SIAM J. Matrix Anal. Appl.
Vol. , No. , pp. 1–27

qd-TYPE METHODS FOR QUASISEPARABLE MATRICES*

ROBERTO BEVILACQUA[†], ENRICO BOZZO[‡], and GIANNA M. DEL CORSO[†]

Abstract. In the last few years many numerical techniques for computing eigenvalues of structured rank matrices have been proposed. Most of them are based on QR iterations since, in the symmetric case, the rank structure is preserved and high accuracy is guaranteed. In the unsymmetric case, however, the QR algorithm destroys the rank structure, which is instead preserved if LR iterations are used. We consider a wide class of quasiseparable matrices which can be represented in terms of the same parameters involved in their Neville factorization. This class, if assumptions are made to prevent possible breakdowns, is closed under LR steps. Moreover, we propose an implicit shifted LR method with a linear cost per step, which resembles the qd method for tridiagonal matrices. We show that for totally nonnegative quasiseparable matrices the algorithm is stable and breakdowns cannot occur if the Laguerre shift, or other shift strategy preserving nonnegativity, is used. Computational evidence shows that good accuracy is obtained also when applied to symmetric positive definite matrices.

Key words. qd algorithms, LR for eigenvalues, quasiseparable matrices

AMS subject classification. 65F15

DOI. 10.1137/100794481

1. Introduction.

In the recent literature quasiseparable matrices have received a great deal of interest (see [8], [9], [4], and references therein). In fact, matrices with this structure appear naturally in many application fields such as systems theory, signal processing, or integral equations. Also covariance matrices, or matrices involved in multivariate statistics or discretization of elliptic PDEs, often have the quasiseparable structure.

The class of quasiseparable matrices includes many important matrices such as companion matrices of polynomials, tridiagonal matrices and their inverses (Green's quasiseparable), unitary Hessenberg matrices, and banded matrices. In [8] the class is proved to be closed under inversion, and a linear complexity inversion method is proposed.

An interesting research topic is the development of fast algorithms, both for the solution of linear systems and for eigenvalue and eigenvector computations, taking advantage of the representation of the matrix in terms of a small number of parameters.

The main purpose of this paper is to propose an LR scheme for eigenvalue computations of a quasiseparable matrix not necessarily Hermitian. In fact, for unsymmetric quasiseparable matrices, it is well known that the QR algorithm destroys the rank structure with an increase of the cost of the computation of the eigenvalues. The LR algorithm, on the contrary, maintains the rank structure providing a valid alternative once the stability is guaranteed. The main objection to the use of LR iterations is the possible instability. However, Fernando and Parlett [10] and Parlett in [17] suggested applying the LR algorithm to symmetric positive definite tridiagonal matrices, showing the good

*Received by the editors May 7, 2010; accepted for publication (in revised form) by M. Van Barel April 19, 2011; published electronically DATE. This research was partially supported by the Italian Ministry for Education, University and Scientific Research under program PRIN 2008: 20083KLJEZ.

http://www.siam.org/journals/simax/x-x/79448.html

[†]Department of Computer Science, Università di Pisa, Largo Pontecorvo, 3, 56127 Pisa, Italy (bevilacq@di.unipi.it, delcorso@di.unipi.it).

[‡]Department of Mathematics and Computer Science, Università di Udine, Via delle Scienze, 206, 33100 Udine, Italy (enrico.bozzo@uniud.it).

performance and stability of the qd-type methods over the standard QR method. The idea behind qd-type algorithms, first proposed by Rutishauser [20], is to represent tridiagonal matrices as the product of the bidiagonal factors of the LU factorization and to update the bidiagonal factors with formulas requiring only quotients and sums. The algorithm is highly accurate and has become one of LAPACK's main tools for computing eigenvalues of symmetric tridiagonal matrices. Since many interesting algorithms for semiseparable and quasiseparable matrices have been derived from similar techniques employed on tridiagonal matrices (see, for example [16], [19], [23]), our idea is to design an algorithm inspired by the qd-type algorithms. While these algorithms for tridiagonals perform well on symmetric positive definite matrices, it turns out that the methods we propose in this paper achieve a high accuracy and stability when applied to totally nonnegative (TN) quasiseparable matrices.

The association between TN and quasiseparable matrices was recently made by Dopico, Bella, and Olshevsky in two different talks [6], [7] and by Gemignani in [14]. They presented necessary and sufficient conditions to verify if a quasiseparable matrix is TN, and they proposed fast and stable algorithms for the solution of linear systems. In [14] the more general case of order-r semiseparable matrices has been exploited.

A historical example of a TN and quasiseparable matrix is the discrete Green function for a string with both ends fastened [11].

For TN matrices, we are able to prove that the algorithms here proposed for the computation of the eigenvalues are subtraction-free and turn out to be very effective when combined with the Laguerre shift strategy. The formulation of the algorithms in terms of recurrences, where some intermediate variables are introduced to avoid possible cancellations, makes these methods similar to the qd-type algorithms for tridiagonal matrices. For quasiseparable, TN matrices Gemignani in [14], following an idea of Koev [15], sketched an algorithm for the reduction into a similar tridiagonal form. We extend this algorithm for the matrices that we call Neville-representable (see section 3), showing the effectiveness when associated with a qd scheme.

