

Article

The Application of the Bidiagonal Factorization of Totally Positive Matrices in Numerical Linear Algebra

José-Javier Martínez 

Departamento de Física y Matemáticas, Universidad de Alcalá, 28871 Alcalá de Henares, Spain;
jjavier.martinez@uah.es

Abstract: The approach to solving linear systems with structured matrices by means of the bidiagonal factorization of the inverse of the coefficient matrix is first considered in this review article, the starting point being the classical Björck–Pereyra algorithms for Vandermonde systems, published in 1970 and carefully analyzed by Higham in 1987. The work of Higham briefly considered the role of total positivity in obtaining accurate results, which led to the generalization of this approach to totally positive Cauchy, Cauchy–Vandermonde and generalized Vandermonde matrices. Then, the solution of other linear algebra problems (eigenvalue and singular value computation, least squares problems) is addressed, a fundamental tool being the bidiagonal decomposition of the corresponding matrices. This bidiagonal decomposition is related to the theory of Neville elimination, although for achieving high relative accuracy the algorithm of Neville elimination is not used. Numerical experiments showing the good behavior of these algorithms when compared with algorithms that ignore the matrix structure are also included.

Keywords: bidiagonal decomposition; Björck–Pereyra algorithm; structured matrix; totally positive matrix; high relative accuracy

MSC: 65F05; 65F15; 65F20; 65F35; 15A23; 15B48; 15B05



Citation: Martínez, J.-J. The Application of the Bidiagonal Factorization of Totally Positive Matrices in Numerical Linear Algebra. *Axioms* **2024**, *13*, 258. <https://doi.org/10.3390/axioms13040258>

Academic Editors: Jorge Delgado Gracia and Florin Felix Nichita

Received: 27 February 2024

Revised: 6 April 2024

Accepted: 11 April 2024

Published: 14 April 2024



Copyright: © 2024 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The second edition of the Handbook of Linear Algebra [1], edited by L. Hogben, is substantially expanded from the first edition of 2007 and, in connection with our work, it contains a new chapter by M. Stewart entitled *Fast Algorithms for Structured Matrix Computations* (Chapter 62) [2]. This chapter includes, as Section 62.7, the subject of fast algorithms for Vandermonde systems. Among these algorithms the author incorporates the Björck–Pereyra algorithm for solving Vandermonde linear systems, indicating the relationship with polynomial interpolation by using the Newton basis, and the interpretation of this process as a factorization in terms of bidiagonal matrices.

Section 62.7 also recalls the high relative accuracy of the Björck–Pereyra algorithm when the (nonnegative) nodes are ordered in increasing order, a fact observed in the error analysis presented by N. J. Higham in [3].

In connection with *high relative accuracy*, the book [1] also contains a chapter (no. 59), due to Z. Drmač, entitled *Computing Eigenvalues and Singular Values to High Relative Accuracy* (already available in the first edition as Chapter 46) [4]. In this chapter, the author includes references to the work of Demmel and Kahan [5] and of Fernando and Parlett [6] on the computation of singular values of bidiagonal matrices, fundamental references for the subsequent work of Demmel and Koev (see [7–9]). A brief comment on the bidiagonal factorization of totally nonnegative matrices and to the work of Koev on the computation of eigenvalues and singular values is also included in Section 59.3 of that chapter.

We find in that book [1] another chapter closely related to our subject: Chapter 29 due to S. M. Fallat [10]. Section 29.2 of that chapter (entitled *Factorizations*) explicitly considers

the bidiagonal factorization of totally positive matrices. The author writes: *Recently, there has been renewed interest in total positivity partly motivated by the so-called “bidiagonal factorization”, namely, the fact that any totally positive matrix can be factored into entry-wise bidiagonal matrices. This result has proven to be a very useful and tremendously powerful property for this class.*

Nevertheless, the approach of the chapter is mainly theoretical, and the author does not include references to numerical linear algebra papers that exploit the bidiagonal factorization of totally positive matrices (the part of the Handbook [1] devoted to numerical methods is Part III, which includes Chapter 50 through Chapter 64).

We observe the same situation in two other recent relevant books on numerical linear algebra: the fourth edition of Golub and Van Loan’s *Matrix Computations* [11] and the new book by A. Björck *Numerical Methods in Matrix Computations* [12]. In both works the authors pay attention to the description of the Björck–Pereyra algorithm for solving Vandermonde linear systems, in the corresponding chapters devoted to linear system solving: Chapter 4 (*Special Linear Systems*) in [11], and Chapter 1 (*Direct Methods for Linear Systems*) in [12]. Other separate chapters from both books are devoted to eigenvalue and singular value problems.

Consequently, to the reader, these problems appear as very different problems: linear system solving, or eigenvalue and singular value problems. The purpose of the present work is to show, in the important cases where total positivity is present, the unifying role of the bidiagonal decompositions of the corresponding matrices to develop a first stage in algorithms for solving various linear algebra problems with high relative accuracy.

The analysis of the Björck–Pereyra algorithms presented by Higham in [3] showed (among many other things) the important role of total positivity in the accuracy of the algorithms. So this property will be of fundamental importance in our account.

Let us recall that, according to the classical respectable definition, a matrix is said to be *totally positive* if all its minors are nonnegative, and when all the minors are positive the matrix is called *strictly totally positive* [13]. More recently, matrices with all their minors nonnegative have been called *totally nonnegative* matrices, this being mathematically more precise. Recent books on this subject are [14,15]. These monographs cover many aspects of the theory and applications of totally positive matrices and, although they do not develop the topic of accurate computations with this type of matrices, they contain useful references to the work of Demmel and Koev in this field.

The classical Björck–Pereyra algorithms date back to 1970, and now we want to highlight the important work of Demmel and Koev in the development of these algorithms with high relative accuracy. In [7] Demmel and Koev called some of these methods Björck–Pereyra-type methods, acknowledging in this way the importance of the work of Björck and Pereyra, and in the present work we want to carry out a necessary explanation of the analogies and differences between Björck–Pereyra-type methods and related methods, the main analogy being the role of bidiagonal factorization of matrices.

One key fact is that while Björck–Pereyra algorithms for linear systems were related to the factorization of the inverse A^{-1} of the coefficient matrix, the use of the bidiagonal factorization of A considered in the work of Koev allowed to extend the accurate computations to various linear algebra problems different from linear system solving.

This bidiagonal factorization is related to Neville elimination (see [16,17]), and early applications of it were the solution of Cauchy–Vandermonde linear systems in [18] (in this case by using the related factorization of A^{-1}) and the computation of eigenvalues and singular values in [8].

It is clear that a very important concept in our presentation is that of *high relative accuracy*. By using the bidiagonal factorization related to Neville elimination (a key result being a theorem of [17] included in [8] as Theorem 2.1), Koev presented in [8] algorithms to compute eigenvalues and singular values of totally positive matrices to *high relative accuracy*. The definition of this concept can be seen in the recent paper [19]:

For a computed quantity \hat{x} , to have *high relative accuracy* means that it satisfies an error bound with its true counterpart x

$$|\hat{x} - x| \leq \theta|x|,$$

where θ is a modest multiple of the *machine precision* ϵ . In other words, the sign and most significant digits of x must be correct.

