Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 26, pp. 357-380, June 2013

ELA

ACCURATE COMPUTATIONS WITH TOTALLY POSITIVE BERNSTEIN–VANDERMONDE MATRICES*

ANA MARCO† AND JOSÉ-JAVIER MARTÍNEZ†

Abstract. The accurate solution of some of the main problems in numerical linear algebra (linear system solving, eigenvalue computation, singular value computation and the least squares problem) for a totally positive Bernstein–Vandermonde matrix is considered. Bernstein–Vandermonde matrices are a generalization of Vandermonde matrices arising when considering for the space of the algebraic polynomials of degree less than or equal to n the Bernstein basis instead of the monomial basis.

The approach in this paper is based on the computation of the bidiagonal factorization of a totally positive Bernstein–Vandermonde matrix or of its inverse. The explicit expressions obtained for the determinants involved in the process make the algorithm both fast and accurate. The error analysis of this algorithm for computing this bidiagonal factorization and the perturbation theory for the bidiagonal factorization of totally positive Bernstein–Vandermonde matrices are also carried out.

Several applications of the computation with this type of matrices are also pointed out.

Key words. Bernstein–Vandermonde matrix, Bernstein basis, Total positivity, Neville elimination, Bidiagonal decomposition, High relative accuracy.

AMS subject classifications. 15A18, 15A23, 65D05, 65F15, 65F35.

1. Introduction.

The design of accurate and efficient algorithms for structured matrices is a relevant field in numerical linear algebra which in recent years has received a growing attention (see, for instance, the survey paper [12] and references therein). In particular, in Section 2.3 of [12] several different classes of structured matrices are considered, among them the class of totally positive matrices.

Recent books on the subject of totally positive matrices are [17, 34] (see also [16] in connection with Chapter 2 of [17]). These monographs cover many aspects of the theory and applications of totally positive matrices and, although they do not include the topic of accurate computations with this type of matrices, they provide useful

*Received by the editors on February 26, 2013. Accepted for publication on May 26, 2013. Handling Editor: Michael Tsatsomeros.

†Departamento de Física y Matemáticas, Universidad de Alcalá, Alcalá de Henares, Madrid 28871, Spain (ana.marco@uah.es, jjavier.martinez@uah.es). This research has been partially supported by Spanish Research Grant MTM2012-31544 from the Spanish Ministerio de Educación, Cultura y Deporte. The authors are members of the Research Group ASYNACS (Ref. CCEE2011/R34) of Universidad de Alcalá.

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 26, pp. 357-380, June 2013

ELA

references to the work of Demmel and Koev [14, 28, 29].

In our work, we will consider one special class of totally positive matrices which we have analyzed in [30] in the context of linear system solving: Bernstein–Vandermonde matrices. They have also been used in [32] for solving the problem of least squares fitting in the Bernstein basis.

Bernstein–Vandermonde matrices are a generalization of Vandermonde matrices arising when considering the *Bernstein basis* instead of the monomial basis for the space of the algebraic polynomials of degree less than or equal to n.

As recalled in the recent survey [20], the Bernstein polynomials were originally introduced one century ago by Sergei Natanovich Bernstein to facilitate a constructive proof of the Weierstrass approximation theorem. Many years later, in the early 1960s, the work of Bézier and de Casteljau introduced the Bernstein polynomials in the field now known as Computer Aided Geometric Design (see [18]).

In connection with the study of the good properties of the Bernstein polynomials, in [8], it was proved that "the Bernstein basis has optimal shape preserving properties among all normalized totally positive bases for the space of polynomials of degree less than or equal to n over a compact interval".

Other results which are relevant to our work have recently been found for the Bernstein basis. Firstly, the fact that Bernstein–Vandermonde matrices have optimal (i.e., lowest) condition number among all the corresponding collocation matrices of normalized totally positive bases of the space of polynomials of degree less than or equal to n on $[0, 1]$ has been proved in [10]. In addition, the good properties of the Bernstein basis for polynomial evaluation when one uses the de Casteljau algorithm are analyzed in [11]. This last fact is useful, for instance, in regression problems (see [32]) when the regression polynomial is computed in the Bernstein basis: this polynomial can be evaluated to compute the residuals without the need to convert it to the monomial basis.

Other recent applications of the Bernstein basis are shown in Section 9 of [20], for instance to finite element analysis [1, 26]. The explicit conversion between the Bernstein basis and the power basis is exponentially ill-conditioned as the polynomial degree increases [19, 20], and so it is very important that when designing any algorithm for performing numerical computations with polynomials expressed in Bernstein form, all the intermediate operations are developed using this form only.

As recalled in [12], a crucial preliminary stage of the algorithms considered there for the class of totally positive matrices is the decomposition of the matrix as a product of bidiagonal factors. In this sense, one can learn from [28, 29] that the problem of performing accurate computations with totally positive matrices is very

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 26, pp. 357-380, June 2013

ELA

much a *representation problem*: instead of representing a matrix by its entries, it is represented as a product of nonnegative bidiagonal matrices. And this is one of our contributions for the case of Bernstein–Vandermonde matrices: the fast and accurate computation of this *bidiagonal factorization*.

Working along this line, in this work in addition to reviewing the problem of linear system solving considered in [30] and the least squares problem addressed in [32], we show the application of this approach to the computation of eigenvalues and singular values of Bernstein–Vandermonde matrices. As we will see in Section 7, our approach is accurate even for computing the smallest eigenvalues and the smallest singular values of this type of matrices. The computation of the smallest eigenvalue of a Bernstein–Vandermonde matrix is essential for determining, when the Bernstein basis is used, the convergence rate of the iterative method for approximating interpolation curves based on the *progressive iteration approximation property*. It must be observed that the Bernstein basis provides for this procedure the fastest convergence rate among all normalized totally positive basis of the space of the polynomials of degree less than or equal to n [9]. As for the singular values, it will be shown how its accurate computation provides an appropriate procedure for estimating the 2-norm condition number of a Bernstein–Vandermonde matrix.

The algorithms for solving all these problems are based on the bidiagonal factorization of the Bernstein–Vandermonde matrix or of its inverse. The error analysis of our algorithm for computing this bidiagonal factorization and the perturbation theory for the bidiagonal factorization of Bernstein–Vandermonde matrices are also considered in this work.

Factorizations in terms of bidiagonal matrices are very useful when working with Vandermonde [5, 25], Cauchy [6], Cauchy-Vandermonde [33] and generalized Vandermonde matrices [14]. An application of the bidiagonal decomposition to the solving of linear systems with a different class of matrices (the so-called ϵ-BD matrices) has recently been presented in [3].

As for the other stages of our algorithms, we will use several algorithms developed by Plamen Koev [27, 28, 29]. An important condition for an algorithm to be accurate is the so called NIC (no inaccurate cancellation) condition, which is emphasized in [12] (see also Section 5 in [13]):

NIC: The algorithm only multiplies, divides, adds (resp., subtracts) real numbers with like (resp., differing) signs, and otherwise only adds or subtracts input data.

The rest of the paper is organized as follows. Some basic results on Neville elimination and total positivity are recalled in Section 2. In Section 3, the bidiagonal factorization of a Bernstein–Vandermonde matrix and of its inverse are considered.

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 26, pp. 357-380, June 2013

ELA

The algorithm for computing the bidiagonal factorization of those matrices is also included in this section. In Section 4, the algorithms for solving linear systems, computing eigenvalues, computing singular values and solving least squares problems for the case of totally positive Bernstein–Vandermonde matrices are presented. Section 5 is devoted to the study of the error analysis of the bidiagonal factorization algorithm presented in Section 3. In Section 6, the perturbation theory is carried out. Finally, Section 7 is devoted to illustrate the accuracy of the algorithms by means of some numerical experiments.

2. Basic facts on Neville elimination and total positivity. To make this paper as self-contained as possible, we will briefly recall in this section some basic results on Neville elimination and total positivity which will be essential for obtaining the results presented in Section 3. Our notation follows the notation used in [21] and [22]. Given k, $n \in \mathbb{N}$ $(1 \leq k \leq n)$, $Q_{k,n}$ will denote the set of all increasing sequences of k positive integers less than or equal to n.

