Linear Algebra and its Applications 651 (2022) 312–331

Contents lists available at ScienceDirect

Linear Algebra and its Applications

www.elsevier.com/locate/laa

Accurate computations with collocation matrices of the Lupaş-type (p,q)-analogue of the Bernstein basis ☆

Ana Marco *, José-Javier Martínez, Raquel Viaña

Universidad de Alcalá, Departamento de Física y Matemáticas, Alcalá de Henares, Madrid 28871, Spain

A R T I C L E I N F O

Article history:
Received 4 March 2022
Accepted 19 June 2022
Available online 23 June 2022
Submitted by S. Fallat

MSC:
65F05
65F15
65F20
65D05
65D17
15A23
15B48

Keywords:
Bidiagonal decomposition
(p,q)-Lupaş matrix
Totally positive matrix
Neville elimination
High relative accuracy

A B S T R A C T

A fast and accurate algorithm to compute the bidiagonal decomposition of collocation matrices of the Lupaş-type (p,q)-analogue of the Bernstein basis is presented. The error analysis of the algorithm and the perturbation theory for the bidiagonal decomposition are also included. Starting from this bidiagonal decomposition, the accurate and efficient solution of several linear algebra problems involving these matrices is addressed: linear system solving, eigenvalue and singular value computation, and computation of the inverse and the Moore-Penrose inverse. The numerical experiments carried out show the good behaviour of the algorithm.

☆ This research has been partially supported by Spanish Research Grant PGC2018-096321-B-I00 from the Spanish Ministerio de Ciencia, Innovación y Universidades. The authors are members of the Research Group ASYNACS (Ref. CT-CE2019/683) of Universidad de Alcalá.

* Corresponding author.
E-mail addresses: ana.marco@uah.es (A. Marco), jjavier.martinez@uah.es (J.-J. Martínez), raquel.viana@uah.es (R. Viaña).

https://doi.org/10.1016/j.laa.2022.06.023

1. Introduction

The Lupaş-type (p,q)-analogue of the Bernstein basis was introduced in [14] as a generalization of the Lupaş-type q-analogue of the Bernstein basis [18]. In that paper [14], the authors studied curves and surfaces associated to this new basis, and called them (p,q)-Bézier curves and surfaces. They involve (p,q)-integers as shape parameters, and in comparison to q-Bézier curves and surfaces based on Lupaş q-Bernstein functions they provide more flexibility in controlling the shapes of the associated curves and surfaces.

As it can be seen for instance in Section 5 of [8], in the field of Computer Aided Geometric Design the shape-preserving property of curves is closely related to the total positivity of the corresponding collocation matrices. Such matrices are defined as follows.

Definition 1.1. The *collocation matrix* of a system of functions $(u_0(x), u_1(x), \dots, u_n(x))$ on an interval I at $x_0 < x_1 < \cdots < x_m$ in I is given by

$$M\begin{pmatrix} u_0, \dots, u_n \\ x_0, \dots, x_m \end{pmatrix} := (u_j(x_i))_{i=0,\dots,m;j=0,\dots,n}.$$

We recall that, classically, a matrix is called *totally positive* if all its minors are nonnegative and *strictly totally positive* if all its minors are positive [11,26]. Although we will follow such terminology, we must note that totally positive matrices and strictly totally positive matrices are also called *totally nonnegative* and *totally positive* matrices, respectively [7].

The properties of the (p,q)-Lupaş system of functions presented in [14], in particular the end-point interpolation property, the convex-hull property and the variation diminishing property, suggest that the corresponding collocation matrices are strictly totally positive when $p, q > 0$ and the nodes are in the interval $(0, 1)$ sorted in increasing ordering [2]. Following a different approach, we will give a proof of this fact in Theorem 2. Let us notice that the technique we have chosen provides not only an elegant demonstration of this theorem, but also a clearer proof of Theorem 3 and Theorem 4.

Although numerical linear algebra for structured strictly totally positive matrices can be done by using standard numerical linear algebra algorithms, the ill conditioning of such matrices is the reason why these algorithms do not give accurate results (see, for example, [16]). Because of this, the design of efficient numerical linear algebra algorithms that, taking into account the structure of these totally positive matrices, provide accurate answers is a current topic of research in this area.

In his work in [16] and [17] P. Koev showed that, having the bidiagonal decomposition of a nonsingular totally positive matrix computed with high relative accuracy, virtually all linear algebra problems can be solved accurately for that matrix. That is, the bidiagonal decomposition is the adequate representation for doing accurate numerical linear algebra with nonsingular totally positive matrices. Given a nonsingular totally positive matrix A, this decomposition can be stored in a matrix, denoted by P. Koev $\mathcal{BD}(A)$, of the same size as A.

Examples of structured totally positive matrices whose bidiagonal decomposition has been computed in a fast and accurate way are Bernstein-Vandermonde matrices [20], Cauchy-Vandermonde matrices [23], Lupaş matrices [4], collocation and Wronskian matrices of Jacobi polynomials [19] or generalized Pascal matrices [3].

Our aim in this work is obtaining a fast and accurate algorithm for computing the bidiagonal decomposition $\mathcal{BD}(A)$ of a strictly totally positive collocation matrix of a Lupaş-type (p,q)-analogue of the Bernstein basis, and apply it to solve some fundamental linear algebra problems involving A. In our approach the main theoretical tool for obtaining $\mathcal{BD}(A)$ is Neville elimination, but it must be pointed out that we do not apply the Neville elimination algorithm since it does not provide accurate results. This fact is noted in [15], where the author indicates that the function `TNBD` for computing the $\mathcal{BD}(A)$ by means of Neville elimination does not guarantee high relative accuracy. Let us remind that an algorithm computes to high relative accuracy if it satisfies the so called NIC (no inaccurate cancellation) condition [5]:

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

Finally, let us recall some basic concepts on (p,q)-calculus needed in our work. More details on the subject can be found for instance in [13].

For any $p > 0$ and $q > 0$ the (p,q) *integer* $[n]_{p,q}$ is defined by

$$[n]_{p,q} = p^{n-1} + p^{n-2}q + p^{n-3}q^2 + \cdots + pq^{n-2} + q^{n-1}, \quad n = 0, 1, 2, \ldots$$

The (p,q) *factorial* $[n]_{p,q}!$ is defined as

$$[n]_{p,q}! = [n]_{p,q}[n-1]_{p,q} \cdots [2]_{p,q}[1]_{p,q}, \quad n = 0, 1, 2, \ldots,$$

and the (p,q) *binomial coefficients* $\begin{bmatrix} n \\ k \end{bmatrix}_{p,q}$ as

$$\begin{bmatrix} n \\ k \end{bmatrix}_{p,q} = \frac{[n]_{p,q}!}{[k]_{p,q}![n-k]_{p,q}!}, \quad n = 0, 1, 2, \ldots; \ k = 0, 1, \ldots, n.$$

The rest of the paper is organized as follows. In Section 2, basic results on Neville elimination and total positivity are given. Section 3 is devoted to obtain the bidiagonal decomposition of collocation matrices of Lupaş (p,q)-analogue of the Bernstein functions, the corresponding algorithm being presented in Section 4. The error analysis of such algorithm is done in Section 5, while the perturbation theory of the bidiagonal decomposition is studied in Section 6. Finally, in Section 7 numerical experiments showing the good results obtained when applying our approach to solve several linear algebra problems involving this type of matrices are included.

