Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Linear Algebra and its Applications 424 (2007) 435–447
LINEAR ALGEBRA AND ITS APPLICATIONS
www.elsevier.com/locate/laa

Accurate eigenvalues of certain sign regular matrices

Plamen Koev [a,*,1], Froilán Dopico [b,2]

[a] *Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*
[b] *Departamento de Matemáticas, Universidad Carlos III de Madrid, Avda. Universidad 30, 28911 Leganés, Spain*

Received 29 June 2006; accepted 5 February 2007
Available online 20 February 2007
Submitted by V. Mehrmann

Abstract

We present a new $\mathcal{O}(n^3)$ algorithm for computing all eigenvalues of certain sign regular matrices to high relative accuracy in floating point arithmetic. The accuracy and cost are unaffected by the conventional eigenvalue condition numbers.

A matrix is called sign regular when the signs of its nonzero minors depend only of the order of the minors. The sign regular matrices we consider are the ones which are nonsingular and whose kth order nonzero minors are of sign $(-1)^{k(k-1)/2}$ for all k. This class of matrices can also be characterized as "nonsingular totally nonnegative matrices with columns in reverse order".

We exploit a characterization of these particular sign regular matrices as products of nonnegative bidiagonals and the reverse identity. We arrange the computations in such a way that no subtractive cancellation is encountered, thus guaranteeing high relative forward accuracy.

Keywords: Eigenvalues; High relative accuracy; Sign regular matrices; Totally nonnegative matrices

1. Introduction

An $n \times n$ matrix all of whose nonzero kth order minors have the same sign $\rho_k \in \{-1, +1\}$, $k = 1, 2, \ldots, n$, is called *sign regular*. All eigenvalues of a sign regular matrix are real. Our primary

[*] Corresponding author.
E-mail addresses: plamen@math.mit.edu (P. Koev), dopico@math.uc3m.es (F. Dopico).

[1] The work of this author was supported by National Science Foundation grant DMS-0314286.

[2] The work of this author was supported by the Ministerio de Educación y Ciencia of Spain through grant BFM-2003-00223 and the PRICIT program of Comunidad de Madrid through grant SIMUMAT (S-0505/ESP/0158).

0024-3795/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.laa.2007.02.012

concern in this paper is the accurate computation of the eigenvalues of one particular class of nonsingular sign regular matrices, namely those with $\rho_k = (-1)^{k(k-1)/2}$, $k = 1, 2, \ldots, n$.

We call this class TN^J. The choice of notation stems from the fact that the TN^J matrices can be characterized as nonsingular totally nonnegative (TN) matrices with the columns in reverse order (see Section 2).

The TN^J matrices can be very ill conditioned; in fact, they are as ill conditioned as TN matrices (since reversing the order of the columns does not change the conventional two-norm condition number $\kappa(A) \equiv \|A\|_2 \cdot \|A^{-1}\|_2$). In particular, the class of TN^J matrices includes includes notorious examples such as (column-reversed) TN Vandermonde, Cauchy, and Pascal matrices. Large condition numbers have dire implications for numerical computations: the conventional eigenvalue algorithms deliver only high *absolute* accuracy and, in particular, the computed eigenvalues $\hat{\lambda}_i$ satisfy a bound of the form

$$|\lambda_i - \hat{\lambda}_i| \leqslant \mathcal{O}(\varepsilon)\frac{\|A\|}{|y^*x|}, \tag{1}$$

where ε is the machine precision and x and y are the left and right normalized eigenvectors corresponding to λ_i, respectively.

The expression (1) implies that only the largest eigenvalues of an ill conditioned TN^J matrix may be well conditioned with respect to unstructured normwise perturbations in the matrix and thus be computed accurately by the conventional algorithms [2,9]. The problem is with the tiny eigenvalues—they are very ill conditioned with respect to such unstructured perturbations and are most often lost to roundoff. This is unfortunate, since all eigenvalues of the TN^J matrices, including the tiniest ones, are very accurately determined by the data, are very well conditioned with respect to *structure-preserving* perturbations, and thus deserve to be computed accurately (see Section 2).

Our main contribution in this paper is a new accurate algorithm for computing all eigenvalues of TN^J matrices. It costs $\mathcal{O}(n^3)$—about the same as the conventional algorithms—but computes all eigenvalues with guaranteed high relative accuracy. Namely, every computed eigenvalue $\hat{\lambda}_i$ will have a correct sign and leading digits, and will satisfy the bound

$$|\lambda_i - \hat{\lambda}_i| \leqslant \mathcal{O}(\varepsilon)|\lambda_i|.$$

The constant hidden in the big-$\mathcal{O}$ notation is a modestly growing function of n.

We believe that this is the first example of an $\mathcal{O}(n^3)$ algorithm to accurately compute all eigenvalues of a nonsymmetric matrix with both positive and negative eigenvalues without using extra precision.

The caveat in our algorithm is that we require that the TN^J structure be explicitly revealed. Namely, a TN^J matrix must be represented by the n^2 independent, nonnegative entries of its bidiagonal decomposition

$$A = L^{(1)}L^{(2)}\cdots L^{(n-1)}DU^{(n-1)}U^{(n-2)}\cdots U^{(1)} \cdot J, \tag{2}$$

where D is diagonal, $L^{(k)}$ and $U^{(k)}$ are lower and upper unit bidiagonal matrices, respectively, and $J \equiv [\delta_{i,n+1-j}]_{i,j=1}^n$ is the reverse identity.