As we learn from [20], an algorithm computes to high relative accuracy if it satisfies the so-called **NIC** (no inaccurate cancellation) condition. **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. In Section 2 the problem of linear system solving is considered, a problem to which Björck–Pereyra algorithms and their generalizations were devoted. Section 3 reviews Neville elimination and the bidiagonal factorization associated with it, a key theoretical tool for the development of algorithms. The extension of these algorithms based on a bidiagonal factorization to the problems of eigenvalue and singular value computation, including a look at the initial history of this approach, is addressed in Section 4, while Section 5 considers the extension to the rectangular case, which allows solving least squares problems. The brief Section 6 considers the class of totally positive tridiagonal matrices, Section 7 includes the reference to very recent applications to matrices different from collocation matrices as well as the singular case, and finally, Section 8 is devoted to conclusions.

2. Linear System Solving: The Vandermonde Case and Some Extensions

As recently recalled in [12], it was in 1970 when Björck and Pereyra showed that the Newton–Horner algorithm for solving a Vandermonde linear system related to polynomial interpolation could be expressed in terms of a factorization of the inverse of the Vandermonde matrix as a product of diagonal and lower bidiagonal matrices. In their paper [21] those authors included, after the numerical experiments, the following sentence:

It seems as if at least some problems connected with Vandermonde systems, which traditionally have been considered too ill-conditioned to be attacked, actually can be solved with good precision.

Many years later, at the beginning of his brilliant contributions in the field of numerical linear algebra, Higham gave in [3] an analysis of Björck–Pereyra algorithms and indicated that when the interpolation nodes are nonnegative and ordered in increasing order then the corresponding Vandermonde matrix

$$\begin{pmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & x_n^n \end{pmatrix}$$

is *totally positive*. If, in addition, the components of f (see next paragraph) alternate in sign, then there is no cancellation and high relative accuracy is obtained.

We follow the clear presentation of [11,12] to show the matrix interpretation of the Björck–Pereyra algorithm to solve the dual Vandermonde system $V^T a = f$, i.e., the linear system corresponding to polynomial interpolation, where f is the vector of interpolation data. These authors call *Vandermonde matrix* to the transpose V of that matrix, whose first row is $(1, 1, \dots, 1)$.

So we will consider the linear system with a coefficient matrix (of order $n + 1$)

$$V^T = \begin{pmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & x_n^n \end{pmatrix},$$

where x_0, \dots, x_n are the interpolation nodes.

As we read in Section 4.6 of [11] and in Section 1.8.3 of [12], the solution a of $V^T a = f$ is given as

$$a = (V^T)^{-1}f = L^T U^T f =$$

$$L_0(x_0)^T \cdots L_{n-1}(x_{n-1})^T D_{n-1}^{-1} L_{n-1}(1) \cdots D_0^{-1} L_0(1).$$

For $k = 0, \dots, n-1$, D_k are diagonal matrices:

$$D_k = \text{diag}(1, \dots, 1, x_{k+1} - x_0, \dots, x_n - x_{n-k-1})$$

(with the $k+1$ initial diagonal entries equal to 1).

On the other hand, for $k = 0, \dots, n-1$, $L_k(\alpha)$ are lower bidiagonal matrices with all the diagonal entries equal to 1. For the subdiagonal entries, we have

$$L_k(\alpha)(j+1, j) = 0 \quad \text{for } j = 1, \dots, k$$

and

$$L_k(\alpha)(j+1, j) = -\alpha \quad \text{for } j = k+1, \dots, n+1.$$

In Section 3 we will illustrate with an example the precise structure of those matrices and we will show the differences between this factorization and the factorization related to Neville elimination.

Let us observe that we can identify two stages in the algorithm, since $a = L^T U^T f = L^T c$, where $c = U^T f$. This vector c is the vector of divided differences, i.e., the coefficients of the interpolating polynomial in the *Newton basis*. The second stage ($a = L^T c$) corresponds to the change of basis from the Newton basis to the monomial basis.

This approach was later applied by Boros, Kailath and Olshevsky to design a Björck–Pereyra-type algorithm for solving Cauchy linear systems. As we read in [22], “*for the class of totally positive Cauchy matrices the new algorithm is forward and backward stable, producing a remarkable high relative accuracy. In particular, Hilbert linear systems, often considered to be too ill-conditioned to be attacked, can be rapidly solved with high precision*”.

Another generalization (in this case to confluent Vandermonde-like systems) was presented by Higham (see [23] and Chapter 22 of his book [24]), but in general, the bidiagonal structure of the matrices involved in the factorization is lost. More recently a new generalization of Björck–Pereyra algorithms (now to the case of Szegö–Vandermonde matrices) was introduced in [25], but the authors admit that some matrices involved in the factorization are not sparse (i.e., they are far from being bidiagonal).

A bidiagonal factorization of the inverse of the coefficient matrix (in this case for a Cauchy–Vandermonde matrix) related to Neville elimination was presented in [18], a pioneering paper on this subject which has recently been completed in the light of the work of Koev (see [26]). Also, in [7] the authors extend the Björck–Pereyra algorithm to solve totally positive *generalized Vandermonde* linear systems $Gy = b$ with

$$G = \begin{pmatrix} x_1^{a_1} & x_1^{a_2} & \cdots & x_1^{a_n} \\ x_2^{a_1} & x_2^{a_2} & \cdots & x_2^{a_n} \\ \vdots & \vdots & \ddots & \vdots \\ x_n^{a_1} & x_n^{a_2} & \cdots & x_n^{a_n} \end{pmatrix}$$

(where $0 \leq a_1 < a_2 < \cdots < a_n$ are integers and $0 < x_1 < x_2 < \cdots < x_n$) by computing a bidiagonal decomposition of G^{-1} and computing $y = G^{-1}b$. The authors indicate that this decomposition, related to Neville elimination [17], reveals the total positivity of G and yields a Björck–Pereyra-type algorithm for the solution of $Gy = b$. They also comment that when b has an alternating sign pattern then their algorithm is subtraction-free so that the solution y is very accurate, the same fact observed by Higham in [3] when analyzing the

Björck–Pereyra algorithm. This is a consequence of the fact that the inverse of a totally positive matrix G has a checkerboard sign pattern:

$$G^{-1} = \begin{pmatrix} + & - & + & - \\ - & + & - & + \\ + & - & + & - \\ - & + & - & + \end{pmatrix}.$$

In their work [20] Demmel and co-workers have briefly indicated the importance of these Björck–Pereyra-type methods for solving linear systems with several classes of totally positive matrices (Vandermonde, Cauchy, Pascal, Cauchy–Vandermonde, generalized Vandermonde, Bernstein–Vandermonde, ...). The main fact, however, is that when the bidiagonal factorization of the coefficient matrix A is related to *Neville elimination*, then there are exactly n^2 independent nonnegative parameters in the bidiagonal decomposition which are stored in a matrix $B = \mathcal{BD}(A)$ (bidiagonal decomposition of A). Then, starting from an accurate $\mathcal{BD}(A)$, virtually all linear algebra with totally nonnegative matrices can be performed accurately (i.e., *to high relative accuracy*) by using algorithms due to P. Koev (see [27], where P. Koev has also included some algorithms from other authors).

Unfortunately the great book of Björck [12] only considers the papers [7,22], which include (in the title or in the abstract) the name Björck–Pereyra, and so only the problem of linear system solving is analyzed, while the application of the bidiagonal factorization to the computation of eigenvalues and singular values is not addressed.

The great achievement of the work of Koev was to show that, while all these Björck–Pereyra-type methods pay attention to the factorization of the inverse A^{-1} of the coefficient matrix A (which seems natural when solving a linear system $Ax = f$), if one uses the (unique) bidiagonal decomposition of A related to Neville elimination various linear algebra problems can additionally be solved.