2. Neville elimination and total positivity

In this section we will briefly recall 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 [9] and [10]. 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 a square real matrix of order n. For $k \leq n$, $m \leq n$, and for any $\alpha \in Q_{k,n}$ 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* [9–11], a procedure that makes zeros in a matrix by adding to a given row an appropriate multiple of the previous one.

Let $A = (a_{i,j})_{1 \leq i,j \leq n}$ be a square matrix of order n. The Neville elimination of A consists of $n-1$ steps resulting in a sequence of matrices $A_1 := A \to A_2 \to \ldots \to A_n$, where $A_t = (a_{i,j}^{(t)})_{1 \leq i,j \leq n}$ 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:

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

In this process the element

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

is called (i,j) *pivot* of the Neville elimination of A. The process would break down if any of the pivots $p_{i,j}$ $(j \leq i < n)$ is zero. In that case we can move the corresponding rows to the bottom and proceed with the new matrix, as described in [9]. 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*. If all the pivots $p_{i,j}$ are nonzero, then $p_{i,1} = a_{i,1}\ \forall i$ and, by Lemma 2.6 of [9]

$$p_{i,j} = \frac{\det A[i-j+1, \ldots, i|1, \ldots, j]}{\det A[i-j+1, \ldots, i-1|1, \ldots, j-1]} \qquad 1 < j \leq i \leq n. \tag{2}$$

The element

$$m_{i,j} = \frac{p_{i,j}}{p_{i-1,j}} \qquad 1 \leq j \leq n-1, \quad j < i \leq n, \tag{3}$$

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 $j \geq i$ (see p. 116 of [11]).

The Neville elimination characterizes the strictly totally positive matrices as follows [9]:

Theorem 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.*

3. Bidiagonal decomposition

Lupaş (p,q)-*analogue of the Bernstein functions (rational)* of degree n for $p > 0, q > 0$ were introduced in [14], where they were defined as

$$b^{r,n}_{p,q}(t) = \frac{\begin{bmatrix} n \\ r \end{bmatrix}_{p,q} p^{\frac{(n-r)(n-r-1)}{2}} q^{\frac{r(r-1)}{2}} t^r (1-t)^{n-r}}{\prod_{k=1}^{n} (p^{k-1}(1-t) + q^{k-1}t)}, \quad t \in [0, 1], \quad r = 0, 1, \ldots, n.$$

Lupaş (p,q)-Bézier curves of degree n are defined also in [14] as

$$\mathbf{P}(t; p, q) = \sum_{r=0}^{n} \mathbf{P}_r \ b^{r,n}_{p,q}(t),$$

where $\mathbf{P}_r \in \mathbb{R}^3$ $(r = 0, \ldots, n)$ are the control points.

Let us observe that when $p = 1$, Lupaş (p,q)-analogue of the Bernstein functions coincide with the Lupaş q-analogue of the Bernstein functions [18], and when $p = q = 1$ with the classical Bernstein polynomials [8].

The matrix

$$\begin{pmatrix}
\frac{\begin{bmatrix} n \\ 0 \end{bmatrix}_{p,q} p^{\frac{n(n-1)}{2}} (1-t_1)^n}{\prod_{k=1}^{n}(p^{k-1}(1-t_1)+q^{k-1}t_1)} & \frac{\begin{bmatrix} n \\ 1 \end{bmatrix}_{p,q} p^{\frac{(n-1)(n-2)}{2}} t_1(1-t_1)^{n-1}}{\prod_{k=1}^{n}(p^{k-1}(1-t_1)+q^{k-1}t_1)} & \cdots & \frac{\begin{bmatrix} n \\ n \end{bmatrix}_{p,q} q^{\frac{n(n-1)}{2}} t_1^n}{\prod_{k=1}^{n}(p^{k-1}(1-t_1)+q^{k-1}t_1)} \\
\frac{\begin{bmatrix} n \\ 0 \end{bmatrix}_{p,q} p^{\frac{n(n-1)}{2}} (1-t_2)^n}{\prod_{k=1}^{n}(p^{k-1}(1-t_2)+q^{k-1}t_2)} & \frac{\begin{bmatrix} n \\ 1 \end{bmatrix}_{p,q} p^{\frac{(n-1)(n-2)}{2}} t_2(1-t_2)^{n-1}}{\prod_{k=1}^{n}(p^{k-1}(1-t_2)+q^{k-1}t_2)} & \cdots & \frac{\begin{bmatrix} n \\ n \end{bmatrix}_{p,q} q^{\frac{n(n-1)}{2}} t_2^n}{\prod_{k=1}^{n}(p^{k-1}(1-t_2)+q^{k-1}t_2)} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\begin{bmatrix} n \\ 0 \end{bmatrix}_{p,q} p^{\frac{n(n-1)}{2}} (1-t_{l+1})^n}{\prod_{k=1}^{n}(p^{k-1}(1-t_{l+1})+q^{k-1}t_{l+1})} & \frac{\begin{bmatrix} n \\ 1 \end{bmatrix}_{p,q} p^{\frac{(n-1)(n-2)}{2}} t_{l+1}(1-t_{l+1})^{n-1}}{\prod_{k=1}^{n}(p^{k-1}(1-t_{l+1})+q^{k-1}t_{l+1})} & \cdots & \frac{\begin{bmatrix} n \\ n \end{bmatrix}_{p,q} q^{\frac{n(n-1)}{2}} t_{l+1}^n}{\prod_{k=1}^{n}(p^{k-1}(1-t_{l+1})+q^{k-1}t_{l+1})}
\end{pmatrix}$$

is the collocation matrix of the Lupaş (p,q)-analogue of the Bernstein functions $(b^{0,n}_{p,q}, b^{1,n}_{p,q}, \ldots, b^{n,n}_{p,q})$ at $\{t_i\}_{1 \leq i \leq l+1}$. From now on, we assume $l \geq n$ and $0 < t_1 < t_2 <$

$\cdots < t_{l+1} < 1$, and we call the matrix above (p,q)-*Lupaş matrix* for the (p,q)-analogue of the Bernstein functions $(b_{p,q}^{r,n})_{0\le r\le n}$ and the nodes $\{t_i\}_{1\le i\le l+1}$.