We review the bidiagonal decomposition (2) in Section 2. It is inherited by the bidiagonal decompositions of the TN matrices and has the desirable property that its n^2 nontrivial entries parameterize the set of TN^J matrices, reveal the TN^J structure of A, and determine the eigenvalues of A to high relative accuracy.

For the column reversed versions of classical TN matrices, including Vandermonde, generalized Vandermonde, Bernstein Vandermonde, Cauchy–Vandermonde, Cauchy, Hilbert, Pascal, etc., as well as their products, Schur complements, etc., it is known how to compute the decomposition (2) accurately in $\mathcal{O}(n^3)$ time, as discussed in Section 2.

The ideas in deriving our accurate TN^J eigenvalue algorithm are:

1. Since subtractive cancellation is the source of loss of high relative accuracy in the conventional eigenvalue algorithms, our goal is to avoid it. This idea has already been used successfully in other accurate algorithms [1,3,4,6,8,15,16,19]. It is based on the observation that multiplications, divisions, and sums of same-sign quantities preserve the high relative accuracy, whereas expressions involving subtractions may not;
2. We use similarity transformations to reduce the TN^J matrix A to a nonnegative anti-bidiagonal matrix
$$C = \begin{bmatrix} & & & b_1 & a_1 \\ & & \cdot & a_2 & \\ & \cdot & \cdot & & \\ b_{n-1} & a_{n-1} & & & \\ a_n & & & & \end{bmatrix}.$$
The matrix C has the same eigenvalues as a *symmetric* anti-bidiagonal matrix
$$G = \begin{bmatrix} & & & b_1' & a_1' \\ & & \cdot & a_2' & \\ & \cdot & \cdot & & \\ b_1' & a_2' & & & \\ a_1' & & & & \end{bmatrix}.$$
The matrices C and G may not be similar (see Section 3.2), but they have the same eigenvalues. The transformation from A to G is performed implicitly by transforming the bidiagonal decomposition (2) of A instead of A itself and involve *no subtractions* (see Section 4);
3. Finally, we compute the eigenvalues λ_i of G to high relative accuracy as follows. We know from theory that $\mathrm{sign}(\lambda_i) = (-1)^{i-1}$ (see Section 2), thus we need to only compute the magnitudes $|\lambda_i|$. Since G is symmetric, it suffices to compute its singular values $\sigma_i = |\lambda_i|$. The matrices G and GJ have the same singular values, and GJ is *bidiagonal*. The singular values of the latter can then be easily computed to high relative accuracy (see, e.g., [5]).

This paper is organized as follows. In Section 2, we review some properties of TN^J matrices and their (bidiagonal) decompositions. In Section 3, we describe how to reduce a TN^J matrix to an anti-bidiagonal using similarity transformations. We describe how to do so in a subtraction-free fashion in Section 4. We present our main algorithm in Section 5 and present numerical experiments in Section 6. The conclusions and open problems are in Section 7.

2. Properties of TN^J matrices

In this section, we recall some properties of TN^J matrices and their decompositions [10].[3]

The eigenvalues λ_i of the TN^J matrices are always real whether the latter are symmetric or not. If we order the eigenvalue by magnitude

$$|\lambda_1| \geqslant |\lambda_2| \geqslant \cdots \geqslant |\lambda_n| > 0,$$

then $\mathrm{sign}(\lambda_i) = (-1)^{i-1}$ [10, Theorem 13, p. 269].

[3] Please note than the sign regular matrices are called “sign-definite” in [10].

The TN^J and TN matrices are closely related. From the Cauchy–Binet identity we immediately conclude that if A is TN^J, then AJ is nonsingular TN and vice versa (recall that $J \equiv [\delta_{i,n+1-j}]_{i,j=1}^n$ is the reverse identity). Since multiplication on the right by the reverse identity is equivalent to reversing the order of the columns, the TN^J matrices are "nonsingular TN matrices with columns in reverse order". The choice of reversing the *columns* of a TN matrix to obtain a TN^J matrix is unimportant—since A is TN^J whenever JAJ is, we may as well have reversed *the rows* of a TN matrix to obtain a TN^J one.

By using a representation of the nonsingular TN matrices as products of nonnegative bidiagonals [12] one obtains the following structure theorem for TN^J matrices.

Theorem 1. *A nonsingular $n \times n$ matrix A is* TN^J *if and only if it can be uniquely factored as*

$$A = L^{(1)} \cdots L^{(n-1)} \cdot D \cdot U^{(n-1)} \cdots U^{(1)} \cdot J, \tag{3}$$

where $D = \mathrm{diag}(d_1, d_2, \ldots, d_n)$, and $L^{(k)}$ and $U^{(k)}$ are lower and upper unit bidiagonal matrices with offdiagonal elements $l_i^{(k)}$ and $u_i^{(k)}$, respectively, such that

1. $d_i > 0$ *for all* i;
2. $l_i^{(k)} = u_i^{(k)} = 0$ *for* $i < n - k$;
3. $l_i^{(k)} \geqslant 0$, $u_i^{(k)} \geqslant 0$ *for* $i \geqslant n - k$;
4. $l_i^{(k)} = 0$ *implies* $l_{i+s}^{(k-s)} = 0$ *for* $s = 1, \ldots, k-1$; *and* $u_i^{(k)} = 0$ *implies* $u_{i+s}^{(k-s)} = 0$ *for* $s = 1, \ldots, k-1$.