Therefore, our following section is devoted to presenting this bidiagonal decomposition of A , and to showing the analogies and differences between classical Björck–Pereyra algorithms and algorithms related to Neville elimination when dealing with totally positive matrices.

3. Bidiagonal Factorization and Neville Elimination

As recently recalled by J. M. Peña in [28], the work by Gasca and Mühlbach on the connection between interpolation formulas and elimination techniques made it clear that what they named *Neville elimination* had special interest for totally positive matrices.

The paper [28] also considers the history of initial approaches to bidiagonal factorization (different from Neville elimination) and addresses (following [29]) the question of its uniqueness, presenting in its Subsection 5.1 the details of the bidiagonal factorization by means of Neville elimination.

In the survey paper [30], Gasca and Mühlbach recall the early history of this elimination technique: “one strategy which we called Neville elimination proved to be well suited to work with some special classes of matrices, in particular totally positive matrices”. They also remark on one of the first important applications of this elimination technique: “using the Neville elimination strategy, in [31,32] tests of algorithmic complexity $O(N^4)$ for matrices being strictly totally positive were derived for the first time”. These two papers were published in 1987, and later on in [17,33,34] Gasca and Peña greatly improved the previous results on Neville elimination and total positivity. Their work was one of the starting points for the work of Demmel and Koev in developing algorithms, starting from the appropriate bidiagonal decomposition associated with Neville elimination (which is the theoretical tool but not the algorithm being used with the various classes of structured matrices), made the accurate solution of many linear algebra problems for totally positive matrices possible, including eigenvalue and singular value computation (see [7–9]).

For the sake of completeness, we will now recall (following [26,35]) several basic facts on Neville elimination and total positivity which are very important to obtain the results

presented in this section. The notation follows the one used in [33,34]. 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 .

Neville elimination is a procedure that makes zeros in a matrix adding to a given row an appropriate multiple of the previous one.

Let $A = (a_{i,j})_{1 \leq i,j \leq n}$ be a square matrix of order n . The Neville elimination of A consists of $n - 1$ steps resulting in a sequence of matrices $A_1 := A \rightarrow A_2 \rightarrow \dots \rightarrow A_n$, 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 ($t = 1, \dots, n - 1$) by using the following formula:

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

In this process, the element

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

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 [33]. The Neville elimination can be conducted 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 [33],

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

The element

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

is called *multiplier* of the Neville elimination of A . The matrix $U := A_n$ is upper triangular and has 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 continuing 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 conducted 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 [17]).

Neville elimination characterizes nonsingular totally positive matrices, according to the results of [17] recalled as Theorem 8 in [28]:

Theorem 1. *A square matrix A is nonsingular totally positive if and only if the Neville elimination of A and A^T can be performed without row exchanges, all the multipliers of the Neville elimination of A and A^T are nonnegative and all the diagonal pivots of the Neville elimination of A are positive.*

The bidiagonal decomposition of A and of its inverse A^{-1} are given in the following theorems (see [8,9,17,28,33,34], and see [26] for the case of Cauchy–Vandermonde matrices):

Theorem 2. Let A be a nonsingular totally positive matrix of size $n \times n$. Then, A^{-1} admits a factorization in the form

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

where G_i ($1 \leq i \leq n-1$) are $n \times n$ upper triangular bidiagonal matrices, F_i ($i = 1, \dots, n-1$) are $n \times n$ lower triangular bidiagonal matrices, and D is a diagonal matrix of order n .

The structure of these matrices is the following:

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

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

and D is the diagonal matrix of order n

$$D = \text{diag}\{p_{1,1}, p_{2,2}, \dots, p_{n,n}\}.$$

The quantities $m_{i,j}$ are the multipliers of the Neville elimination of the matrix A . The quantities $\tilde{m}_{i,j}$ are the multipliers of the Neville elimination of A^T . Finally, the i th diagonal element of D ($i = 1, \dots, n$) is the diagonal pivot p_{ii} of the Neville elimination of A .

Theorem 3. Let A be a nonsingular totally positive matrix of size $n \times n$. Then, A admits a factorization in the form

$$A = \bar{F}_{n-1} \cdots \bar{F}_1 D \bar{G}_1 \cdots \bar{G}_{n-1}$$

where \bar{F}_i ($1 \leq i \leq n-1$) are $n \times n$ lower triangular bidiagonal matrices, \bar{G}_i ($1 \leq i \leq n-1$) are $n \times n$ upper triangular bidiagonal matrices, and D is a diagonal matrix of order n .

The structure of these matrices is now the following:

$$\bar{F}_i = \begin{bmatrix} 1 & & & & \\ 0 & 1 & & & \\ & \ddots & \ddots & & \\ & & 0 & 1 & \\ & & & m_{i+1,1} & 1 \\ & & & & m_{i+2,2} & 1 \\ & & & & & \ddots & \ddots \\ & & & & & & m_{n,n-i} & 1 \end{bmatrix}, \bar{G}_i^T = \begin{bmatrix} 1 & & & & \\ 0 & 1 & & & \\ & \ddots & \ddots & & \\ & & 0 & 1 & \\ & & & \tilde{m}_{i+1,1} & 1 \\ & & & & \tilde{m}_{i+2,2} & 1 \\ & & & & & \ddots & \ddots \\ & & & & & & \tilde{m}_{n,n-i} & 1 \end{bmatrix}$$

and D is the diagonal matrix of order n

$$D = \text{diag}\{p_{1,1}, p_{2,2}, \dots, p_{n,n}\}.$$

As in the previous theorem, the quantities $m_{i,j}$ are the multipliers of the Neville elimination of the matrix A , the quantities $\tilde{m}_{i,j}$ are the multipliers of the Neville elimination of A^T , and the i th diagonal element of D ($i = 1, \dots, n$) is the diagonal pivot p_{ii} of the Neville elimination of A .

Remark 1. The algorithm TNBD in [27] computes the matrix denoted as $\mathcal{BD}(A)$ in [9], which represents the bidiagonal decomposition of A (its entries are the n^2 parameters $m_{i,j}$, $\tilde{m}_{i,j}$ and p_{ii} of Theorem 3, as illustrated in the next example). But it is a remarkable fact that the same matrix $\mathcal{BD}(A)$ also serves to represent the bidiagonal decomposition of A^{-1} (see Example 3.4 in [35]). The

algorithm TNBD computes $\mathcal{BD}(A)$ by performing Neville elimination on A , which involves true subtractions, and therefore, does not guarantee high relative accuracy.

As we have seen, the matrix $\mathcal{BD}(A)$ serves to have a new parameterization of a totally nonnegative matrix A . Although it does not guarantee high relative accuracy, the algorithm TNBD of the package TNTool [27] computes $\mathcal{BD}(A)$ starting from A . But, remarkably we also have the other way: starting from $B = \mathcal{BD}(A)$, the algorithm TNExpand of [27] computes (now with high relative accuracy) the matrix A . In addition, the algorithm TNInverseExpand of Marco and Martínez (also included in [27]) computes, starting from $B = \mathcal{BD}(A)$, the inverse matrix A^{-1} , again with high relative accuracy.

The following example illustrates these facts and the accuracy obtained by using the algorithm TNInverseExpand.