Let us point out that, when $l = n$, collocation matrices arise in Lagrange interpolation problems as the coefficient matrices of the corresponding linear systems. In our case the collocation matrix A is the coefficient matrix of the linear system associated with the following interpolation problem in the (p,q)-analogue of the Bernstein basis $(b_{p,q}^{r,n})_{0\le r\le n}$: given the interpolation nodes $\{t_i : i = 1, \cdots, n+1\}$, where $0 < t_1 < t_2 < \cdots < t_{n+1} < 1$, and the interpolation data $\{b_i : i = 1, \cdots, n+1\}$ find the polynomial

$$p(t) = \sum_{r=0}^{n} a_r b_{p,q}^{r,n}(t)$$

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

In the next theorem, the strict total positivity of (p,q)-Lupaş matrices is established.

Theorem 2. *The (p,q)-Lupaş matrix for the (p,q)-analogue of the Bernstein functions* $(b_{p,q}^{r,n})_{0\le r\le n}$ *and the nodes* $\{t_i\}_{1\le i\le l+1}$, *where* $0 < t_1 < t_2 < \cdots < t_{l+1} < 1$, *is strictly totally positive.*

Proof. Let A be the *(p,q)-Lupaş matrix* for the (p,q)-analogue of the Bernstein functions $(b_{p,q}^{r,n})_{0\le r\le n}$ and the nodes $\{t_i\}_{1\le i\le l+1}$, where $0 < t_1 < t_2 < \cdots < t_{l+1} < 1$. It can be seen that A can be factorized as $A = WCZ$, where W is the $(l+1)\times(l+1)$ diagonal matrix whose i-th diagonal entry is

$$w_{i,i} = \Big(\prod_{k=1}^{n}(p^{k-1}(1-t_i) + q^{k-1}t_i\Big)^{-1},$$

$$C = \begin{pmatrix} (1-t_1)^n & t_1(1-t_1)^{n-1} & \cdots & t_1^n \\ (1-t_2)^n & t_2(1-t_2)^{n-1} & \cdots & t_2^n \\ \vdots & \vdots & \ddots & \vdots \\ (1-t_{l+1})^n & t_{l+1}(1-t_{l+1})^{n-1} & \cdots & t_{l+1}^n \end{pmatrix}$$

and Z is the $(n+1)\times(n+1)$ diagonal matrix whose i-th diagonal entry is

$$z_{i,i} = \begin{bmatrix} n \\ i-1 \end{bmatrix}_{p,q} p^{\frac{(n-i+1)(n-i)}{2}} q^{\frac{(i-1)(i-2)}{2}}.$$

Taking into account that $0 < t_1 < \ldots < t_{l+1} < 1$, $p > 0$ and $q > 0$, the diagonal matrices W and Z are nonsingular totally positive matrices.

Since each minor of C has the same strict sign as the corresponding minor of the Bernstein-Vandermonde matrix at the nodes $\{t_i\}_{1\le i\le l+1}$ $(0 < t_1 < t_2 < \cdots < t_{l+1} < 1)$, and this Bernstein-Vandermonde matrix is strictly totally positive (see [20]), the matrix C is strictly totally positive.

Therefore, since the product of a strictly totally positive matrix by a nonsingular totally positive matrix is a strictly totally positive matrix [1], $A = WCZ$ is strictly totally positive and the proof is complete. □

In the following theorem a bidiagonal factorization of a (p,q)-Lupaş matrix is stated. It must be observed that this bidiagonal decomposition is the one needed to compute $\mathcal{BD}(A)$, it is not derived directly from the factorization $A = WCZ$ and it is unique.

Theorem 3. *Let $A = (a_{i,j})_{1\le i\le l+1;1\le j\le n+1}$ be the (p,q)-Lupaş matrix for the (p,q)-analogue of the Bernstein functions $(b^{r,n}_{p,q})_{0\le r\le n}$ and the nodes $\{t_i\}_{1\le i\le l+1}$, where $0 < t_1 < t_2 < \cdots < t_{l+1} < 1$. Then A admits a factorization in the form*

$$A = F_l F_{l-1} \cdots F_1 D G_1 \cdots G_{n-1} G_n \tag{4}$$

where F_i $(1 \le i \le l)$ are bidiagonal matrices of order $l+1$ of the form

$$F_i = \begin{pmatrix} 1 & & & & & & & \\ 0 & 1 & & & & & & \\ & \ddots & \ddots & & & & & \\ & & 0 & 1 & & & & \\ & & & m_{i+1,1} & 1 & & & \\ & & & & m_{i+2,2} & 1 & & \\ & & & & & \ddots & \ddots & \\ & & & & & & m_{l+1,l+1-i} & 1 \end{pmatrix}, \tag{5}$$

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

$$G_i^T = \begin{pmatrix} 1 & & & & & & & \\ 0 & 1 & & & & & & \\ & \ddots & \ddots & & & & & \\ & & 0 & 1 & & & & \\ & & & \widetilde{m}_{i+1,1} & 1 & & & \\ & & & & \widetilde{m}_{i+2,2} & 1 & & \\ & & & & & \ddots & \ddots & \\ & & & & & & \widetilde{m}_{n+1,n+1-i} & 1 \end{pmatrix}, \tag{6}$$

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

$$D = \mathrm{diag}\{p_{1,1}, p_{2,2}, \ldots, p_{n+1,n+1}\}. \tag{7}$$

The quantities $m_{i,j}$ are the multipliers of the Neville elimination of the (p,q)-Lupaş matrix A, and have the expression

$$m_{i,j} = \frac{(1-t_i)^{n-j+1}(1-t_{i-j})\prod_{k=1}^{j-1}(t_i - t_{i-k})\prod_{k=2}^{n}(p^{k-1}(1-t_{i-1}) + q^{k-1}t_{i-1})}{(1-t_{i-1})^{n-j+2}\prod_{k=2}^{j}(t_{i-1} - t_{i-k})\prod_{k=2}^{n}(p^{k-1}(1-t_i) + q^{k-1}t_i)}, \quad (8)$$

where $j = 1, \ldots, n+1$, $i = j+1, \ldots, l+1$. The quantities $\widetilde{m}_{i,j}$ are the multipliers of the Neville elimination of A^T, and their expression is

$$\widetilde{m}_{i,j} = \frac{[n-i+2]_{p,q}\; t_j q^{i-2}}{[i-1]_{p,q}(1-t_j)p^{n-i+1}}, \quad (9)$$

where $j = 1, \ldots, n$, $i = j+1, \ldots, n+1$. Finally, the i-th diagonal entry of D is the diagonal pivot of the Neville elimination of A and its expression is

$$p_{i,i} = \frac{\begin{bmatrix} n \\ i-1 \end{bmatrix}_{p,q} p^{\frac{(n-i+1)(n-i)}{2}} q^{\frac{(i-1)(i-2)}{2}} (1-t_i)^{n-i+1}\prod_{k<i}(t_i - t_k)}{\prod_{k=2}^{n}(p^{k-1}(1-t_i) + q^{k-1}t_i)\prod_{k=1}^{i-1}(1-t_k)}, \quad (10)$$

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

Proof. As we have seen in Theorem 2, the matrix A is strictly totally positive and therefore, by Theorem 1, the complete Neville elimination of A can be performed without row and column exchanges providing the factorization of A in (4), (5), (6) and (7) (see [9,10]).