The bidiagonal decomposition (3) is the ultimate unique representation of a TN^J matrix in this paper and we denote it by $\mathcal{BD}^J(A)$. The trick in our algorithm, as we will see, is that by transforming $\mathcal{BD}^J(A)$ instead of A, we end up performing no subtractions.

The idea of using bidiagonal decompositions with TN^J matrices comes from the theory of TN matrices [12] according to which any nonsingular TN matrix P can be written as

$$P = L^{(1)} \cdots L^{(n-1)} \cdot D \cdot U^{(n-1)} \cdots U^{(1)}. \tag{4}$$

The bidiagonal decompositions of TN matrices have proven extremely useful in designing accurate algorithms for those matrices: virtually all linear algebra with TN matrices can be performed accurately [15,16] starting with the decomposition (4). We follow the notation of this earlier work [15,16] and denote the decomposition (4) by $\mathcal{BD}(P)$ (the bidiagonal decompositions of TN matrices are thus denoted by $\mathcal{BD}$ and those of TN^J matrices by $\mathcal{BD}^J$).

Following [15, Section 4], it is convenient to store the n^2 nontrivial entries of the bidiagonal decomposition (3) as an $n \times n$ array $B = \mathcal{BD}^J(A)$ where

$$b_{ij} = \begin{cases} d_i, & i = j, \\ l_{i-1}^{(n-i+j)}, & i > j, \\ u_{j-1}^{(n-j+i)}, & i < j. \end{cases}$$

The bidiagonal decomposition $\mathcal{BD}^J(A)$ of a TN^J matrix A and the bidiagonal decomposition $\mathcal{BD}(AJ)$ of the TN matrix AJ is thus represented by the same $n \times n$ array B since the nontrivial entries in their corresponding bidiagonal decompositions (3) and (4), respectively, are the same. This should cause no confusion since we use the notation $\mathscr{BD}^J$ to denote the bidiagonal decomposition of TN^J matrices and $\mathscr{BD}$ to denote the bidiagonal decomposition of TN matrices.

There are other representations of TN (and thus of TN^J) matrices as products of bidiagonals. They are theoretically [12] and numerically [16] analogous, so the particular choice in Theorem 1 does not result in any loss of generality.

The reader may wonder why we choose to represent the TN^J matrices by the entries of their bidiagonal decompositions instead of their entries. The reasons are several. The matrix entries are a poor choice to represent a TN^J matrix, since small relative perturbations in the entries can lead to enormous relative perturbations in the eigenvalues. For example, an $\varepsilon > 0$ perturbation in the (1,2) entry from $1+\varepsilon$ to $1+2\varepsilon$ of the following 2×2 TN^J matrix:

$$\begin{bmatrix} 1 & 1+\varepsilon \\ 1 & 1 \end{bmatrix}$$

perturbs the smaller eigenvalue from about $-\varepsilon/2$ to about $-\varepsilon$. Unsurprisingly, the factorization (2) of this matrix is also poorly determined in its diagonal (second) factor

$$\begin{bmatrix} 1 & 1+\varepsilon \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{1+\varepsilon} & 1 \end{bmatrix} \begin{bmatrix} 1+\varepsilon & 0 \\ 0 & \frac{\varepsilon}{1+\varepsilon} \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{1+\varepsilon} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Therefore, in order to use our new eigenvalue algorithm, it is critical to first obtain $\mathscr{BD}^J(A)$ accurately. There are necessary and sufficient conditions for being able to compute accurate $\mathscr{BD}(A)$ (and thus $\mathscr{BD}^J(A)$), as well as determinantal formulas for the entries of $\mathscr{BD}^J(A)$ in [15, Section 3]. In particular, for several structured TN^J matrices resulting from reordering the columns of a TN matrix, such as Vandermonde, Cauchy–Vandermonde, generalized Vandermonde, Bernstein–Vandermonde, and Pascal matrices, accurate formulas for their bidiagonal decompositions $\mathscr{BD}^J$ are readily available [7,15–17]. We used these formulas to produce our numerical experiments for TN^J Vandermonde matrices in Section 6.

In contrast, the n^2 nontrivial entries of the bidiagonal decomposition $\mathscr{BD}^J$ of a TN^J matrix immediately reveal its TN^J structure (Theorem 1). Additionally, every eigenvalue is determined very accurately by the entries of $\mathscr{BD}^J$ which we formulate in the following proposition.