Let A be an $l \times n$ real matrix. For $k \leq l$, $m \leq n$, and for any $\alpha \in Q_{k,l}$ and $\beta \in Q_{m,n}$, we will denote by $A[\alpha|\beta]$ the $k \times m$ submatrix of A containing the rows numbered by α and the columns numbered by β.

The fundamental theoretical tool for obtaining the results presented in this paper is the *Neville elimination* [21, 22], a procedure that makes zeros in a matrix adding to a given row an appropriate multiple of the previous one.

Let $A = (a_{i,j})_{1\leq i\leq l;1\leq j\leq n}$ be a matrix where $l \geq n$. The Neville elimination of A consists of $n-1$ steps resulting in a sequence of matrices $A_1 := A \to A_2 \to \cdots \to A_n$, where $A_t = (a_{i,j}^{(t)})_{1\leq i\leq l;1\leq j\leq n}$ has zeros below its main diagonal in the $t-1$ first columns. The matrix A_{t+1} is obtained from A_t $(t = 1, \ldots, n-1)$ by using the following formula:

$$a_{i,j}^{(t+1)} := \begin{cases} a_{i,j}^{(t)} , & \text{if} \quad i \leq t, \\ a_{i,j}^{(t)} - (a_{i,t}^{(t)}/a_{i-1,t}^{(t)})a_{i-1,j}^{(t)} , & \text{if} \quad i \geq t+1 \text{ and } j \geq t+1, \\ 0 , & \text{otherwise.} \end{cases}$$

In this process, the element

$$p_{i,j} := a_{i,j}^{(j)}, \qquad 1 \leq j \leq n, \quad j \leq i \leq l$$

is called (i,j) *pivot* of the Neville elimination of A. The process would break down if any of the pivots $p_{i,j}$ $(1 \leq j \leq n, \quad j \leq i \leq l)$ is zero. In that case we can move the corresponding rows to the bottom and proceed with the new matrix, as described in [21]. The Neville elimination can be done without row exchanges if all the pivots are nonzero, as it will happen in our situation. The pivots $p_{i,i}$ are called *diagonal pivots*.

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 26, pp. 357-380, June 2013

ELA

If all the pivots $p_{i,j}$ are nonzero, then $p_{i,1} = a_{i,1}\ \forall i$ and, by Lemma 2.6 of [21]

$$p_{i,j} = \frac{\det A[i-j+1,\ldots,i|1,\ldots,j]}{\det A[i-j+1,\ldots,i-1|1,\ldots,j-1]}, \qquad 1 < j \leq n,\ j \leq i \leq l.$$

The element

$$m_{i,j} = \frac{p_{i,j}}{p_{i-1,j}}, \qquad 1 \leq j \leq n,\ \ j < i \leq l,$$

is called *multiplier* of the Neville elimination of A. The matrix $U := A_n$ is upper triangular and has the diagonal pivots on its main diagonal.

The *complete Neville elimination* of a matrix A consists of performing the Neville elimination of A for obtaining U and then continue with the Neville elimination of U^T. The (i,j) pivot (respectively, multiplier) of the complete Neville elimination of A is the (j,i) pivot (respectively, multiplier) of the Neville elimination of U^T, if $j \geq i$. When no row exchanges are needed in the Neville elimination of A and U^T, we say that the complete Neville elimination of A can be done without row and column exchanges, and in this case, the multipliers of the complete Neville elimination of A are the multipliers of the Neville elimination of A if $i \geq j$ and the multipliers of the Neville elimination of A^T if $j \geq i$.

A detailed error analysis of Neville elimination has been carried out in [2]. However, our approach uses Neville elimination as a theoretical tool but it does not apply that algorithm for obtaining the bidiagonal factorization of Bernstein–Vandermonde matrices.

A matrix is called *totally positive* (respectively, *strictly totally positive*) if all its minors are nonnegative (respectively, positive). The Neville elimination characterizes the strictly totally positive matrices as follows [21]:

THEOREM 2.1. *A matrix is strictly totally positive if and only if its complete Neville elimination can be performed without row and column exchanges, the multipliers of the Neville elimination of A and A^T are positive, and the diagonal pivots of the Neville elimination of A are positive.*

It is well known [8] that the Bernstein–Vandermonde matrix is a strictly totally positive matrix when the nodes satisfy $0 < x_1 < x_2 < \cdots < x_{l+1} < 1$, but this result is also a consequence of our Theorem 3.2.

3. Bidiagonal factorizations. The *Bernstein basis* of the space $\Pi_n(x)$ of polynomials of degree less than or equal to n on the interval $[0,1]$ is:

$$\mathcal{B}_n = \left\{ b_i^{(n)}(x) = \binom{n}{i}(1-x)^{n-i}x^i, \quad i = 0,\ldots,n \right\}.$$

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 26, pp. 357-380, June 2013

The matrix

$$A = \begin{bmatrix} \binom{n}{0}(1-x_1)^n & \binom{n}{1}x_1(1-x_1)^{n-1} & \cdots & \binom{n}{n}x_1^n \\ \binom{n}{0}(1-x_2)^n & \binom{n}{1}x_2(1-x_2)^{n-1} & \cdots & \binom{n}{n}x_2^n \\ \vdots & \vdots & \ddots & \vdots \\ \binom{n}{0}(1-x_{l+1})^n & \binom{n}{1}x_{l+1}(1-x_{l+1})^{n-1} & \cdots & \binom{n}{n}x_{l+1}^n \end{bmatrix}$$

is the $(l+1)\times(n+1)$ *Bernstein–Vandermonde matrix* for the Bernstein basis $\mathcal{B}_n$ and the nodes $\{x_i\}_{1\leq i\leq l+1}$. From now on, we will assume $0 < x_1 < x_2 < \cdots < x_{l+1} < 1$. In this situation, the Bernstein–Vandermonde matrix is a strictly totally positive matrix [8].

Let us observe that when $l = n$, the matrix A is the coefficient matrix of the linear system associated with the following interpolation problem in the Bernstein basis $\mathcal{B}_n$: given the interpolation nodes $\{x_i : \ i = 1, \ldots, n+1\}$, where $0 < x_1 < x_2 < \cdots < x_{n+1} < 1$, and the interpolation data $\{b_i : \ i = 1, \ldots, n+1\}$ find the polynomial

$$p(x) = \sum_{k=0}^{n} a_k b_k^{(n)}(x)$$

such that $p(x_i) = b_i$ for $i = 1, \ldots, n+1$.

In the situation we are considering the following two theorems hold:

THEOREM 3.1. *Let $A = (a_{i,j})_{1\leq i,j\leq n+1}$ be a Bernstein–Vandermonde matrix whose nodes satisfy $0 < x_1 < x_2 < \cdots < x_n < x_{n+1} < 1$. Then A^{-1} admits a factorization in the form*

$$A^{-1} = G_1 G_2 \cdots G_n D^{-1} F_n F_{n-1} \cdots F_1,$$

where F_i $(1 \leq i \leq n)$ are $(n+1)\times(n+1)$ bidiagonal matrices of the form

$$F_i = \begin{bmatrix} 1 & & & & & & & & \\ 0 & 1 & & & & & & & \\ & \ddots & \ddots & & & & & & \\ & & 0 & 1 & & & & & \\ & & & -m_{i+1,i} & 1 & & & & \\ & & & & -m_{i+2,i} & 1 & & & \\ & & & & & \ddots & \ddots & & \\ & & & & & & -m_{n+1,i} & 1 \end{bmatrix},$$

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 26, pp. 357-380, June 2013

ELA

G_i^T $(1 \leq i \leq n)$ are $(n+1) \times (n+1)$ bidiagonal matrices of the form

$$G_i^T = \begin{bmatrix} 1 & & & & & & & & \\ 0 & 1 & & & & & & & \\ & \ddots & \ddots & & & & & & \\ & & 0 & 1 & & & & & \\ & & & -\widetilde{m}_{i+1,i} & 1 & & & & \\ & & & & -\widetilde{m}_{i+2,i} & 1 & & & \\ & & & & & \ddots & \ddots & & \\ & & & & & & -\widetilde{m}_{n+1,i} & 1 \end{bmatrix},$$

and D is a diagonal matrix of order $n+1$

$$D = \operatorname{diag}\{p_{1,1}, p_{2,2}, \ldots, p_{n+1,n+1}\}.$$

The quantities $m_{i,j}$ are the multipliers of the Neville elimination of the Bernstein–Vandermonde matrix A, and have the expression

$$m_{i,j} = \frac{(1-x_i)^{n-j+1}(1-x_{i-j})\prod_{k=1}^{j-1}(x_i - x_{i-k})}{(1-x_{i-1})^{n-j+2}\prod_{k=2}^{j}(x_{i-1} - x_{i-k})}, \tag{3.1}$$

where $j = 1, \ldots, n$ and $i = j+1, \ldots, n+1$.