Example. Starting from

$$B = \begin{pmatrix} 16 & 3 & 2 & 13 \\ 5 & 10 & 11 & 8 \\ 9 & 6 & 7 & 12 \\ 4 & 15 & 14 & 1 \end{pmatrix}$$

(the magic square appearing in Dürer's engraving *Melencolia I*) we obtain, by means of the statement $A = \text{TNExpand}(B)$, the following matrix (which we can call the Dürer totally positive matrix):

$$A = \begin{pmatrix} 16 & 48 & 96 & 1248 \\ 80 & 250 & 610 & 8810 \\ 720 & 2310 & 6277 & 94941 \\ 2880 & 10140 & 37011 & 617764 \end{pmatrix}.$$

This means that the bidiagonal decomposition of A is

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 4 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 9 & 1 & 0 \\ 0 & 0 & 15 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 5 & 1 & 0 & 0 \\ 0 & 6 & 1 & 0 \\ 0 & 0 & 14 & 1 \end{pmatrix} \begin{pmatrix} 16 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 3 & 0 & 0 \\ 0 & 1 & 11 & 0 \\ 0 & 0 & 1 & 12 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 1 & 8 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 13 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Although of size 4×4 , the condition number of this matrix is $\kappa_2(A) = 1.4 \times 10^{11}$, which means the computation of the inverse (which we call $A^{-1}\text{Im}$) by using MATLAB R2018b will suffer the effect of the high condition number. In fact, by using the exact

inverse (which we call $A\mathbf{I}e$) computed by Maple we find that the relative error computed in the spectral norm is

$$\frac{\|A\mathbf{I}m - A\mathbf{I}e\|_2}{\|A\mathbf{I}e\|_2} = 7.1 \times 10^{-10},$$

while for the inverse matrix $A\mathbf{I}k$ computed by using the algorithm `TNInverseExpand(B)` we have full accuracy:

$$\frac{\|A\mathbf{I}k - A\mathbf{I}e\|_2}{\|A\mathbf{I}e\|_2} = 1.2 \times 10^{-16}.$$

Now, let us show the differences (also the analogies) between the bidiagonal factorization associated with Neville elimination (stored in $B = \mathcal{BD}(A)$) and the factorization corresponding to the Björck–Pereyra algorithm, by using as an example the Vandermonde matrix of order $n = 4$ with nodes 2, 3, 5, 8:

$$A = \begin{pmatrix} 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \\ 1 & 5 & 25 & 125 \\ 1 & 8 & 64 & 512 \end{pmatrix}.$$

For this matrix we have

$$B = \mathcal{BD}(A) = \begin{pmatrix} 1 & 2 & 2 & 2 \\ 1 & 1 & 3 & 3 \\ 1 & 2 & 6 & 5 \\ 1 & 3/2 & 5/2 & 90 \end{pmatrix},$$

which means

$$A = \bar{F}_3 \bar{F}_2 \bar{F}_1 D \bar{G}_1 \bar{G}_2 \bar{G}_3 =$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & \frac{3}{2} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & \frac{5}{2} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 90 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 3 & 0 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

and

$$A^{-1} = G_1 G_2 G_3 D^{-1} F_3 F_2 F_1.$$

$$A^{-1} = \begin{pmatrix} 1 & -2 & 0 & 0 \\ 0 & 1 & -2 & 0 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -3 & 0 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/6 & 0 \\ 0 & 0 & 0 & 1/90 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{5}{2} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -2 & 1 & 0 \\ 0 & 0 & -\frac{3}{2} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix}.$$

On the other hand, the factorization of A^{-1} corresponding to the matrix interpretation of the Björck–Pereyra algorithm is the following (see [11]):

$$A^{-1} = G_1 G_2 G_3 D_2^{-1} L_2 D_1^{-1} L_1 D_0^{-1} L_0 =$$

$$G_1 G_2 G_3 \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{1}{6} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & \frac{1}{5} \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{1}{3} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

Let us observe that Stage II of the algorithms (corresponding to the product $G_1 G_2 G_3$) is the same for both factorizations. This matrix $G_1 G_2 G_3$ is the matrix of change of basis from the Newton basis to the monomial basis.

On the contrary, Stage I of the algorithm (corresponding to the factorizations $D^{-1} F_3 F_2 F_1 = D_2^{-1} L_2 D_1^{-1} L_1 D_0^{-1} L_0$) is different in the Björck–Pereyra algorithm and in the algorithm TNSolve which starts from $B = \mathcal{BD}(A)$. This first stage corresponds to the computation of the *divided differences*, i.e., the coefficients of the interpolating polynomial in the Newton basis.

For instance, if we solve the linear system $Ax = f$ with $f = (9, -9, 9, -9)^T$ the solution is $x = (159, -125, 29, -2)^T$. The solution of Stage I is $c = (9, -18, 9, -2)^T$ (the coefficients in the Newton basis), which means that

$$(9) \cdot 1 + (-18)(t - 2) + (9)(t - 2)(t - 3) + (-2)(t - 2)(t - 3)(t - 5) =$$

$$159 - 125t + 29t^2 - 2t^3.$$

A recent study of polynomial Lagrange interpolation by using the Newton basis (instead of the monomial basis), including the bidiagonal factorization of the corresponding collocation matrices is presented in [36].

We finish this section with some numerical experiments of linear system solving with Vandermonde and Cauchy matrices. We only include simple experiments with matrices of small size, and using available algorithms so that the interested reader can easily reproduce them. All the algorithms of P. Koev put together in the package TNTool are available in [27].

We begin with the Vandermonde case, with a Vandermonde matrix A of order $n = 7$ corresponding to the nodes $1, 2, 3, 4, 5, 6, 7$ (these nodes are the second column of the matrix), whose condition number is $\kappa_2(A) = 2.4 \times 10^7$. We will solve the linear system $Ax = f$, with

$$f = [1/21, -1/21, 1/23, -1/23, 1/29, -1/29, 1/31]^T.$$

We will compare the approximate solutions with the exact solution x_e computed in exact rational arithmetic by using *Maple*, and the relative errors will be computed in MATLAB as $\text{norm}(x - xe, 2) / \text{norm}(xe, 2)$, i.e., we are computing the relative error in the Euclidean norm.

First, we compute the solution by means of the MATLAB function $A \setminus f$, and the corresponding relative error is 4.1×10^{-14} . Next, the system is solved by the Björck-Pereyra algorithm by using the algorithm VTsolve, which can be obtained from the m-files of Chapter 4 of the book of Golub and Van Loan [11], available in this web page: www.cs.cornell.edu/cv/GVL4/M-Files/M-Home.htm (accessed on 1 April 2024).

In this case, the relative error is 2.6×10^{-16} , which confirms the high relative accuracy to be expected of this algorithm when f has an alternating sign pattern.

Finally, we compute the solution of the linear system by using the algorithm TNSolve of P. Koev, by previously computing $B = \mathcal{BD}(A)$ by means of the algorithm TNVandBD of P. Koev. In this case, the relative error is 2.6×10^{-16} , which again confirms the high relative accuracy to be expected of this algorithm when f has an alternating sign pattern.

In the second example, the coefficient matrix is a Hilbert matrix A of order $n = 7$, constructed in MATLAB by means of the instruction $A = \text{hilb}(7)$, whose condition number is $\kappa_2(A) = 4.7 \times 10^{86}$. We will solve the linear system $Ax = f$, with

$$f = [1/21, -1/21, 1/23, -1/23, 1/29, -1/29, 1/31]^T.$$

As before, we will compare the approximate solutions with the exact solution x_e computed in exact rational arithmetic by using *Maple*, and the relative errors will be computed in MATLAB as $\text{norm}(x - xe, 2) / \text{norm}(xe, 2)$.