Proposition 1. *If A is TN^J then small δ $(0 < \delta \ll 1)$ componentwise relative perturbations in $\mathscr{BD}^J(A)$ can cause at most $2n^2\delta/(1-2n^2\delta)$ relative perturbations in any eigenvalue.*

Proof. The argument is analogous to that of Theorem 7.2 in [15], so we only sketch it here. Let x be any nontrivial entry of $\mathscr{BD}^J(A)$. From the Cauchy–Binet identity, any minor of A is a linear function of x with either nonpositive, or nonnegative coefficients. Either way, a δ relative perturbation in x causes at most a δ relative perturbation in any ith order minor of A and in turn at most a δ componentwise relative perturbation in $\mathscr{A}^{(i)}$—the ith compound matrix of A. Since $\mathscr{A}^{(i)}$ is either nonpositive or nonnegative, these perturbations cause at most a δ relative perturbation in its largest (in magnitude) eigenvalue $\lambda_1\lambda_2\cdots\lambda_i$ [9]. The same argument for $\mathscr{A}^{(i-1)}$ implies that the relative perturbation in $\lambda_1\lambda_2\cdots\lambda_{i-1}$ also does not exceed δ, and thus the relative perturbation in λ_i does not exceed $2\delta/(1-2\delta)$. Accumulating such perturbations in each of the n^2 entries of $\mathscr{BD}^J(A)$ gives us the final result. □

We refer the reader to [12] for a detailed account of the bidiagonal decompositions of TN matrices.

3. Reduction to a symmetric anti-bidiagonal

In this section, we constructively show that a TN^J matrix is similar to a nonnegative anti-bidiagonal matrix.[4] The anti-bidiagonal then has the same eigenvalues as a *symmetric* anti-bidiagonal matrix.

In Section 4, we will show how to preserve the relative accuracy in these transformations.

3.1. Reduction to an anti-bidiagonal

The first step is to similarity reduce the TN^J matrix to an anti-bidiagonal. For this task we use only two operations:

1. Subtracting a positive multiple of a row from the previous or next one in order to create a zero;
2. Adding a positive multiple of one column to the next or previous one.

Each of the above operations, written in matrix form, represents a multiplication by the unit bidiagonal matrix

$$E_i(x) = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & x & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}$$

or its transpose, $E_i^{\mathrm{T}}(x)$. The matrix $E_i(x)$ differs from the identity only in the entry x in position $(i, i-1)$. If $x > 0$ then $E_i(x)$ is TN. We will use the identities

$$E_i^{-1}(x) = E_i(-x) \quad \text{and} \quad J \cdot E_i(x) \cdot J = E_{n-i+2}^{\mathrm{T}}(x).$$

The matrices E_i play a major role in the theory of TN matrices (see. e.g., [12]) since every $L^{(k)}$ and $U^{(k)}$ in (3) is a product of E_i's and E_i^{T}'s, respectively.

In this section, it is more convenient to think of a TN^J matrix A as a product $A = PJ$ of a TN matrix P and the reverse identity J. The idea is to apply the similarity transformations in such a way that they affect only the TN factor P, preserving its TN structure, eventually reducing it to an upper bidiagonal.

First, we apply similarity transformations to reduce the factor P in $A = PJ$ to upper triangular form.

We create zeros in the lower triangular part of P by columns from the bottom up by starting with the $(n, 1)$ entry. Here we present the procedure to eliminate the entry in position (i, j) (where $i > j$) and assume that $p_{i-1,j}$ and p_{ij} are the only nonzero entries in the submatrix $P(i-1{:}n, 1{:}j)$.

We subtract a multiple of row $i-1$ from row i by multiplying on the left by the matrix $E_i(-x)$ (where $x = p_{ij}/p_{i-1,j}$). We then complete the similarity transformation on the right

$$PJ \to E_i(-x)PJE_i^{-1}(-x) = E_i(-x)P(JE_i(x)J)J = E_i(-x) \cdot P \cdot E_{n-i+2}^{\mathrm{T}}(x) \cdot J.$$

[4] The idea was inspired by the similarity reduction of a matrix to a tridiagonal described in [13].

The similarity transformation can thus be seen as an operation on the TN factor P only

$$P \to E_i(-x) \cdot P \cdot E^{\mathrm{T}}_{n-i+2}(x). \tag{5}$$

In this transformation, forming the product $E_i(-x)P$ means creating a zero in P in position (i, j) by subtracting of the appropriate multiple of row $i-1$ from row i. Completing the similarity by multiplying P on the right by $E^{\mathrm{T}}_{n-i+2}(x)$ means adding a positive multiple of the $(n-i+1)$st column to the following one—the $(n-i+2)$nd—which does not disturb the newly created zero in position (i, j) nor the zeros previously created in the lower triangular part of P. The results in [11] imply that this similarity transformation preserves the TN structure of P.

Once P is upper triangular, we similarity transform the product PJ into a product of a nonnegative upper bidiagonal matrix and the reverse identity. We do so by introducing zeros in P by columns from the top down starting with the last column.

Next, we describe the similarity transformation that creates a zero in P above the first superdiagonal in position $(i, j), i < j-1$, assuming that p_{ij} and $p_{i+1,j}$ are the only nonzero entries in the submatrix $P(1{:}\,i+1, j{:}\,n)$.

