Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 42, pp. 58-83, February 2026.

ACCURATE COMPUTATION OF ALL EIGENVALUES OF A TOTALLY NONNEGATIVE MATRIX BY THE CAUCHON ALGORITHM*

FATIMA RASHEED†, MOHAMMAD ADM†, AND JÜRGEN GARLOFF‡§

Abstract. Totally nonnegative matrices are considered, i.e., matrices having all their minors nonnegative. Any $n \times n$ nonsingular totally nonnegative matrix can be represented as the product of $2n-1$ entry-wise nonnegative bidiagonal matrices. The n^2 nontrivial entries of this factorization parameterize the set of the nonsingular totally nonnegative matrices. This bidiagonal factorization has been used by Plamen Koev in *SIAM J. Matrix Anal. Appl.* 27, pp. 1–23, 2005, to design an algorithm for the computation of all the eigenvalues of nonsingular totally nonnegative matrices with high relative accuracy. In this paper, a different approach is employed: A matrix is used, which could be obtained by running the Cauchon algorithm, but for the sake of high relative accuracy this matrix is computed from the bidiagonal factorization. The effect of elementary operations applied to reduce this matrix to tridiagonal form is determined. It is shown that the computations can be performed without any subtraction of numbers of equal sign. This provides the basis for an algorithm needing $O(n^3)$ arithmetic operations for the computation of all the eigenvalues of a nonsingular totally nonnegative matrix with guaranteed high relative accuracy, independently of the condition number of the given problem.

Key words. Totally positive matrix, Totally nonnegative matrix, High relative accuracy, Bidiagonal factorization, Cauchon algorithm, Restoration algorithm, Eigenvalue.

AMS subject classifications. 15B48, 65F15.

1. Introduction.

The development of algorithms for computing eigenvalues, eigenvectors, singular values, the inverse matrix, and related quantities with high relative accuracy has found increasing interest, see the survey article [24]. Here, it is meant by that an algorithm can be performed *with high relative accuracy* (or *accurately* for short in this paper) that the relative distance of the computed solution from the exact solution is bounded by a constant times the unit roundoff, provided that initial data are exact. The main source of the loss of relative accuracy is the cancellation of significant digits when floating point numbers of equal signs and approximate magnitude are subtracted. In contrast, the addition of numbers of equal signs, the multiplication, division, and taking the square root are not critical. Therefore, to achieve the goal to perform an algorithm accurately, any subtraction has to be avoided throughout the computations. Obviously, this can be accomplished in a direct way only in simplest cases. Therefore, we cannot expect to accurately compute, e.g., all the eigenvalues of a matrix, by only using its entries. Fortunately, for some classes of matrices, more suitable parametrizations can be employed in which the computations can be performed subtraction-free and hence accurately.

Such classes are formed by the totally positive and the nonsingular totally nonnegative matrices. A matrix is called totally positive (abbreviated TP henceforth) and totally nonnegative (TN) if all its minors are positive and nonnegative, respectively. These matrices arise in a remarkable variety of ways within

---

*Received by the editors on September 4, 2024. Accepted for publication on January 19, 2026. Handling Editor: Dario Bini. Corresponding Author: Jürgen Garloff.

†Department of Applied Mathematics and Physics, Palestine Polytechnic University, Hebron, Palestine (176022@ppu.edu, moh_95@ppu.edu)

‡Department of Mathematics and Statistics, University of Konstanz, Konstanz, Germany

§HTWG Konstanz – University of Applied Sciences, Faculty for Computer Science, Konstanz, Germany (garloff@htwg-konstanz.de).

mathematics and in many areas of its applications. These include differential and integral equations, probability, algebraic and enumerative combinatorics, approximation theory, Lie theory, computer-aided geometric design, and mechanics. For applications and for properties of the TN matrices, see the monographs [8, 10, 15, 22, 26]. We mention only the sequel relevant property that all eigenvalues of a TN matrix A are real and nonnegative. If A is furthermore nonsingular and irreducible (it is then termed *oscillatory*), in particular, when A is TP, then they are in addition distinct.

A nonsingular TN matrix A can be represented as a product of entry-wise nonnegative bidiagonal matrices [11],

$$A = L^{(1)} \dots L^{(n-1)} \cdot D \cdot U^{(n-1)} \dots U^{(1)}, \tag{1.1}$$

where D is a diagonal matrix and $L^{(k)}$ and $U^{(k)}$ are lower and upper unit bidiagonal matrices, $k = 1, \dots, n-1$, see Subsection 3.2 below. The n^2 entries in the bidiagonal factorization (1.1), which are the only ones which can be nonzero, parameterize the set of the nonsingular TN matrices. Koev [16], [17] developed a procedure based on (1.1) requiring $O(n^3)$ arithmetic operations to accurately compute all the eigenvalues and the singular values of a nonsingular TN matrix. In contrast, by traditional algorithms only the largest eigenvalues and largest singular values of an ill-conditioned TN matrix can be computed with guaranteed relative accuracy, whereas the smallest eigenvalues and singular values may be computed with no relative accuracy at all, see [16, Section 1]. Often they are the only quantities of practical interest, e.g., in some applications in computer aided geometric design it is important to know the smallest eigenvalue of a row stochastic TN matrix [23].

An algorithm, very appropriate for working with TN matrices is the Cauchon algorithm (originally called *deleting derivations algorithm*) and the restoration algorithm [13]. In our paper, we are using condensed forms of both algorithms [3], [27, Section 2.1] which reduces the number of required arithmetic operations from $O(n^4)$ of the original algorithms to $O(n^3)$. We consider the following operations on a matrix A,

(i) multiplication of a row of A by a positive scalar,
(ii) multiplication of the ith row of A by the positive scalar x and addition of the resulting row to the next row multiplied by y, followed by the multiplication of row i by $1/y$.

We explicitly describe the effect of the operations (i) and (ii) under the performance of the Cauchon algorithm with a TP and nonsingular TN matrix. Although this algorithm itself is not subtraction-free, we prove that the computations can be performed subtraction-free. Therefore, we are able to proceed similarly as in [16] to compute accurately with $O(n^3)$ arithmetic operations *all* the eigenvalues of a nonsingular TN matrix, independently of the condition number of the problem.

The organization of our paper is as follows. In the next section, we introduce our notation. In Section 3, we recall the condensed form of the Cauchon and restoration algorithms and show how the matrix obtained by running the Cauchon algorithm can be computed from the bidiagonal factorization (1.1). In Section 4, we study the effect of the elementary operations (i) and (ii) when this algorithm is applied to a TP matrix and to a nonsingular TN matrix. The accurate computation of the eigenvalues of nonsingular TN matrices is the theme of Section 5, and in Section 6, we present a perturbation and error analysis of our approach. In Section 7, we draw some conclusions. Lengthy technical parts are delegated to Appendices A (proof of Theorem 4.2) and B (function `AddToNextRow`).

2. Notation. Let $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_p)$ and $\beta = (\beta_1, \beta_2, \ldots, \beta_q)$ be strictly increasing index sequences in $\{1, 2, \ldots, n\}$ and $\{1, 2, \ldots, m\}$, respectively. For $A = [a_{ij}] \in \mathbb{R}^{n\times m}$, the submatrix of A lying in the rows indexed by α and the columns indexed by β will be denoted by $A[\alpha|\beta] = A\left[\alpha_1, \alpha_2, \ldots, \alpha_p|\beta_1, \beta_2, \ldots, \beta_q\right]$. If $\alpha = \beta$, we write $A[\alpha]$. The *converse matrix* $A^\#$ of A is defined as

$$A^\# := P_n A P_m,$$

where P_n is the permutation matrix of order n defined by

$$(P_n)_{i,n+1-i} = 1, \text{ for } i = 1, \ldots n,$$

and similarly P_m. Then, the entries of $A^\#$ are given by $(A^\#)_{ij} = a_{n+1-i,m+1-j}$, $i = 1, \ldots, n$, $j = 1, \ldots, m$. The symbol T denotes transposition. $A \geq 0$ (> 0) means that all entries of A are nonnegative (positive).

A matrix is called *totally nonnegative* [abbreviated by TN] (respectively, *totally positive* [TP]) if all its minors are nonnegative (positive). If $A \in \mathbb{R}^{n\times m}$ is TN, then so are $A^\#$ and $(A^\#)^T$ [8, Theorem 1.4.1].

3. The Cauchon algorithm. In this section, we recall the condensed form of the Cauchon algorithm originally developed by Gérard Cauchon in [6]. This algorithm provides an efficient method for checking a given matrix for total nonnegativity or total positivity [4, 19]. Further applications include to find the bidiagonal factorization of a given nonsingular TN matrix [4] and to determine the rank of matrices that are connected with Cauchon diagrams [2].

3.1. Condensed form of the Cauchon algorithm and the restoration algorithm. In this subsection, we recall the condensed form of the Cauchon algorithm [3] and introduce its inverse algorithm, the condensed form of the restoration algorithm. Both algorithms reduce the number of arithmetic operations from $\mathcal{O}(n^4)$ of the original versions to $\mathcal{O}(n^3)$. Examples for the performance of both algorithms can be found in [27, Section 2.1].

Definition 3.1. *(Cauchon matrix) A matrix $A = [a_{ij}] \in \mathbb{R}^{n\times m}$ is called a* Cauchon matrix *if for all $i \in \{1, \ldots, n\}, j \in \{1, \ldots, m\}$, $a_{ij} = 0$ implies that $a_{kj} = 0$, $k = 1, \ldots, i-1$, or $a_{il} = 0$, $l = 1, \ldots, j-1$.*

Algorithm 3.1. *(Condensed form of the Cauchon algorithm) Let $A = [a_{ij}] \in \mathbb{R}^{n\times m}$.*

(i) Set $A^{(n)} := A$.

(ii) For $k = n-1, \ldots, 1$, define $A^{(k)} = [a^{(k)}_{ij}] \in \mathbb{R}^{n\times m}$ as follows:
For $j = 1, \ldots, m-1$,
set $u_j := \min\left\{h \in \{j+1, \ldots, m\} \,|\, a^{(k+1)}_{k+1,h} \neq 0\right\}$ (we set $u_j := \infty$, if this set is empty);
for $i = 1, \ldots, k$,

$$a^{(k)}_{ij} = \begin{cases} a^{(k+1)}_{ij} - \frac{a^{(k+1)}_{k+1,j}\, a^{(k+1)}_{i,u_j}}{a^{(k+1)}_{k+1,u_j}} & , \text{ if } \quad u_j < \infty, \\ a^{(k+1)}_{ij} & , \text{ if } \quad u_j = \infty. \end{cases}$$

For $i = k+1, \ldots, n, j = 1, \ldots, m$, and $i = 1, \ldots, k, j = m$,

$$a^{(k)}_{ij} = a^{(k+1)}_{ij}.$$

(iii) Put $\tilde{A} = A^{(1)}$.

We note that the entries in the last column and row of A remain unchanged through the run of the algorithm.