The multipliers $m_{i,j}$ and the diagonal pivots $p_{i,i}$ of the Neville elimination of A will be computed by means of formulas (2) and (3), so $det(A[i-j+1, \ldots, i|1, \ldots, j])$ has to be computed first. Factorizing $A[i-j+1, \ldots, i|1, \ldots, j]$ analogously as we did for A in the proof of Theorem 2, and taking into account that

$$\det \begin{pmatrix} (1-t_1)^k & t_1(1-t_1)^{k-1} & \cdots & t_1^k \\ (1-t_2)^k & t_2(1-t_2)^{k-1} & \cdots & t_2^k \\ \vdots & \vdots & \ddots & \vdots \\ (1-t_{k+1})^k & t_{k+1}(1-t_{k+1})^{k-1} & \cdots & t_{k+1}^k \end{pmatrix} = \prod_{1 \le u < s \le k+1}(t_s - t_u) \quad (11)$$

(see Corollary 3.2 in [20]), we obtain that

$$det(A[i-j+1, \ldots, i|1, \ldots, j]) = \frac{\prod_{k=0}^{j-1}\begin{bmatrix} n \\ k \end{bmatrix}_{p,q} p^{\frac{(n-k)(n-k-1)}{2}} q^{\frac{k(k-1)}{2}} \prod_{k=0}^{j-1}(1-t_{i-k})^{n-j+1}\prod_{i-j+1\le u<s\le i}(t_s - t_u)}{\prod_{s=0}^{j-1}\left(\prod_{k=1}^{n}(p^{k-1}(1-t_{i-s}) + q^{k-1}t_{i-s})\right)}. \quad (12)$$

Using this formula and (2) we obtain the expression (10) for the diagonal pivots $p_{i,i}$, and using it, as well as (2) and (3), the expression (8) for the multipliers $m_{i,j}$ of the Neville elimination of A is established.

Now, in order to compute the multipliers of the Neville elimination of A^T we have to compute $det(A^T[i-j+1, \ldots, i|1, \ldots, j])$. Proceeding similarly as we have done in the case of $det(A[i-j+1, \ldots, i|1, \ldots, j])$ we get

$$det(A^T[i-j+1,\ldots,i|1,\ldots,j]) =$$
$$\frac{\prod_{k=i-j}^{i-1}\begin{bmatrix}n\\k\end{bmatrix}_{p,q} p^{\frac{(n-k)(n-k-1)}{2}} q^{\frac{k(k-1)}{2}} \prod_{k=1}^{j} t_k^{i-j} \prod_{k=1}^{j}(1-t_k)^{n-i+1} \prod_{1\le u<s\le j}(t_s-t_u)}{\prod_{s=1}^{j}\left(\prod_{k=1}^{n}(p^{k-1}(1-t_s)+q^{k-1}t_s)\right)}. \tag{13}$$

Considering this last formula, (2) and (3), the expression (9) for the multipliers $\widetilde{m}_{i,j}$ of the Neville elimination of A^T is obtained. $\square$

Next, a bidiagonal factorization of the inverse of a (p,q)-Lupaş matrix is given.

Theorem 4. *Let $A = (a_{i,j})_{1\le i,j\le n+1}$ be the (p,q)-Lupaş matrix for the (p,q)-analogue of the Bernstein functions $(b_{p,q}^{r,n})_{0\le r\le n}$ and the nodes $\{t_i\}_{1\le i\le n+1}$, where $0 < t_1 < t_2 < \cdots < t_{n+1} < 1$. Then A^{-1} admits a factorization in the form*

$$A^{-1} = G_1G_2\cdots G_nD^{-1}F_nF_{n-1}\cdots F_1,$$

where F_i, G_i $(i = 1,\ldots,n)$ are $(n+1)\times(n+1)$ bidiagonal matrices of the form

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

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

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

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

The expressions of the multipliers $m_{i,j}$ $(1\le j<i\le n+1)$ of the Neville elimination of A, the multipliers $\widetilde{m}_{i,j}$ $(1\le j<i\le n+1)$ 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 (8) and (9), and (10), respectively.

Proof. Since, by Theorem 2, the matrix A is strictly totally positive, by Theorem 1 the complete Neville elimination of A can be performed without row and column exchanges, providing the bidiagonal factorization of A^{-1} in the statement of this theorem (see [11]). The rest of the proof is the same as in Theorem 3. □

It must be observed that although the Neville elimination of A and A^T gives us the bidiagonal factorization of A and A^{-1}, the matrices F_i $(i = 1, \dots, n)$ and the matrices G_j $(j = 1, \dots, n)$ 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 and Theorem 4). A detailed explanation of this fact can be found in [11].

4. Algorithm for the bidiagonal decomposition

In this section we present a fast and accurate algorithm to compute the bidiagonal factorization of a strictly totally positive (p,q)-Lupaş matrix A for the (p,q)-analogue of the Bernstein functions $(b^{r,n}_{p,q})_{0\le r\le n}$ and the nodes $\{t_i\}_{1\le i\le l+1}$ $(0 < t_1 < t_2 < \cdots < t_{l+1} < 1)$ introduced in Section 3. Previously we include an auxiliary algorithm (Algorithm 1) that computes recursively the (p,q)-integers $[k]_{p,q}$ for $k = 0, 1, \dots, n$ by taking into account the relationship:

$$[i]_{p,q} = p[i-1]_{p,q} + q^{i-1}, \quad i = 1, \dots, n.$$

Algorithm 1

Input: p, q and n, where $n = 0, 1, 2, \dots$
Output: $[k]_{p,q}$, $k = 0, 1, \dots, n$

```
function pq=pqNumbers(p, q, n)
pq=zeros(1,n);
pq(1)=1;
qpot=1;
for j=2:n
   qpot=qpot*q;
   pq(j)=p*(pq(j-1))+qpot;
end;
```

Just by looking at the code of Algorithm 1, we can assert that it preserves high relative accuracy because it satisfies NIC condition.