The quantities $\widetilde{m}_{i,j}$ are the multipliers of the Neville elimination of A^T and their expression is

$$\widetilde{m}_{i,j} = \frac{(n-i+2)\cdot x_j}{(i-1)(1-x_j)}, \tag{3.2}$$

where $j = 1, \ldots, n$ and $i = j+1, \ldots, n+1$.

Finally, the ith diagonal element of D is the diagonal pivot of the Neville elimination of A and its expression is

$$p_{i,i} = \frac{\binom{n}{i-1}(1-x_i)^{n-i+1}\prod_{k<i}(x_i - x_k)}{\prod_{k=1}^{i-1}(1-x_k)} \tag{3.3}$$

for $i = 1, \ldots, n+1$.

Proof. It can be found in [30]. □

THEOREM 3.2. *Let $A = (a_{i,j})_{1\leq i\leq l+1; 1\leq j\leq n+1}$ be a Bernstein–Vandermonde matrix for the Bernstein basis $\mathcal{B}_n$ whose nodes satisfy $0 < x_1 < x_2 < \cdots < x_l < x_{l+1} < 1$. Then A admits a factorization in the form*

$$A = F_l F_{l-1} \cdots F_1 D G_1 \cdots G_{n-1} G_n$$

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 26, pp. 357-380, June 2013

where F_i $(1 \leq i \leq l)$ *are* $(l+1) \times (l+1)$ *bidiagonal matrices of the form*

$$F_i = \begin{bmatrix} 1 & & & & & & & & & \\ 0 & 1 & & & & & & & & \\ & \ddots & \ddots & & & & & & & \\ & & 0 & 1 & & & & & & \\ & & & m_{i+1,1} & 1 & & & & & \\ & & & & m_{i+2,2} & 1 & & & & \\ & & & & & \ddots & \ddots & & & \\ & & & & & & m_{l+1,l+1-i} & 1 & & \end{bmatrix},$$

G_i^T $(1 \leq i \leq n)$ *are* $(n+1) \times (n+1)$ *bidiagonal matrices of the form*

$$G_i^T = \begin{bmatrix} 1 & & & & & & & & & \\ 0 & 1 & & & & & & & & \\ & \ddots & \ddots & & & & & & & \\ & & 0 & 1 & & & & & & \\ & & & \widetilde{m}_{i+1,1} & 1 & & & & & \\ & & & & \widetilde{m}_{i+2,2} & 1 & & & & \\ & & & & & \ddots & \ddots & & & \\ & & & & & & \widetilde{m}_{n+1,n+1-i} & 1 & & \end{bmatrix},$$

and D *is the* $(l+1) \times (n+1)$ *diagonal matrix*

$$D = (d_{i,j})_{1 \leq i \leq l+1; 1 \leq j \leq n+1} = \operatorname{diag}\{p_{1,1}, p_{2,2}, \ldots, p_{n+1,n+1}\}.$$

The expressions of the multipliers $m_{i,j}$ $(j = 1, \ldots, n+1; \quad i = j+1, \ldots, l+1)$ *of the Neville elimination of* A*, the multipliers* $\widetilde{m}_{i,j}$ $(j = 1, \ldots, n; \quad i = j+1, \ldots, n+1)$ *of the Neville elimination of* A^T*, and the diagonal pivots* $p_{i,i}$ $(i = 1, \ldots, n+1)$ *of the Neville elimination of* A *are also in this case given by Eq.* (3.1)*, Eq.* (3.2) *and Eq.* (3.3)*, respectively.*

When in the expression of F_i *we have some multiplier* $m_{k,j}$ *which does not exist (because* $j > n+1$*), then the corresponding entry is equal to zero, and analogously for the entries in* G_i^T*.*

Proof. It can be found in [32]. □

It must be observed that in the square case, the matrices F_i $(i = 1, \ldots, l)$ and the matrices G_j $(j = 1, \ldots, n)$ that appear in the bidiagonal factorization of A are not the same bidiagonal matrices that appear in the bidiagonal factorization of A^{-1}, nor their inverses (see Theorem 3.1 and Theorem 3.2). The multipliers of the Neville elimination of A and A^T give us the bidiagonal factorization of A and A^{-1}, but

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 26, pp. 357-380, June 2013

ELA

obtaining the bidiagonal factorization of A from the bidiagonal factorization of A^{-1} (or vice versa) is not straightforward. The structure of the bidiagonal matrices that appear in both factorizations is not preserved by the inversion, that is, in general, F_i^{-1} $(i = 1, \ldots, l)$ and G_j^{-1} $(j = 1, \ldots, n)$ are not bidiagonal matrices. See [23] for a more detailed explanation and Example 3.4 below.

A fast and accurate algorithm for computing the bidiagonal factorization of the totally positive Bernstein–Vandermonde matrix A and of its inverse (when it exists) has been developed by using the expressions (3.1), (3.2) and (3.3) for the computation of the multipliers $m_{i,j}$ and $\widetilde{m}_{i,j}$, and the diagonal pivots $p_{i,i}$ of its Neville elimination [30, 32]. Given the nodes $\{x_i\}_{1\leq i\leq l+1} \in (0,1)$ and the degree n of the Bernstein basis, it returns a matrix $M \in \mathbb{R}^{(l+1)\times(n+1)}$ such that

$$\begin{array}{ll} M_{i,i} = p_{i,i} & i = 1, \ldots, n+1, \\ M_{i,j} = m_{i,j} & j = 1, \ldots, n+1;\ i = j+1, \ldots, l+1, \\ M_{i,j} = \widetilde{m}_{j,i} & i = 1, \ldots, n;\ j = i+1, \ldots, n+1. \end{array}$$

The algorithm, which we have called `TNBDBV` in the square case (and `TNBDBVR` in the rectangular case), does not construct the Bernstein–Vandermonde matrix, it only works with the nodes $\{x_i\}_{1\leq i\leq l+1}$. Its computational cost is of $O(ln)$ arithmetic operations, and has high relative accuracy because it only involves arithmetic operations that avoid inaccurate cancellation. The implementation in MATLAB of the algorithm can be taken from [27].

In order to facilitate the understanding of the error analysis presented in Section 5, we include here the pseudocode corresponding to the algorithm `TNBDBVR` for the case of a Bernstein–Vandermonde matrix for the Bernstein basis $\mathcal{B}_n$ and the nodes $\{x_i\}_{1\leq i\leq l+1}$, where $l \geq n$:

Computation of the $m_{i,j}$ given by equation (3.1):

`for` $i = 2 : l+1$

$M = \frac{(1-x_i)^n}{(1-x_{i-1})^{n+1}}$

$m_{i,1} = (1 - x_{i-1}) \cdot M$

$k = \min(i-2, n)$

`for` $j = 1 : k$

$M = \frac{(1-x_{i-1})(x_i - x_{i-j})}{(1-x_i)(x_{i-1} - x_{i-j-1})} \cdot M$

$m_{i,j+1} = (1 - x_{i-j-1}) \cdot M$

`end`

`end`

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 26, pp. 357-380, June 2013

ELA

Computation of the $\widetilde{m}_{i,j}$ given by equation (3.2):

`for` $j = 1 : n$

$c_j = \frac{x_j}{1-x_j}$

`for` $i = j + 1 : n + 1$

$\widetilde{m}_{i,j} = \frac{n-i+2}{i-1} \cdot c_j$

`end`

`end`

Computation of the $p_{i,i}$ of D given by equation (3.3):

$q = 1$

$p_{1,1} = (1 - x_1)^n$

`for` $i = 1 : n$

$q = \frac{(n-i+1)}{i(1-x_i)} \cdot q$

$aux = 1$

`for` $k = 1 : i$

$aux = (x_{i+1} - x_k) \cdot aux$

`end`

$p_{i+1,i+1} = q \cdot (1 - x_{i+1})^{n-i} \cdot aux$

`end`

REMARK 3.3. The algorithms `TNBDBV` and `TNBDBVR` compute the matrix M, denoted as $\mathcal{BD}(A)$ in [28], which represents the *bidiagonal decomposition* of A. But it is a remarkable fact that in the square case the same matrix $\mathcal{BD}(A)$ also serves to represent the bidiagonal decomposition of A^{-1}. The following example illustrates this fact.