Theorem 3.1. *[2, 19] Assume that $A \in \mathbb{R}^{n\times m}$. Then, A is TP (TN) if and only if $\tilde{A} > 0$ ($\tilde{A} \geq 0$ and a Cauchon matrix). In addition, if $n = m$, A is nonsingular if and only if $\tilde{a}_{ii} > 0, i = 1, \ldots, n$.*

In [18, Proposition 3.5], an algorithm called restoration algorithm is introduced, which can be considered as the inverse algorithm of the Cauchon algorithm. In the following, we present the condensed form of this algorithm.

Algorithm 3.2. *(The condensed form of the restoration algorithm) Let $A = [a_{ij}] \in \mathbb{R}^{n\times m}$.*

(i) *Set $A^{(1)} := A$.*

(ii) *For $k = 2, \ldots, n$, define $A^{(k)} = [a_{ij}^{(k)}] \in \mathbb{R}^{n\times m}$ as follows:*
For $i = 1, \ldots, k-1$, $j = m-1, \ldots, 1$,
if $j = m-1$,

$$a_{i,m-1}^{(k)} = \begin{cases} a_{i,m-1}^{(k-1)} + \frac{a_{k,m-1}^{(k-1)} a_{i,m}^{(k-1)}}{a_{k,m}^{(k-1)}} & , \text{ if } \quad a_{k,m} \neq 0, \\ a_{i,m-1}^{(k-1)} & , \text{ if } \quad a_{k,m} = 0; \end{cases}$$

else, for $j = m-2, \ldots, 1$,
set $u_j := \min\left\{h \in \{j+1, \ldots, m-1\} \mid a_{k,h}^{(k-1)} \neq 0\right\}$ (we set $u_j := \infty$, if this set is empty);

$$a_{ij}^{(k)} = \begin{cases} a_{ij}^{(k-1)} + \frac{a_{k,j}^{(k-1)} a_{i,u_j}^{(k)}}{a_{k,u_j}^{(k-1)}} & , \text{ if } \quad u_j < \infty, \\ a_{ij}^{(k-1)} & , \text{ if } \quad u_j = \infty. \end{cases}$$

For $i = k, \ldots, n, j = 1, \ldots, m$, and $i = 1, \ldots, k-1, j = m$,

$$a_{ij}^{(k)} = a_{ij}^{(k-1)}.$$

(iii) *Put $\bar{A} = A^{(n)}$.*

We can represent each entry of the matrix $\tilde{A}$, which is obtained by the application of Algorithm 3.1 to a given nonsingular TN matrix A as a ratio of two minors formed from consecutive rows and columns.

Lemma 3.1. *[3, Proposition 2.10] Let $A \in \mathbb{R}^{n\times n}$ be nonsingular TN. Then the entries $\tilde{a}_{kj}$ of the matrix $\tilde{A}$ can be represented as*

$$\tilde{a}_{kj} = \frac{\det A\left[k, \ldots, k+w | j, \ldots, j+w\right]}{\det A\left[k+1, \ldots, k+w | j+1, \ldots, j+w\right]}, \quad k, j = 1, \ldots, n, \tag{3.1}$$

where $w = \min\{n-k, n-j\}$.

We denote the right-hand side of (3.1) by $A\left\langle k, \ldots, k+w | j, \ldots, j+w\right\rangle$ and employ the similar notation

$$A\langle k, k+2, \ldots, k+w \mid j+1, \ldots, j+w\rangle := \frac{\det A[k, k+2, \ldots, k+w \mid j+1, \ldots, j+w]}{\det A[k+2, \ldots, k+w \mid j+2, \ldots, j+w]}.$$

In Section 5, we will consider the computation of all eigenvalues of a nonsingular TN matrix. In passing, we deduce from (3.1) the following eigenvalue bound. Let A be an oscillatory matrix, see the Introduction. By (3.1), the representation

$$\tilde{a}_{11} = \frac{\det A}{\det A[2,\dots,n]},$$

follows. By using the strict interlacing of the eigenvalues of A and $A[2,\dots,n]$, see, e.g., [8, Theorem 5.5.2], we conclude that $\tilde{a}_{11}$ provides a strict lower bound for the largest eigenvalue and a strict upper bound for the smallest eigenvalue of A. By taking ratios of consecutive principal minors, we see that these strict bounds hold also for the other diagonal entries of $\tilde{A}$. Since nonsingular TN matrices may be approximated by TP matrices, see Remark 3.1 below, we obtain by continuity analogous non-strict bounds if we assume only that A is nonsingular TN.

3.2. Bidiagonal factorization of totally nonnegative matrices.

Investigating when a class of matrices admits a particular type of factorization is an important study. Often any intrinsic properties about a particular class of matrices can be deduced from certain factorization results. A factorization that is especially suited for the class of nonsingular TN matrices is elementary bidiagonal factorization. This allows one to obtain algorithms with high relative accuracy for the computation of singular values, eigenvalues, and inverses of totally nonnegative matrices [16, 17].

Definition 3.2. *(Elementary bidiagonal matrices) Let I_n denote the n-by-n identity matrix, and for $i, j = 1,\dots,n$, let E_{ij} denote the n-by-n matrix whose only nonzero entry is a 1 that occurs in position (i, j). For any real numbers s, t, define*

$$L_i(s) := I_n + sE_{i,i-1} \text{ and } U_j(t) := I_n + tE_{j-1,j},\ i, j = 2,\dots,n.$$

Such matrices are called elementary bidiagonal matrices.

According to [11], any nonsingular TN (TP) matrix $A \in \mathbb{R}^{n\times n}$ can be written as

$$A = L^{(1)} \dots L^{(n-1)} \cdot D \cdot U^{(n-1)} \dots U^{(1)}, \tag{3.2}$$

with $D = diag(d_1, d_2,\dots,d_n)$, and

$$L^{(r)} = L_{n-r+1}(l_1^{(r)}) L_{n-r+2}(l_2^{(r)}) \dots L_n(l_r^{(r)}),\ r = 1,\dots,n-1,$$

$$= \begin{array}{r} \\ \\ \text{n-r+1} \rightarrow \\ \\ \\ \\ \end{array} \begin{bmatrix} 1 & & & & \\ & \ddots & & \text{\Large 0} & \\ & l_1^{(r)} & 1 & & \\ & & l_2^{(r)} & 1 & \\ \text{\Large 0} & & & \ddots & \ddots \\ & & & l_r^{(r)} & 1 \end{bmatrix},$$

and

$$U^{(r)} = U_n(u_1^{(r)}) U_{n-1}(u_2^{(r)}) \dots U_{r+1}(u_r^{(r)}),\ r = 1,\dots,n-1,$$

$$
= \begin{array}{r} \\ \text{n-r} \rightarrow \\ \\ \\ \\ \\ \end{array}
\left[\begin{array}{cccccc} 1 & & & & & \\ & \ddots & u_r^{(r)} & & \text{\Large 0} & \\ & & \ddots & \ddots & & \\ & & & 1 & u_2^{(r)} & \\ & \text{\Large 0} & & & 1 & u_1^{(r)} \\ & & & & & 1 \end{array}\right],
$$

where $d_i > 0$, $i = 1, \dots, n$, and $l_j^{(r)}, u_j^{(r)} \geq 0 \ (> 0)$, $j = 1, \dots, r$.

If A is TP, then these numbers can be determined by application of Algorithm 3.1 to $G = (A^{\#})^T$ similarly as in [1, Section 3.4].

Algorithm 3.3. *(Computation of the bidiagonal factorization from $\tilde{G}$) Let A be a TP matrix. We find the d_i's, $l_j^{(r)}$'s, and $u_j^{(r)}$'s given in (3.2) as follows:*

(i) Run Algorithm 3.1 on $G = (A^{\#})^T$.
(ii) $l_1^{(r)} = \frac{\tilde{g}_{nr}}{\tilde{g}_{n,r+1}}$, $l_2^{(r)} = \frac{\tilde{g}_{n-1,r-1}}{\tilde{g}_{n-1,r}}$, $\dots$, $l_r^{(r)} = \frac{\tilde{g}_{n-r+1,1}}{\tilde{g}_{n-r+1,2}}$, $r = 1, \dots, n-1$;
$d_{ii} = \tilde{g}_{n+1-i,n+1-i}$, $i = 1, \dots, n$;
$u_1^{(r)} = \frac{\tilde{g}_{1,n-r+1}}{\tilde{g}_{2,n-r+1}}$, $u_2^{(r)} = \frac{\tilde{g}_{2,n-r+2}}{\tilde{g}_{3,n-r+2}}, \dots$, $u_r^{(r)} = \frac{\tilde{g}_{rn}}{\tilde{g}_{r+1,n}}$, $r = 1, \dots, n-1$.

If in the upper triangular part of $\tilde{G}$ one entry is zero, then the entry above it is zero, and if one entry is zero in the lower triangular part, then the entry to the left of it is zero. Therefore, in accordance with [11], the respective values in (ii) are set as zero, see [4, Section 4.3].

Conversely, we can employ the given bidiagonal factorization of A to get $\tilde{G}$. This can be accomplished by Algorithm 3.4 below as follows: Since the diagonal entries of D are supposed to be given, we know the diagonal entries of $\tilde{G}$. The entries in the lower triangular part of $\tilde{G}$ can be calculated as follows. In each row i, $i = 1, \dots, n-1$, we compute recursively, starting for $k = 1$ from the entry on the subdiagonal, the entries to the left of it in row i. In the upper triangular part, we proceed similarly by starting in each column i, $i = 1, \dots, n-1$, with the entry on the superdiagonal and calculate recursively the entries above it in column i.

Algorithm 3.4. *(Computation of $\tilde{G}$ from the bidiagonal factorization) Let A be a TP matrix and assume that the bidiagonal factorization (3.2) with the parameters d_{ii}, $l_j^{(r)}$, $u_j^{(r)}$'s is given. Then, we find the entries $\tilde{g}_{ij}$ of $\tilde{G}$ as follows:*

$$
\tilde{g}_{ii} = d_{n+1-i,\, n+1-i}, \ \ i = 1, \dots, n;
$$

$$
\left.\begin{array}{r} \tilde{g}_{n+1-i,\, r+1-i} = l_i^{(r)} \ \tilde{g}_{n+1-i,\, r+2-i}, \\[1ex] \tilde{g}_{i,\, n-r+i} = u_i^{(r)} \ \tilde{g}_{i+1,\, n-r+i}, \end{array}\right\} \begin{array}{c} r = 1, \dots, n-1, \\ i = 1, \dots, r. \end{array}
$$

Algorithm 3.4 requires $n^2 - n$ arithmetic operations and is subtraction-free.

To extend Algorithms 3.3 and 3.4 to the nonsingular TN case, we approximate the nonsingular TN matrix G by the TP matrix G_ϵ as follows: By the application of Algorithm 3.1, we get $\tilde{G}$. Then by Theorem 3.1, $\tilde{G}$ is a nonnegative Cauchon matrix with positive diagonal entries. For a zero entry in the upper triangular part of $\tilde{G}$, all entries in the same column above it are zero, and for a zero entry in the lower triangular part, all entries in the same row to the left of it are zero, too. We replace the zero entries in each column of the upper triangular part from bottom to top and in each row of the lower triangular part from right to left by increasing integer powers of a positive number ϵ. We call the resulting matrix $\tilde{G}_\epsilon$. We apply Algorithm 3.2 to $\tilde{G}_\epsilon$ and obtain by Theorem 3.1 the TP matrix G because all entries in $\tilde{G}_\epsilon$ are positive. Since $\tilde{G}_\epsilon$ tends to $\tilde{G}$ as ϵ tends to 0, G_ϵ tends to G. So we can approximate the nonsingular TN matrix G arbitrarily closely by the TP matrix G_ϵ.