We create the zero in position (i, j) in P by subtracting the appropriate multiple of the $(i+1)$st row from the ith. In other words, we form the product $E^{\mathrm{T}}_{i+1}(-x) \cdot (PJ)$, where $x = p_{ij}/p_{i+1,j}$. We then complete the similarity on the right as before

$$\begin{aligned} PJ \to E^{\mathrm{T}}_{i+1}(-x)PJE^{-\mathrm{T}}_{i+1}(-x) &= E^{\mathrm{T}}_{i+1}(-x)P\big(JE^{\mathrm{T}}_{i+1}(x)J\big)J \\ &= E^{\mathrm{T}}_{i+1}(-x)PE_{n-i+1}(x)J. \end{aligned}$$

The similarity thus amounts to the transformation

$$P \to E^{\mathrm{T}}_{i+1}(-x)PE_{n-i+1}(x). \tag{6}$$

While the multiplication of P on the left by $E^{\mathrm{T}}_{i+1}(-x)$ creates the desired zero in position (i, j), the multiplication on the right by $E_{n-i+1}(x)$ represents the addition of a positive multiple of the $(n-i+1)$st column in P to the previous one—the $(n-i)$th—thus introducing an undesired "bulge" just below the diagonal in position $(n-i+1, n-i)$.

This bulge is only a minor nuisance—we kill it as described in the first step above. Killing the bulge does not disturb any of the zeros we have created and especially does not disturb the newly created zero in position (i, j) in P.

We continue this process until P is reduced to a (nonnegative, nonsingular) upper bidiagonal matrix.

With the thus transformed P, the matrix $C \equiv PJ$ is nonnegative, anti-bidiagonal, TN^J matrix which is similar to A.

3.2. Making the anti-bidiagonal symmetric

The last step is to make C symmetric. It consists of two parts. First, we make the zeros just above the antidiagonal symmetric (this part, if necessary, is not a similarity, but regardless, it does not change the eigenvalues of C). Then we apply a diagonal similarity transformation to the thus changed C to make it symmetric.

Next, we show that if an entry $c_{i,n-i}$ in C just above the antidiagonal is zero, then setting the corresponding entry $c_{n-i,i}$ on the other side of the main diagonal to zero does not change the eigenvalues of C.

Lemma 1. *Let C be a* TN^J *anti-bidiagonal matrix such that* $c_{i,n-i} = 0$ *for some* $i \in \{1, 2, \ldots, n-1\}$ *and* $c_{n-i,i} \neq 0$. *Let G be obtained from C by setting* $c_{n-i,i}$ *to zero. Then C and G have the same eigenvalues.*

Proof. The proof becomes nearly obvious once we consider the tridiagonal matrices $F = C^2$ and $H = G^2$. The matrices F and G only differ in positions $(i+1, i)$ and $(n-i, n-i+1)$, which are zero in G and nonzero in F. The values of these entries have no bearing on the eigenvalues, since both F and G are reducible ($c_{i,n-i} = g_{i,n-i} = 0$ imply $f_{i,i+1} = f_{n-i+1,n-i} = 0 = h_{i,i+1} = h_{n-i+1,n-i}$). Therefore, the eigenvalues of C^2 and G^2 are the same. We conclude the proof by observing that the signs of the eigenvalues of C and G are the same (since both are TN^J). □

Once the zeros above the anti-bidiagonal of C are in symmetric pairs, we make C symmetric by replacing the elements on the antidiagonal, $c_{i,n-i+1}$, by $\sqrt{c_{i,n-i+1}c_{n-i+1,i}}$ for $i = 1, 2, \ldots, n$, and the ones just above them, $c_{i,n-i}$, by $\sqrt{c_{i,n-i}c_{n-i,i}}$ for $i = 1, 2, \ldots, n-1$. This is a simple diagonal similarity which, again, does not affect the eigenvalues of C.

4. Accurate reduction to a symmetric anti-bidiagonal

In this section, we describe how to accurately apply the transformations from the previous section by implicitly transforming the bidiagonal decomposition $\mathscr{BD}^J$ without performing any subtractions.

To this end, we need to show how to accurately perform the transformations (5) and (6):

$$P \to E_i(-x) \cdot P \cdot E_{n-i+2}^{\mathrm{T}}(x) \quad \text{and} \quad P \to E_{i+1}^{\mathrm{T}}(-x) P E_{n-i+1}(x).$$

The transformation $P \to E_i(-x) \cdot P$ is equivalent to setting an entry in $\mathscr{BD}(P)$ to zero (see [15, Section 4.1]). The transformation $P \to P \cdot E_{n-i+2}^{\mathrm{T}}(x)$ is performed using the procedure `TNAddToNext` from [16]. The similarity (5) is thus performed as

$$x = b_{ij}, \quad b_{ij} = 0, \quad B = \texttt{TNAddToNext}(B^{\mathrm{T}}, x, n-i+2)^{\mathrm{T}}. \tag{7}$$

For the similarity (6), the transformation $P \to P \cdot E_{n-i+1}(x)$ is performed using Algorithm 4.2 from [15].

Finally, we show how to perform the transformation $P \to E_{i+1}^{\mathrm{T}}(-x) \cdot P$, where P is upper triangular and TN.

Let $\mathscr{BD}(P)$ be given as

$$P = DU^{(n-1)}U^{(n-2)} \cdots U^{(1)}. \tag{8}$$

We demonstrate only the first step, i.e., how to kill the $(1, n)$ entry in P, the rest being analogous. In other words, let P' be obtained from P by subtracting a multiple, $x = p_{1n}/p_{2n}$, of the second row from the first in order to create a zero in position $(1, n)$. We show how to compute $\mathscr{BD}(P')$ from $\mathscr{BD}(P)$ in $\mathscr{O}(n)$ operations without performing any subtractions (from the theory developed in [11] we know that P' is also TN).