EXAMPLE 3.4. Let us consider the Bernstein–Vandermonde matrix A for the Bernstein basis $\mathcal{B}_2$ and the nodes $\frac{1}{4} < \frac{1}{2} < \frac{3}{4}$:

$$A = \begin{bmatrix} 9/16 & 3/8 & 1/16 \\ 1/4 & 1/2 & 1/4 \\ 1/16 & 3/8 & 9/16 \end{bmatrix}.$$

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 26, pp. 357-380, June 2013

ELA

The bidiagonal decomposition of A is

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1/4 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 4/9 & 1 & 0 \\ 0 & 3/4 & 1 \end{bmatrix} \begin{bmatrix} 9/16 & 0 & 0 \\ 0 & 1/3 & 0 \\ 0 & 0 & 1/3 \end{bmatrix} \begin{bmatrix} 1 & 2/3 & 0 \\ 0 & 1 & 1/2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1/6 \\ 0 & 0 & 1 \end{bmatrix},$$

the bidiagonal decomposition of A^{-1} is

$$\begin{bmatrix} 1 & -2/3 & 0 \\ 0 & 1 & -1/6 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1/2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 16/9 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -3/4 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -4/9 & 1 & 0 \\ 0 & -1/4 & 1 \end{bmatrix}$$

and the matrix $\mathcal{BD}(A)$ is

$$\mathcal{BD}(A) = \begin{bmatrix} 9/16 & 2/3 & 1/6 \\ 4/9 & 1/3 & 1/2 \\ 1/4 & 3/4 & 1/3 \end{bmatrix}.$$

It must be noted that while Neville elimination has been the key theoretical tool for the analysis of the bidiagonal decomposition of A, it generally fails to provide an accurate algorithm for computing $\mathcal{BD}(A)$ (see [30] and Example 7.1 below). Consequently, the accurate computation of $\mathcal{BD}(A)$ is an important task, a task which we have carried out for the case of Bernstein–Vandermonde matrices.

4. Accurate computations with Bernstein–Vandermonde matrices. In this section, four fundamental problems in numerical linear algebra (linear system solving, eigenvalue computation, singular value computation and the least squares problem) are considered for the case of a totally positive Bernstein–Vandermonde matrix. The bidiagonal factorization of the Bernstein–Vandermonde matrix (or its inverse) led us to develop accurate and efficient algorithms for solving each one of these problems.

Let us observe here that, of course, one could try to solve these problems by using standard algorithms. However the solution provided by them will generally be less accurate since Bernstein–Vandermonde matrices are ill conditioned (see [30]) and these algorithms can suffer from inaccurate cancellation, since they do not take into account the structure of the matrix, which is crucial in our approach.

Some results concerning the conditioning of the Bernstein-Vandermonde matrices can be found in [10]. The condition number of other types of interpolation matrices is also analyzed using singular values, in [7].

4.1. Linear system solving. The fast and accurate solution of structured linear systems is a problem that has been studied in the field of numerical linear algebra for different types of structured matrices (see, for example, [5, 6, 14, 33]). Now we

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 26, pp. 357-380, June 2013

ELA

will consider this problem for the case of totally positive Bernstein–Vandermonde matrices.

Let $Ax = b$ be a linear system whose coefficient matrix A is a square Bernstein–Vandermonde matrix of order $n+1$ generated by the nodes $\{x_i\}_{1\leq i\leq n+1}$, where $0 < x_1 < \cdots < x_{n+1} < 1$. An application which involves the solution of this type of linear system, in the context of Lagrange bivariate interpolation by using the bivariate tensor-product Bernstein basis, has been presented in [31].

The following algorithm solves $Ax = b$ accurately with a computational cost of $O(n^2)$ arithmetic operations (see [30] for the details):

INPUT: The nodes $\{x_i\}_{1\leq i\leq n+1}$ and the data vector $b \in \mathbb{R}^{n+1}$.

OUTPUT: The solution vector $x \in \mathbb{R}^{n+1}$.

- *Step 1:* Computation of the bidiagonal decomposition of A^{-1} by using `TNBDBV`.
- *Step 2:* Computation of

$$x = A^{-1}b = G_1G_2\cdots G_nD^{-1}F_nF_{n-1}\cdots F_1b.$$

Step 2 can be carried out by using the algorithm `TNSolve` of P. Koev [27]. Given the bidiagonal factorization of the matrix A, `TNSolve` solves $Ax = b$ accurately by using backward substitution.

Several examples illustrating the good behaviour of our algorithm can be found in [30].

4.2. Eigenvalue computation. Let A be a square Bernstein–Vandermonde matrix of order $n+1$ generated by the nodes $\{x_i\}_{1\leq i\leq n+1}$, where $0 < x_1 < \cdots < x_{n+1} < 1$. The following algorithm computes accurately the eigenvalues of A.

INPUT: The nodes $\{x_i\}_{1\leq i\leq n+1}$.

OUTPUT: A vector $x \in \mathbb{R}^{n+1}$ containing the eigenvalues of A.

- *Step 1:* Computation of the bidiagonal decomposition of A by using `TNBDBV`.
- *Step 2:* Given the result of Step 1, computation of the eigenvalues of A by using the algorithm `TNEigenvalues`.

`TNEigenvalues` is an algorithm of P. Koev [28] which computes accurate eigenvalues of a totally positive matrix starting from its bidiagonal factorization. The computational cost of `TNEigenvalues` is of $O(n^3)$ arithmetic operations (see [28]) and its implementation in MATLAB can be taken from [27]. In this way, as the computational cost of Step 1 is of $O(n^2)$ arithmetic operations, the cost of the whole algorithm is of $O(n^3)$ arithmetic operations.

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 26, pp. 357-380, June 2013

ELA

4.3. The least squares problem. Let $A \in \mathbb{R}^{(l+1)\times(n+1)}$ be a Bernstein–Vandermonde matrix generated by the nodes $\{x_i\}_{1\leq i\leq l+1}$, where $0 < x_1 < \cdots < x_{l+1} < 1$ and $l > n$. Let $b \in \mathbb{R}^{l+1}$ be a data vector. The least squares problem associated to A and b consists of computing a vector $x \in \mathbb{R}^{n+1}$ minimizing $\| Ax - b \|_2$.

Taking into account that in the situation we are considering A is a strictly totally positive matrix, it has full rank, and the method based on the QR decomposition due to Golub [24] is adequate [4]. For the sake of completeness, we include the following result (see Section 1.3.1 in [4]) which will be essential in the construction of our algorithm.

THEOREM 4.1. *Let $Ac = f$ a linear system where $A \in \mathbb{R}^{(l+1)\times(n+1)}$, $l \geq n$, $c \in \mathbb{R}^{n+1}$ and $f \in \mathbb{R}^{l+1}$. Assume that $rank(A) = n+1$, and let the QR decomposition of A be given by*

$$A = Q \begin{bmatrix} R \\ 0 \end{bmatrix},$$

where $Q \in \mathbb{R}^{(l+1)\times(l+1)}$ is an orthogonal matrix and $R \in \mathbb{R}^{(n+1)\times(n+1)}$ is an upper triangular matrix with positive diagonal entries. Then the solution of the least squares problem $min_c \| f - Ac \|_2$ is obtained from

$$\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = Q^T f, \quad Rc = d_1, \quad r = Q \begin{bmatrix} 0 \\ d_2 \end{bmatrix},$$

where $d_1 \in \mathbb{R}^{n+1}$, $d_2 \in \mathbb{R}^{l-n}$ and $r = f - Ac$. In particular $\| r \|_2 = \| d_2 \|_2$.