Algorithm 2 is the main algorithm of this section. Starting from the nodes $\{t_i\}_{1\le i\le l+1}$, $p > 0$, $q > 0$ and the number of columns of matrix A ($n1$ in Algorithm 2) it computes the matrix $\mathcal{BD}(A)$ [16] containing the bidiagonal decomposition of A given by Theorem 3. This matrix, which in the pseudocode of our algorithm is denoted by M, satisfies

$$\begin{aligned}
M_{i,i} &= p_{i,i} && i = 1, \dots, n+1,\\
M_{i,j} &= m_{i,j} && j = 1, \dots, n+1;\ i = j+1, \dots, l+1,\\
M_{i,j} &= \widetilde{m}_{j,i} && i = 1, \dots, n;\ j = i+1, \dots, n+1,
\end{aligned}$$

where $p_{i,i}$ are the diagonal pivots of the Neville elimination of A, $m_{i,j}$ are the multipliers of the Neville elimination of A and $\widetilde{m}_{i,j}$ are the multipliers of the Neville elimination of A^T.

Taking into account Theorem 3 and Theorem 4, we note that the matrix $\mathcal{BD}(A)$ stores factorizations of both A and A^{-1}.

Algorithm 2

Input: The vector c containing the nodes $\{t_i\}_{1\le i\le l+1}$, $p > 0$, $q > 0$ and $n1$ the number of columns of matrix A.

Output: $M = \mathcal{BD}(A)$

function M=`TNBDpqLupas`(c, p, q, $n1$)

n=n1-1
M=zeros(l+1,n+1);

- $[k]_{p,q}$ computation:

pq=`pqNumbers`(p,q,n);

- $\prod_{k=1}^{n}(p^{k-1}(1-t_i)+q^{k-1}t_i)$ computation:

α=zeros(1,l+1);
for i=1:l+1
  $\alpha_i = 1$;
  for k=2:n
  $\alpha_i = \alpha_i(p^{k-1}(1-t_i)+q^{k-1}t_i)$;
  end
end

- $m_{i,j}$ computation:

for i=2:l+1
  $auxM = \frac{(1-t_i)^n}{(1-t_{i-1})^{n+1}}\frac{\alpha_{i-1}}{\alpha_i}$;
  $M_{i,1} = (1-t_{i-1})auxM$;
  k=min(i-2,n);
  for j=1:k
    $auxM = auxM\frac{(1-t_{i-1})(t_i-t_{i-j})}{(1-t_i)(t_{i-1}-t_{i-j-1})}$;
    $M_{i,j+1} = (1-t_{i-j-1})auxM$;
  end
end

- $\widetilde{m}_{i,j}$ computation:

for j=1:n
  $cj = \frac{t_j}{1-t_j}$;
  for i=j+1:n+1
  $ai = \frac{pq_{n-i+2}q^{i-2}}{pq_{i-1}p^{n-i+1}}$;
  $M_{j,i} = cj\cdot ai$;
  end
end

Observing Algorithm 2, we notice that it preserves high relative accuracy because it satisfies NIC condition. It has a computational cost of $O(n^2)$ arithmetic operations and it does not construct the (p,q)-Lupaş matrix A since it only works with the nodes $\{t_i\}_{1\le i\le l+1}$, what implies an additional saving in storage space.

- $p_{i,i}$ computation:

$r = p^{\frac{n(n-1)}{2}}$;
$M_{1,1} = r\frac{(1-t_1)^n}{\alpha_1}$;
for i=1:n
$r = \frac{pq_{n-i+1}}{pq_i}\frac{p^{i-n}q^{i-1}}{1-t_i}r$;
aux=1;
for k=1:i
$aux = aux(t_{i+1} - t_k)$;
end
$M_{i+1,i+1} = r\frac{(1-t_{i+1})^{n-i}}{\alpha_{i+1}}aux$;
end
M;

5. Error analysis

The error analysis of the algorithm `TNBDpqLupas` (Algorithm 2) to compute the bidiagonal decomposition of a totally positive (p,q)-Lupaş matrix presented in the previous section is carried out in this section.

In our error analysis we use the standard model of floating point arithmetic (see section 2.2 of [12]):

Let x, y be floating point numbers, $fl(\cdot)$ be the evaluation of an expression in floating point arithmetic, and ϵ be the *machine precision*. Then we have

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

Our error analysis of algorithm `TNBDpqLupas` is summarized in the following theorem:

Theorem 5. *Let A be a (p,q)-Lupaş matrix for the (p,q)-analogue of the Bernstein functions $(b^{r,n}_{p,q})_{0\le r\le n}$ and the nodes $\{t_i\}_{1\le i\le l+1}$, where $0 < t_1 < t_2 < \cdots < t_{l+1} < 1$. Let $\mathcal{BD}(A) = (b_{i,j})_{1\le i\le l+1; 1\le j\le n+1}$ be the matrix representing the exact bidiagonal decomposition of A and $(\widehat{b}_{i,j})_{1\le i\le l+1;1\le j\le n+1}$ be the matrix representing the computed bidiagonal decomposition of A by means of the algorithm `TNBDpqLupas` in floating point arithmetic with machine precision ϵ. Then*

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

Proof. Accumulating the relative errors in the style of Higham (see Chapter 3 of [12]) in the computation of the multipliers $m_{i,j}$ by means of the algorithm `TNBDpqLupas` included in Section 4 we obtain

$$|\widehat{m}_{i,j} - m_{i,j}| \le \frac{(n^2+17n-3)\epsilon}{1-(n^2+17n-3)\epsilon} m_{i,j}, \quad j = 1, \dots, n+1; \; i = j+1, \dots, l+1, \tag{14}$$

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

$$|\widehat{\widetilde{m}}_{i,j} - \widetilde{m}_{i,j}| \leq \frac{(3n-1)\epsilon}{1-(3n-1)\epsilon}\widetilde{m}_{i,j}, \quad j = 1,\dots,n; \; i = j+1,\dots,n+1, \tag{15}$$

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

$$|\widehat{p}_{i,i} - p_{i,i}| \leq \frac{(4n^2+4n-4)\epsilon}{1-(4n^2+4n-4)\epsilon}p_{i,i}, \quad i = 1,\dots,n+1, \tag{16}$$

where $\widehat{p}_{i,i}$ are the diagonal pivots $p_{i,i}$ computed in floating point arithmetic.

Therefore, looking at the inequalities given by (14), (15) and (16) and taking into account that $\widehat{m}_{i,j}$, $\widehat{\widetilde{m}}_{i,j}$ 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

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

This result shows that `TNBDpqLupas` computes the bidiagonal decomposition of a (p,q)-Lupaş matrix very accurately in floating point arithmetic.

6. Perturbation theory

Given a totally positive matrix A represented as a product of nonnegative bidiagonal matrices, in [16] it was proved that small relative perturbations in the entries of the bidiagonal factors produce only small relative perturbations in both the eigenvalues and the singular values of matrix A. Furthermore, as stated in Corollary 7.3 of [16], the bidiagonal decomposition $\mathcal{BD}(A)$ accurately determines both the singular values and the eigenvalues of A, and the appropriate structured condition number of each eigenvalue and/or singular value with respect to perturbations in $\mathcal{BD}(A)$ is at most $2n^2$. The analysis of the perturbation theory for $\mathcal{BD}(A)$ of a totally positive matrix A is thus of great relevance.