The paper is organized as follows. In section 2 some preliminary definitions and results are provided. In section 3 the class of Neville-representable quasiseparable matrices is introduced, and structural results for the L and R factors of the LU factorization of matrices in this class are given. A complete characterization of the class of the Neville-representable quasiseparable matrices is given in terms of the generators of the quasiseparable matrix. Section 4 contains a description of the shifted LR iterations and theoretical results about the preservation of the structure. The study of the numerical stability and of the computational cost of the proposed methods is addressed in section 5. In section 6, as an alternative for the computation of the eigenvalue, we show a tridiagonalization procedure that can be followed by qd-type iterations as well as any other eigensolver for unsymmetric tridiagonal matrices. Section 7 contains the numerical experiments. In particular, we tested our methods on both random unsymmetric matrices and TN matrices. The results show a good performance in terms of time required and accuracy achieved, also for matrices not TN. A comparison for symmetric matrices with EIGSSD routine[1]—implementing implicit QR steps—is performed, showing the better behavior of our methods still achieving a comparable accuracy.

2. Preliminary results.

In this section we present some preliminary results that will be useful in the remaining parts of the paper.

[1] EIGSSD is a function included in the MATLAB package SSPACK, accompanying the book [25], and it is available online at http://www.cs.kuleuven.be/~mase/.

In the paper we will use the MATLAB-like notation to denote the lower (upper) triangular part of a matrix. In particular, $\mathrm{tril}(A, k)$ denotes the lower triangular part of the matrix A below and including the kth diagonal, that is, $\mathrm{tril}(A, k) = \{a_{ij} | j - i \leq k\}$, and $\mathrm{triu}(A, k)$ denotes the upper triangular part of the matrix A above and including the kth diagonal, that is, $\mathrm{triu}(A, k) = \{a_{ij} | j - i \geq k\}$. Note that the index k in these definitions can also be negative; hence $k = 0$ is the main diagonal, $k > 0$ is above the main diagonal, and $k < 0$ is below the main diagonal. Similarly $\mathrm{diag}(\mathbf{v}, k)$ denotes the matrix with the elements of vector $\mathbf{v}$ on the kth diagonal. We denote by $v_{i:j}$ the partial product of the entries of vector $\mathbf{v}$, from index i to index j, that is, $v_{i:j} = v_i v_{i+1} \cdots v_j$, where $i \leq j$.

Definition 1. *An $n \times n$ matrix S is called a semiseparable matrix if the following properties are satisfied:*

$$\mathrm{rank}\, S(i\colon n, 1\colon i) \leq 1, \qquad \mathrm{rank}\, S(1\colon i, i\colon n) \leq 1 \quad for\ i = 1, \ldots, n-1.$$

All semiseparable matrices $S = (s_{ij})$ can be represented by using six vectors $\mathbf{u}$, $\mathbf{v}$, $\mathbf{t}$, $\mathbf{p}$, $\mathbf{q}$, and $\mathbf{r}$ in this way (see Definition 2.14 in [24]):

$$s_{ij} = \begin{cases} u_i t_{i-1:j} v_j, & 1 \leq j < i \leq n, \\ u_i v_i = p_i q_i, & 1 \leq j = i \leq n, \\ p_i r_{i:j-1} q_j, & 1 \leq i < j \leq n. \end{cases} \tag{2.1}$$

Definition 2. *A matrix S is called a generator-representable semiseparable matrix if there exist four vectors $\mathbf{u}$, $\mathbf{v}$, $\mathbf{p}$, and $\mathbf{q}$ such that*

$$s_{ij} = \begin{cases} u_i v_j, & 1 \leq j < i \leq n, \\ u_i v_i = p_i q_i, & 1 \leq j = i \leq n, \\ p_i q_j, & 1 \leq i < j \leq n. \end{cases} \tag{2.2}$$

In the case S is irreducible, t and r in (2.1)can be chosen as unit vectors, and S is then generator-representable. If some t_i or r_i is zero, S is reducible. However, if S is reducible but symmetric or triangular, then it can always be expressed as the direct sum of two or more generator-representable matrices. See [2] for the details.

The class of matrices that can be represented as the sum of a semiseparable and a diagonal matrix is called the class of the *semiseparable plus diagonal* matrices.

In this paper a generalization of the semiseparable plus diagonal matrices is considered, that is, the class of quasiseparable matrices introduced in [8], [21].

There are many definitions of quasiseparable matrices [24]. The most general is the following.

Definition 3. *An $n \times n$ matrix S is called a quasiseparable matrix if the following conditions are satisfied:*

$$\mathrm{rank}\, S(i+1\colon n, 1\colon i) \leq 1, \qquad \mathrm{rank}\, S(1\colon i, i+1\colon n) \leq 1 \quad for\ i = 1, \ldots, n-1.$$

This definition captures semiseparable matrices, tridiagonal matrices, and semiseparable plus diagonal matrices. Note that singular matrices as well as block diagonal matrices are included in the class, while this does not happen if other definitions are chosen.

A very convenient way to represent quasiseparable matrices is the one introduced in [5], [8]. The Givens-vector representation as well as the generator representation can both be considered as special cases of this quasiseparable representation [24].