For that purpose we "push" the factor $U^{(1)} = E_n^{\mathrm{T}}(x_1)$, where $x_1 = u_{n-1}^{(1)}$, to the left of the factorization (8):

$$\begin{aligned} P &= DU^{(n-1)}U^{(n-2)} \cdots U^{(3)}U^{(2)}U^{(1)} \\ &= DU^{(n-1)}U^{(n-2)} \cdots U^{(3)}\underline{U^{(2)}E_n^{\mathrm{T}}(x_1)} \\ &= DU^{(n-1)}U^{(n-2)} \cdots \underline{U^{(3)}E_{n-1}^{\mathrm{T}}(x_2)}\overline{U}^{(2)} \end{aligned} \tag{9}$$

$$
\begin{aligned}
&= DU^{(n-1)}U^{(n-2)}\cdots E_{n-2}^{\mathrm{T}}(x_3)\overline{U}^{(3)}\overline{U}^{(2)}\\
&= \cdots\\
&= D\underline{U^{(n-1)}E_3^{\mathrm{T}}(x_{n-2})}\overline{U}^{(n-2)}\cdots\overline{U}^{(3)}\overline{U}^{(2)}\\
&= \underline{DE_2^{\mathrm{T}}(x_{n-1})}\overline{U}^{(n-1)}\overline{U}^{(n-2)}\cdots\overline{U}^{(3)}\overline{U}^{(2)} \qquad (10)\\
&= E_2^{\mathrm{T}}(x_n)D\overline{U}^{(n-1)}\overline{U}^{(n-2)}\cdots\overline{U}^{(3)}\overline{U}^{(2)}. \qquad (11)
\end{aligned}
$$

The matrices being transformed on each step are underlined. Each transformation step from (9) to (10) proceeds according to the identity

$$
E_{i-1}^{\mathrm{T}}(x_{k+1})\cdot\begin{bmatrix}1 & \bar{u}_1 & & \\ & \ddots & \ddots & \\ & & 1 & \bar{u}_{n-1}\\ & & & 1\end{bmatrix} = \begin{bmatrix}1 & u_1 & & \\ & \ddots & \ddots & \\ & & 1 & u_{n-1}\\ & & & 1\end{bmatrix}\cdot E_i^{\mathrm{T}}(x_k), \qquad (12)
$$

where

$$
x_{k+1} = \frac{u_{i-2}x_k}{u_{i-1}+x_k}, \quad \bar{u}_{i-2} = \frac{u_{i-2}u_{i-1}}{u_{i-1}+x_k}, \quad \text{and} \quad \bar{u}_{i-1} = u_{i-1}+x_k. \qquad (13)
$$

The formulas (13) are easily verified by writing (12) as

$$
\begin{bmatrix}\ddots & & & & \\ & 1 & \bar{u}_{i-2}+x_{k+1} & \bar{u}_{i-1}x_{k+1} & \\ & & 1 & \bar{u}_{i-1} & \\ & & & 1 & \\ & & & & \ddots\end{bmatrix} = \begin{bmatrix}\ddots & & & & \\ & 1 & u_{i-2} & u_{i-2}x_k & \\ & & 1 & u_{i-1}+x_k & \\ & & & 1 & \\ & & & & \ddots\end{bmatrix}.
$$

The final step (11) is easy; for $x_n = x_{n-1}d_1/d_2$ we have

$$
E_2^{\mathrm{T}}(x_n)D = DE_2^{\mathrm{T}}(x_{n-1}).
$$

Now $x_n = p_{1n}/p_{2n}$ is the multiple needed to kill the $(1, n)$ entry of P by subtracting that multiple of the second row from the first. Therefore, the bidiagonal decomposition $\mathcal{BD}(P')$ is

$$
P' = E_2^{\mathrm{T}}(-x_n)P = D\overline{U}^{(n-1)}\overline{U}^{(n-2)}\cdots\overline{U}^{(3)}\overline{U}^{(2)}. \qquad (14)
$$

We utilize the same idea in the further reduction of the matrix P to bidiagonal form by extracting factors E_i^{T} from the right of the factorization (14) and pushing them through all the way on the left.

Algorithm 1 (`TNSubtractFromPrevious`). Given the bidiagonal decomposition $B = \mathcal{BD}(P)$ of an upper triangular TN matrix P, the following subtraction-free algorithm computes $\mathcal{BD}(P')$, where P' is obtained from P by subtracting the multiple $p_{ij}/p_{i+1,j}$ of the $(i+1)$st row from the ith in order to create a zero in P in position (i, j), $i < j - 1$. We further assume that p_{ij} and $p_{i+1,j}$ are the only nonzero entries in the upper right corner submatrix $P(1{:}\,i+1, j{:}\,n)$.

```
function $[B, x]$ =TNSubtractFromPrevious $(B, i, j)$
$x = b_{ij}$
$b_{ij} = 0$
for $k = j-1 : -1 : i+1$
    $z = b_{ik}/(b_{i+1,k+1} + x)$
```

```
        $b_{ik} = b_{i+1,k+1}z$
        $b_{i+1,k+1} = b_{i+1,k+1} + x$
        $x = zx$
    end
    $x = xb_{ii}/b_{i+1,i+1}$
```

The cost of Algorithm 1 is O(n) operations.

