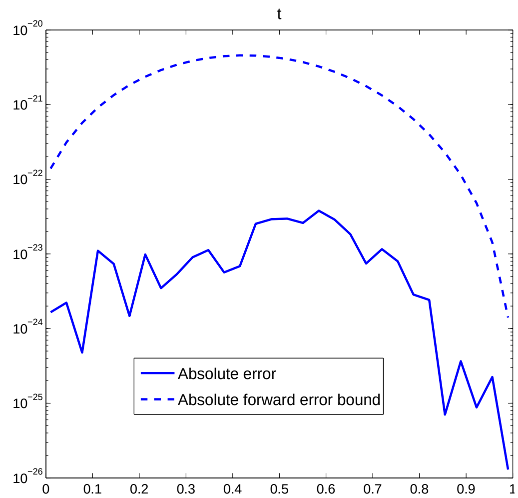

Figure 1: Absolute errors and absolute forward error bounds evaluating $p(t)$

Algorithm 2 *De Casteljau algorithm* with running error bound of the absolute error for the evaluation of a polynomial p at a point t

Require: $t \in [0,1]$, $n \geq 0$ and $(c_i)_{i=0}^n$
Ensure: $\widehat{p}(t) \approx p(t) = \sum_{i=0}^n c_i b_i^n(t)$ and μ such that $|p(t) - \widehat{p}(t)| \leq \mu$
  for $i = 0$ to n **do**
    $c_i^0 = c_i$
    $M_i^0 = |c_i|$
  end for
  for $r = 1$ to n **do**
    for $i = 0$ to $n - r$ **do**
      $c_i^r = (1-t)c_i^{r-1} + tc_{i+1}^{r-1}$
      $M_i^r = (1-t)M_i^{r-1} + tM_{i+1}^{r-1} + |c_i^r|$
    end for
  end for
  $\widehat{p}(t) = c_0^n$
  $\mu = (2M_0^n - \widehat{p}(t))u$

analysis gives rise to an extension of the de Casteljau algorithm, which provides the approximation $\widehat{p}(t)$ to $p(t)$ and, at the same time, an upper bound of the absolute error.

Example 3.3 *Let us consider again the* 20*th-degree polynomial defined in formula* (1)*. In this example we have repeated the process followed in Example 3.2 at the same points, but calculating the running error bounds instead of the absolute error bound provided by the forward error through Algorithm 2. Figure 2 shows the absolute errors, the running error bounds and the absolute error bounds provided by the forward error* *analysis in Theorem 3.1, which had already been computed in Example 3.2. In this figure we can observe that the running error bounds also have a good behaviour and, in addition, these bounds of the absolute errors are sharper and more realistic than the corresponding ones to the forward error analysis.*

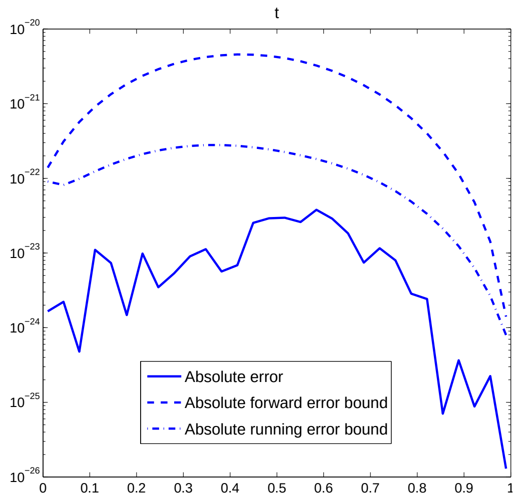

Figure 2: Absolute errors and absolute running and forward error bounds when evaluating $p(t)$