The following algorithm, which is based on the previous theorem, solves in an accurate and efficient way our least squares problem:

INPUT: The nodes $\{x_i\}_{1\leq i\leq l+1}$, the data vector f, and the degree n of the Bernstein basis.

OUTPUT: The vector $x \in \mathbb{R}^{n+1}$ minimizing $\| Ax-b \|_2$ and the minimum residual $r = b - Ax$.

- *Step 1*: Computation of the bidiagonal factorization of A by means of `TNBDBV`.
- *Step 2*: Given the result of Step 1, computation of the QR decomposition of A by using `TNQR`.
- *Step 3*: Computation of

$$d = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = Q^T f.$$

- *Step 4*: Solution of the upper triangular system $Rc = d_1$.

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 26, pp. 357-380, June 2013

ELA

- *Step 5*: Computation of

$$r = Q \begin{bmatrix} 0 \\ d_2 \end{bmatrix}.$$

The algorithm `TNQR` has been developed by P. Koev, and given the bidiagonal factorization of A, it computes the matrix Q and the bidiagonal factorization of the matrix R. Let us point out here that if A is strictly totally positive, then R is strictly totally positive. `TNQR` is based on Givens rotations, has a computational cost of $O(l^2 n)$ arithmetic operations if the matrix Q is required, and its high relative accuracy comes from the avoidance of inaccurate cancellation [29]. Its implementation in MATLAB can be obtained from [27].

Steps 3 and 5 are carried out by using the standard matrix multiplication command of MATLAB. As for Step 4, it is done by means of the algorithm `TNSolve` of Koev.

The computational cost of the whole algorithm is led by the cost of computing the QR decomposition of A, and therefore, it is of $O(l^2 n)$ arithmetic operations.

Some numerical experiments which show the good behaviour of our algorithm when solving problems of polynomial regression in the Bernstein basis have been presented in [32].

4.4. Singular value computation. Let $A \in \mathbb{R}^{(l+1)\times(n+1)}$ be a Bernstein–Vandermonde matrix generated by the nodes $\{x_i\}_{1\leq i\leq l+1}$, where $0 < x_1 < \cdots < x_{l+1} < 1$ and $l > n$. The following algorithm computes in an accurate and efficient way the singular values of A.

INPUT: The nodes $\{x_i\}_{1\leq i\leq l+1}$ and the degree n of the Bernstein basis.

OUTPUT: A vector $x \in \mathbb{R}^{n+1}$ containing the singular values of A.

- *Step 1:* Computation of the bidiagonal decomposition of A by using `TNBDBV`.
- *Step 2:* Given the result of Step 1, computation of the singular values by using `TNSingularValues`.

`TNSingularValues` is an algorithm of P. Koev that computes accurate singular values of a totally positive matrix starting from its bidiagonal factorization [28]. Its computational cost is of $O(ln^2)$ and its implementation in MATLAB can be found in [27]. Taking this complexity into account, the computational cost of our algorithm for computing the singular values of a totally positive Bernstein–Vandermonde matrix is of $O(ln^2)$ arithmetic operations.

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 26, pp. 357-380, June 2013

ELA

5. Error analysis. In this section, the error analysis of the algorithm `TNBDBV` (`TNBDBVR` in the rectangular case) for computing the bidiagonal factorization of a totally positive Bernstein–Vandermonde matrix (included in Section 3) is carried out. For our error analysis, we use the standard model of floating point arithmetic (see Section 2.2 of [25]):

Let x, y be floating point numbers and ϵ be the *machine precision*,

$$fl(x \odot y) = (x \odot y)(1+\delta)^{\pm 1}, \quad \text{where } |\delta| \leq \epsilon, \quad \odot \in \{+, -, \times, /\}.$$

The following theorem shows that `TNBDBVR`, and in consequence `TNBDBV`, compute the bidiagonal decomposition of a Bernstein–Vandermonde matrix accurately in floating point arithmetic.

THEOREM 5.1. *Let A be a strictly totally positive Bernstein–Vandermonde matrix for the Bernstein basis $\mathcal{B}_n$ and the nodes $\{x_i\}_{1\leq i\leq l+1}$. Let $\mathcal{BD}(A) = (b_{i,j})_{1\leq i\leq l+1;\ 1\leq j\leq n+1}$ be the matrix representing the exact bidiagonal decomposition of A and $(\widehat{b}_{i,j})_{1\leq i\leq l+1;1\leq j\leq n+1}$ be the matrix representing the computed bidiagonal decomposition of A by means of the algorithm* `TNBDBVR` *in floating point arithmetic with machine precision ϵ. Then*

$$|\widehat{b}_{i,j} - b_{i,j}| \leq \frac{(8nl - 4n^2 + 2n)\epsilon}{1 - (8nl - 4n^2 + 2n)\epsilon} b_{i,j}, \quad i = 1, \ldots, l+1;\ j = 1, \ldots, n+1.$$

Proof. Accumulating the relative errors in the style of Higham (see Chapter 3 of [25], [15] and [28]) in the computation of the $m_{i,j}$ by means of the algorithm `TNBDBVR` included in Section 3, we obtain

$$|\widehat{m}_{i,j} - m_{i,j}| \leq \frac{(8nl - 4n^2 + 2n)\epsilon}{1 - (8nl - 4n^2 + 2n)\epsilon} m_{i,j}, \tag{5.1}$$

for $j = 1, \ldots, n+1$ and $i = j+1, \ldots, l+1$, where $\widehat{m}_{i,j}$ are the multipliers $m_{i,j}$ computed in floating point arithmetic. Proceeding in the same way for the computation of the $\widetilde{m}_{i,j}$ we derive

$$|\widehat{\widetilde{m}}_{i,j} - \widetilde{m}_{i,j}| \leq \frac{4\epsilon}{1 - 4\epsilon} \widetilde{m}_{i,j}, \quad j = 1, \ldots, n;\ i = j+1, \ldots, n+1, \tag{5.2}$$

where $\widehat{\widetilde{m}}_{i,j}$ are the multipliers $\widetilde{m}_{i,j}$ computed in floating point arithmetic. Analogously

$$|\widehat{p}_{i,i} - p_{i,i}| \leq \frac{(8n+1)\epsilon}{1 - (8n+1)\epsilon} p_{i,i}, \quad i = 1, \ldots, n+1, \tag{5.3}$$

where $\widehat{p}_{i,i}$ are the diagonal pivots $p_{i,i}$ computed in floating point arithmetic. Therefore, looking at the inequalities given by (5.1), (5.2) and (5.3) and taking into account

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 26, pp. 357-380, June 2013

that $\widehat{m}_{i,j}$, $\widehat{\widehat{m}}_{i,j}$ and $\widehat{p}_{i,i}$ are the entries of $(\widehat{b}_{i,j})_{1\leq i\leq l+1;1\leq j\leq n+1}$, we conclude that

$$|\widehat{b}_{i,j}-b_{i,j}|\leq \frac{(8nl-4n^2+2n)\epsilon}{1-(8nl-4n^2+2n)\epsilon}b_{i,j},\quad i=1,\ldots,l+1;\ j=1,\ldots,n+1.\ \square$$

6. Perturbation theory. In Section 7 of [28] it is proved that if a totally nonnegative matrix A is represented as a product of nonnegative bidiagonal matrices, then small relative perturbations in the entries of the bidiagonal factors cause only small relative perturbations in the eigenvalues and singular values of A. More precisely (see Corollary 7.3 in [28]), $\mathcal{BD}(A)$ determines the eigenvalues and the singular values of A accurately, and the appropriate structured condition number of each eigenvalue and/or singular value with respect to perturbations in $\mathcal{BD}(A)$ is at most $2n^2$.

These results make clear the importance, in the context of our work, of the study of the sensitivity of the $\mathcal{BD}(A)$ of a Bernstein-Vandermonde matrix with respect to perturbations in the nodes x_i, and so in this section, we prove that small relative perturbations in the nodes of a Bernstein–Vandermonde matrix A produce only small relative perturbations in its bidiagonal factorization $\mathcal{BD}(A)$.

We begin by defining the quantities which lead to the finding of an appropriate condition number, in a similar way to the work carried out in [28, 15].