An unsymmetric quasiseparable matrix A can be expressed by means of $7n-8$ parameters, as follows:

$$A = \begin{pmatrix} \delta_1 & q_2p_1 & q_3r_2p_1 & q_4r_{3:2}p_1 & \cdots & q_nr_{n-1:2}p_1 \\ u_2v_1 & \delta_2 & q_3p_2 & q_4r_3p_2 & \cdots & q_nr_{n-1:3}p_2 \\ u_3t_2v_1 & u_3v_2 & \delta_3 & q_4p_3 & \cdots & q_nr_{n-1:4}p_3 \\ u_4t_{3:2}v_1 & u_4t_3v_2 & u_4v_3 & \ddots & & \vdots \\ \vdots & \vdots & \vdots & & \delta_{n-1} & q_np_{n-1} \\ u_nt_{n-1:2}v_1 & u_nt_{n-1:3}v_2 & u_nt_{n-1:4}v_3 & \cdots & u_nv_{n-1} & \delta_n \end{pmatrix}. \tag{2.3}$$

Note that the redundancy of parameters allows us to express quasiseparable matrices with zero subblocks. The relevance of this representation is proved by the following theorem proven in [24].

THEOREM 1. *A matrix A is quasiseparable if and only if it is representable as in* (2.3)*.*

In what follows, we prove some preliminary results about representations of quasiseparable matrices, which will be useful in the remaining part of the paper.

COROLLARY 2. *A quasiseparable matrix A can be decomposed as $A = S^{(u)} + Q$, where Q is a lower triangular matrix and*

$$S^{(u)} = \left[\begin{array}{c|ccc} 0 & & & \\ \vdots & & S_{n-1} & \\ 0 & & & \\ \hline 0 & 0 & \cdots & 0 \end{array}\right].$$

S_{n-1} is an $(n-1)\times(n-1)$ symmetric semiseparable matrix, representable with parameters $\mathbf{q} = (q_2, \ldots, q_n)^T$, $\mathbf{p} = (p_1, \ldots, p_{n-1})^T$, $\mathbf{r} = (r_2, \ldots, r_{n-1})^T$, and in view of the symmetry, $\mathbf{u} = \mathbf{q}$, $\mathbf{v} = \mathbf{p}$, and $\mathbf{t} = \mathbf{r}$ (see (2.3)*). Similarly $A = S^{(l)} + P$, where $S^{(l)}$ embeds a symmetric semiseparable matrix of size $n-1$ with zeros in the first row and in the last column, and P is upper triangular.*

Proof. From (2.3) we see that the upper right $(n-1)\times(n-1)$ minor of A has a semiseparable structure (2.1) in the upper triangular part. We set $S_{n-1} = \mathrm{triu}(A,1) + \mathrm{tril}(A^T,-2)$. Matrix Q is defined as the difference between A and $S^{(u)}$, and it is easy to see that it is lower triangular. □

LEMMA 3. *If a quasiseparable matrix A is such that $r_i \neq 0$ for $i = 2, \ldots, n-1$ in the representation* (2.3)*, then A can be decomposed into the sum of a lower triangular matrix and a rank-one matrix.*

Proof. Using Corollary 2, $A = S^{(u)} + Q$. If $r_i \neq 0$, we can define the vectors $\hat{\mathbf{q}}$ and $\hat{\mathbf{p}}$ as follows:

$$\begin{cases} \hat{q}_1 = 0, \\ \hat{q}_2 = q_2, \\ \hat{q}_i = q_ir_{i-1}r_{i-2}\cdots r_2, \quad i = 3, \ldots, n, \end{cases} \qquad \begin{cases} \hat{p}_1 = p_1, \\ \hat{p}_i = p_i/(r_ir_{i-1}\cdots r_2), \quad i = 2, \ldots, n-1, \\ \hat{p}_n = 0. \end{cases}$$

We have $S^{(u)} = \hat{\mathbf{p}}\hat{\mathbf{q}}^T + Z$, where Z is a lower triangular matrix. Then $A = \hat{\mathbf{p}}\hat{\mathbf{q}}^T + K$ with $K = Q + Z$, and then it is the sum of the rank-one matrix $\hat{\mathbf{p}}\hat{\mathbf{q}}^T$ and the lower triangular matrix K. □

3. The Neville representation.

3. The Neville representation. Neville elimination is a classical elimination technique which, differently from the standard Gaussian method, uses consecutive rows (columns) to reduce a matrix into an upper (lower) triangular form. When this elimination can be completely accomplished over rows and columns to reduce the matrix to diagonal form without interchanges, its formulation in terms of Gauss elementary matrices allows us to represent the matrix as the product of $O(n)$ bidiagonal matrices. These factors give the *Neville representation* of the matrix, which is called *Neville-representable.* Neville elimination for rank-structured matrices is considered in [14].

In this section we introduce a subclass of quasiseparable matrices which are Neville-representable.

We first present some general results about the LU factorization of a quasiseparable matrix, and then we consider the Neville representation of a quasiseparable matrix, showing conditions for its existence.

With the LU factorization, the standard factorization of a matrix into the product of a unit lower triangular matrix and of an upper triangular matrix is meant. In the following, we will denote with L unit lower triangular matrices and with R unit upper triangular matrices.