5. Eigenvalue algorithm

In this section, we present our main TN^J eigenvalue algorithm. It takes as input the bidiagonal decomposition $\mathscr{BD}^J(A)$ of a TN^J matrix A and returns its eigenvalues to high relative accuracy.

It applies the transformations of Section 3 implicitly to $\mathscr{BD}^J(A)$ as described in Section 4 reducing A to a symmetric anti-bidiagonal matrix. The singular values $\sigma_i = |\lambda_i|$ of the latter are computed to high relative accuracy using the LAPACK [2] routine DLASQ1. The signs of the eigenvalues we know from theory (see Section 2): $\mathrm{sign}(\lambda_i) = (-1)^{i-1}$ and do not compute them.

Algorithm 2 (TN^{J}Eigenvalues). Given the decomposition $B = \mathscr{BD}^J(A)$ of the TN^J matrix A, the following algorithm computes all eigenvalues of A to high relative accuracy in $\mathscr{O}(n^3)$ time:

```
    function TN$^J$Eigenvalues(B)
    $n$ =size of the matrix
    … first kill the lower triangular part
    for $j = 1 : n - 1$
        for $i = n : -1 : j + 1$
            kill $b_{ij}$ using the similarity (7)
        end
    end
    …then kill everything above the superdiagonal
    for $j = n: -1: 3$
        for $i = 1: j - 2$
            … kill $b_{ij}$ using the similarity (6):
            $[B, x]$ =TNSubtractFromPrevious($B, i, j$)
            $B$ =TNAddToPrevious($B, x, 1, n - i + 1$)
            kill the bulge $b_{n-i+1,n-i}$ using the similarity (7)
        end
    end
    … $B$ is now the $\mathscr{BD}$ of a bidiagonal; next, recover that bidiagonal:
    for $i = 1: n - 1$
        $b_{i,i+1} = b_{i,i+1}b_{ii}$
    end
    … make the anti-bidiagonal $BJ$ symmetric:
```

```
for $i = 1, 2, \ldots, \lfloor n/2 \rfloor$
    replace $b_{ii}$ and $b_{n-i+1,n-i+1}$ by $\sqrt{b_{ii} b_{n-i+1,n-i+1}}$
    replace $b_{i,i+1}$ and $b_{n-i,n-i+1}$ by $\sqrt{b_{i,i+1} b_{n-i,n-i+1}}$
end
Compute the singular values $\{\sigma_i\}_{i=1}^n$ of $B$ using DLASQ1
The eigenvalues are $\{(-1)^{i-1}\sigma_i\}_{i=1}^n$
```

The cost of Algorithm 2 is $O(n^3)$.

The error analysis of Algorithm 2 is identical to that of Algorithm 5.1 in [15]. The main argument, again, is that every arithmetic operation of Algorithm 2 results in an at most ε relative error of one entry of the bidiagonal decomposition $\mathscr{B}\mathscr{D}^J$ of some intermediate TN^J matrix. Each such error causes at most a $2\varepsilon/(1-2\varepsilon)$ perturbation in any eigenvalue (see Section 2 and [15, Section 7]). Accumulating the perturbations throughout the algorithm results in an at most $O(n^3)\varepsilon$ error in the final computed eigenvalues.

6. Numerical experiments

We performed a number of numerical experiments and verified the correctness and accuracy of Algorithm 2, an implementation of which we made available online [14]. We report one of our experiments here.

We computed the eigenvalues of the Vandermonde matrix $V = [x_i^{j-1}]_{i,j=1}^{40}$ with nodes $(x_1, x_2, \ldots, x_{40}) = (4, 3.9, 3.8, \ldots, 0.1)$. The matrix V a fairly ill conditioned (nonsymmetric)

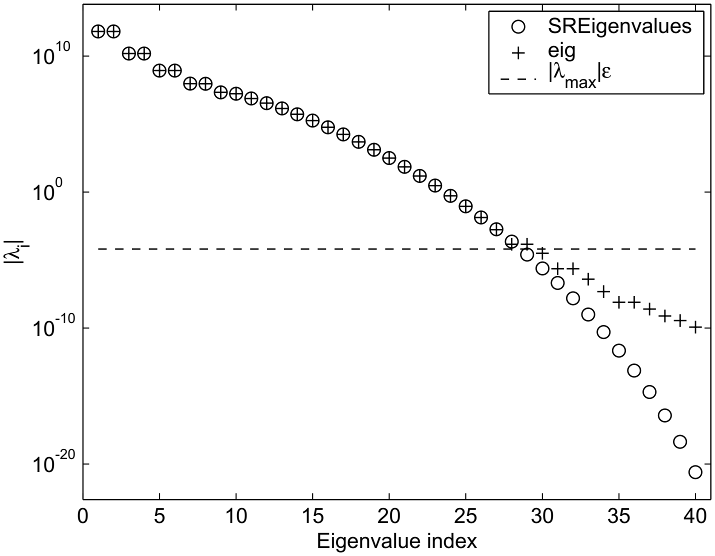

Fig. 1. The absolute values of the eigenvalues the 40×40 Vandermonde matrix with nodes $(4.0, 3.9, \ldots, 0.1)$. "O" = accurate, "+" = conventional.