DEFINITION 6.1. Let A be a strictly totally positive Bernstein–Vandermonde matrix for the Bernstein basis $\mathcal{B}_n$ and the nodes $\{x_i\}_{1\leq i\leq l+1}$ and let $x_i'=x_i(1+\delta_i)$ be the perturbed nodes for $1\leq i\leq l+1$, where $|\delta_i|<<1$. We define:

$$rel\_gap_x\equiv\min_{i\neq j}\frac{|x_i-x_j|}{|x_i|+|x_j|},$$

$$rel\_gap_1\equiv\min_i\frac{|1-x_i|}{|x_i|},$$

$$\theta\equiv\max_i\frac{|x_i-x_i'|}{|x_i|}=\max_i|\delta_i|,$$

$$\alpha\equiv min\{rel\_gap_x,rel\_gap_1\},$$

$$\kappa_{BV}\equiv\frac{1}{\alpha},$$

where $\theta<<rel\_gap_x,rel\_gap_1$.

THEOREM 6.2. *Let A and A' be strictly totally positive Bernstein–Vandermonde matrices for the Bernstein basis $\mathcal{B}_n$ and the nodes $\{x_i\}_{1\leq i\leq l+1}$ and $x_i'=x_i(1+\delta_i)$*

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 26, pp. 357-380, June 2013

ELA

for $1 \leq i \leq l+1$, *where* $|\delta_i| \leq \theta << 1$. *Let* $\mathcal{BD}(A)$ *and* $\mathcal{BD}(A')$ *be the matrices representing the bidiagonal decomposition of* A *and the bidiagonal decomposition of* A', *respectively. Then*

$$|(\mathcal{BD}(A'))_{i,j} - (\mathcal{BD}(A))_{i,j}| \leq \frac{(2n+2)\kappa_{BV}\theta}{1-(2n+2)\kappa_{BV}\theta}(\mathcal{BD}(A))_{i,j}.$$

Proof. Taking into account that $|\delta_i| \leq \theta$, it can be easily shown that

$$1 - x_i' = (1-x_i)(1+\delta_i'), \quad |\delta_i'| \leq \frac{\theta}{rel\_gap_1} \tag{6.1}$$

and

$$x_i' - x_j' = (x_i - x_j)(1+\delta_{i,j}), \quad |\delta_{i,j}| \leq \frac{\theta}{rel\_gap_x}. \tag{6.2}$$

Accumulating the perturbations in the style of Higham (see Chapter 3 of [25], [15] and [28]) using the formula (3.1) for the $m_{i,j}$, and (6.1) and (6.2) we obtain

$$m_{i,j}' = m_{i,j}(1+\bar{\delta}), \quad |\bar{\delta}| \leq \frac{(2n+2)\kappa_{BV}\theta}{1-(2n+2)\kappa_{BV}\theta},$$

where $m_{i,j}'$ are the entries of $\mathcal{BD}(A')$ below the main diagonal. Proceeding in the same way by using the formula (3.2) we get

$$\widetilde{m}_{i,j}' = \widetilde{m}_{i,j}(1+\bar{\delta}), \quad |\bar{\delta}| \leq \frac{2\frac{\theta}{rel\_gap_1}}{1-2\frac{\theta}{rel\_gap_1}},$$

where $\widetilde{m}_{i,j}'$ are the entries of $\mathcal{BD}(A')$ above the main diagonal. Analogously, and using in this case the formula (3.3), we get

$$p_{i,i}' = p_{i,i}(1+\bar{\delta}), \quad |\bar{\delta}| \leq \frac{(n+i-1)\kappa_{BV}\theta}{1-(n+i-1)\kappa_{BV}\theta},$$

where $p_{i,i}'$ are the diagonal elements of $\mathcal{BD}(A')$. Finally, considering the last three inequalities we conclude that

$$|(\mathcal{BD}(A'))_{i,j} - (\mathcal{BD}(A))_{i,j}| \leq \frac{(2n+2)\kappa_{BV}\theta}{1-(2n+2)\kappa_{BV}\theta}(\mathcal{BD}(A))_{i,j}. \quad \square$$

So, we see that the quantity $(2n+2)\kappa_{BV}$ is an appropriate structured condition number of A with respect to the relative perturbations in the data x_i. These results are analogous to the results of [28, 15], in the sense that the relevant quantities for the

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 26, pp. 357-380, June 2013

ELA

determination of an structured condition number are the *relative* separations between the nodes (in our case also the relative distances to 1).

Combining this theorem with Corollary 7.3 in [28], which states that small componentwise relative perturbations of $\mathcal{BD}(A)$ cause only small relative perturbation in the eigenvalues λ_i and singular values σ_i of A, we obtain that

$$|\lambda_i' - \lambda_i| \leq O(n^3\kappa_{BV}\theta)\lambda_i \quad \text{and} \quad |\sigma_i' - \sigma_i| \leq O(n^3\kappa_{BV}\theta)\sigma_i,$$

where λ_i' and σ_i' are the eigenvalues and the singular values of A'. That is to say, small relative perturbation in the nodes of a Bernstein–Vandermonde matrix A produce only small relative perturbations in its eigenvalues and in its singular values.

7. Numerical experiments. In this last section, we include several numerical experiments illustrating the high relative accuracy of the algorithms we have presented for the problems of eigenvalue computation and singular value computation. Numerical experiments for the cases of linear system solving and of least squares problems have been included in [30, 32].

As it can be read in [28], the traditional algorithms for computing eigenvalues or singular values of ill-conditioned totally positive matrices only compute the largest eigenvalues and the largest singular values with guaranteed relative accuracy, and the tiny eigenvalues and singular values may be computed with no relative accuracy at all, even though they may be the only quantities of practical interest. This is the reason why, for computing with high relative accuracy all the eigenvalues and singular values of an ill-conditioned totally positive matrix, it is very important to develop algorithms that exploit the structure of the specific matrix we are considering. This is what our algorithms for computing the eigenvalues or singular values of a totally positive Bernstein–Vandermonde matrix does in its two stages: the computation of the $\mathcal{BD}(A)$, that is, the bidiagonal decomposition of A, and the computation of the eigenvalues or singular values of A starting from $\mathcal{BD}(A)$.

EXAMPLE 7.1. Let $\mathcal{B}_{20}$ be the Bernstein basis of the space of polynomials with degree less than or equal to 20 on $[0, 1]$ and let A be the square Bernstein–Vandermonde matrix of order 21 generated by the nodes:

$$\frac{1}{12} < \frac{1}{11} < \frac{1}{10} < \frac{1}{9} < \frac{1}{8} < \frac{1}{7} < \frac{1}{6} < \frac{1}{5} < \frac{1}{4} < \frac{1}{3} < \frac{1}{2}$$

$$< \frac{7}{12} < \frac{13}{22} < \frac{3}{5} < \frac{11}{18} < \frac{5}{8} < \frac{9}{14} < \frac{2}{3} < \frac{7}{10} < \frac{3}{4} < \frac{5}{6}.$$

The condition number of A is $\kappa_2(A) = 1.9e + 12$. In Table 7.1, we present the eigenvalues λ_i of A and the relative errors obtained when computing them by means of:

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 26, pp. 357-380, June 2013

ELA

1. The algorithm presented in Section 4.2 (column labeled by `MM`).
2. The command `eig` from MATLAB.

The relative error of each computed eigenvalue is obtained by using the eigenvalues calculated in *Maple* with 50-digit arithmetic.

λ_i	`MM`	`eig`
$1.0e+00$	0	$4.0e-15$
$8.4e-01$	$4.0e-16$	$1.3e-16$
$2.8e-01$	$9.9e-16$	$3.0e-15$
$2.1e-01$	0	$3.7e-15$
$1.2e-01$	$6.0e-16$	$3.6e-15$
$6.6e-02$	$8.5e-16$	$5.5e-15$
$3.8e-02$	$3.7e-16$	$1.2e-14$
$2.2e-02$	$4.8e-16$	$2.0e-14$
$9.4e-03$	0	$4.6e-14$
$4.6e-03$	$1.9e-16$	$9.3e-14$
$1.5e-03$	$2.9e-16$	$6.1e-14$
$5.9e-04$	$1.8e-16$	$2.5e-13$
$1.7e-04$	$3.2e-16$	$1.3e-12$
$4.3e-05$	$9.4e-16$	$3.6e-12$
$1.3e-05$	$1.1e-15$	$6.9e-12$
$1.7e-06$	$6.1e-16$	$5.1e-11$
$5.6e-07$	0	$1.9e-10$
$3.5e-08$	$2.8e-15$	$1.8e-09$
$1.1e-08$	$5.9e-16$	$5.9e-09$
$2.7e-10$	$2.1e-15$	$6.4e-08$
$1.3e-12$	$9.0e-16$	$1.0e-05$

TABLE 7.1
Example 7.1: Eigenvalues of a Bernstein–Vandermonde matrix of order 21.