In this section, the sensitivity of the bidiagonal factorization $\mathcal{BD}(A)$ with respect to perturbations in the nodes t_i will be studied, where A is the (p,q)-Lupaş matrix for the (p,q)-analogue of the Bernstein functions $(b^{r,n}_{p,q})_{0\leq r\leq n}$ and the nodes $\{t_i\}_{1\leq i\leq l+1}$ $(0 < t_1 < t_2 < \dots < t_{l+1} < 1)$. In particular, we show that small relative perturbations in the nodes of A produce only small relative perturbations in its $\mathcal{BD}(A)$.

Following the approaches in [16,21–23,25] we define next some quantities required to state an appropriate condition number.

Definition 1. Let A be a strictly totally positive (p,q)-Lupaş matrix for the (p,q)-analogue of the Bernstein functions $(b^{r,n}_{p,q})_{0\leq r\leq n}$ and the nodes $\{t_i\}_{1\leq i\leq l+1}$ $(0 < t_1 < t_2 < \dots < t_{l+1} < 1)$ and let $t'_i = t_i(1+\delta_i)$ be the perturbed nodes for $1 \leq i \leq l+1$, where $|\delta_i| << 1$. We define:

$$rel\_gap_t \equiv \min_{i\neq j} \frac{|t_i - t_j|}{|t_i| + |t_j|},$$

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

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

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

$$\kappa_{p,q} \equiv \frac{1}{\alpha},$$

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

The following theorem is the major result of this section.

Theorem 6. *Let A and A' be strictly totally positive (p,q)-Lupaş matrices for the (p,q)-analogue of the Bernstein functions $(b_{p,q}^{r,n})_{0\leq r\leq n}$ and the nodes $\{t_i\}_{1\leq i\leq l+1}$ and $t_i' = t_i(1+\delta_i)$ 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*

$$|(\mathcal{BD}(A'))_{i,j} - (\mathcal{BD}(A))_{i,j}| \leq \frac{4n\kappa_{p,q}\theta}{1 - 4n\kappa_{p,q}\theta}(\mathcal{BD}(A))_{i,j}.$$

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

$$t_i' - t_j' = (t_i - t_j)(1 + \delta_{i,j}), \quad |\delta_{i,j}| \leq \frac{\theta}{rel\_gap_t}, \tag{17}$$

$$1 - t_i' = (1 - t_i)(1 + \delta_i'), \quad |\delta_i'| \leq \frac{\theta}{rel\_gap_1}, \tag{18}$$

and

$$p^k(1 - t_i') + q^k t_i' = (p^k(1 - t_i) + q^k t_i)(1 + \delta_i''), \quad |\delta_i''| \leq \frac{\theta}{rel\_gap_1}. \tag{19}$$

Accumulating the perturbations in the style of Higham (see Chapter 3 of [12]) using the formulae (8) for the $m_{i,j}$, and (17), (18) and (19) we obtain

$$m_{i,j}' = m_{i,j}(1 + \bar{\delta}), \quad |\bar{\delta}| \leq \frac{4n\kappa_{p,q}\theta}{1 - 4n\kappa_{p,q}\theta},$$

where $m_{i,j}'$ are the entries of $\mathcal{BD}(A')$ below the main diagonal. Proceeding similarly but now using (9) and (18) we have

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

where $\widetilde{m}'_{i,j}$ are the entries of $\mathcal{BD}(A')$ above the main diagonal. Analogously, and using in this case formulae (10), and (17), (18) and (19) we get

$$p'_{i,i} = p_{i,i}(1+\bar{\delta}), \quad |\bar{\delta}| \le \frac{(3n-1)\kappa_{p,q}\theta}{1-(3n-1)\kappa_{p,q}\theta},$$

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

$$|(\mathcal{BD}(A'))_{i,j} - (\mathcal{BD}(A))_{i,j}| \le \frac{4n\kappa_{p,q}\theta}{1-4n\kappa_{p,q}\theta}(\mathcal{BD}(A))_{i,j}. \quad \square$$

From this result we infer that the quantity $4n\kappa_{p,q}$ is an adequate *structured condition number* of A with respect to the perturbations in the data t_i and so, important quantities in the determination of a structured condition number are the relative separation between the nodes and the relative distances between the nodes and 1 [21,25]. Similar results involving only in the determination of the structured condition number the relative separation between the nodes can be found in [6,16].

Combining this theorem with Corollary 7.3 in [16], which states that small componentwise relative perturbations of $\mathcal{BD}(A)$ cause only small relative perturbations in the eigenvalues λ_i and singular values σ_i of A, we obtain that

$$|\lambda'_i - \lambda_i| \le O(n^3\kappa_{p,q}\theta)\lambda_i \quad \text{and} \quad |\sigma'_i - \sigma_i| \le O(n^3\kappa_{p,q}\theta)\sigma_i,$$

where λ'_i and σ'_i are the eigenvalues and the singular values of A'. That is, small relative perturbations in the nodes of a (p,q)-Lupaş matrix A cause only small relative perturbations in its eigenvalues and singular values.

7. Numerical experiments

Let A be the strictly totally positive (p,q)-Lupaş matrix for the (p,q)-analogue of the Bernstein functions $(b^{r,n}_{p,q})_{0\le r\le n}$, with nodes $\{t_i\}_{1\le i\le l+1}$, $0 < t_1 < t_2 < \cdots < t_{l+1} < 1$. In this section, algorithm `TNBDpqLupas` (Algorithm 2) to compute the bidiagonal factorization of A is applied to solve some fundamental problems in linear algebra. In the square case, the problems chosen have been linear system solving and computation of both the eigenvalues and the inverse of matrix A. In the rectangular case, the problems considered have been the computation of both the singular values and the Moore-Penrose inverse of A. All the problems in this section have been solved in an efficient and accurate way.

7.1. Linear system

Given the linear system

$$Ax = b,$$

where b has an alternating sign pattern, we obtain its solution by first computing accurately the bidiagonal decomposition of A with the algorithm `TNBDpqLupas`. Then, taking into account that A is a strictly totally positive matrix, the command `TNSolve`, provided by P. Koev [15], can be applied. It takes as input the matrix $\mathcal{BD}(A)$ containing the bidiagonal decomposition of A and vector b, and gives as output the solution of the linear system $Ax = b$. Provided b has an alternating sign pattern, the high relative accuracy of the solution is ensured [17].

Regarding the computational cost of the process, it depends on the costs of both algorithms `TNBDpqLupas` and `TNSolve`. As shown in Section 4, the cost of applying the former is $O(n^2)$ arithmetic operations. The implementation of the latter in Matlab requires also $O(n^2)$ arithmetic operations. Thus, obtaining the solution of the linear system with our method has a total computational cost of $O(n^2)$ arithmetic operations.