First, we compute the solution by means of the MATLAB function $A \setminus f$, and the corresponding relative error is 4.0×10^{-9} . Next, the system is solved by the BKO algorithm of Boros, Kailath and Olshevsky, which can be taken from page 277 of their paper [22]. In this case, the relative error is 1.3×10^{-16} , which confirms the high relative accuracy to be expected of this algorithm when f has an alternating sign pattern.

Finally, we compute the solution of the linear system by using the algorithm TNSolve of P. Koev by previously computing $B = \mathcal{BD}(A)$ by means of the algorithm TNCauchyBD of P. Koev, with the instruction $B = \text{TNCauchyBD}(x, y)$ by defining $x = [0, 1, 2, 3, 4, 5, 6]$ and $y = [1, 2, 3, 4, 5, 6, 7]$. In this case, the relative error is 1.4×10^{-16} , which again confirms the high relative accuracy to be expected of this algorithm when f has an alternating sign pattern.

4. Eigenvalue and Singular Value Problems

In Section 59.3 of the already commented chapter [4], Z. Drmač refers to the work of Koev [8,9] to recall that if the totally nonnegative matrix A is given implicitly by its bidiagonal decomposition then all its singular values (and eigenvalues, too) can be computed to

high relative accuracy. The author also recalls that an accurate bidiagonal representation is possible provided that certain minors can be accurately computed.

These results are not included in the recent books of Björck [12] and Golub–Van Loan [11] but are important to indicate that *the same bidiagonal factorization* (with n^2 parameters stored in the matrix $B = \mathcal{BD}(A)$) is the starting point for accurately computing the eigenvalues and the singular values of A . By using the algorithms `TNEigenValues` and `TNSingularValues` of the package `TNTool` [27], one can compute the eigenvalues and the singular values (respectively) of A to high relative accuracy.

The application of the bidiagonal decomposition for computing eigenvalues and singular values of totally positive matrices developed by P. Koev [8,9] has an important precedent in the work of Demmel and coworkers [37]. In Section 9 of [37], devoted to the class of totally positive matrices, the authors indicate that “*achieving high relative accuracy requires not just total positivity but an appropriate parameterization that permits minors to be evaluated to high relative accuracy*”.

Some years later, in Section 2 of [8] the author acknowledges these contributions of [37], and in Section 3 he adds a fundamental tool: the results on Neville elimination and total positivity introduced by Gasca and Peña in [17,33,34]. Koev shows that being able to compute to high relative accuracy all n^2 initial minors is a necessary and sufficient condition for accurately computing the bidiagonal decomposition $\mathcal{BD}(A)$ of a totally positive matrix.

In Section 7 of [8] we find the main idea for the construction of accurate algorithms: “*In other words, $\mathcal{BD}(A)$ determines the eigenvalues and the singular values of A accurately*”. In addition, it is indicated that the final step of the algorithms `TNEigenValues` and `TNSingularValues` is the computation of the singular values of a bidiagonal matrix by using the LAPACK routine `DLASQ1` ([5,6]) (which must be compiled to be used in the MATLAB algorithms), and it is known to introduce only a small additional relative error ([6]).

As we read in the introduction of [8], “when traditional algorithms are used to compute the eigenvalues or the singular values of an ill-conditioned TN matrix, only the largest eigenvalues and the largest singular values are computed with guaranteed relative accuracy. 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”. We will illustrate with two simple examples the good behavior of the algorithms `TNEigenValues` and `TNSingularValues` when applied to ill-conditioned totally positive matrices.

The first matrix we consider is a Hilbert matrix of order 10, constructed in MATLAB by means of the instruction `A = hilb(10)`. Let us recall that a Hilbert matrix is a special case of a Cauchy matrix with generic entries $c_{ij} = 1/(x_i - y_j)$. The condition number of this matrix is $\kappa_2(A) = 1.6 \times 10^{13}$.

We can compute $B = \mathcal{BD}(A)$ by means of the algorithm `TNCauchyBD` of P. Koev, with the instruction `B = TNCauchyBD(x, y)`, by defining

$$x = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], y = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$$

(P. Koev is using as entries of the Cauchy matrix $c_{ij} = 1/(x_i + y_j)$). Then the eigenvalues are computed by means of the instruction `TNEigenValues(B)`.

Then these eigenvalues are also computed by means of the standard MATLAB instruction `eig(A)`. We compare the approximate value of the smallest eigenvalue with the “exact” value obtained by using *Maple* with extended precision, and we obtain for the value computed by means of `eig(A)` a relative error of 9.3×10^{-5} , while for the algorithm `TNEigenValues` the relative error is 3.4×10^{-16} .

In the second example, we compute the singular values of the Pascal matrix of order 10, constructed in MATLAB by means of the instruction `A = pascal(10)`. Now the condition number is $\kappa_2(A) = 4.1 \times 10^9$.

It is interesting to observe that for Pascal matrices we have an exact $B = \mathcal{BD}(A)$: the matrix with all the entries equal to 1 (see [38]), which is constructed in MATLAB as `B = ones(n, n)`. Then, the singular values are computed by means of the instruction `TNSingularValues(B)`.

These singular values are also computed by means of the standard MATLAB instruction `svd(A)`. We compare the approximate value of the smallest singular value with the “exact” value obtained by using *Maple* with extended precision, and we obtain for the value computed by means of `svd(A)` a relative error of 1.7×10^{-9} , while for the algorithm `TNSingularValues` the relative error is 6.5×10^{-16} .

We see the effect of the ill-conditioning of the Hilbert and Pascal matrices when using the standard MATLAB functions `eig` and `svd`, while the algorithms `TNEigenValues` and `TNSingularValues` (starting from an accurate $\mathcal{BD}(A)$) give high relative accuracy.

The availability of the algorithms of Koev in [27] has encouraged the search for new algorithms for the bidiagonal decomposition of various classes of totally positive structured matrices, including matrices from new application fields such as [39]. The financial applications considered in [39] have recently been addressed in [40] where for the class of Green matrices, an algorithm of linear complexity is presented to compute (for a full matrix) the bidiagonal factorization.

It can be interesting to end this section with a look at the initial history of the application of $\mathcal{BD}(A)$ to solve linear algebra problems. The last paragraph of the introduction of [22] (published in 1999) contained the following sentence: “*The results of this paper were available since 1994 as ISL reports at Stanford University, and they were reported at several conferences. They have influenced a recent interest in the connections between accuracy and total positivity (e.g., [18,37]).*”

In fact, both the very important paper [37] (which was still a LAPACK working note) and our early contribution on the use of the bidiagonal factorization related to Neville elimination [18] cited the preprint version of [22], but it is illustrative to see how its authors defended their priority in this work, which is an indication of the importance they were giving to it.

Six years later, Demmel and Koev cited in their paper [7] the papers [18,22,41], related to the solution of structured linear systems. Curiously, although the title of [41] contains the term Björck–Pereyra, it is not related to bidiagonal factorization, while [18] uses it. As explained in the introduction of [18] (the editor needed to know if there was a new algorithm), “*the main difference between this algorithm and the algorithm proposed in [41] comes from the fact that the algorithm presented here uses the factorization in terms of bidiagonal matrices, extending to the case of Cauchy–Vandermonde matrices the results about the factorization of Cauchy matrices given in [22]. In contrast, in [41] the main tool is the construction of the triangular matrices L and U by using the connection between a Cauchy–Vandermonde linear system and the rational interpolation problem associated with it*”.