The results appearing in Table 7.1 show that, while the command `eig` from MATLAB only computes the largest eigenvalues with high relative accuracy, our algorithm computes all the eigenvalues with high relative accuracy. In particular, the smallest eigenvalue, the one whose accurate computation is crucial, for example, for obtaining an accurate convergence rate for the progressive iteration approximation method, is computed by `eig` with a relative error of $1.0e-05$, while using our procedure it is computed with a relative error of $9.0e-16$.

As for the first stage of the algorithm, let us observe that in this example the greatest relative error obtained in the computation of the entries of $\mathcal{BD}(A)$ by using

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 26, pp. 357-380, June 2013

our algorithm is $1.7e{-}14$, while the greatest relative error obtained in the computation of the entries of $\mathcal{BD}(A)$ by means of `TNBD` [27], an algorithm that computes the Neville elimination of A without taking into account its structure is $1.0e-5$. The advantage of our algorithm `TNBDBV` compared with Neville elimination was expected since, as indicated in [27], algorithm `TNBD` does not guarantee high relative accuracy.

The example below shows how the situation of computing the singular values of a totally positive Bernstein–Vandermonde matrix is analogous to the situation of computing its eigenvalues.

EXAMPLE 7.2. Let $\mathcal{B}_{15}$ be the Bernstein basis of the space of polynomials with degree less than or equal to 15 on $[0,1]$ and let $A \in \mathbb{R}^{21\times 16}$ be the Bernstein–Vandermonde matrix generated by the nodes:

$$\frac{1}{22} < \frac{1}{20} < \frac{1}{18} < \frac{1}{16} < \frac{1}{14} < \frac{1}{12} < \frac{1}{10} < \frac{1}{8} < \frac{1}{6} < \frac{1}{4} < \frac{1}{2}$$
$$< \frac{23}{42} < \frac{21}{38} < \frac{19}{34} < \frac{17}{30} < \frac{15}{26} < \frac{13}{22} < \frac{11}{18} < \frac{9}{14} < \frac{7}{10} < \frac{5}{6}.$$

The condition number of A is $\kappa_2(A) = 5.3e+08$. In Table 7.2, we present the singular values σ_i of A and the relative errors obtained when computing them by means of

1. The algorithm presented in Section 4.4 (column labeled by `MM`).
2. The command `svd` from MATLAB.

The relative error of each computed singular value is obtained by using the singular values calculated in *Maple* with 50-digit arithmetic.

Let us observe that, while the relative error obtained in the computation of the smallest singular value by means of `svd` from MATLAB is $2.5e-10$, the relative error obtained by using our approach is $2.9e-15$.

EXAMPLE 7.3. Let $\mathcal{B}_{20}$ be the Bernstein basis of the space of polynomials with degree less than or equal to 20 on $[0,1]$ and let $A \in \mathbb{R}^{30\times 21}$ be the Bernstein–Vandermonde matrix generated by the nodes:

$$\frac{1}{31} < \frac{1}{30} < \frac{1}{29} < \frac{1}{28} < \frac{1}{27} < \frac{1}{26} < \frac{1}{25} < \frac{1}{24} < \frac{1}{23} < \frac{1}{22} < \frac{1}{21} < \frac{1}{20} < \frac{1}{19} < \frac{1}{18}$$
$$< \frac{1}{17} < \frac{1}{16} < \frac{1}{15} < \frac{1}{14} < \frac{1}{13} < \frac{1}{12} < \frac{1}{11} < \frac{1}{10} < \frac{1}{9} < \frac{1}{8} < \frac{1}{7} < \frac{1}{6} < \frac{1}{5} < \frac{1}{4} < \frac{1}{3} < \frac{1}{2}.$$

In this example, we are interested in estimating the condition number with respect to the 2-norm of the matrix A, which is

$$\kappa_2(A) = \frac{\sigma_{max}(A)}{\sigma_{min}(A)},$$

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 26, pp. 357-380, June 2013

σ_i	MM	svd
$1.6e+00$	$1.4e-16$	$1.4e-16$
$1.2e+00$	$9.4e-16$	$9.4e-16$
$5.8e-01$	$5.8e-16$	$3.8e-16$
$4.8e-01$	$5.8e-16$	$2.3e-16$
$2.5e-01$	$2.2e-16$	0
$1.8e-01$	$6.0e-16$	$4.5e-16$
$6.3e-02$	$2.2e-16$	0
$3.5e-02$	0	$1.0e-15$
$7.6e-03$	$1.1e-16$	$9.8e-15$
$3.4e-03$	$1.1e-15$	$2.0e-15$
$3.9e-04$	$1.4e-16$	$1.5e-14$
$1.3e-04$	$2.0e-16$	$8.4e-15$
$7.5e-06$	$9.0e-16$	$1.6e-12$
$1.6e-06$	$1.3e-16$	$4.3e-12$
$4.8e-08$	$1.8e-15$	$3.9e-10$
$3.0e-09$	$2.9e-15$	$2.5e-10$

TABLE 7.2

Example 7.2: Singular values of a Bernstein–Vandermonde matrix 21×16.

where $\sigma_{max}(A)$ is the largest singular value of A and $\sigma_{min}(A)$ is the smallest singular value of A.

In Table 7.3, we show the condition number $\kappa_2(A)$ obtained in *Maple* by dividing the greatest by the smallest singular value of A (computed by using 50-digit arithmetic) and the relative errors obtained when computing it by means of

1. The ratio of the largest by the smallest singular value of A obtained by means of the algorithm presented in Section 4.4 (column labeled by `MM`).
2. The command `cond(A,2)` from MATLAB.

$\kappa_2(A)$	MM	cond(A,2)
$2.0879e+27$	$3.8e-15$	1

TABLE 7.3

Example 7.3: Condition number of a Bernstein–Vandermonde matrix 30×21.

The relative error is obtained by using the value of $\kappa_2(A)$ presented in the first column of Table 7.3.

It must be observed that, taking into account that the Bernstein–Vandermonde matrices we are considering are totally positive, their rank is maximum and therefore

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 26, pp. 357-380, June 2013

ELA

all their singular values are nonzero. In this way, the condition number with respect to the 2-norm of A is obtained by simply dividing the first singular value of A by the last one, if we consider them in descending order.

In conclusion, the experiments show our algorithm is adequate for estimating the condition number of totally positive Bernstein–Vandermonde matrices with respect to the 2-norm.

In the following example, we test the results concerning the perturbation theory developed in Section 6.

EXAMPLE 7.4. Let A be the same Bernstein–Vandermonde matrix as in Example 7.1. In this case, the minimum relative gap between the $\{x_i\}_{1\leq i\leq 21}$ is $rel\_gap_x = 6.5e-3$, and the minimum relative gap between the $\{x_i\}_{1\leq i\leq 21}$ and 1 is $rel\_gap_1 = 2.0e-1$. We introduce random perturbation in the 10th digit of the $\{x_i\}_{1\leq i\leq 21}$ resulting in $\theta = 7.2e-9$.

We then compute the exact bidiagonal factorization B of A and the exact bidiagonal factorization B' of the perturbed matrix A' in *Maple*, obtaining that the maximum relative componentwise distance between the entries of B and the entries of B' is $1.7e-7$, while the bound we get when using our Theorem 6.2 is $4.7e-5$.

As for the eigenvalues, we compute the eigenvalues of A and of A' in in *Maple* by using 50-digit arithmetic. The maximum of the relative distance between the eigenvalues of A and of A' is $2.2e-8$, while the bound obtained for this example when applying Corollary 7.3 in [28] with the value $\delta = 1.7e-7$ is $1.5e-4$.