THEOREM 4. *Let A be a quasiseparable matrix, and assume there exist L, R, and D such that $A = LDR$, where L and R are unit lower and upper triangular and D is diagonal. Then L and R can be chosen having quasiseparable structure. Moreover, L and R can be represented, according to* (2.3)*, with the same parameters u_i, t_i, q_i, r_i appearing in the representation of A.*

Proof. Since A is LU-factorizable, if A is nonsingular, then it is also strongly nonsingular, and the thesis follows from a known result, which states that in this case L and R must be quasiseparable (see [24, p. 171]).

In the case A is singular, we have that at least one of the diagonal entries d_i of D is zero. Writing $A = S^{(l)} + P$, where $S^{(l)}$ is strictly lower triangular and quasiseparable, and P is upper triangular, we have $LD = AR^{-1} = S^{(l)}R^{-1} + PR^{-1}$, and looking at the $\mathrm{tril}(LD, -1) = \mathrm{tril}(S^{(l)}R^{-1}, -1)$, we see that LD is quasiseparable, and the generators can be expressed in terms of the generators of A. In detail, denote by $\mathbf{u}$, $\mathbf{v}$, $\mathbf{t}$ the generators of the lower triangular part of A, by $\tilde{\mathbf{u}}$, $\tilde{\mathbf{v}}$, $\tilde{\mathbf{t}}$ the generators of $\mathrm{tril}(LD, -1)$, and by w_{ij} the $(i,\, j)$th entry of R^{-1}, we can take, e.g.,

$$\tilde{u}_i = u_i, \quad \tilde{t}_i = t_i, \quad \tilde{v}_i = \sum_{j=1}^{i} t_{i:j+1} v_j w_{ji}.$$

Thus L can be chosen, in infinitely many ways, as a unit lower triangular semiseparable matrix, generated by the same $\tilde{\mathbf{u}}$, $\tilde{\tilde{\mathbf{v}}}$, $\tilde{\mathbf{t}}$, where $\tilde{\tilde{v}}_i d_i = \tilde{v}_i$.

Similarly, R can be chosen as a unit upper triangular semiseparable matrix, generated by the same $\mathbf{q}$, $\mathbf{r}$ generating the upper triangular part of A. □

Now we will consider a set of quasiseparable matrices which are Neville-representable. A Neville-representable matrix can be expressed as a product of this form (see [12]):

$$L^{(n-1)} \cdots L^{(1)} D R^{(1)} \cdots R^{(n-1)},$$

where D is diagonal, and the factors $L^{(i)}$, $R^{(i)}$ are unit bidiagonal, lower and upper, respectively, with zero entries in these positions:

$$L^{(i)}_{k+1,k} = R^{(i)}_{k,k+1} = 0 \quad \text{for } k = 1, \ldots, i-1.$$

Definition 4. *Let $\mathcal{S}$ be the set of matrices A admitting the following factorization:*

$$A = L_s L_1 D R_1 R_s, \tag{3.1}$$

where

$$L_s^{-1} = \begin{pmatrix} 1 & & & & \\ -x_1 & 1 & & & \\ & -x_2 & 1 & & \\ & & \ddots & \ddots & \\ & & & -x_{n-1} & 1 \end{pmatrix}, \qquad L_1 = \begin{pmatrix} 1 & & & & \\ -a_1 & 1 & & & \\ & -a_2 & 1 & & \\ & & \ddots & \ddots & \\ & & & -a_{n-1} & 1 \end{pmatrix},$$

$$R_1 = \begin{pmatrix} 1 & -b_1 & & & \\ & 1 & -b_2 & & \\ & & \ddots & \ddots & \\ & & & 1 & -b_{n-1} \\ & & & & 1 \end{pmatrix}, \qquad R_s^{-1} = \begin{pmatrix} 1 & -y_1 & & & \\ & 1 & -y_2 & & \\ & & \ddots & \ddots & \\ & & & 1 & -y_{n-1} \\ & & & & 1 \end{pmatrix},$$

and D is a diagonal matrix.

It is straightforward to see that the matrices introduced by Definition 4 are LU-factorizable and quasiseparable. Moreover, they are also Neville-representable, because if we set

$$\begin{aligned} &L^{(i)} = I + \operatorname{diag}(x_i \mathbf{e}_i, -1), \quad R^{(i)} = I + \operatorname{diag}(y_i \mathbf{e}_i, 1), \quad i = n-1, \ldots, 2, \\ &L^{(1)} = (I + \operatorname{diag}(x_1 \mathbf{e}_1, -1)) L_1, \qquad R^{(1)} = R_1 (I + \operatorname{diag}(y_1 \mathbf{e}_1, 1)), \end{aligned}$$

where $\mathbf{e}_i$ is the ith vector of the canonical basis of $\mathbf{R}^{n-1}$, we have

$$A = L_s L_1 D R_1 R_s = L^{(n-1)} \cdots L^{(1)} D R^{(1)} \cdots R^{((n-1))},$$

that is, the Neville representation of A. Therefore (3.1) can be seen as a variant of the Neville representation. The factorization (3.1) is not unique: even when A is strongly nonsingular and the products $L = L_s L_1$ and $R = R_1 R_s$ are uniquely determined, there are infinitely many values of x_1, a_1, b_1, y_1 giving the same matrices $L^{(1)}$ and $R^{(1)}$, and therefore the same A. They can be freely chosen according to the conditions $x_1 - a_1 = l_{21}$ and $y_1 - b_1 = r_{12}$.