TN^J matrix ($\kappa_2(V) = 2.4 \times 10^{44}$). The entries of $\mathscr{BD}^J(V)$ were computed to high relative accuracy using the explicit formulas in [15, Section 3].

We compared the output of our Algorithm 2 with that of a conventional eigenvalue algorithm as implemented in the MATLAB [18] command `eig` both in double precision arithmetic and in 80-decimal-digit arithmetic using the MATLAB command `vpa`. The output of Algorithm 2 agreed with the one computed using 80-decimal-digit arithmetic to at least 14 decimal significant digits confirming the accuracy of Algorithm 2.

Since the signs of the eigenvalues of V are not in question, in Fig. 1 we plotted the absolute values of the eigenvalues of V computed by Algorithm 2 and those computed by `eig` in double precision arithmetic.

We observe, as expected, that `eig` in double precision computes only the largest eigenvalues of V accurately. The lack of symmetry in V and the effect of the angle between the left and right eigenvectors explains the deterioration of the accuracy in the eigenvalue closest to, but larger than $|\lambda_{\max}|\varepsilon$ in absolute value.

7. Conclusions and open problems

We presented a new accurate algorithm for computing all eigenvalues of TN^J matrices. This is the second accurate algorithm for sign regular matrices after Algorithm 5.1 from [15] for TN matrices (the TN matrices are also sign-regular with $\rho_k = 1$ for all k). These two algorithms handle 4 out of the 2^n different classes of nonsingular sign regular matrices (the TN^J and TN matrices, and the negatives thereof).

The eigenvector matrices can certainly be accumulated in Algorithm 2 as products of factors each of which is highly accurate in the appropriate sense. This process, however, is prone to massive subtractive cancellation and we guarantee no accuracy in the computed eigenvectors.

The problem of computing accurate eigenvalues of sign regular matrices with other "signatures" $\{\rho_k\}$ remains open. One major obstacle is that there is no known parameterization of the sign-regular matrices with other signatures. In fact, we know of no efficient way to even generate sign regular matrices other than TN, TN^J or their negatives. The only method we are aware of is the limiting argument in Gantmacher and Krein [10, Chapter V, Section 5], which does not appear to be numerically feasible.

The eigenvalue problem for singular TN^J (as well as TN) matrices is also open. The singular TN and TN^J matrices do not possess unique bidiagonal decompositions, but it appears likely that accurate algorithms can still be designed—this is a topic of current research.

The solutions to the TN^J singular value, QR, least squares, and linear systems problems follow trivially from the solutions to the corresponding problems for TN matrices and can be found in the papers [15,16].

References

[1] A.S. Alfa, J. Xue, Q. Ye, Entrywise perturbation theory for diagonally dominant M-matrices with applications, Numer. Math. 90 (2002) 401–414.

[2] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, LAPACK Users' Guide, Software Environ. Tools 9, third ed., SIAM, Philadelphia, 1999.

[3] J. Demmel, Accurate singular value decompositions of structured matrices, SIAM J. Matrix Anal. Appl. 21 (2) (1999) 562–580.

[4] J. Demmel, M. Gu, S. Eisenstat, I. Slapničar, K. Veselić, Z. Drmač, Computing the singular value decomposition with high relative accuracy, Linear Algebra Appl. 299 (1–3) (1999) 21–80.

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

[6] J. Demmel, P. Koev, Accurate SVDs of weakly diagonally dominant *M*-matrices, Numer. Math. 98 (1) (2004) 99–104.

[7] J. Demmel, P. Koev, The accurate and efficient solution of a totally positive generalized Vandermonde linear system, SIAM J. Matrix Anal. Appl. 27 (1) (2005) 142–152.

[8] F. Dopico, P. Koev, Accurate symmetric rank revealing and eigen decompositions of symmetric structured matrices, SIAM J. Matrix Anal. Appl. 28 (4) (2006) 1126–1156.

[9] L. Elsner, I. Koltracht, M. Neumann, D. Xiao, On accurate computations of the Perron root, SIAM J. Matrix Anal. Appl. 14 (2) (1993) 456–467.

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

[11] M. Gasca, J.M. Peña, Total positivity and Neville elimination, Linear Algebra Appl. 165 (1992) 25–44.

[12] M. Gasca, J.M. Peña, On factorizations of totally positive matrices, Total Positivity and its Applications, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996, pp. 109–130.

[13] G.A. Geist, G.W. Howell, D.S. Watkins, The BR eigenvalue algorithm, SIAM J. Matrix Anal. Appl. 20 (4) (1999) 1083–1098.

[14] P. Koev. <http://www-math.mit.edu/∼plamen>.

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

[16] P. Koev, Accurate computations with totally nonnegative matrices, SIAM J. Matrix Anal. Appl. in press.

[17] J.J. Martínez, J.M. Peña, Factorizations of Cauchy–Vandermonde matrices, Linear Algebra Appl. 284 (1–3) (1998) 229–237.

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

[19] J.M. Peña, LDU decompositions with L and U well conditioned, ETNA (Electron. Trans. Numer. Anal.) 18 (2004) 198–208.