Example 1. To show the performance of our proposal we have taken $p = 2.5$, $q = 0.5$ and $n = 15$. The nodes considered are

$$\{t_i\}_{1\leq i\leq 16} = \left\{\frac{1}{17}, \frac{2}{17}, \frac{3}{17}, \frac{4}{17}, \frac{5}{17}, \frac{6}{17}, \frac{7}{17}, \frac{8}{17}, \frac{9}{17}, \frac{10}{17}, \frac{11}{17}, \frac{12}{17}, \frac{13}{17}, \frac{14}{17}, \frac{15}{17}, \frac{16}{17}, \right\},$$

and the vector b of the linear system is

$$b = (1, -2, 4, -1, 3, -2, 5, -1, 3, -4, 2, -5, 2, -2, 6, -1)^T .$$

We have compared the exact solution of the linear system, computed in *Mathematica* with exact arithmetic, with the approximated solutions obtained with both our algorithm and the command $A\backslash b$ of Matlab. The comparison is performed by means of the relative errors of each approximated solution, that is

$$\text{Error} = \frac{||x - x_e||_2}{||x_e||_2},$$

where x_e is the exact solution, and x is the corresponding approximated solution. The results are shown in the first two columns of Table 1. As we can observe, the solution obtained with the command $A\backslash b$ of Matlab is not accurate at all. With our proposal (MMV), on the contrary, a very accurate result is obtained.

7.2. Eigenvalues

Now we consider the problem of computing the eigenvalues of the (p,q)-Lupaş matrix A. They are obtained by first computing accurately its bidiagonal decomposition with the command `TNBDpqLupas`, and then applying the command `TNEigenValues`, provided

Table 1
In this table the results of the experiments performed in Examples 1, 2 and 3 are shown.

Linear system		Eigenvalues		Inverse	
$A\backslash b$	MMV	eig	MMV	inv	MMV
1.2e+00	5.6e-16	2.6e-08	6.2e-15	6.6e+01	7.1e-15

by P. Koev [15], which takes as input the matrix $\mathcal{BD}(A)$ containing the bidiagonal decomposition of a totally positive matrix A and gives as output the eigenvalues of A with high relative accuracy.

The computational cost of `TNEigenValues` is of $O(n^3)$ arithmetic operations [16], which dominates the $O(n^2)$ computational cost of `TNBDpqLupas` (see Section 4). Therefore, the total cost of computing the eigenvalues os A is $O(n^3)$.

Example 2. Taking the same matrix A as in the previous example, that is, the (p,q)-Lupaş matrix with $p = 2.5$, $q = 0.5$, $n = 15$ and the equidistant nodes $\left\{\frac{i}{17}\right\}$, $i = 1, \cdots, 16$, its eigenvalues are computed in *Mathematica* with a precision of 100 digits, and they are taken for the sake of comparison as the exact eigenvalues. The maxima of the relative errors of the approximated eigenvalues of A obtained with both our method (MMV) and the `eig` command of MATLAB are computed. The results obtained are shown in columns three and four of Table 1, where we observe that the proposed method outperforms the result offered by MATLAB.

7.3. Inverse

We consider now the last of the problems involving the strictly totally positive square matrix A. As in the two previous problems, the structure of A is exploited by first using algorithm `TNBDpqLupas` to obtain accurately a matrix $\mathcal{BD}(A)$ containing its bidiagonal decomposition. Then, the command `TNInverseExpand`, developed by A. Marco and J.J. Martínez in [24] and implemented in MATLAB in [15], is applied to $\mathcal{BD}(A)$ and provides the inverse of A with high relative accuracy.

The computational cost of algorithms `TNBDpqLupas` and `TNInverseExpand` is $O(n^2)$, what determines the same cost for computing the inverse of A.

Example 3. To evaluate the performance of our proposal, the same matrix A as the one appearing in the two previous experiments is taken. The exact inverse of A is computed with *Mathematica* with exact arithmetic. The approximate inverses of A are computed by using our proposal (MMV) and with the command `inv` of MATLAB. In the last two columns of Table 1 the corresponding maxima of the componentwise relative errors are included. They show that the algorithm proposed in this paper computes accurately the inverse of A, whereas the `inv` command produces an inverse not accurate at all.

7.4. Singular values

Let us assume now that A is a (p,q)-Lupaş rectangular matrix of dimension $(l+1)\times(n+1)$. The first step for computing the singular values of A consists of applying algorithm `TNBDpqLupas` to obtain matrix $\mathcal{BD}(A)$, which contains the bidiagonal decomposition of A. Then, the `TNSingularValues` command [15] is applied to $\mathcal{BD}(A)$, providing with high relative accuracy the singular values of A.

As for the computational cost, the cost of algorithm `TNBDpqLupas` is $O(ln)$, and the cost of `TNSingularValues` is $O(ln^2)$ arithmetic operations. Therefore, our proposal computes the singular values of A at a cost of $O(ln^2)$ arithmetic operations.

As a byproduct, the accurate computation of the 2-norm condition number of A can be calculated accurately, since it can be computed by dividing the greatest by the smallest singular values of A.

Example 4. To illustrate the good performance of our algorithm, we take the particular values $p = 0.7$, $q = 2.5$, $n = 10$, $l = 15$, and the vector of nodes

$$\{t_i\}_{1\le i\le 16} = \left\{\frac{1}{10}, \frac{1}{9}, \frac{1}{8}, \frac{1}{7}, \frac{1}{6}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{9}{16}, \frac{5}{8}, \frac{2}{3}, \frac{7}{10}, \frac{7}{8}, \frac{9}{10}, \frac{14}{15}\right\}.$$

The first two columns of Table 2 show the maximum relative errors obtained by our method (MMV) and by the command `svd` of MATLAB. The singular values considered as exact are obtained with the command SingularValueList of *Mathematica* with a precision of 100 digits. Besides, Table 2 shows in its third and fourth columns the relative errors obtained when computing the 2-norm condition number of A with our approach (MMV) and with the command `cond` of MATLAB. To perform the comparison, the considered exact condition number $\kappa_2(A)$ is obtained in *Mathematica* by dividing the largest singular value of A by the smallest one.

Looking at Table 2 we notice that our approach computes both the singular values and the condition number of A accurately, and that commands `cond` and `svd` of MATLAB do not produce accurate results at all.

7.5. Moore-Penrose inverse

The matrix A we consider in this section is the same rectangular matrix as the matrix in the previous section. To compute accurately the Moore-Penrose inverse of A, first the bidiagonal decomposition of A, stored in matrix $\mathcal{BD}(A)$, is obtained with high relative accuracy using algorithm `TNBDpqLupas`. Matrix $\mathcal{BD}(A)$ is the input of the algorithm `TNQR` [15], which gives with high relative accuracy the QR decomposition of A. Finally, the inverse of the submatrix of the nonsingular totally positive triangular matrix R formed by its first $n+1$ rows is computed by the algorithm `TNInverseExpand` [24], and such inverse is multiplied by the transpose of the submatrix of Q formed by its last $n+1$ columns. The result is the Moore-Penrose inverse of A.