Remark 3.1. *As a by-product, we have just proven that the set of the TP matrices is dense in the set of the nonsingular TN matrices. More generally, the set of the TP matrices is dense in the set of the TN matrices [28].*

After cancellation of common powers of ϵ, the denominators of the entries of G_ϵ do not contain ϵ. By letting ϵ tend to 0, the extension of Algorithms 3.3 and 3.4 to the nonsingular TN case follows.

4. Elementary operations and the Cauchon algorithm.

4.1. Elementary operations under performance of the Cauchon algorithm. In this section, we perform the following elementary operations on a matrix A:

1. multiplying a row by a positive scalar,
2. adding a positive multiple of a row to the next row.

Each of these elementary operations preserves the total nonnegativity and total positivity, see [26, Proposition 1.5]. In the following, we show how the entries of $\tilde{A}$ will be changed when performing the above elementary operations on a TP matrix A. After that, we will consider the case of a nonsingular TN matrix.

4.1.1. Case of a totally positive matrix. Multiplying a row by a positive scalar.

Theorem 4.1. *Let $A = [a_{kj}] \in \mathbb{R}^{n\times n}$ be a TP matrix, let $\tilde{A}$ be the matrix obtained from A by running Algorithm 3.1 on A, and let $B = [b_{kj}]$ be the matrix obtained from A by multiplying row i by a positive scalar x. When we apply Algorithm 3.1 on B, we get the matrix $\tilde{B}$ with entries given as follows:*

$$\tilde{b}_{kj} = \begin{cases} x\,\tilde{a}_{kj}, & \textit{for} \quad k = i, \\ \tilde{a}_{kj}, & \textit{for} \quad k \neq i, \end{cases}$$

$k, j = 1, 2, \ldots, n$. The matrix $\tilde{B}$ can be computed from $\tilde{A}$ with exactly n multiplications without performing any subtractions.

Proof. Since the matrix B is TP, its entries can be represented as, see Lemma 3.1,

$$\begin{aligned}\tilde{b}_{kj} &= \frac{\det B[k,\ldots,k+w|j,\ldots,j+w]}{\det B[k+1,\ldots,k+w|j+1,\ldots,j+w]} \\ &= B\left\langle k,\ldots,k+w|j,\ldots,j+w\right\rangle,\ w = \min\{n-k, n-j\}.\end{aligned}$$

To find the entries of $\tilde{B}$ in terms of the entries of $\tilde{A}$, we distinguish two cases:

Case 1: $k \neq i$. Since row i is involved both in the denominator and numerator of $\tilde{b}_{kj}$, or is not involved in both, we obtain

$$\begin{aligned}\tilde{b}_{kj} &= B\left\langle k,\ldots,k+w|j,\ldots,j+w\right\rangle\\ &= A\left\langle k,\ldots,k+w|j,\ldots,j+w\right\rangle\\ &= \tilde{a}_{kj}.\end{aligned}$$

Case 2: $k = i$. Since row i is only involved in the numerator of $\tilde{b}_{kj}$, we get

$$\begin{aligned}\tilde{b}_{kj} &= B\left\langle k,\ldots,k+w|j,\ldots,j+w\right\rangle\\ &= x\,A\left\langle k,\ldots,k+w|j,\ldots,j+w\right\rangle\\ &= x\ \tilde{a}_{kj}.\end{aligned}$$

□

Adding a positive multiple of a row to the next row.

Similarly as in [16], we consider the following operations. For $A = [a_{kj}] \in \mathbb{R}^{n\times n}$, we define the matrix $B = [b_{kj}]$ as follows:

$$b_{kj} := \begin{cases} \frac{1}{y}\ a_{kj}, & \text{for} \quad k = i,\\ x\ a_{k-1,j} + y\ a_{kj}, & \text{for} \quad k = i+1,\\ a_{kj}, & \text{for} \quad k \neq i \text{ and } k \neq i+1, \end{cases} \tag{4.1}$$

$j = 1, 2, \ldots, n$, and $x, y > 0$. That is, B is obtained by the multiplication of the ith row of A by the positive scalar x and addition of the resulting row to the next row multiplied by y, followed by the multiplication of the row i by $1/y$. We want to find the entries of the matrix $\tilde{B}$ in terms of the entries of the matrix $\tilde{A}$.

THEOREM 4.2. *Let $A \in \mathbb{R}^{n\times n}$ be TP, $x, y > 0$, let $\tilde{A}$ be the matrix obtained by running Algorithm 3.1 on A, and let B be defined as in (4.1). When we apply Algorithm 3.1 on B, we get the matrix $\tilde{B}$ with entries as follows:*

$$\tilde{b}_{kj} = \begin{cases} \dfrac{\tilde{a}_{kj}}{y + x\sum_{m=j+1}^{n}\frac{\tilde{a}_{km}}{\tilde{a}_{k+1,m}}}, & \textit{for} \quad k = i, 1 \le j < n,\\ \frac{1}{y}\ \tilde{a}_{kj}, & \textit{for} \quad k = i, j = n,\\ y\ \tilde{a}_{kj} + x\ \left(\tilde{a}_{k-1,j} + \tilde{a}_{kj}\sum_{m=j+1}^{n}\frac{\tilde{a}_{k-1,m}}{\tilde{a}_{km}}\right), & \textit{for} \quad k = i+1, 1 \le j < n,\\ x\ \tilde{a}_{k-1,j} + y\ \tilde{a}_{kj}, & \textit{for} \quad k = i+1, j = n,\\ \tilde{a}_{kj}, & \textit{for} \quad k \neq i \textit{ and}\\ & \qquad\quad k \neq i+1, 1 \le j \le n. \end{cases}$$

We delegate the lengthy proof to Appendix A.

In the following theorem, we rewrite the entries of the matrix $\tilde{B}$ that have been obtained in Theorem 4.2 so that we reduce the number of required arithmetic operations. For an example, for the recursive computation of the entries of $\tilde{B}$ see Example 4.1 below.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 42, pp. 58-83, February 2026.

Theorem 4.3. *The entries of the matrix $\tilde{B}$ in Theorem 4.2 can be represented as follows:*

$$\tilde{b}_{kj} = \begin{cases} x\,\tilde{a}_{k-1,j} + \frac{\tilde{a}_{kj}\,\tilde{b}_{k,j+1}}{\tilde{a}_{k,j+1}}, & \text{for} \quad k = i+1, 1 \le j < n, \\ x\,\tilde{a}_{k-1,j} + y\,\tilde{a}_{kj}, & \text{for} \quad k = i+1, j = n, \\ \frac{\tilde{a}_{kj}\tilde{a}_{k+1,j+1}}{\tilde{b}_{k+1,j+1}}, & \text{for} \quad k = i, 1 \le j < n, \\ \frac{1}{y}\,\tilde{a}_{kj}, & \text{for} \quad k = i, j = n, \\ \tilde{a}_{kj}, & \text{for} \quad k \neq i \text{ and } k \neq i+1, 1 \le j \le n. \end{cases}$$

Therefore, the matrix $\tilde{B}$ can be computed from $\tilde{A}$ with at most $5n-1$ arithmetic operations without performing any subtractions.

Proof. Concerning the number of arithmetic operations, it should be noted that for $j < n$, the factors of $\tilde{a}_{kj}$ are reciprocal for $k = i$ and $k = i+1$. By Theorem 4.2, the relations for $k < i$ or $k > i+1$ and $1 \le j \le n$ as well as for $k = i, i+1$ and $j = n$ hold.

In the following, we investigate the cases $k = i+1$ and $k = i$.

Case 1: $k = i+1$ and $1 \le j < n$.

From (A.5), we get

$$A\langle i, i+2, \ldots, i+1+w \mid n-\gamma, \ldots, n-\gamma+w\rangle = \frac{\tilde{b}_{i+1,n-\gamma} - y\,\tilde{a}_{i+1,n-\gamma}}{x}. \tag{4.2}$$

If $n-\gamma > i+1$, then for $j = n-\gamma-1$, we are in the upper triangular part or on the diagonal of the matrix $\tilde{B}$. From (A.7), we obtain

$$\tilde{b}_{i+1,n-\gamma-1} = y\,\tilde{a}_{i+1,n-\gamma-1} + x\,A\langle i, i+2, \ldots, i+2+w \mid n-\gamma-1, \ldots, n-\gamma+w\rangle. \tag{4.3}$$

Also, from (A.10), we get

$$\begin{aligned} &A\langle i, i+2, \ldots, i+2+w \mid n-\gamma-1, \ldots, n-\gamma+w\rangle \\ &= \tilde{a}_{i,n-\gamma-1} + \frac{\tilde{a}_{i+1,n-\gamma-1}}{\tilde{a}_{i+1,n-\gamma}} A\langle i, i+2, \ldots, i+1+w \mid n-\gamma, \ldots, n-\gamma+w\rangle \\ &= \tilde{a}_{i,n-\gamma-1} + \frac{\tilde{a}_{i+1,n-\gamma-1}}{\tilde{a}_{i+1,n-\gamma}} \frac{\tilde{b}_{i+1,n-\gamma} - y\,\tilde{a}_{i+1,n-\gamma}}{x}, \text{from (4.2)} \end{aligned} \tag{4.4}$$

Substitution of (4.4) into (4.3) results in

$$\begin{aligned} \tilde{b}_{i+1,n-\gamma-1} &= y\,\tilde{a}_{i+1,n-\gamma-1} + x\left(\tilde{a}_{i,n-\gamma-1} + \frac{\tilde{a}_{i+1,n-\gamma-1}}{\tilde{a}_{i+1,n-\gamma}} \frac{\tilde{b}_{i+1,n-\gamma} - y\,\tilde{a}_{i+1,n-\gamma}}{x}\right) \\ &= x\,\tilde{a}_{i,n-\gamma-1} + \frac{\tilde{a}_{i+1,n-\gamma-1}}{\tilde{a}_{i+1,n-\gamma}} \tilde{b}_{i+1,n-\gamma}. \end{aligned}$$

If $n-\gamma \le i+1$, then for $j = n-\gamma-1$, we are in the lower triangular part of $\tilde{B}$, and by (A.12), we obtain

$$\tilde{b}_{i+1,n-\gamma-1} = y\,\tilde{a}_{i+1,n-\gamma-1} + x\,A\langle i, i+2, \ldots, i+1+w \mid n-\gamma-1, \ldots, n-\gamma-1+w\rangle. \tag{4.5}$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 42, pp. 58-83, February 2026.