Also, it must be observed that [7] does not contain a reference to [37], since it only deals with linear system solving. The great idea of Koev, presented in [8] was to extend the use of the bidiagonal factorization to some other linear algebra problems, like eigenvalue and singular value computation. In this fundamental work [8] the author acknowledges the inspiration received from [37]. Although the term $\mathcal{BD}(A)$ (and its precise meaning) is introduced in [8], Koev writes that “*by using the Cauchy–Binet identity, Demmel et al. [37] established that all minors of a TN matrix are determined accurately by the entries of $\mathcal{BD}(A)$* .” On the other hand, the paper by Fernando and Parlett [6], which is the origin of the algorithm `DLASQ1` (a relevant issue in the work of Koev) was an important reference in [37].

5. The Rectangular Case

Linear system solving and eigenvalue computation correspond to the case in which A is a square matrix but, as seen in [9], this bidiagonal decomposition also exists for rectangular matrices. For instance, an algorithm for computing it for the case of Bernstein–Vandermonde matrices is presented in [35] (see algorithm `TNBDBVR` in [27]).

Let us recall that in the square case, the matrices must be nonsingular. For the extension to the rectangular case, we must take into account the following comment in the Introduction of [9]:

"The existence and uniqueness of the bidiagonal decomposition is critical to the design of our algorithms. Therefore, we restrict the class of totally nonnegative matrices under consideration to only those that are *leading contiguous submatrices of square nonsingular totally nonnegative matrices*".

These rectangular matrices arise in a natural way when solving least squares problems. In this context, and returning to our guide book [1], we find this subject studied in Chapter 52, entitled *Least Squares Solution of Linear Systems* [42]. In Sections 52.4 and 52.5, mainly following the classical book of Björck [43], the authors recall the use of the QR factorization of A (of size $m \times n$ with $m \geq n$), which leads to solving a (square) linear system $Rx = Q^T b$. As read in [42], the algorithm that is least sensitive to the influence of rounding errors is based on the QR factorization of A , as first suggested by G. H. Golub in [44].

For this purpose, P. Koev includes in [9] a new gem for computing with totally positive matrices: the QR factorization of a totally positive matrix A . Starting from $B = \mathcal{BD}(A)$ (the bidiagonal decomposition of A), the algorithm TNQR computes Q and $\mathcal{BD}(R)$ (the bidiagonal decomposition of the triangular factor R). This algorithm has been used in [45] to solve least squares problems by using the Bernstein basis, including the accurate computation of the projection matrix (the *hat matrix* of statistics). An extension to the bivariate setting has recently been carried out in [46].

An important concept in the extension to the bivariate setting is the *Kronecker product* of two matrices (which has recently been analyzed in connection with optimal properties of the tensor product of B-bases in [47]), although in [46] a *generalized Kronecker product* is used. In [47] optimal properties of collocation matrices of the tensor product of normalized B-bases are proved, by extending to the bivariate case the results of [48].

For the sake of completeness, we recall following [48], the basic notions on B-bases, an important concept in Computer Aided Geometric Design. A system of functions is TP (totally positive) when all its collocations matrices are TP. In CAGD, the functions u_0, \dots, u_n also satisfy $\sum_{i=0}^n u_i(t) = 1$ for all $t \in I$ (i.e., the system (u_0, \dots, u_n) is *normalized*), and a normalized TP system is denoted by NTP. It is known that shape-preserving representations are associated with NTP bases. A TP basis (b_0, \dots, b_n) of a space of functions \mathcal{U} defined on a real interval I is a *B-basis* of \mathcal{U} if for any TP basis (u_0, \dots, u_n) of \mathcal{U} we can write $(u_0, \dots, u_n) = (b_0, \dots, b_n)A$, with A being a real nonsingular TP matrix of order $n + 1$. Given a space with an NTP basis, there exists a unique TP basis of the space with optimal shape-preserving properties, which is the *normalized B-basis* of the space. An important normalized B-basis is the Bernstein basis of the space of polynomials of degree less than or equal to n on $[0, 1]$.

Finally, we observe that another important situation where the bidiagonal factorization for the rectangular case is used is the computation of singular values of rectangular matrices (one of its applications being the computation of the spectral condition number of a matrix), as seen in Section 7 of [35], and also the computation of the Moore–Penrose inverse, as carried out in [13].

6. The Tridiagonal Case

In this section, we briefly consider the case in which the totally positive matrix A is also symmetric and tridiagonal. An important example of symmetric and tridiagonal matrices are the Jacobi matrices associated with a family of orthogonal polynomials, and in this context, the eigenvalues of the Jacobi matrices are the zeros of the corresponding orthogonal polynomials ([49,50]), which consequently can be computed with high relative accuracy starting from an accurate $\mathcal{BD}(A)$.

An example of this situation (where the Jacobi matrices are totally positive) has been analyzed in [51], and more recently in [52]. Although in [51] Gaussian quadrature formulae were considered, only the nodes were computed in that paper, since the accurate computation of the eigenvectors of the corresponding Jacobi matrices (necessary to compute the weights) could not be achieved with MATLAB.

While the LAPACK subroutine DLASQ1 (used by the algorithm TNEigenValues) provides high relative accuracy in the computation of the nodes, for the computation of the weights (presented in Section 4 of [52]) the LAPACK subroutine DBDSQR is used to compute the right singular vectors of a bidiagonal matrix. The initial stage is again the accurate computation of the bidiagonal factorization of the corresponding totally positive Jacobi matrices.

An interesting aspect of this symmetric and tridiagonal case is the fact that now Neville elimination is the same as Gaussian elimination, and the bidiagonal decomposition of A is precisely $A = LDL^T$, where L is now not only lower triangular but also bidiagonal. Although the author does not pay explicit attention to total positivity, the algorithmic advantages of using the LDL^T factorization have been shown by B. N. Parlett in [53].

The problem of accurate computations with tridiagonal matrices by using the bidiagonal factorization has also been addressed in a different context in [54].

7. Recent Extensions

Recent research on the application of the bidiagonal factorization of totally positive matrices deals with matrices that are not collocation matrices but different classes of matrices, namely *Wronskian matrices* and *Gram matrices*, defined as follows (see [55]).

Let (u_0, \dots, u_n) be the basis of a space U of functions defined on a real interval I . If the space U is formed by n -times continuously differentiable functions and $t \in I$, the *Wronskian matrix* at t is defined by

$$W(u_0, \dots, u_n)(t) := (u_{j-1}^{(i-1)}(t))_{1 \leq i, j \leq n+1},$$

where $u^{(i)}(t), i \leq n$, denotes the i th derivative of u at t .

Now, let U be a Hilbert space of functions on the interval $[0, T]$, $T \leq +\infty$, with a given inner product

$$\langle u, v \rangle = \int_0^T u(t)v(t)dt,$$

defined for any $u, v \in U$. Then, given linearly independent functions v_0, \dots, v_n in U , the corresponding *Gram matrix* is the symmetric matrix G given by

$$G(v_0, \dots, v_n) := (\langle v_{i-1}, v_{j-1} \rangle)_{1 \leq i, j \leq n+1}.$$

Some applications of the Gram matrices of the Bernstein basis are considered in [56]. On the other hand, although Wronskian and Gram matrices are very different, interestingly several applications of these matrices are presented together in [57]. The use of the bidiagonal decomposition to achieve accurate computations with different classes of totally positive Wronskian and Gram matrices has been presented, for instance, in [55,58–60].