Table 2
This table shows the results of the experiments 4 and 5.

Singular Values		Condition Number		Moore-Penrose inverse	
`svd`	MMV	`cond`	MMV	`pinv`	MMV
1.0e+01	5.7e-16	9.5e-01	3.5e-15	1.9e+01	9.2e-14

The computational cost of both `TNBDpqLupas` and `TNInverseExpand` is $O(n^2)$. The command `TNQR` requires $O(l^2n)$ arithmetic operations [16]. Thus, the total cost to accurately compute the Moore-Penrose inverse of A is $O(l^2n)$ arithmetic operations.

Example 5. The numerical experiment we perform here considers the same particular values as in Example 4, that is, $p = 0.7$, $q = 2.5$, $n = 10$, $l = 15$, and the same vector of nodes. The exact Moore-Penrose inverse of A is obtained with *Mathematica* in exact arithmetic, and it is used to compute the maximum componentwise relative error obtained with our method (called MMV) and with the command `pinv` of Matlab. Such relative errors are shown in Table 2. We notice that the Moore-Penrose inverse of A obtained with the method proposed in this paper is accurate, whereas the application of the `pinv` command does not produce an accurate result.

To conclude, we observe that in all the experiments performed in this section for the five considered problems our proposal has given accurate results, clearly outperforming the corresponding commands of Matlab. The reason of this is that we take into account the structure of matrix A, whereas Matlab does not. The ill-conditioning of matrix A makes that the results obtained with Matlab are far from being accurate. Specifically, in the first three problems the square matrix A taken to perform the experiments has 2-norm condition number κ_2 =1.5e+75, and for the rectangular matrix of the last two problems, the 2-norm condition number of A is κ_2 =2.2e+22. The ill-conditioning of A makes that general methods are not appropriate to make computations directly on A. Our method, on the contrary, takes into account the structure of A to compute accurately its bidiagonal decomposition. Moreover, none of the algorithms presented in this paper requires the matrix A to be constructed, working in all of them only with the nodes $\{t_i\}_{1\leq i\leq l+1}$. As a result, very accurate results are obtained.

Declaration of competing interest

None declared.

References

[1] T. Ando, Totally positive matrices, Linear Algebra Appl. 90 (1987) 165–219.
[2] J.M. Carnicer, J.M. Peña, Total positivity and optimal bases, in: M. Gasca, C.A. Michelli (Eds.), Total Positivity and Its Applications, Jaca, 1994, Kluwer Academic Publishers, Dordrecht, 1996, pp. 133–155.

[3] J. Delgado, H. Orera, J.M. Peña, Accurate bidiagonal decomposition and computations with generalized Pascal matrices, J. Comput. Appl. Math. 391 (2021) 113443.
[4] J. Delgado, J.M. Peña, Accurate computations with Lupaş matrices, Appl. Math. Comput. 303 (2017) 171–177.
[5] J. Demmel, I. Dumitriu, O. Holtz, P. Koev, Accurate and efficient expression evaluation and linear algebra, Acta Numer. 17 (2008) 87–145.
[6] J. Demmel, P. Koev, Accurate SVDs of polynomial Vandermonde matrices involving orthonormal polynomials, Linear Algebra Appl. 417 (2006) 382–396.
[7] S.M. Fallat, C.R. Johnson, Totally Nonnegative Matrices, Princeton University Press, Princeton, NJ, 2011.
[8] R.T. Farouki, The Bernstein polynomial basis: a centennial retrospective, Comput. Aided Geom. Des. 29 (2012) 379–419.
[9] M. Gasca, J.M. Peña, Total positivity and Neville elimination, Linear Algebra Appl. 165 (1992) 25–44.
[10] M. Gasca, J.M. Peña, A matricial description of Neville elimination with applications to total positivity, Linear Algebra Appl. 202 (1994) 33–45.
[11] M. Gasca, J.M. Peña, On factorizations of totally positive matrices, in: M. Gasca, C.A. Michelli (Eds.), Total Positivity and Its Applications, Jaca, 1994, Kluwer Academic Publishers, Dordrecht, 1996, pp. 109–130.
[12] N.J. Higham, Accuracy and Stability of Numerical Algorithms, 2nd ed., SIAM, Philadelphia, 2002.
[13] M.N. Hounkonnou, J.D.B. Kyemba, $\mathcal{R}$(p,q)-calculus: differentiation and integration, SUT J. Math. 49 (2013) 145–167.
[14] K. Khan, D.K. Lobiyal, Bézier curves based on Lupaş (p,q)-analogue of Bernstein functions in CAGD, J. Comput. Appl. Math. 317 (2017) 458–477.
[15] P. Koev, http://www.math.sjsu.edu/~koev.
[16] P. Koev, Accurate eigenvalues and SVDs of totally nonnegative matrices, SIAM J. Matrix Anal. Appl. 21 (2005) 1–23.
[17] P. Koev, Accurate computations with totally nonnegative matrices, SIAM J. Matrix Anal. Appl. 29 (2007) 731–751.
[18] A. Lupaş, A q-analogue of the Bernstein operator, Semin. Numer. Stat. Calc. 9 (1987) 85–92.
[19] E. Mainar, J.M. Peña, B. Rubio, Accurate computations with collocation and Wronskian matrices of Jacobi polynomials, J. Sci. Comput. 87 (2021) 77.
[20] A. Marco, J.J. Martínez, A fast and accurate algorithm for solving Bernstein-Vandermonde linear systems, Linear Algebra Appl. 422 (2007) 616–628.
[21] A. Marco, J.-J. Martínez, Accurate computations with totally positive Bernstein–Vandermonde matrices, Electron. J. Linear Algebra 26 (2013) 357–380.
[22] A. Marco, J.-J. Martínez, Bidiagonal decomposition of rectangular totally positive Said-Ball-Vandermonde matrices: error analysis, perturbation theory and applications, Linear Algebra Appl. 495 (2016) 90–107.
[23] A. Marco, J.-J. Martínez, J.M. Peña, Accurate bidiagonal decomposition of totally positive Cauchy-Vandermonde matrices and applications, Linear Algebra Appl. 517 (2017) 63–84.
[24] A. Marco, J.-J. Martínez, Accurate computation of the Moore-Penrose inverse of strictly totally positive matrices, J. Comput. Appl. Math. 350 (2019) 299–308.
[25] A. Marco, J.J. Martínez, R. Viaña, Error analysis, perturbation theory and applications of the bidiagonal decomposition of rectangular totally positive h-Bernstein-Vandermonde matrices, Linear Algebra Appl. 613 (2021) 377–392.
[26] A. Pinkus, Totally Positive Matrices, Cambridge University Press, Cambridge, UK, 2010.