From (A.13), we get

$$\begin{aligned}
&A\langle i, i+2, \ldots, i+1+w \mid n-\gamma-1, \ldots, n-\gamma-1+w\rangle \\
&= \tilde{a}_{i,n-\gamma-1} + \frac{\tilde{a}_{i+1,n-\gamma-1}}{\tilde{a}_{i+1,n-\gamma}} A\langle i, i+2, \ldots, i+1+w \mid n-\gamma, \ldots, n-\gamma+w\rangle \\
&= \tilde{a}_{i,n-\gamma-1} + \frac{\tilde{a}_{i+1,n-\gamma-1}}{\tilde{a}_{i+1,n-\gamma}} \frac{\tilde{b}_{i+1,n-\gamma} - y\,\tilde{a}_{i+1,n-\gamma}}{x}, \text{ by (4.2).} \qquad (4.6)
\end{aligned}$$

Hence, by substituting (4.6) into (4.5), we conclude that

$$\tilde{b}_{i+1,j} = x\,\tilde{a}_{ij} + \frac{\tilde{a}_{i+1,j}\,\tilde{b}_{i+1,j+1}}{\tilde{a}_{i+1,j+1}}, \qquad 1 \leq j < n. \tag{4.7}$$

The expression in (4.7) involves no subtractions and requires $4n-4$ arithmetic operations such that together with the three arithmetic operations for the computation of $\tilde{b}_{i+1,n}$ the total number of arithmetic operations to find all the entries in the row $i+1$ is $4n-1$.

Case 2: $k = i$ and $1 \leq j < n$.

For $n-\gamma > i$, (A.15) reads

$$\tilde{b}_{i,n-\gamma-1} = \frac{\tilde{a}_{i,n-\gamma-1}}{y + x\,\dfrac{A\langle i, i+2, \ldots, i+1+w \mid n-\gamma, \ldots, n-\gamma+w\rangle}{\tilde{a}_{i+1,n-\gamma}}}.$$

and by substituting (4.2) into (A.15), we obtain

$$\tilde{b}_{i,n-\gamma-1} = \frac{\tilde{a}_{i,n-\gamma-1}\,\tilde{a}_{i+1,n-\gamma}}{\tilde{b}_{i+1,n-\gamma}}.$$

For $n-\gamma \leq i$, we proceed similarly as in the case $n-\gamma > i$ and get

$$\tilde{b}_{ij} = \frac{\tilde{a}_{ij}\,\tilde{a}_{i+1,j+1}}{\tilde{b}_{i+1,j+1}}, \qquad 1 \leq j < n. \tag{4.8}$$

The expression in (4.8) involves no subtractions and requires $2n-2$ arithmetic operations, which reduces to $n--1$ because the factors of $\tilde{a}_{ij}$ are reciprocal for $k=i$ and $k=i+1$. Together with the one arithmetic operation for $j=n$, the total number of arithmetic operations to find all the entries in the row i is n. Therefore, the total number of arithmetic operations for the computation of $\tilde{B}$ is $4n-1+n=5n-1$. ☐

Example 4.1. We consider the following Pascal matrix

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 6 & 10 \\ 1 & 4 & 10 & 20 \end{bmatrix}.$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 42, pp. 58-83, February 2026.

Application of Algorithm 3.1 yields

$$\tilde{A} = \begin{bmatrix} 1/4 & 1/6 & 1/4 & 1 \\ 1/6 & 2/10 & 6/10 & 4 \\ 1/4 & 6/10 & 1 & 10 \\ 1 & 4 & 10 & 20 \end{bmatrix}.$$

We multiply the third row of A by 2 and add to it the second row multiplied by 10. Then, we scale the second row by 1/2. The resulting matrix is

$$B = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1/2 & 1 & 3/2 & 2 \\ 12 & 26 & 42 & 60 \\ 1 & 4 & 10 & 20 \end{bmatrix}.$$

With $x = 10$ and $y = 2$, we compute recursively the entries of $\tilde{B}$ from the entries of $\tilde{A}$. The entries in the first and last row of $\tilde{A}$ do not change. First, we calculate the entries in the third row of $\tilde{B}$ and start with

$$\tilde{b}_{34} = x\,\tilde{a}_{24} + y\,\tilde{a}_{34} = 60.$$

The other entries are

$$\tilde{b}_{33} = x\,\tilde{a}_{23} + \frac{\tilde{a}_{33}\,\tilde{b}_{34}}{\tilde{a}_{34}} = 12,$$

$$\tilde{b}_{32} = x\,\tilde{a}_{22} + \frac{\tilde{a}_{32}\,\tilde{b}_{33}}{\tilde{a}_{33}} = \frac{46}{5},$$

$$\tilde{b}_{31} = x\,\tilde{a}_{21} + \frac{\tilde{a}_{31}\,\tilde{b}_{32}}{\tilde{a}_{32}} = \frac{11}{2}.$$

The entries in the second row of $\tilde{B}$ are calculated as follows:

$$\tilde{b}_{24} = \frac{1}{y}\,\tilde{a}_{24} = 2,$$

$$\tilde{b}_{23} = \frac{\tilde{a}_{23}\tilde{a}_{34}}{\tilde{b}_{34}} = \frac{1}{10},$$

$$\tilde{b}_{22} = \frac{\tilde{a}_{22}\tilde{a}_{33}}{\tilde{b}_{33}} = \frac{1}{60},$$

$$\tilde{b}_{21} = \frac{\tilde{a}_{21}\tilde{a}_{32}}{\tilde{b}_{32}} = \frac{1}{92}.$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 42, pp. 58-83, February 2026.

Therefore, the matrix $\tilde{B}$ is

$$\tilde{B} = \begin{bmatrix} 1/4 & 1/6 & 1/4 & 1 \\ 1/92 & 1/60 & 1/10 & 2 \\ 11/2 & 46/5 & 12 & 60 \\ 1 & 4 & 10 & 20 \end{bmatrix}.$$

4.1.2. Case of a nonsingular totally nonnegative matrix. In this section, we extend Theorem 4.2 to the case of a nonsingular totally nonnegative matrix A. We proceed as in Section 3.2 and arrive at the following formulae for the entries of $\tilde{B}$.

$$\tilde{b}_{kj} = \begin{cases} \dfrac{\tilde{a}_{kj}}{y+x\left(\sum_{m=j+1}^{n} \frac{\tilde{a}_{km}}{\tilde{a}_{k+1,m}}\right)}, & \text{for } k=i, i \leq j < n, \\ \dfrac{\tilde{a}_{kj}}{y+x\left(\sum_{m=j+1}^{n} \frac{\tilde{a}_{km}}{\tilde{a}_{k+1,m}}\right)}, & \text{for } k=i, 1 \leq j < i \\ & \quad \text{and } \tilde{a}_{k+1,j+1} \neq 0, \\ \dfrac{\tilde{a}_{kj}\ \tilde{a}_{k+1,j}}{y\ \tilde{a}_{k+1,j}+x\left(\sum_{m=j+1}^{n} \frac{\tilde{a}_{k+1,j}\ \tilde{a}_{km}}{\tilde{a}_{k+1,m}}\right)} = 0, & \text{for } k=i, 1 \leq j < i \\ & \quad \text{and } \tilde{a}_{k+1,j+1} = 0, \\ \frac{1}{y}\ \tilde{a}_{kj}, & \text{for } k=i, j=n, \\ y\ \tilde{a}_{kj} + x\ \left(\tilde{a}_{k-1,j} + \tilde{a}_{kj}\left(\sum_{m=j+1}^{n} \frac{\tilde{a}_{k-1,m}}{\tilde{a}_{km}}\right)\right), & \text{for } k=i+1, i \leq j < n, \\ y\ \tilde{a}_{kj} + x\ \left(\tilde{a}_{k-1,j} + \tilde{a}_{kj}\left(\sum_{m=j+1}^{n} \frac{\tilde{a}_{k-1,m}}{\tilde{a}_{km}}\right)\right), & \text{for } k=i+1, 1 \leq j < i \\ & \quad \text{and } \tilde{a}_{k,j+1} \neq 0, \\ y\ \tilde{a}_{kj} + x\ \left(\tilde{a}_{k-1,j} + \left(\sum_{m=j+1}^{n} \frac{\tilde{a}_{kj}\ \tilde{a}_{k-1,m}}{\tilde{a}_{km}}\right)\right) & \\ \qquad = x\ \tilde{a}_{k-1,j}, & \text{for } k=i+1, 1 \leq j < i \\ & \quad \text{and } \tilde{a}_{k,j+1} = 0, \\ x\ \tilde{a}_{k-1,j} + y\ \tilde{a}_{kj}, & \text{for } k=i+1, j=n, \\ \tilde{a}_{kj}, & \text{for } k \neq i \text{ and } k \neq i+1, \\ & \quad 1 \leq j \leq n. \end{cases} \tag{4.9}$$

In the following algorithm, we extend Theorem 4.3 to the nonsingular TN case.

Algorithm 4.1. *For a given nonsingular TN matrix $A \in \mathbb{R}^{n\times n}$, and given $x, y > 0$, and row index i, the following algorithm implements the extension of the calculation according to Theorem 4.3 and computes $\tilde{B}$ from $\tilde{A}$ with at most $5n-1$ arithmetic operations.*

(i) $\tilde{b}_{kj} := \tilde{a}_{kj}$, for $k \neq i, i+1, j = 1, \ldots, n$;
(ii) $\tilde{b}_{i+1,n} := x\ \tilde{a}_{in} + y\ \tilde{a}_{i+1,n}$ and $\tilde{b}_{in} = \frac{\tilde{a}_{in}}{y}$.
(iii) For $j = n-1, \ldots, 1$ put:

(a) If $\tilde{a}_{i+1,j+1} \neq 0$, then

$$\tilde{b}_{kj} = \begin{cases} x\ \tilde{a}_{k-1,j} + \frac{\tilde{a}_{kj}\ \tilde{b}_{k,j+1}}{\tilde{a}_{k,j+1}}, & \text{if} \quad k = i+1, \\ \frac{\tilde{a}_{kj}\tilde{a}_{k+1,j+1}}{\tilde{b}_{k+1,j+1}}, & \text{if} \quad k = i. \end{cases}$$

(b) If $\tilde{a}_{i+1,j+1} = 0$ and $\tilde{a}_{i+1,j} = 0$, then

$$\tilde{b}_{kj} = \begin{cases} x\ \tilde{a}_{k-1,j}, & \text{if} \quad k = i+1, \\ 0, & \text{if} \quad k = i. \end{cases}$$

(c) If $\tilde{a}_{i+1,j+1} = 0$ and $\tilde{a}_{i+1,j} \neq 0$, then
set $u_j := \min\{h \in \{j+2, \ldots, n\} | \tilde{a}_{i+1,h} \neq 0\}$ (we set $u_j := \infty$, if the set is empty).
If $u_j \neq \infty$, then

$$\tilde{b}_{kj} = \begin{cases} x\ \tilde{a}_{k-1,j} + \frac{\tilde{a}_{kj}\ \tilde{b}_{k,u_j}}{\tilde{a}_{k,u_j}}, & \text{if} \quad k = i+1, \\ \frac{\tilde{a}_{kj}\tilde{a}_{k+1,u_j}}{\tilde{b}_{k+1,u_j}}, & \text{if} \quad k = i. \end{cases}$$

If $u_j = \infty$, then

$$\tilde{b}_{kj} = \begin{cases} x\ \tilde{a}_{k-1,j} + y\ \tilde{a}_{kj}, & \text{if} \quad k = i+1, \\ \frac{\tilde{a}_{kj}}{y}, & \text{if} \quad k = i. \end{cases}$$

A MATLAB [21] function called `AddToNextRow` for the implementation of Algorithm 4.1 is given in Appendix B.