Finally, it must be remarked that another recent line of research related to our subject is to consider the *singular case*, i.e., to allow matrices of arbitrary rank. An example of this work is the paper [19], in which new explicit expressions for bidiagonal decompositions of totally positive Vandermonde and related matrices are derived.

8. Conclusions

The starting point of our work has been the paper of Björck and Pereyra of 1970 [21], where they showed that the Newton–Horner algorithm for solving a Vandermonde system associated with polynomial interpolation could be interpreted in connection with a factorization of the inverse of the Vandermonde matrix as a product of bidiagonal matrices.

Our main goal has been to show how this approach to solving linear systems has been extended to solve several other linear algebra problems, such as eigenvalue and singular value computation, or least squares problems. The main tool is the use of the bidiagonal decomposition of the corresponding matrix (not only of its inverse, as in the Björck–Pereyra algorithm), stored as $B = \mathcal{BD}(A)$.

This idea, first presented in the work of Koev [8] (see also [20]), along with the availability of his algorithms, has encouraged the search for new algorithms for the bidiagonal decomposition of various classes of totally positive structured matrices.

It must be remarked that although a superficial reading (or careless writing) of our references could suggest that Neville elimination is the method to construct those algorithms, the fact is that Neville elimination is a fundamental *theoretical tool* but not the algorithmic tool (see Section 5 in [13], where a necessary comment on the paper [61] is included). For achieving high relative accuracy, the algorithms must be adapted to the specific structure of each class of matrices.

Funding: This research has been partially supported by Spanish Research Grants PID2022-138569NB-I00 and RED2022-134176-T (MCI/AEI). J.-J. Martínez is member of the Research Group ASYNACS (Ref. CT-CE2019/683) of Universidad de Alcalá.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Hogben, L. (Ed.) *Handbook of Linear Algebra*, 2nd ed.; CRC Press: Boca Raton, FL, USA, 2014.
- Stewart, M. Fast Algorithms for Structured Matrix Computations. Chapter 62. In *Handbook of Linear Algebra*, 2nd ed.; Hogben, L., Ed.; CRC Press: Boca Raton, FL, USA, 2014.
- Higham, N.J. Error analysis of the Björck–Pereyra algorithms for solving Vandermonde systems. *Numer. Math.* **1987**, *50*, 613–632. [[CrossRef](#)]
- Drmač, Z. Computing Eigenvalues and Singular Values to High Relative Accuracy. Chapter 59. In *Handbook of Linear Algebra*, 2nd ed.; Hogben, L., Ed.; CRC Press: Boca Raton, FL, USA, 2014.
- Demmel, J.; Kahan, W. Accurate singular values of bidiagonal matrices. *SIAM J. Sci. Stat. Comput.* **1990**, *11*, 873–912. [[CrossRef](#)]
- Fernando, K.; Parlett, B. Accurate singular values and differential qd algorithms. *Numer. Math.* **1994**, *67*, 191–229. [[CrossRef](#)]
- Demmel, J.; Koev, P. The accurate and efficient solution of a totally positive generalized Vandermonde linear system. *Siam J. Matrix Anal. Appl.* **2005**, *27*, 142–152. [[CrossRef](#)]
- Koev, P. Accurate eigenvalues and SVDs of totally nonnegative matrices. *SIAM J. Matrix Anal. Appl.* **2005**, *27*, 1–23. [[CrossRef](#)]
- Koev, P. Accurate computations with totally nonnegative matrices. *SIAM J. Matrix Anal. Appl.* **2007**, *29*, 731–751. [[CrossRef](#)]
- Fallat, S.M. Totally Positive and Totally Nonnegative Matrices. Chapter 29. In *Handbook of Linear Algebra*, 2nd ed.; Hogben, L., Ed.; CRC Press: Boca Raton, FL, USA, 2014.
- Golub, G.H.; Van Loan, C.F. *Matrix Computations*, 4th ed.; Johns Hopkins University Press: Baltimore, MD, USA, 2013.
- Björck, A. *Numerical Methods in Matrix Computations*; Texts in Applied Mathematics, Volume 59; Springer International Publishing: Cham, Switzerland, 2015.
- Marco, A.; Martínez, J.-J. Accurate computation of the Moore–Penrose of strictly totally positive matrices. *J. Comput. Appl. Math.* **2019**, *350*, 299–308. [[CrossRef](#)]
- Fallat, S.M.; Johnson, C.R. *Totally Nonnegative Matrices*; Princeton University Press: Princeton, NJ, USA; Oxford, UK, 2011.
- Pinkus, A. *Totally Positive Matrices*; Cambridge Tracts in Mathematics, Num. 181; Cambridge University Press: Cambridge, UK, 2010.
- Gasca, M.; Micchelli, C.A. (Eds.) *Total Positivity and Its Applications* (Jaca, 1994); Kluwer Academic Publishers: Dordrecht, The Netherlands, 1996.
- Gasca, M.; Peña, J.M. On Factorizations of Totally Positive Matrices. In *Total Positivity and Its Applications*; Gasca, M., Michelli, C.A., Eds.; Kluwer Academic Publishers: Dordrecht, The Netherlands, 1996; pp. 109–130.
- Martínez, J.-J.; Peña, J.M. Factorizations of Cauchy–Vandermonde matrices. *Linear Algebra Appl.* **1998**, *284*, 229–237. [[CrossRef](#)]
- Delgado, J.; Koev, P.; Marco, A.; Martínez, J.-J.; Peña, J.M.; Persson, P.-O.; Spasov, S. Bidiagonal decompositions of Vandermonde-type matrices of arbitrary rank. *J. Comput. Appl. Math.* **2023**, *426*, 115064. [[CrossRef](#)]
- Demmel, J.; Dumitriu, I.; Holtz, O.; Koev, P. Accurate and efficient expression evaluation and linear algebra. *Acta Numer.* **2008**, *17*, 87–145. [[CrossRef](#)]
- Björck, A.; Pereyra, V. Solution of Vandermonde systems of equations. *Math. Comp.* **1970**, *24*, 893–903. [[CrossRef](#)]
- Boros, T.; Kailath, T.; Olshevsky, V. A fast parallel Björck–Pereyra-type algorithm for solving Cauchy linear equations. *Linear Algebra Appl.* **1999**, *302/303*, 265–293. [[CrossRef](#)]
- Higham, N.J. Stability analysis of algorithms for solving confluent Vandermonde-like systems. *SIAM J. Matrix Anal. Appl.* **1990**, *11*, 23–41. [[CrossRef](#)]
- Higham, N.J. *Accuracy and Stability of Numerical Algorithms*, 2nd ed.; SIAM: Philadelphia, PA, USA, 2002.
- Bella, T.; Eidelman, Y.; Gohberg, I.; Koltracht, I.; Olshevsky, V. A Björck–Pereyra-type algorithm for Szegő–Vandermonde matrices based on properties of unitary Hessenberg matrices. *Linear Algebra Appl.* **2007**, *420*, 634–647. [[CrossRef](#)]