Finally, we include an example in which we show the sensitivity of the bidiagonal decomposition of a Bernstein–Vandermonde matrix to perturbations in the $\{x_i\}_{1\leq i\leq l+1}$ when x_{l+1} is very close to 1 (in this case, the value of κ_{BV} in Theorem 6.2 will be large since rel_gap_1 is small), and the sensitivity to perturbations in the $\{x_i\}_{1\leq i\leq l+1}$ when x_1 is very close to 0 (in this case, the value of θ in Theorem 6.2 will be large since $|x_1|$ is small).

EXAMPLE 7.5. First, let A be the Bernstein–Vandermonde matrix of order 21 generated by the first twenty nodes in Example 7.1 and $x_{21} = \frac{9999}{10000}$. The minimum relative gap between the $\{x_i\}_{1\leq i\leq 21}$ is $rel\_gap_x = 6.5e-3$, the same as in Example 7.4, while the minimum relative gap between the $\{x_i\}_{1\leq i\leq 21}$ and 1 is in this case $rel\_gap_1 = 1.0e-4$. We introduce the same perturbation as in Example 7.4 in the 10th digit of the $\{x_i\}_{1\leq i\leq 21}$ resulting in $\theta = 7.2e-9$, the same value as in the previous example.

When computing the exact bidiagonal factorization B of A and the exact bidiagonal factorization B' of the perturbed matrix A' in *Maple*, we get that the maximum

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 26, pp. 357-380, June 2013

ELA

relative componentwise distance between the entries of B and the entries of B' is $1.4e-4$, while the bound we get when using Theorem 6.2 is $3.0e-3$.

Now let A be the Bernstein–Vandermonde matrix of order 21 generated by $x_1 = \frac{1}{10000}$ and the last twenty nodes in Example 7.1. The minimum relative gap between the $\{x_i\}_{1\leq i\leq 21}$ is $rel\_gap_x = 6.5e-3$, the same as in Example 7.4, and the minimum relative gap between the $\{x_i\}_{1\leq i\leq 21}$ and 1 is $rel\_gap_1 = 2.0e-1$, also the same value as in Example 7.4. We introduce the same perturbation as in Example 7.4 in the 10th digit of the $\{x_i\}_{1\leq i\leq 21}$, resulting in this case $\theta = 6.0e-6$.

When we compute the exact bidiagonal factorization B of A and the exact bidiagonal factorization B' of the perturbed matrix A' in *Maple*, the bound we obtain when using Theorem 6.2 is $4.0e-2$, and the maximum relative componentwise distance between the entries of B and the entries of B' is $6.0e-6$.

Acknowledgment. The authors are grateful to one anonymous referee and to the associate editor for suggesting useful ideas to extend and improve the initial version of the paper.

REFERENCES

[1] M. Ainsworth, G. Andriamaro, and O. Davydov. Bernstein–Bézier finite elements of arbitrary order and optimal assembly procedures. *SIAM Journal on Scientific Computing*, 33:3087–3109, 2011.
[2] P. Alonso, M. Gasca, and J.M. Peña. Backward error analysis of Neville elimination. *Applied Numerical Mathematics*, 23:193–204, 1997.
[3] A. Barreras and J.M. Peña. Bidiagonal decompositions, minors and applications. *Electronic Journal of Linear Algebra*, 25:60–71, 2012.
[4] A. Björck. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, 1996.
[5] A. Björck and V. Pereyra. Solution of Vandermonde Systems of equations. *Mathematics of Computation*, 24:893–903, 1970.
[6] T. Boros, T. Kailath, and V. Olshevsky. A fast parallel Björck-Pereyra-type algorithm for solving Cauchy linear equations. *Linear Algebra and its Applications*, 302/303:265–293, 1999.
[7] J.P. Boyd and A.M. Sousa. An SVD analysis of equispaced polynomial interpolation. *Applied Numerical Mathematics*, 59:2534–2547, 2009.
[8] J.M. Carnicer and J.M. Peña. Shape preserving representations and optimality of the Bernstein basis. *Advances in Computational Mathematics*, 1:173–196, 1993.
[9] J. Delgado and J.M. Peña. Progressive iterative approximation and bases with the fastest convergence rates. *Computer Aided Geometric Design*, 24:10–18, 2007.
[10] J. Delgado and J.M. Peña. Optimal conditioning of Bernstein collocation matrices. *SIAM Journal on Matrix Analysis and Applications*, 31:990–996, 2009.
[11] J. Delgado and J.M. Peña. Optimality of Bernstein representations for computational purposes. *Reliable Computing*, 17:1–10, 2012.
[12] J. Demmel, I. Dumitriu, O. Holtz, and P. Koev. Accurate and efficient expression evaluation and linear algebra. *Acta Numerica*, 17:87–145, 2008.
[13] J. Demmel, M. Gu, S. Eisenstat, I. Slapničar, K. Veselić, and Z. Drmač. Computing the singular value decomposition with high relative accuracy. *Linear Algebra and its Applications*,

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 26, pp. 357-380, June 2013

ELA

299:21–80, 1999.
[14] J. Demmel and P. Koev. The accurate and efficient solution of a totally positive generalized Vandermonde linear system. *SIAM Journal on Matrix Analysis and Applications*, 27:142–152, 2005.
[15] J. Demmel and P. Koev. Accurate SVDs of polynomial Vandermonde matrices involving orthonormal polynomials. *Linear Algebra and its Applications*, 417:382–396, 2006.
[16] S.M. Fallat. Bidiagonal factorizations of totally nonnegative matrices. *American Mathematical Monthly*, 108:697–712, 2001.
[17] S.M. Fallat and C.R. Johnson. *Totally Nonnegative Matrices*. Princeton University Press, Princeton, NJ, 2011.
[18] G. Farin. *Curves and Surfaces for CAGD: A Practical Guide*, fifth edition. Academic Press, San Diego, CA, 2002.
[19] R.T. Farouki. On the stability of transformations between power and Bernstein polynomial forms. *Computer Aided Geometric Design*, 8:29–36, 1991.
[20] R.T. Farouki. The Bernstein polynomial basis: a centennial retrospective. *Computer Aided Geometric Design*, 29:379–419, 2012.
[21] M. Gasca and J.M. Peña. Total positivity and Neville elimination. *Linear Algebra and its Applications*, 165:25–44, 1992.
[22] M. Gasca and J.M. Peña. A matricial description of Neville elimination with applications to total positivity. *Linear Algebra and its Applications*, 202:33–45, 1994.
[23] M. Gasca and J.M. Peña. On Factorizations of Totally Positive Matrices. In: M. Gasca and C.A. Michelli (editors), *Total Positivity and Its Applications*, Kluwer Academic Publishers, Dordrecht, 109–130, 1996.
[24] G.H. Golub. Numerical methods for solving linear least squares problems. *Numerische Mathematik*, 7:206–216, 1965.
[25] N.J. Higham. *Accuracy and Stability of Numerical Algorithms*, second edition. SIAM, Philadelphia, 2002.
[26] R.C. Kirby and K.T. Thinh. Fast simplicial quadrature-based finite element operators using Bernstein polynomials. *Numerische Mathematik*, 121:261–279, 2012.
[27] P. Koev. `http://www.math.sjsu.edu/~koev/`.
[28] P. Koev. Accurate eigenvalues and SVDs of totally nonnegative matrices. *SIAM Journal on Matrix Analysis and Applications*, 27:1–23, 2005.
[29] P. Koev. Accurate computations with totally nonnegative matrices. *SIAM Journal on Matrix Analysis and Applications*, 29:731–751, 2007.
[30] A. Marco and J.J. Martínez. A fast and accurate algorithm for solving Bernstein–Vandermonde linear systems. *Linear Algebra and its Applications*, 422:616–628, 2007.
[31] A. Marco and J.J. Martínez. Bernstein–Bezoutian matrices and curve implicitization. *Theoretical Computer Science*, 377:65–72, 2007.
[32] A. Marco and J.J. Martínez. Polynomial least squares fitting in the Bernstein basis. *Linear Algebra and its Applications*, 433:1254–1264, 2010.
[33] J.J. Martínez and J.M. Peña. Factorizations of Cauchy-Vandermonde matrices. *Linear Algebra and its Applications*, 284:229–237, 1998.
[34] A. Pinkus. *Totally Positive Matrices*. Cambridge University Press, Cambridge, UK, 2010.