Nevertheless, there are LU-factorizable quasiseparable matrices which are not Neville-representable. For instance, the matrix

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

is factorizable $A = LDR$, where

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix},$$

but we cannot find any x_i and a_i such that $L_s L_1 = L$.

Definition 5. *Let $\mathcal{S}_1$ be the set of matrices in $\mathcal{S}$ for which there exists a choice of parameters in the representation* (2.3) *satisfying the following conditions:*

(a) $u_i \neq 0$ *for* $i = 2, 3, \ldots n$;

(b) $q_i \neq 0$ *for* $i = 2, 3, \ldots, n$.

Remark 1. One can see some ambiguity in Definition 5 of $\mathcal{S}_1$, due to the fact that the same quasiseparable matrix has infinitely many representations (2.3). For instance, a quasiseparable matrix having zero entries only in the last row is in $\mathcal{S}_1$, because it can be represented according to (2.3) with $u_n = 1$, $t_{n-1} = v_{n-1} = \delta_n = 0$, but it can also be represented with $u_n = \delta_n = 0$ for arbitrary choices of t_{n-1} and v_{n-1}. Two simple equivalent conditions for a matrix in $\mathcal{S}$ to be in $\mathcal{S}_1$ are the following:

(i) if there is an entry $a_{ij} = 0$ in the strictly lower triangular part, then $a_{kj} = 0$ for $k = i + 1, \ldots, n$;

(ii) if there is an entry $a_{ij} = 0$ in the strictly upper triangular part, then $a_{ik} = 0$ for $k = j + 1, \ldots, n$.

A quasiseparable matrix which violates (i) or (ii) cannot be represented with all nonzero u_i and q_i. We could overcome the question by saying that the matrices in $\mathcal{S}_1$ are all those that admit a representation (2.3) with $u_i = q_i = 1$ for every i, as we will see in Corollary 6.

Theorem 5. *The class $\mathcal{S}$ coincides with the class $\mathcal{S}_1$.*

Proof. We prove the theorem by showing that $\mathcal{S} \subseteq \mathcal{S}_1$ and $\mathcal{S}_1 \subseteq \mathcal{S}$.

Let us start by proving that if $A \in \mathcal{S}$, then $A \in \mathcal{S}_1$. First, A is factorizable $A = LDR$ with $L = L_s L_1$ and $R = R_1 R_s$. To prove that A is quasiseparable it is sufficient to prove that it is quasiseparable in the lower and upper triangular parts.

Observe that L_s is semiseparable, and in fact the rank-one structure propagates to the main diagonal. We distinguish two cases according to the possible reducibility of L_s.

If L_s is irreducible, then all $x_i \neq 0$. Hence it can be written as $L_s = \bar{\mathbf{u}}\bar{\mathbf{v}}^T + P$, where $\bar{u}_i = \prod_{k=1}^{i-1} x_k$, $\bar{v}_i = \bar{u}_i^{-1}$; P is a strictly upper triangular matrix with superdiagonal entries $p_{i,i+1} = x_i^{-1}$. Writing the bidiagonal matrix L_1 as $L_1 = I - \mathrm{diag}(\mathbf{a}, -1)$, we have

$$\begin{aligned} A &= (\bar{\mathbf{u}}\bar{\mathbf{v}}^T + P)(I - \mathrm{diag}(\mathbf{a}, -1))DR \\ &= (\bar{\mathbf{u}}\bar{\mathbf{v}}^T + P - \bar{\mathbf{u}}\bar{\mathbf{v}}^T\mathrm{diag}(\mathbf{a}, -1) - P\mathrm{diag}(\mathbf{a}, -1))DR \\ &= \bar{\mathbf{u}}(\bar{\mathbf{v}}^T - \bar{\mathbf{v}}^T\mathrm{diag}(\mathbf{a}, -1))DR + (P - P\mathrm{diag}(\mathbf{a}, -1))DR. \end{aligned}$$

Setting $\tilde{\mathbf{v}}^T = (\bar{\mathbf{v}}^T - \bar{\mathbf{v}}^T\mathrm{diag}(\mathbf{a}, -1))DR$ and $\tilde{P} = (P - P\mathrm{diag}(\mathbf{a}, -1))DR$, we have that $A = \bar{\mathbf{u}}\tilde{\mathbf{v}}^T + \tilde{P}$. $\tilde{P}$ is an upper triangular matrix having $-a_i d_i x_i^{-1}$ as ith diagonal entry. So A is the sum of a rank-one matrix and an upper triangular matrix, and hence its lower triangular part is quasiseparable. In a similar way we can show that A has the upper triangular part with a quasiseparable structure. Note that all the entries of $\bar{\mathbf{u}}$ are nonzero, since $\bar{u}_i = \prod_{k=1}^{i-1} x_k$. Hence, condition (a) of Definition 5 is satisfied.