Example 4.2. Let A be the following tridiagonal matrix

$$A = \begin{bmatrix} 3 & 2 & 0 & 0 & 0 \\ 1 & 4 & 4 & 0 & 0 \\ 0 & 1 & 3 & 1 & 0 \\ 0 & 0 & 6 & 10 & 2 \\ 0 & 0 & 0 & 20 & 10 \end{bmatrix}.$$

Application of Algorithm 3.1 yields

$$\tilde{A} = \begin{bmatrix} 2 & 2 & 0 & 0 & 0 \\ 1 & 2 & 4 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 6 & 6 & 2 \\ 0 & 0 & 0 & 20 & 10 \end{bmatrix}.$$

We multiply the fourth row of A by 3 and add it to the third row multiplied by 4. After scaling the third row by 1/3, we obtain

$$B = \begin{bmatrix} 3 & 2 & 0 & 0 & 0 \\ 1 & 4 & 4 & 0 & 0 \\ 0 & 1/3 & 1 & 1/3 & 0 \\ 0 & 4 & 30 & 34 & 6 \\ 0 & 0 & 0 & 20 & 10 \end{bmatrix}.$$

Application of (4.9) with $x = 4$ and $y = 3$, yields

$$\tilde{B} = \begin{bmatrix} 2 & 2 & 0 & 0 & 0 \\ 1 & 2 & 4 & 0 & 0 \\ 0 & 1/5 & 6/11 & 1/3 & 0 \\ 0 & 4 & 30 & 22 & 6 \\ 0 & 0 & 0 & 20 & 10 \end{bmatrix}.$$

4.2. Elementary operations and bidiagonal factorization. In [16], it was studied how the bidiagonal factorization of a nonsingular TN matrix A changes when we multiply its ith column by a positive scalar x/y and add it to the column $i-1$, and scale column i and the resulting column $i-1$ by $1/y$ and $y > 0$, respectively. In matrix language, this is equivalent to form the product $A' := AJ_i(x, y)$, where $J_i(x, y)$ is the lower bidiagonal matrix having the entries from y, x, and $1/y$ in the positions $(i-1, i-1)$, $(i, i-1)$, and (i, i), respectively.

$$J_i(x, y) = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & y & & & \\ & & x & 1/y & & \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix}.$$

As an alternative to the procedure in [16], we form[*] $B := (A'^{\#})^T$ and apply Theorem 4.3 or Algorithm 4.1 to compute $\tilde{B}$. Finally, we obtain by Algorithm 3.3, the bidiagonal factorization of $\tilde{B}$. The worked example in [27, Section 3.2] illustrates the procedures in [16] as well as our approach.

5. Accurate computation of the eigenvalues of totally nonnegative matrices.

In this section, we present an algorithm for computing the eigenvalues of a nonsingular TN matrix A, given $\tilde{G}$, where $G = \left(A^{\#}\right)^T$, with high relative accuracy. The computation of $\tilde{G}$ by the Cauchon algorithm is inadequate, because this algorithm is not subtraction-free. Instead, we calculate $\tilde{G}$ from the bidiagonal factorization of A; this can be done accurately by Algorithm 3.4. For many kinds of structured matrices, a bidiagonal factorizations can be found, for a large list see [20]. Koev [16, Section 3] presents a necessary and sufficient condition for a TP matrix to allow an accurate computation of its bidiagonal factorization. This is possible, e.g., for Vandermonde and Cauchy matrices. We assume here that $\tilde{G}$ is accurately computed.

Let $A = [a_{ij}] \in \mathbb{R}^{n \times n}$ be a nonsingular TN matrix. To reduce A to a tridiagonal matrix T, we proceed similarly as the BR algorithm [12]. We start by introducing a zero in the $(n, 1)$ entry by subtracting a

---

[*] Note that by this transformation elementary operations applied to columns change to elementary operations on rows.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 42, pp. 58-83, February 2026.

multiple of the second column from the first. Then, we add the same multiple of the first row to the second to complete the similarity transformation. In the next step, we subtract a multiple of the second row from the first to create a zero in position $(1, n)$; then again, we complete the similarity transformation by adding the same multiple of the first column to the second. The zero in position $(n, 1)$ is not disturbed by the last operation. Next, we introduce zeros in positions $(n, 2)$ and $(2, n)$ in a similar manner, which does not disturb the zeros already created in positions $(n, 1)$ and $(1, n)$. We continue the same process until all but the last two entries of the last row and column are zero. Then, we apply the same process to the principal submatrix $A[1, \ldots, n-1]$ and so on until the matrix is reduced to the tridiagonal matrix T.

This process uses the following elementary operations:

1. subtracting of a positive multiple of a row (column) from the previous one;
2. adding of a positive multiple of a row (column) to the next one.

To preserve the accuracy, instead of performing these elementary operations on A, we perform the elementary operations on $\tilde{G}$ in such a way that subtractions are not required. For simplicity, we explain the procedure in the case that A is TP. Then, if $n = 4$, G is given by

$$G = \left(A^{\#}\right)^T = \begin{pmatrix} a_{44} & a_{34} & a_{24} & a_{14} \\ a_{43} & a_{33} & a_{23} & a_{13} \\ a_{42} & a_{32} & a_{22} & a_{12} \\ a_{41} & a_{31} & a_{21} & a_{11} \end{pmatrix},$$

and $\tilde{G}$ can be represented as, see Lemma 3.1,

$$\tilde{G} = \begin{pmatrix} \dfrac{\det A\left[1,2,3,4\right]}{\det A\left[1,2,3\right]} & \dfrac{\det A\left[1,2,3|2,3,4\right]}{\det A\left[1,2|2,3\right]} & \dfrac{\det A\left[1,2|3,4\right]}{a_{13}} & a_{14} \\[2ex] \dfrac{\det A\left[2,3,4|1,2,3\right]}{\det A\left[2,3|1,2\right]} & \dfrac{\det A\left[1,2,3\right]}{\det A\left[1,2\right]} & \dfrac{\det A\left[1,2|2,3\right]}{a_{12}} & a_{13} \\[2ex] \dfrac{\det A\left[3,4|1,2\right]}{a_{31}} & \dfrac{\det A\left[2,3|1,2\right]}{a_{21}} & \dfrac{\det A\left[1,2\right]}{a_{11}} & a_{12} \\[2ex] a_{41} & a_{31} & a_{21} & a_{11} \end{pmatrix}. \tag{5.1}$$

We start with $\tilde{G}$ as input. Since the value of a determinant of a matrix does not alter when one adds a multiple of one of its rows (or columns) to another row (or column), creating a zero in position $(4, 1)$ in A does not change the remaining entries in the first column of $\tilde{G}$ (note that with the exception of a_{31} and a_{41}, all minors in the first column of $\tilde{G}$ in (5.1) contain the second column of A). In the resulting matrix, we perform the addition of the multiple of its first row to the second according to Theorem 4.3. Similarly, creating a zero in position $(1, 4)$ does not alter the remaining entries in the first row.

In the extension to nonsingular TN matrices, note that by Theorem 3.1 $\tilde{G}$ is a nonnegative Cauchon matrix with positive diagonal entries. As a consequence, making a nonzero entry zero is always possible because the entry to the right of it in the same row in the lower triangular part or below it in the same column in the upper triangular part is nonzero.

Continuing in this way, see lines 3 to 16 in Algorithm 5.1 below, we obtain the matrix $\tilde{H}$ with $H = (T^{\#})^T$, where T is the desired tridiagonal matrix.

By Algorithm 3.3, we obtain from the matrix $\tilde{H}$ the bidiagonal factorization of the tridiagonal matrix T, which has only three nontrivial factors, see (3.2),

$$T = L^{(n-1)} \cdot D \cdot U^{(n-1)}.$$

The matrix T may not be symmetric, so we form as in [16] the symmetric tridiagonal matrix $\bar{T} := \bar{L}^{(n-1)} \cdot D \cdot \bar{U}^{(n-1)}$, where

$$\bar{l}_i^{(n-1)} = \bar{u}_i^{(n-1)} := \sqrt{l_i^{(n-1)} u_i^{(n-1)}}, i = 1, 2, \ldots, n-1.$$

The matrices T and $\bar{T}$ have the same eigenvalues since they have the same characteristic polynomial. We compute the eigenvalues of $\bar{T}$ accurately as the squares of the singular values of its Cholesky factor $C = D^{1/2}\bar{U}^{(n-1)}$, determined in lines 17 to 22 in Algorithm 5.1. Here, we use the function `bidsvd` [25], see line 23, which is based on the `LAPACK` [5] routine `dbdsqr` and requires $\mathcal{O}(n^2)$ arithmetic operations. The eigenvalues of A are the same as the eigenvalues of $\bar{T}$.

Algorithm 5.1. *Let $B = \tilde{G}$ be given, which is the matrix obtained from the bidiagonal factorization of A by Algorithm 3.4. The following algorithm computes the eigenvalues of A accurately.*

```
1  function TNEigenvalues(B)
2  n = size(B,1)
3  for i = n:-1:3
4    for k = 1:i-2
5      if B(i,k+1) ~= 0
6        x = B(i,k)/B(i,k+1);
7        B(i,k) = 0;
8        B = AddToNextRow(B,x,1,k);
9      end
10     if B(k+1,i) ~= 0
11       x = B(k,i)/B(k+1,i);
12       B(k,i) = 0;
13       B = (AddToNextRow((B)^T,x,1,k))^T;
14     end
15   end
16 end
17 for i = 1:n
18   D(i) = sqrt(B(n-i+1,n-i+1));
19 end
20 for i = 1:n-1
21   C(i) = sqrt(B(n-i,n-i+1)*B(n-i+1,n-i)/B(n-i+1,n-i+1));
22 end
23 TNEigenvalues = (bidsvd(D,C))^2;
```

The computation of $\tilde{H}$ from $\tilde{G}$ requires at most

$$2\sum_{i=n}^{3}\sum_{k=1}^{i-2}(5i-1) = \frac{10}{3}n^3 + \mathcal{O}(n^2),$$

arithmetic operations, see Theorem 4.3 and Algorithm 4.1, respectively, and Algorithm 5.1. Additional $\mathcal{O}(n^2)$ operations are needed for the calculation of $\tilde{G}$ from the bidiagonal factorization of A, see Algorithm 3.4, and for the final computation of the eigenvalues of the tridiagonal matrix T by using the function `bidsvd`. In contrast, Koev's approach requires at most $\frac{16}{3}n^3 + \mathcal{O}(n^2)$ operations [16].