Nevertheless, the relative error is the best measure of the error when approximating a value a by $\widehat{a}$. But the error analyses presented before provide bounds of the absolute error corresponding to the approximation $\widehat{p}(t)$ to $p(t)$ obtained when evaluating a polynomial $p(t)$ at a point t by the de Casteljau algorithm. In [5] the problem of finding bounds of the relative errors corresponding to approximations $\widehat{p}(t)$ to the value of a function $p(t)$ at a point t obtained by an evaluation algorithm was faced. In Theorem 3.1 of [5], Delgado and Peña assumed an absolute error bound $|\widehat{p}(t)-p(t)| \leq u\,K+\mathcal{O}(u^2)$, and provide a sufficient condition on $\widehat{p}(t)$ in order to assure a relative error bound. Applying that theorem to the case of polynomials represented with Bernstein polynomials we obtain the following result.

Theorem 3.4 *Let us consider an evaluation algorithm of a polynomial of the form $p(t) = \sum_{i=0}^{n} c_i b_i^n(t)$ such that $p(t) \neq 0$, let $\widehat{p}(t)$ be the computed value, and let us assume that*

$$|\widehat{p}(t) - p(t)| \leq u\,K + \mathcal{O}(u^2), \tag{3}$$

where u is the unit roundoff. If $|\widehat{p}(t)| > u\,K$, then

$$\left|\frac{\widehat{p}(t) - p(t)}{p(t)}\right| \leq u\,\frac{K}{|\widehat{p}(t)|} + \mathcal{O}(u^2). \tag{4}$$

Moreover, assuming that $\widehat{p}(t)p(t) > 0$, a necessary condition for expecting

$$\left|\frac{\widehat{p}(t) - p(t)}{p(t)}\right| < \frac{1}{2} \tag{5}$$

using a formula with the absolute error bound (3) *is* $|\widehat{p}(t)| > u\,K$.

The previous theorem can be applied to both bounds, the one provided by the forward error analysis and the one provided by the running error analysis. Taking into account the results obtained in Example 3.3 and the experiments in [5], we can observe that running error bounds are more realistic and sharper than forward error bounds. Hence, we apply Theorem 3.4 to the running error bound of the absolute error computed in Algorithm 2, obtaining the following algorithm, which computes an approximation $\widehat{p}(t)$ to the value $p(t)$ at t by using the de Casteljau algorithm and, at the same time, calculates an approximation of the bound of the relative error $|p(t) - \widehat{p}(t)|/|p(t)|$ given by (4).

Algorithm 3 *De Casteljau algorithm* with running error bound of the relative error for the evaluation of a polynomial p at a point t

---

Require: $t \in [0,1]$, $n \geq 0$ and $(c_i)_{i=0}^n$
Ensure: $\widehat{p}(t) \approx p(t) = \sum_{i=0}^n c_i b_i^n(t)$ and μ such that $|p(t) - \widehat{p}(t)|/|p(t)| \leq \mu$ if $|\widehat{p}(t)| > u\,(2M_0^n - \widehat{p}(t))$ or $\mu = -1$ in otherwise if $|\widehat{p}(t)| > u\,\mu$
  for $i = 0$ to n **do**
    $c_i^0 = c_i$
    $M_i^0 = |c_i|$
  end for
  for $r = 1$ to n **do**
    for $i = 0$ to $n - r$ **do**
      $c_i^r = (1-t)c_i^{r-1} + tc_{i+1}^{r-1}$
      $M_i^r = (1-t)M_i^{r-1} + tM_{i+1}^{r-1} + |c_i^r|$
    end for
  end for
  $\widehat{p}(t) = c_0^n$
  $\mu = (2M_0^n - \widehat{p}(t))u$
  if $|\widehat{p}(t)| > \mu$ **then**
    $\mu = \mu/\widehat{p}(t)$
  else
    $\mu = -1$
  end if