If L_s is reducible, then one or more $x_i = 0$. For simplicity let us consider only the case for which $x_i = 0$ and $x_j \neq 0$ for all $j \neq i$. The generalization to multiple blocks is straightforward. We already observed that reducible triangular semiseparable matrices can be expressed as the direct sum of generator-representable semiseparable matrices. In this case $L_s = L_s^{(1)} \bigoplus L_s^{(2)}$, where $L_s^{(1)}$ and $L_s^{(2)}$ are generator-representable semiseparable irreducible matrices of sizes n_1 and n_2, respectively, and hence $L_s^{(1)} = \mathbf{u}^{(1)}\mathbf{v}^{(1)\,T} + P_1$, and $L_s^{(2)} = \mathbf{u}^{(2)}\mathbf{v}^{(2)\,T} + P_2$, with P_1 and P_2 being strictly upper triangular matrices. Moreover, $\mathbf{u}^{(1)}$ and $\mathbf{u}^{(2)}$ have no zero entries; since $u_i^{(1)}v_i^{(1)} = 1$, $u_i^{(2)}v_i^{(2)} = 1$. Let us partition all the relevant matrices according to the partitioning of L_s; thus we have $D = D_1 \bigoplus D_2$, and

$$R = R_1 R_s = \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix}.$$

In the same way as before we get

$$A = \begin{bmatrix} \mathbf{u}^{(1)}\mathbf{w}^{(1)\,T} & 0 \\ \kappa \mathbf{u}^{(2)}\mathbf{e}_{n_1}^T & \mathbf{u}^{(2)}\mathbf{w}^{(2)\,T} \end{bmatrix} + \begin{bmatrix} P_{11} & P_{12} \\ 0 & P_{22} \end{bmatrix},$$

where $\mathbf{e}_{n_1}$ is the n_1th vector of the canonical basis in $\mathbf{R}^{n_1}$, $\kappa = -v_1^{(2)} d_{n_1} a_i$, $\mathbf{w}^{(1)\,T} = (\mathbf{v}^{(1)\,T} - \mathbf{v}^{(1)\,T}\mathrm{diag}(\mathbf{a}(1\colon i-1), -1))D_1 R_{11}$, and $\mathbf{w}^{(2)\,T} = (\mathbf{v}^{(2)\,T} - \mathbf{v}^{(2)\,T}\mathrm{diag}(\mathbf{a}(i+1\colon n-1), -1))D_2 R_{22} + \kappa \mathbf{e}_{n_1}^T R_{12}$. Moreover, P_{11} and P_{22} are upper triangular matrices. Note that the block in position $(2,1)$ of A is null except for the last column, and this column is proportional to $\mathbf{u}^{(2)}$. Then $\mathrm{tril}(A, 0)$ is quasiseparable, and the following is a possible choice of generators for A:

$$t_j = \begin{cases} 0 & \text{for } j = i, \\ 1 & \text{for } j \neq i, \end{cases} \qquad u_j = \begin{cases} u_j^{(1)} & \text{for } 1 \leq j \leq i, \\ u_{j-i}^{(2)} & \text{for } i+1 \leq j \leq n, \end{cases}$$

$$v_j = \begin{cases} w_j^{(1)} & \text{for } 1 \leq j < i, \\ \kappa & \text{for } j = i, \\ w_{j-i}^{(2)} & \text{for } j+1 \leq j \leq n. \end{cases}$$

Clearly all u_j's are nonzero. A similar proof can be given for the upper triangular part of A.

Let us prove that $\mathcal{S}_1 \subseteq \mathcal{S}$; that is, for every quasiseparable, LU-factorizable matrix with $u_i \neq 0$ and $q_i \neq 0$, we can find parameters $\mathbf{x}$, $\mathbf{a}$, $\mathbf{b}$, $\mathbf{y}$, $\mathbf{d}$ such that $A = L_s L_1 D R_1 R_s$.

By assumption we have $A = LDR$, where L is unit lower triangular, D diagonal, and R is unit upper triangular. We want to prove that it is possible to factorize L as $L_s L_1$ and R as $R_1 R_s$. By Theorem 4, we have that L and R can be chosen quasiseparable, so in detail,

$$L = \begin{pmatrix} 1 & & & & & \\ \tilde u_2 \tilde v_1 & 1 & & & & \\ \tilde u_3 \tilde t_2 \tilde v_1 & \tilde u_3 \tilde v_2 & 1 & & & \\ \tilde u_4 \tilde t_{3:2} \tilde v_1 & \tilde u_4 \tilde t_3 \tilde v_2 & \tilde u_4 \tilde v_3 & 1 & & \\ \vdots & \vdots & \vdots & & \ddots & \\ \tilde u_n \tilde t_{n-1:2} \tilde v_1 & \tilde u_n \tilde t_{n-1:3} \tilde v_2 & \tilde u_n \tilde t_{n-1:4} \tilde v_3 & \cdots & \tilde u_n \tilde v_{n-1} & 1 \end{pmatrix},$$

where $\tilde u_i \neq 0$ since $\tilde u_i = u_i$, which are assumed to be nonzero. It is now easy to prove that L can be factorized as the product of L_s and L_1 by observing that the Neville elimination can be applied to the rows of L. In particular, L_s is the inverse of the bidiagonal matrix

$I-\operatorname{diag}(\mathbf{x},-1)$ having the Neville multipliers as codiagonal entries, i.e., $x_{i-1} = -\tilde{u}_i\tilde{t}_{i-1}/\tilde{u}_{i-1}$, $i=3,\dots,n$.