26. Marco, A.; Martínez, J.-J.; Peña, J.M. Accurate bidiagonal decomposition of totally positive Cauchy–Vandermonde matrices and applications. *Linear Algebra Appl.* **2017**, *517*, 63–84. [[CrossRef](#)]
27. Koev, P. Available online: <https://math.mit.edu/~plamen/software/TNTool.html> (accessed on 1 April 2024).
28. Peña, J.M. Accurate Computations and Applications of Some Classes of Matrices. In *Computational Mathematics, Numerical Analysis and Applications*; Mateos, M., Alonso, P., Eds.; SEMA SIMAI Springer Series; Springer: Cham, Switzerland, 2017; Volume 13.
29. Barreras, A.; Peña, J.M. Accurate computations of matrices with bidiagonal decomposition using methods for totally positive matrices. *Numer. Linear Algebra Appl.* **2013**, *20*, 413–424. [[CrossRef](#)]
30. Gasca, M.; Mühlbach, G. Elimination techniques: From extrapolation to totally positive matrices and CAGD. *J. Comput. Appl. Math.* **2000**, *122*, 37–50. [[CrossRef](#)]
31. Gasca, M.; Mühlbach, G. Generalized Schur complements and a test for total positivity. *Appl. Numer. Math.* **1987**, *3*, 215–232. [[CrossRef](#)]
32. Mühlbach, G.; Gasca, M. A test for strict total positivity via Neville elimination. In *Current trends in Matrix Theory (Auburn, Ala., 1986)*; Uhlig, F., Grone, R., Eds.; North Holland: Amsterdam, The Netherlands, 1987; pp. 225–232.
33. Gasca, M.; Peña, J.M. Total positivity and Neville elimination. *Linear Algebra Appl.* **1992**, *165*, 25–44. [[CrossRef](#)]
34. Gasca, M.; Peña, J.M. A matricial description of Neville elimination with applications to total positivity. *Linear Algebra Appl.* **1994**, *202*, 33–45. [[CrossRef](#)]
35. Marco, A.; Martínez, J.-J. Accurate computations with totally positive Bernstein–Vandermonde matrices. *Electron. J. Linear Algebra* **2013**, *26*, 357–380. [[CrossRef](#)]
36. Khiar, Y.; Mainar, E.; Royo-Amondarain, E.; Rubio, B. On the accurate computation of the Newton form of the Lagrange interpolant. *arXiv* **2023**, arXiv:2312.14483v1.
37. Demmel, J.; Gu, M.; Eisenstat, S.; Slapničar, I.; Veselić, K.; Drmač, Z. Computing the singular value decomposition with high relative accuracy. *Linear Algebra Appl.* **1999**, *299*, 21–80. [[CrossRef](#)]
38. Fallat, S.M. Bidiagonal factorizations of totally nonnegative matrices. *Am. Math. Mon.* **2001**, *109*, 697–712. [[CrossRef](#)]
39. Delgado, J.; Peña, G.; Peña, J.M. Accurate and fast computations with positive extended Schoenmakers-Coffey matrices. *Numer. Linear Algebra Appl.* **2016**, *23*, 1023–1031. [[CrossRef](#)]
40. Delgado, J.; Peña, G.; Peña, J.M. Accurate and fast computations with Green matrices. *Appl. Math. Lett.* **2023**, *45*, 108778. [[CrossRef](#)]
41. Martínez, J.-J.; Peña, J.M. Fast algorithms of Björck–Pereyra type for solving Cauchy–Vandermonde linear systems. *Appl. Numer. Math.* **1998**, *26*, 343–352. [[CrossRef](#)]
42. Hansen, P.C.; Nielsen, H.B. Least Squares Solution of Linear Systems. Chapter 52. In *Handbook of Linear Algebra*, 2nd ed.; Hogben, L., Ed.; CRC Press: Boca Raton, FL, USA, 2014.
43. Björck, A. *Numerical Methods for Least Squares Problems*; SIAM: Philadelphia, PA, USA, 1996.
44. Golub, G.H. Numerical methods for solving least squares problems. *Numer. Math.* **1965**, *7*, 206–216. [[CrossRef](#)]
45. Marco, A.; Martínez, J.-J. Ajuste polinómico por mínimos cuadrados usando la base de Bernstein. *Gac. RSME* **2015**, *18*, 135–153.
46. Marco, A.; Martínez, J.-J.; Viaña, R. Least squares problems involving generalized Kronecker products and application to bivariate polynomial regression. *Numer. Algorithms* **2019**, *82*, 21–39. [[CrossRef](#)]
47. Delgado, J.; Orera, H.; Peña, J.M. Optimal properties of tensor product of B-bases. *Appl. Math. Lett.* **2021**, *121*, 107473. [[CrossRef](#)]
48. Delgado, J.; Peña, J.M. Extremal and optimal properties of B-bases collocation matrices. *Numer. Math.* **2020**, *146*, 105–118. [[CrossRef](#)]
49. Gautschi, W. *Orthogonal Polynomials. Computation and Approximation*; Oxford University Press: Oxford, UK, 2004.
50. Martínez, J.-J. Polinomios ortogonales, cuadratura gaussiana y problemas de valores propios. In *Margarita Mathematica en memoria de José Javier (Chicho) Guadalupe Hernández*; Español, L., Varona, J.L., Eds.; Servicio de Publicaciones, Universidad de La Rioja: Logroño, Spain, 2001; pp. 595–606.
51. Marco, A.; Martínez, J.-J. A total positivity property of the Marchenko-Pastur law. *Electron. J. Linear Algebra* **2015**, *30*, 106–117. [[CrossRef](#)]
52. Marco, A.; Martínez, J.-J.; Viaña, R. Accurate computations with totally positive matrices applied to the computation of Gaussian quadrature formulae. *Electron. J. Linear Algebra* **2022**, *38*, 777–791. [[CrossRef](#)]
53. Parlett, B.N. For tridiagonals T replace T with LDL^T . *J. Comput. Appl. Math.* **2000**, *123*, 117–130. [[CrossRef](#)]
54. Delgado, J.; Orera, H.; Peña, J.M. Characterizations and accurate computations for tridiagonal Toeplitz matrices. *Linear Multilinear Algebra* **2021**, *70*, 4508–4527. [[CrossRef](#)]
55. Mainar, E.; Peña, J.M.; Rubio, B. Accurate computations with matrices related to bases $\{t^i e^{\lambda t}\}$. *Adv. Comput. Math.* **2022**, *48*, 38. [[CrossRef](#)]
56. Lu, L. Gram matrix of Bernstein basis: Properties and applications. *J. Comput. Appl. Math.* **2015**, *280*, 37–41. [[CrossRef](#)]
57. Hartwig, R.E. Applications of the Wronskian and Gram Matrices of $\{t^i e^{\lambda t}\}$. *Linear Algebra Appl.* **1982**, *43*, 229–241. [[CrossRef](#)]
58. Mainar, E.; Peña, J.M.; Rubio, B. Accurate computations with Wronskian matrices. *Calcolo* **2021**, *58*, 1. [[CrossRef](#)]
59. Mainar, E.; Peña, J.M.; Rubio, B. Accurate and efficient computations with Wronskian matrices of Bernstein and related bases. *Numer. Linear Algebra Appl.* **2022**, *29*, e2423. [[CrossRef](#)]

60. Mainar, E.; Peña, J.M.; Rubio, B. Total positivity and accurate computations with Gram matrices of Bernstein bases. *Numer. Algorithms* **2022**, *91*, 841–859. [[CrossRef](#)]
61. Huang, R.; Zhu, L. Componentwise backward error analysis of Neville elimination. *Linear Algebra Appl.* **2014**, *451*, 33–48. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.