We have run Algorithm 5.1 for Vandermonde matrices up to order 12, which are TP and are known to be badly ill conditioned. Their bidiagonal factorization is known, see, e.g., [16]. The results confirm that all the eigenvalues have been accurately computed.

6. Perturbation and error analysis. In this section, we will show that small perturbations in the parameters of the bidiagonal factorization (1.1) of a nonsingular TN matrix A, and thus in the matrix $\tilde{G}$, cause only small relative perturbations in the eigenvalues of A.

We use the standard model of floating point arithmetic [14, Section 2.2]:

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

Relative perturbations are accumulated as follows [14, Lemmata 3.1 and 3.3]:

If $\rho_i \in \{-1, 1\}$, $|\delta_i| \le \delta$, $i = 1, \ldots, k$, and $k\delta < 1$, then

$$\left|\prod_{i=1}^{k}(1+\delta_i)^{\rho_i} - 1\right| \le \frac{k\delta}{1-k\delta},$$

and

if $|\gamma_1| \le k_1\delta/(1-k_1\delta)$, $|\gamma_2| \le k_2\delta/(1-k_2\delta)$, and $(k_1+k_2)\delta < 1$, then

$$|(1+\gamma_1)(1+\gamma_2) - 1| \le \frac{(k_1+k_2)\delta}{1-(k_1+k_2)\delta}.$$

We follow the arguments in [16, Section 7]. In Theorem 7.2 therein, it is shown that small relative perturbations in the entries of the bidiagonal factorization of a TN matrix cause small relative perturbations in the eigenvalues of this matrix. Specifically, let $A \in \mathbb{R}^{n\times n}$ be a nonsingular TN matrix, and $B \in \mathbb{R}^{n\times n}$ be obtained from A by replacing a single parameter x in the right-hand side of (1.1) by $x(1+\delta_x)$, where $|\delta_x| \le \epsilon$. Denote the eigenvalues of A and B by $\lambda_1 \ge \lambda_2 \ge \ldots \ge \lambda_n$ and $\mu_1 \ge \mu_2 \ge \ldots \ge \mu_n$, respectively. Then, for all $i = 1, \ldots, n$,

$$|\mu_i - \lambda_i| \le \frac{2\epsilon}{1-2\epsilon}\lambda_i. \tag{6.1}$$

In the reduction of the eigenvalue problem to the singular value problem of a bidiagonal matrix by Theorem 4.3 for a TP (or by Algorithm 4.1 for a nonsingular TN) matrix and Algorithm 5.1, the only source of error is in the computation of the entries of the intermediate matrices. All arithmetic is subtraction-free; thus, every single floating point operation causes at most ϵ relative perturbation in an entry of an intermediate matrix. Taking the relative perturbation of any eigenvalue of A according to (6.1) into account, after at most

$$\frac{10}{3}n^3 + O(n^2),$$

such arithmetic operations, accumulation of the errors as described above results in a relative perturbation in each entry of $\tilde{H}$, which does not exceed the bound

$$\frac{\left(\frac{20}{3}n^3 + O(n^2)\right)\epsilon}{1 - \left(\frac{20}{3}n^3 + O(n^2)\right)\epsilon}.$$

The computation of the Cholesky factor C from the tridiagonal matrix $\overline{T}$ requires $2n-2$ multiplications and $2n-1$ square roots and can therefore be performed accurately. Finally, the function `bidsvd` computes accurately each singular value of a bidiagonal matrix because it uses the `LAPACK` [5] routine `dbdsqr` which does this task [7, 9]. Thus, we can conclude that the relative error in each eigenvalue of A, as computed by Algorithm 5.1, does not exceed

$$\left(\frac{20}{3}n^3 + O(n^2)\right)\epsilon.$$

7. Conclusions. In this paper, we have shown that certain arithmetic operations can be performed with the Cauchon algorithm (Algorithm 3.1) with high relative accuracy. By the use of these operations, we are able to accurately compute all eigenvalues of a nonsingular TN matrix independently of its condition number. Naming the Cauchon algorithm in the title of the paper is justified because the matrix $\tilde{G}$ we are using as input for Algorithm 5.1 is indeed the matrix, which is obtained by the application of the Cauchon algorithm to $G = (A^{\#})^T$. However, this computation is not subtraction-free. For the sake of high relative accuracy, we have chosen the calculation of $\tilde{G}$ from the bidiagonal factorization of A. The reduction of the matrix $\tilde{G}$ to the tridiagonal form $\tilde{T}$ is based on the elementary operations considered in Section 4.1 under the performance of the Cauchon algorithm. In addition, the extension of Algorithms 3.3 and 3.4 from the TP to the nonsingular TN case requires the use of the Cauchon algorithm as well as its inverse algorithm, the restoration algorithm (Algorithm 3.2). The use of the matrix $\tilde{G}$ instead of the use of the bidiagonal factorization employed in [16] is the main contribution of this paper. For a large order of the given TN matrix, our approach requires fewer arithmetic operations than the approach in [16].

In our future work, we will extend the approach to the accurate computation of the singular values, eigenvectors, and the inverse matrix of nonsingular TN matrices as well as to the solution of systems of linear equations with such matrices as coefficient matrices.

A. Proof of Theorem 4.2. In the proof of Theorem 4.2, we are making extensive use of the following special form of Sylvester's Determinant Identity.

LEMMA A.1. *Partition $A \in \mathbb{R}^{n\times n}$, $n \geq 3$, as follows:*

$$A = \begin{pmatrix} c & A_{12} & d \\ A_{21} & A_{22} & A_{23} \\ e & A_{32} & f \end{pmatrix},$$

where $A_{22} \in \mathbb{R}^{(n-2)\times(n-2)}$ *and* c, d, e, f *are scalars. Define the submatrices*

$$C := \begin{pmatrix} c & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, D := \begin{pmatrix} A_{12} & d \\ A_{22} & A_{23} \end{pmatrix},$$
$$E := \begin{pmatrix} A_{21} & A_{22} \\ e & A_{32} \end{pmatrix}, F := \begin{pmatrix} A_{22} & A_{23} \\ A_{32} & f \end{pmatrix}.$$

Then, if $\det A_{22} \neq 0$, *we have*

$$\det A = \frac{\det C \det F - \det D \det E}{\det A_{22}}.$$

Proof of Theorem 4.2[†]. The TP matrix B can be written as follows:

$$B = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1,n-1} & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2,n-1} & a_{2n} \\ \vdots & \vdots & & \vdots & \vdots \\ 1/y\ a_{i1} & 1/y\ a_{i2} & \cdots & 1/y\ a_{i,n-1} & 1/y\ a_{in} \\ ya_{i+1,1} + xa_{i1} & ya_{i+1,2} + xa_{i2} & \cdots & ya_{i+1,n-1} + xa_{i,n-1} & ya_{i+1,n} + xa_{in} \\ \vdots & \vdots & & \vdots & \vdots \\ a_{n-1,1} & a_{n-1,2} & \cdots & a_{n-1,n-1} & a_{n-1,n} \\ a_{n1} & a_{n2} & \cdots & a_{n,n-1} & a_{nn} \end{bmatrix}.$$

To find the entries of $\tilde{B}$ in terms of the entries of $\tilde{A}$, we distinguish the following three cases:

Case 1: $k < i$ *or* $k > i+1$.
Since B is TP, we have by Lemma 3.1

$$\begin{aligned} \tilde{b}_{kj} &= B\left\langle k, \ldots, k+w | j, \ldots, j+w \right\rangle \\ &= A\left\langle k, \ldots, k+w | j, \ldots, j+w \right\rangle \\ &= \tilde{a}_{kj}. \end{aligned}$$

The second equality follows in the cases $k > i+1$ and $k+w < i$, since
$B\left[k, \ldots, k+w | j, \ldots, j+w\right] = A\left[k, \ldots, k+w | j, \ldots, j+w\right]$.
Otherwise, $k < i \leq k+w$ holds. If $i+1 \leq k+w$, then by determinantal properties, the second equality also follows. If $i+1 > k+w$, then $i = k+w$ holds. Hence, by determinantal properties, the second equality follows.

Case 2: $k = i+1$.

Since $\tilde{a}_{i+1,n} = a_{i+1,n}$, $\tilde{a}_{in} = a_{in}$, and $\tilde{b}_{i+1,n} = b_{i+1,n}$, it follows from the definition of B that

$$\tilde{b}_{i+1,n} = x\ \tilde{a}_{in} + y\ \tilde{a}_{i+1,n}.$$

[†] A more detailed proof can be found in [27, Section 3.1].

In the following, we will explain how the other entries in row $i+1$ of $\tilde{B}$ are changed. In the sequel, we assume that n is sufficiently large compared to $i+1$ such that sufficiently many rows of B below the $(i+1)$th row are available. If this assumption is not fulfilled, the statement can be proven similarly. By Lemma 3.1 and determinantal properties, we obtain

$$\begin{aligned}\tilde{b}_{i+1,n-1} &= B\left\langle i+1, i+2|n-1, n\right\rangle\\ &= x\ A\left\langle i, i+2|n-1, n\right\rangle + y\ A\left\langle i+1, i+2|n-1, n\right\rangle\\ &= x\ A\left\langle i, i+2|n-1, n\right\rangle + y\ \tilde{a}_{i+1,n-1}.\end{aligned} \tag{A.1}$$

To find the value of the second term in (A.1) in terms of the entries of $\tilde{A}$, we add and subtract the following quantity to it:

$$A\left\langle i+1, i+2|n-1, n\right\rangle\ \frac{a_{in}}{a_{i+1,n}} = \tilde{a}_{i+1,n-1}\ \frac{\tilde{a}_{in}}{\tilde{a}_{i+1,n}}.$$

Hence,

$$\begin{aligned}A\left\langle i, i+2|n-1, n\right\rangle &= A\left\langle i, i+2|n-1, n\right\rangle\\ &\quad \pm A\left\langle i+1, i+2|n-1, n\right\rangle\ \frac{a_{in}}{a_{i+1,n}}\\ &= \frac{a_{i,n-1}a_{i+2,n} - a_{in}a_{i+2,n-1}}{a_{i+2,n}}\\ &\quad + \frac{\tilde{a}_{i+1,n-1}\tilde{a}_{in}}{\tilde{a}_{i+1,n}} - \frac{a_{in}}{a_{i+2,n}a_{i+1,n}}\left(a_{i+2,n}a_{i+1,n-1} - a_{i+1,n}a_{i+2,n-1}\right)\\ &= \frac{a_{i,n-1}a_{i+1,n} - a_{in}a_{i+1,n-1}}{a_{i+1,n}} + \frac{\tilde{a}_{i+1,n-1}\tilde{a}_{in}}{\tilde{a}_{i+1,n}}\\ &= A\left\langle i, i+1|n-1, n\right\rangle + \frac{\tilde{a}_{i+1,n-1}\tilde{a}_{in}}{\tilde{a}_{i+1,n}}\\ &= \tilde{a}_{i,n-1} + \tilde{a}_{i+1,n-1}\frac{\tilde{a}_{in}}{\tilde{a}_{i+1,n}}.\end{aligned} \tag{A.2}$$