Reasoning in the same way for the upper triangular part of A we can complete the proof. □

The following corollary shows how a matrix in $\mathcal{S}$, represented as in Definition 4, can be represented by means of the generators appearing in (2.3).

Corollary 6. *If $A\in\mathcal{S}$, then it can be represented, according to* (2.3)*, with $u_i=q_i=1$, $i=2,\dots,n$, $t_i=x_i$, $r_i=y_i$, $i=2,\dots,n-1$, with the same x_i, y_i, $i=1,2,\dots,n-1$ as in Definition* 4.

Proof. By direct inspection, a simple choice for the parameters involved in the representations of the semiseparable matrices L and R of the factorization $A=LDR$ is the following:

$$u_i=1, \qquad v_i=x_i-a_i, \qquad i=2,\dots,n, \qquad t_i=x_i, \qquad i=2,\dots,n-1 \quad \text{for } L,$$

$$q_i=1, \qquad p_i=y_i-b_i, \qquad i=2,\dots,n, \qquad r_i=y_i, \qquad i=2,\dots,n-1 \quad \text{for } R.$$

Theorem 4 says that the parameters u_i, t_i, q_i, and r_i for A can be taken from the represention of L and R. □

It is easy to prove that $\mathcal{S}$ contains all quasiseparable, Neville-representable matrices.

Theorem 7. *Any quasiseparable Neville-representable matrix is in $\mathcal{S}$.*

Proof. Since $\mathcal{S}=\mathcal{S}_1$, we show that a Neville-representable quasiseparable matrix must be in $\mathcal{S}_1$. Assume by contradiction this is not true; then by Remark 1, one of conditions (i), (ii) would not be obeyed, say, (i). This means that there exist, in the strictly lower triangular part, some $a_{ij}=0$ with $a_{i+1,j}\neq 0$, and, as a consequence of the rank structure, all the entries in the ith row of the strictly lower triangular part would be zero. Then the ith row could not be used to eliminate the $(i+1)$th one, and the Neville row elimination would fail. But this would cause a contradiction. □

3.1. Quasiseparable TN matrices.

Neville elimination is deeply connected with TN matrices [15], [12], [13].

Definition 6. *A matrix A is called TN if all its minors of any order are nonnegative.*

Theorem 8. *If $A\in\mathcal{S}$ is representable as in Definition* 4*, and $x_i, y_i\geq 0$, $a_i, b_i\leq 0$, and $d_i\geq 0$, then A is TN.*

Proof. In the standard hypothesis, A is TN, since it is the product of TN matrices. □

In the nonsingular case, Theorem 8 (and its converse) was proved in [14].

Remark 2. We can easily recognize matrices in $\mathcal{S}$ diagonally similar to TN matrices. The general condition is that $x_iy_i\geq 0$, $d_i\geq 0$, $a_ib_i\geq 0$ with $a_ix_i\leq 0$.

The class of TN matrices has nice properties that resemble those of positive semidefinite Hermitian matrices; in particular, its eigenvalues are real and nonnegative. Another similarity between the two classes of matrices is that also for TN matrices we can give an interlacing theorem [11], [1].

Theorem 9. *Let A be TN with eigenvalues $\lambda_1\geq\lambda_2 \cdots\geq\lambda_n$. Suppose A_k is a $k\times k$ submatrix of A lying in rows and columns with consecutive indices and having eigenvalues $\tilde\lambda_1\geq\tilde\lambda_2 \cdots\geq\tilde\lambda_k$. Then*

$$\lambda_i\geq\tilde\lambda_i\geq\lambda_{i+n-k}, \qquad i=1,\dots,k.$$

4. On the shifted LR algorithm. In this section we present the shifted LR algorithm acting implicitly on the representation (3.1) of A. The shifted LR method proceeds iteratively as follows.[2]. Let $A^{(0)} = A$; we obtain the sequence of matrices $A^{(k)}$ in this way:

$$\begin{cases} A^{(k)} = L^{(k)} D^{(k)} R^{(k)}, \\ A^{(k+1)} = L^{(k)-1} A^{(k)} L^{(k)} - \sigma_{k+1} I = D^{(k)} R^{(k)} L^{(k)} - \sigma_{k+1} I, \quad k = 0, 1, \ldots, \end{cases} \tag{4.1}$$

where, for every k, the parameter σ_{k+1} is chosen in accordance with some shift strategy to accelerate convergence.[3] Usually, instead of computing explicitly the factorization of $A^{(k)}$ and multiplying the factors in reverse order according to (4.1), we proceed implicitly performing the transition $A^{(k)} \to A^{(k+1)}$ in terms of their Neville representations. Note that $A^{(k+1)}$ is no more similar to $A^{(k)}$ because at each step we subtract a shift, but we never restore it. This means that we have to accumulate the shifts during the computation and add them back once the approximation of each eigenvalue becomes available. An important result is that the quasiseparable structure is preserved under LR steps.