Example 3.5 *Let us consider again the* 20*th-degree polynomial defined by* (1)*. First we have computed in exact arithmetic the coefficients* $c_0, c_1, \ldots, c_{20}$ *such that* $p(t) = \sum_{i=0}^{20} c_i b_i^{20}(t)$ *and the exact values of the polynomial at the points of the mesh* M *given by* (2)*. Then we have evaluated the polynomial by Algorithm 3 in floating point arithmetic with double precision at the points in* M*. That algorithm calculates at the same time, under certain conditions, a bound of the corresponding relative error. In addition, we have computed the relative errors corresponding to the obtained approximations. We have checked that, when evaluating polynomial* $p(t)$ *at any point* $t \in M$ *we have that* $|\widehat{p}(t)| > (2M_0^n - \widehat{p}(t))u$ *and, hence, a running relative error bound can be given for all points at* M*. Figure 3 shows the relative errors and its corresponding running bounds. We can observe in the figure that the bounds are very tight. In fact, the polygon formed by the relative running error bounds is very close to the polygon* *formed by the true relative errors and the shape of both polygons are very similar. Hence we can conclude that relative running errors have a good behaviour.*

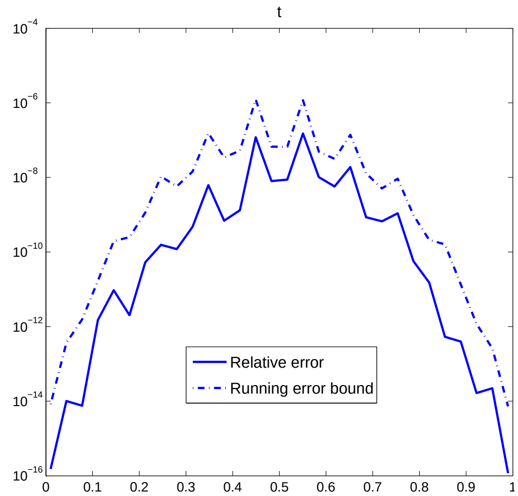

Figure 3: Relative errors and relative running error bounds when evaluating $p(t)$

References

[1] T. Ando. Totally positive matrices. *Linear Algebra Appl.*, 90:165–219, 1987.

[2] J. M. Carnicer and J. M. Peña. Shape preserving representations and optimality of the Bernstein basis. *Adv. Comput. Math.*, 1(2):173–196, 1993.

[3] J. Delgado and J. M. Peña. Progressive iterative approximation and bases with the fastest convergence rates. *Comput. Aided Geom. Design*, 24(1):10–18, 2007.

[4] J. Delgado and J. M. Peña. Optimal conditioning of Bernstein collocation matrices. *SIAM J. Matrix Anal. Appl.*, 31(3):990–996, 2009.

[5] J. Delgado and J. M. Peña. Running relative error for the evaluation of polynomials. *SIAM J. Sci. Comput.*, 31(5):3905–3921, 2009.

[6] G. Farin. *Curves and surfaces for CAGD: a practical guide.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 5th edition, 2002.

[7] R. T. Farouki and T. N. T. Goodman. On the optimal stability of the Bernstein basis. *Math. Comp.*, 65(216):1553–1566, 1996.

[8] P. Koev. Accurate computations with totally nonnegative matrices. *SIAM J. Matrix Anal. Appl.*, 29(3):731–751, 2007.

[9] E. Mainar and J. M. Peña. Error analysis of corner cutting algorithms. *Numer. Algorithms*, 22(1):41–52, 1999.

[10] A. Marco and J.-J. Martínez. A fast and accurate algorithm for solving Bernstein-Vandermonde linear systems. *Linear Algebra Appl.*, 422(2-3):616–628, 2007.

[11] J. M. Peña, editor. *Shape Preserving Representations in Computer-Aided Geoemtric Design.* Nova Science Publishers, Inc., 1999.

[12] J. H. Wilkinson. The evaluation of the zeros of ill-conditioned polynomials. I, II. *Numer. Math.*, 1:150–180, 1959.

[13] J. H. Wilkinson. *Rounding errors in algebraic processes.* Prentice-Hall Inc., Englewood Cliffs, N.J., 1963.