By substituting (A.2) into (A.1), we get

$$\tilde{b}_{i+1,n-1} = y\ \tilde{a}_{i+1,n-1} + x\ \left(\tilde{a}_{i,n-1} + \tilde{a}_{i+1,n-1}\frac{\tilde{a}_{in}}{\tilde{a}_{i+1,n}}\right). \tag{A.3}$$

To prove the relation for the remaining entries in row $i+1$ of $\tilde{B}$, we use decreasing induction on the column index $k = n-2, \ldots, 1$, i.e., we will show

$$\tilde{b}_{i+1,n-\gamma} = y\ \tilde{a}_{i+1,n-\gamma} + x\ \left(\tilde{a}_{i,n-\gamma} + \tilde{a}_{i+1,n-\gamma}\sum_{m=n-\gamma+1}^{n}\frac{\tilde{a}_{im}}{\tilde{a}_{i+1,m}}\right). \tag{A.4}$$

We assume that (A.4) is true. Then, by Lemma 3.1

$$\begin{aligned}\tilde{b}_{i+1,n-\gamma} &= B\left\langle i+1,\dots,i+1+w|n-\gamma,\dots,n-\gamma+w\right\rangle \\ &= y\ \tilde{a}_{i+1,n-\gamma} + x\,A\left\langle i,i+2,\dots,i+1+w|n-\gamma,\dots,n-\gamma+w\right\rangle. \qquad \text{(A.5)}\end{aligned}$$

From (A.4) and (A.5), we obtain

$$A\left\langle i,i+2,\dots,i+1+w|n-\gamma,\dots,n-\gamma+w\right\rangle = \tilde{a}_{i,n-\gamma} + \tilde{a}_{i+1,n-\gamma}\sum_{m=n-\gamma+1}^{n}\frac{\tilde{a}_{im}}{\tilde{a}_{i+1,m}}. \qquad \text{(A.6)}$$

Now, we want to prove that (A.4) holds for $j=\gamma+1$. If $n-\gamma > i+1$, then for $j=n-\gamma-1$, we are in the upper triangular part or on the main diagonal of the matrix $\tilde{B}$.

$$\begin{aligned}\tilde{b}_{i+1,n-\gamma-1} &= B\left\langle i+1,\dots,i+2+w|n-\gamma-1,\dots,n-\gamma+w\right\rangle \\ &= y\ \tilde{a}_{i+1,n-\gamma-1} + x\,A\left\langle i,i+2,\dots,i+2+w|n-\gamma-1,\dots,n-\gamma+w\right\rangle. \qquad \text{(A.7)}\end{aligned}$$

To find the value of the second term in (A.7), we apply Sylvester's Determinant Identity on its numerator and add and subtract the term

$$\begin{aligned}&A\left\langle i+1,\dots,i+2+w|n-\gamma-1,\dots,n-\gamma+w\right\rangle \\ &\cdot\frac{A\left\langle i,i+2,\dots,i+1+w|n-\gamma,\dots,n-\gamma+w\right\rangle}{A\left\langle i+1,\dots,i+1+w|n-\gamma,\dots,n-\gamma+w\right\rangle}.\end{aligned}$$

Then, we obtain

$$\begin{aligned}A\left\langle i,i+2,\dots,i+2+w|n-\gamma-1,\dots,n-\gamma+w\right\rangle &= A\left\langle i,i+2,\dots,i+1+w|n-\gamma-1,\dots,n-\gamma-1+w\right\rangle \\ &-\frac{\det A[i,i+2,\dots,i+1+w|n-\gamma,\dots,n-\gamma+w]\ \det A[i+2,\dots,i+2+w|n-\gamma-1,\dots,n-\gamma-1+w]}{\det A[i+2,\dots,i+1+w|n-\gamma,\dots,n-\gamma-1+w]\ \det A[i+2,\dots,i+2+w|n-\gamma,\dots,n-\gamma+w]} \\ &-A\left\langle i+1,\dots,i+2+w|n-\gamma-1,\dots,n-\gamma+w\right\rangle\ \frac{A\left\langle i,i+2,\dots,i+1+w|n-\gamma,\dots,n-\gamma+w\right\rangle}{A\left\langle i+1,\dots,i+1+w|n-\gamma,\dots,n-\gamma+w\right\rangle} \\ &+\tilde{a}_{i+1,n-\gamma-1}\ \frac{1}{\tilde{a}_{i+1,n-\gamma}}A\left\langle i,i+2,\dots,i+1+w|n-\gamma,\dots,n-\gamma+w\right\rangle. \qquad \text{(A.8)}\end{aligned}$$

Again by Sylvester's Determinant Identity, we conclude

$$\begin{aligned}A\left\langle i+1,\dots,i+2+w|n-\gamma-1,\dots,n-\gamma+w\right\rangle &= A\left\langle i+1,\dots,i+1+w|n-\gamma-1,\dots,n-\gamma-1+w\right\rangle \\ -\frac{\det A[i+1,\dots,i+1+w|n-\gamma,\dots,n-\gamma+w]\ \det A[i+2,\dots,i+2+w|n-\gamma-1,\dots,n-\gamma-1+w]}{\det A[i+2,\dots,i+2+w|n-\gamma,\dots,n-\gamma+w]\ \det A[i+2,\dots,i+1+w|n-\gamma,\dots,n-\gamma-1+w]}&. \qquad \text{(A.9)}\end{aligned}$$

By substituting (A.9) into (A.8) and after simplifications, we get

$$\begin{aligned}&A\left\langle i,i+2,\dots,i+2+w|n-\gamma-1,\dots,n-\gamma+w\right\rangle \\ &\quad= A\left\langle i,i+2,\dots,i+1+w|n-\gamma-1,\dots,n-\gamma-1+w\right\rangle\end{aligned}$$

$$
\begin{aligned}
&- A\left\langle i+1,\dots,i+1+w|n-\gamma-1,\dots,n-\gamma-1+w\right\rangle \\
&\quad\cdot \frac{\det A[i,i+2,\dots,i+1+w|n-\gamma,\dots,n-\gamma+w]}{\det A[i+1,\dots,i+1+w|n-\gamma,\dots,n-\gamma+w]} \\
&+ \frac{\tilde{a}_{i+1,n-\gamma-1}}{\tilde{a}_{i+1,n-\gamma}} A\left\langle i,i+2,\dots,i+1+w|n-\gamma,\dots,n-\gamma+w\right\rangle
\end{aligned}
$$

$$
\begin{aligned}
&= A\left\langle i,\dots,i+1+w|n-\gamma-1,\dots,n-\gamma+w\right\rangle \\
&\quad+\frac{\tilde{a}_{i+1,n-\gamma-1}}{\tilde{a}_{i+1,n-\gamma}} A\left\langle i,i+2,\dots,i+1+w|n-\gamma,\dots,n-\gamma+w\right\rangle \\
&= \tilde{a}_{i,n-\gamma-1} + \frac{\tilde{a}_{i+1,n-\gamma-1}}{\tilde{a}_{i+1,n-\gamma}} \left(\tilde{a}_{i,n-\gamma} + \tilde{a}_{i+1,n-\gamma} \sum_{m=n-\gamma+1}^{n} \frac{\tilde{a}_{im}}{\tilde{a}_{i+1,m}} \right), \text{by (A.6)}
\end{aligned}
\tag{A.10}
$$

$$
= \tilde{a}_{i,n-\gamma-1} + \tilde{a}_{i+1,n-\gamma-1} \sum_{m=n-\gamma}^{n} \frac{\tilde{a}_{im}}{\tilde{a}_{i+1,m}}. \tag{A.11}
$$

Hence by substituting (A.11) into (A.7), we conclude that (A.4) is true when $j=\gamma+1$ in the upper triangular part and on the main diagonal of $\tilde{B}$.

If $n-\gamma \le i+1$, then for $j=n-\gamma-1$, we are in the lower triangular part of $\tilde{B}$, and we obtain similarly as in (A.7)

$$
\begin{aligned}
\tilde{b}_{i+1,n-\gamma-1} &= B\left\langle i+1,\dots,i+1+w|n-\gamma-1,\dots,n-\gamma-1+w\right\rangle \\
&= x\,a + y\;\tilde{a}_{i+1,n-\gamma-1},
\end{aligned}
\tag{A.12}
$$

where $a := A\left\langle i,i+2,\dots,i+1+w|n-\gamma-1,\dots,n-\gamma-1+w\right\rangle$. In (A.12), we rewrite a by using Sylvester's Determinant Identity and add and subtract the following quantity

$$
A\left\langle i,\dots,i+1+w|n-\gamma-1,\dots,n-\gamma+w\right\rangle.
$$

$$
\begin{aligned}
a &= a + A\left\langle i,\dots,i+w+1|n-\gamma-1,\dots,n-\gamma+w\right\rangle \\
&\quad - \frac{\det A[i,i+2,\dots,i+1+w|n-\gamma-1,\dots,n-\gamma-1+w]}{\det A[i+1,\dots,i+1+w|n-\gamma,\dots,n-\gamma+w]} \cdot \\
&\quad\quad \frac{\det A[i+1,\dots,i+1+w|n-\gamma,\dots,n-\gamma+w]}{\det A[i+2,\dots,i+1+w|n-\gamma,\dots,n-\gamma-1+w]} \\
&\quad + \frac{\det A[i,i+2,\dots,i+1+w|n-\gamma,\dots,n-\gamma+w]}{\det A[i+1,\dots,i+1+w|n-\gamma,\dots,n-\gamma+w]} \cdot \\
&\quad\quad A\left\langle i+1,\dots,i+1+w|n-\gamma-1,\dots,n-\gamma-1+w\right\rangle \\
&= A\left\langle i,\dots,i+1+w|n-\gamma-1,\dots,n-\gamma+w\right\rangle \\
&\quad + \frac{\det A[i,i+2,\dots,i+1+w|n-\gamma,\dots,n-\gamma+w]}{\det A[i+1,\dots,i+1+w|n-\gamma,\dots,n-\gamma+w]} \cdot \\
&\quad\quad A\left\langle i+1,\dots,i+1+w|n-\gamma-1,\dots,n-\gamma-1+w\right\rangle
\end{aligned}
$$

$$= \tilde{a}_{i,n-\gamma-1} + \tilde{a}_{i+1,n-\gamma-1} \quad \frac{\frac{\det A[i,i+2,\ldots,i+1+w|n-\gamma,\ldots,n-\gamma+w]}{\det A[i+2,\ldots,i+1+w|n-\gamma+1,\ldots,n-\gamma+w]}}{\frac{\det A[i+1,\ldots,i+1+w|n-\gamma,\ldots,n-\gamma+w]}{\det A[i+2,\ldots,i+1+w|n-\gamma+1,\ldots,n-\gamma+w]}}$$

$$= \tilde{a}_{i,n-\gamma-1} + \frac{\tilde{a}_{i+1,n-\gamma-1}}{\tilde{a}_{i+1,n-\gamma}} \cdot A\left\langle i, i+2, \ldots, i+1+w|n-\gamma, \ldots, n-\gamma+w \right\rangle. \tag{A.13}$$

Similarly as from (A.10), we conclude by (A.6) that (A.4) is true when $j = \gamma + 1$ in the lower triangular part of the matrix $\tilde{B}$.

Case 3: $k = i$.
Because all the entries in the last column of B do not change when we apply Algorithm 3.1 on B, the relation

$$\tilde{b}_{in} = \frac{1}{y}\, \tilde{a}_{in}.$$

holds. To prove

Proof.

$$\tilde{b}_{i,n-j} = \frac{\tilde{a}_{i,n-j}}{y + x\sum_{m=n-j+1}^{n} \frac{\tilde{a}_{im}}{\tilde{a}_{i+1,m}}}, \tag{A.14}$$

for $j = 1, \ldots, n-1$, we use mathematical induction on j. For $j = 1$,

$$\begin{aligned}
\tilde{b}_{i,n-1} &= B\left\langle i, i+1|n-1, n \right\rangle \\
&= \frac{\det A[i,i+1|n-1,n]}{y \det A[i+1|n] + x \det A[i|n]} \\
&= \frac{\det A[i,i+1|n-1,n]/\det A[i+1|n]}{\big(y \det A[i+1|n] + x \det A[i|n]\big)/\det A[i+1|n]} \\
&= \frac{\tilde{a}_{i,n-1}}{y + x\, \frac{\tilde{a}_{in}}{\tilde{a}_{i+1,n}}}.
\end{aligned}$$

Assume that (A.14) is true when $j = \gamma > 0$. We want to prove that (A.14) is true when $j = \gamma + 1$ and consider the case $n - \gamma > i$ first.

$$\begin{aligned}
\tilde{b}_{i,n-\gamma-1} &= B\left\langle i, \ldots, i+1+w|n-\gamma-1, \ldots, n-\gamma+w \right\rangle \\
&= \frac{\det A[i,\ldots,i+1+w|n-\gamma-1,\ldots,n-\gamma+w]}{y\ \det A[i+1,\ldots,i+1+w|n-\gamma,\ldots,n-\gamma+w] + x\ \det A[i,i+2,\ldots,i+1+w|n-\gamma,\ldots,n-\gamma+w]} \\
&= \frac{\det A[i,\ldots,i+1+w|n-\gamma-1,\ldots,n-\gamma+w]/\det A[i+1,\ldots,i+1+w|n-\gamma,\ldots,n-\gamma+w]}{y + x \det A[i,i+2,\ldots,i+1+w|n-\gamma,\ldots,n-\gamma+w]/\det A[i+1,\ldots,i+1+w|n-\gamma,\ldots,n-\gamma+w]} \\
&= \frac{\tilde{a}_{i,n-\gamma-1}}{y + x\frac{\det A[i,i+2,\ldots,i+1+w|n-\gamma,\ldots,n-\gamma+w]/\det A[i+2,\ldots,i+1+w|n-\gamma+1,\ldots,n-\gamma+w]}{\det A[i+1,\ldots,i+1+w|n-\gamma,\ldots,n-\gamma+w]/\det A[i+2,\ldots,i+1+w|n-\gamma+1,\ldots,n-\gamma+w]}} \\
&= \frac{\tilde{a}_{i,n-\gamma-1}}{y + x\frac{A\left\langle i,i+2,\ldots,i+1+w|n-\gamma,\ldots,n-\gamma+w \right\rangle}{\tilde{a}_{i+1,n-\gamma}}}.
\end{aligned} \tag{A.15}$$

By using (A.6), we obtain

$$\begin{aligned}\tilde{b}_{i,n-\gamma-1} &= \frac{\tilde{a}_{i,n-\gamma-1}}{y + x\dfrac{\tilde{a}_{i,n-\gamma}+\tilde{a}_{i+1,n-\gamma}\sum_{m=n-\gamma+1}^{n}\frac{\tilde{a}_{im}}{\tilde{a}_{i+1,m}}}{\tilde{a}_{i+1,n-\gamma}}}\\ &= \frac{\tilde{a}_{i,n-\gamma-1}}{y + x\sum_{m=n-\gamma}^{n}\frac{\tilde{a}_{im}}{\tilde{a}_{i+1,m}}}.\end{aligned}$$

Hence, (A.14) is true when $j=\gamma+1$. When $n-\gamma\leq i$, we proceed similarly. ☐

B. The function `AddToNextRow`. The following function implements Algorithm 4.1, that is, given the matrix $\tilde{A}$, matrix $\tilde{B}$ is computed, where B is obtained from A by multiplication of its ith row by the positive scalar x and adding it to the next row multiplied by y. Finally, the ith row is multiplied by $\frac{1}{y}$.

```
function TildeB = AddToNextRow(TildeA,x,y,i)
n = size(TildeA,1);
TildeB = TildeA;
    % The entries in the row i and i+1
TildeB(i+1,n) = x*TildeA(i,n)+y*TildeA(i+1,n);
TildeB(i,n) = TildeA(i,n)/y;
    for j = n-1:-1:1
        if TildeA(i+1,j+1) ~= 0
            z = TildeB(i+1,j+1)/TildeA(i+1,j+1);
            TildeB(i+1,j) = x*TildeA(i,j) + TildeA(i+1,j)*z;
            TildeB(i,j) = TildeA(i,j)/z;
        else
            if TildeA(i+1,j) == 0
                TildeB(i+1,j) = x*TildeA(i,j);
                TildeB(i,j) = 0;
            else
                h = j+1;
                while TildeA(i+1,h) == 0 && h < n
                     h = h+1;
                end
                if h == n && TildeA(i+1,h) == 0
                    TildeB(i+1,j) = x*TildeA(i,j)+y*TildeA(i+1,j);
                    TildeB(i,j) = TildeA(i,j)/y;
                else
                    z = TildeB(i+1,h)/TildeA(i+1,h);
                    TildeB(i+1,j) = x*TildeA(i,j)+TildeA(i+1,j)*z;
                    TildeB(i,j) = TildeA(i,j)/z;
                end
            end
        end
    end
```

Acknowledgments. The authors thank the reviewers for their constructive comments that have led to an improvement of the manuscript. The third author gratefully acknowledges financial support from the HTWG Konstanz—University of Applied Sciences through the SRP Program.

References.

[1] M. Adm. *Perturbation and Intervals of Totally Nonegative Matrices and Related Properties of Sign Regular Matrices*, Ph.D. Thesis, University of Konstanz, 2016.

[2] M. Adm, K. Al Muhtaseb, A. Abedel Ghani, S. Fallat, and J. Garloff. Further applications of the Cauchon algorithm to rank determination and bidiagonal factorization. *Linear Algebra Appl.*, 545:240–255, 2018.

[3] M. Adm and J. Garloff. Intervals of totally nonnegative matrices. *Linear Algebra Appl.*, 439:3796–3806, 2013.

[4] M. Adm and J. Garloff. Improved tests and characterizations of totally nonnegative matrices. *Electron. J. Linear Algebra*, 27:588–610, 2014.

[5] E. Anderson, Z. Bai, C. Bischof, L.S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*, 3rd ed. Software Environ. Tools, Vol. 9. SIAM, Philadelphia, 1999.

[6] G. Cauchon. Effacement des dérivations et spectres premiers des algèbres quantiques. *J. Algebra*, 260:476–518, 2003.

[7] J. Demmel and W. Kahan. Accurate singular values of bidiagonal matrices. *SIAM J. Sci. Statist. Comput.*, 11(5):873–912, 1990.

[8] S.M. Fallat and C.R. Johnson. *Totally Nonnegative Matrices.* Princeton University Press, Princeton and Oxford, 2011.

[9] K.V. Fernando and B.N. Parlett. Accurate singular values and differential qd algorithms. *Numer. Math.*, 67(2):191–229, 1994.

[10] F. Gantmacher and M. Krein. *Oscillation Matrices and Kernels and Small Vibrations of Mechanical Systems*, revised edition. AMS Chelsea, Providence, RI, 2002.

[11] M. Gasca and J. M. Peña. On factorizations of totally positive matrices. In: M. Gasca and C.A. Micchelli (editors), *Total Positivity and Its Applications.* Kluwer Academic Publishers, Dordrecht, Boston, London, 109–130, 1996.

[12] G. Geist, G. Howell, and D. Watkins. The BR eigenvalue algorithm. *SIAM J. Matrix Anal. Appl.*, 20:1083–1098, 1999.

[13] K.R. Goodearl, S. Launois, and T.H. Lenagan. Totally nonnegative cells and matrix Poisson varieties. *Adv. Math.*, 226:779–826, 2011.

[14] N.J. Higham. *Accuracy and Stability of Numerical Algorithms*, 2nd ed. SIAM, Philadelphia, 2002.

[15] S. Karlin. *Total Positivity*, Vol. I. Stanford University Press, Stanford, CA, 1968.

[16] P. Koev. Accurate eigenvalues and SVDs of totally nonnegative matrices. *SIAM J. Matrix Anal. Appl.*, 27:1–23, 2005.

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

[18] S. Launois. Generators for $\mathcal{H}$-invariant prime ideals in $O_q(\mathcal{M}_{m,p}(\mathbb{C}))$. *Proc. Edinb. Math. Soc.*, 47:163–190, 2004.

[19] S. Launois and T.H. Lenagan. Efficient recognition of totally nonnegative matrix cells. *Found. Comput. Math.*, 14:371–387, 2014.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 42, pp. 58-83, February 2026.

[20] J.-J. Martínez. The application of the bidiagonal factorization of totally positive matrices in numerical linear algebra. *Axioms*, 13(4):article no. 258, 2024.

[21] The MathWorks, Inc. *MATLAB Reference Guide*. Natick, MA, 1992.

[22] J.M. Peña (editor). *Shape Preserving Representations in Computer Aided Geometric Design*. Nova Science Publishers, Commack, NY, 1999.

[23] J.M. Peña. Eigenvalue localization for totally positive matrices. In: B. Bru and S. Romero-Vivó (editors), *Positive Systems*. Lecture Notes in Control and Information Sciences, Vol. 389. Springer, Berlin, Heidelberg, 123–130, 2009.

[24] J.M. Peña. Accurate computations and applications of some classes of matrices. In: M. Mateos and P. Alonso (editors), *Computational Mathematics, Numerical Analysis and Applications*. SEMA SIMAI Springer Series, Vol. 13. Springer International Publishing AG, Cham, Switzerland, 107–151, 2017.

[25] P. Persson. *Eigenvalues of Tridiagonal Matrices in MATLAB*, Department of Mathematics, UC Berkeley, 2024.

[26] A. Pinkus. *Totally Positive Matrices*. Cambridge Facts in Mathematics. Cambridge University Press, Cambridge, 2010.

[27] F. Rasheed. *Some Numerical Aspects of the Cauchon Algorithm*, Master Thesis, Palestine Polytechnic University, Hebron, Palestine, 2021.

[28] A.M. Whitney. A reduction theorem for totally positive matrices. *J. Anal. Math.*, 2:88–92, 1952.