Theorem 10. *If $A^{(0)} = A$ is quasiseperable and LU-factorizable, then the matrix $A^{(1)}$ built by* (4.1) *is quasiseparable.*

Proof. Let $A = LDR$. We want to prove that the matrix $A^{(1)} = DRL$ is still quasiseparable, although the possible symmetry of A is not preserved. Since the addition of a scalar matrix does not affect the quasiseparable structure, we can assume that no shift is performed for notational simplicity. By Corollary 2, there exists a symmetric semiseparable matrix $S^{(u)}$ and a lower triangular matrix Q such that $A = S^{(u)} + Q$, so we have $A^{(1)} = L^{-1}AL = L^{-1}(S^{(u)} + Q)L$. As remarked in section 2, if $S^{(u)}$ is irreducible, then it is generator-representable; otherwise it is the direct sum of generator-representable matrices. Assume that $S^{(u)}$ is irreducible, and therefore $S^{(u)} = \mathbf{p}\mathbf{q}^T + X$, where X is a strictly lower triangular matrix. Then $A^{(1)} = L^{-1}\mathbf{p}\mathbf{q}^T L + L^{-1}XL + L^{-1}QL$. Setting $\tilde{\mathbf{p}} = L^{-1}\mathbf{p}$, $\tilde{\mathbf{q}} = L^T\mathbf{q}$, $\tilde{Q} = L^{-1}(X + Q)L$, we have $A^{(1)} = \tilde{\mathbf{p}}\,\tilde{\mathbf{q}} + \tilde{Q}$. Then the minors taken out of the strictly upper triangular part of $A^{(1)}$ have rank at most one. In the case $S^{(u)}$ is reducible, we can use the same arguments for each of the irreducible diagonal blocks, obtaining that $A^{(1)}$ is reducible too and is quasiseparable in the upper triangular part.

Similarly, if D is nonsingular, we can prove that $A^{(1)}$ has the quasiseparable structure in the strictly lower triangular part. If D is singular, consider the matrix $B = RL$ which is quasiseparable because it is a special case of the theorem with $D = I$. Now, the quasiseparable structure is preserved by multiplication by a diagonal matrix D. □

In this section we present two different methods for performing an implicit LR step through the updating of the parameters involved in the representation of matrices in $\mathcal{S}$ (see Definition 4). The first, presented in section 4.1, works on the parameters a_i, b_i, x_i, y_i, and d_i, and the second one, described in section 4.2, considers the aggregated parameters $m_i = x_i y_i$ and takes advantage of the fact that there are some quantities that are invariant during the steps, as explained in the following corollary.

Corollary 11. *Let A be a quasiseparable matrix in $\mathcal{S}$. If $x_i \neq 0$ and $y_i \neq 0$ for $i = 1, \ldots, n-1$, we can define the quantities*

$$h_i = \frac{a_i d_i}{x_i}, \quad k_i = \frac{b_i d_i}{y_i}, \quad i = 1, \ldots, n-1. \tag{4.2}$$

[2] The superscript notation $\cdot^{(i)}$ will be used only for depicting LR steps performed on matrices. We will omit this superscript as much as possible to avoid overloading the notation.

[3] In section 4.3 we describe in detail the choice of the Laguerre shift, which is particularly well suited when dealing with TN matrices.

These quantities are invariant under LR steps without shift.

Proof. By Corollary 6, in the representation (2.3) of A we have that $r_i \neq 0$ for $i = 2, \ldots, n-1$. By Lemma 3, $A = \hat{\mathbf{p}}\hat{\mathbf{q}}^T + K$, where K is lower triangular. Consider the diagonal entries of A, which for the previous equality are given by the sum of the diagonal entries of $\hat{\mathbf{p}}\hat{\mathbf{q}}^T$ and the diagonal entries of K, say k_i. This means $\delta_i = p_i q_i + k_i$. We have $A^{(1)} = L^{-1}AL = L^{-1}(\hat{\mathbf{p}}\hat{\mathbf{q}}^T + K)L$. Since L is unit lower triangular, we have that the diagonal entries of the lower triangular term $L^{-1}KL$ are still equal to k_i and remain constant during all the iterative steps.

Set $\Delta = \mathrm{dg}(K)$. The matrix $B = A - \Delta$ has a semiseparable structure in the upper triangular part, which extends to the main diagonal. Since $A = L_s L_1 D R_1 R_s$, B can be expressed as

$$B = A - \Delta = L_s T R_s, \tag{4.3}$$

where $T = L_1 D R_1 - L_s^{-1} \Delta R_s^{-1}$ is tridiagonal. The upper triangular matrix R_s is a semiseparable upper triangular matrix satisfying Lemma 3; then it can be written as $R_s = \tilde{\mathbf{p}}\tilde{\mathbf{q}}^T + \tilde{K}$, where $\tilde{K}$ is strictly lower triangular. Substituting in (4.3) we find

$$B = L_s T \tilde{\mathbf{p}}\tilde{\mathbf{q}}^T + L_s T \tilde{K};$$

thus we see that the diagonal of B agrees with the rank-one structure in the upper triangular part only if the lower triangular matrix $L_s T \tilde{K}$ has all zeros as diagonal entries, and this happens only if T is lower bidiagonal. By equating to zero the upper codiagonal entries of $T = L_1 D R_1 - L_s^{-1} \Delta R_s^{-1}$, we readily obtain $k_i = \frac{d_i b_i}{y_i}$.

Repeating the same reasoning for the lower triangular part of A, we obtain that $h_i = \frac{d_i a_i}{x_i}$ are invariants too. □

4.1. A qd-type method. In this section we show how to obtain, starting from a matrix $A = A^{(k)}$ in $\mathcal{S}$, expressed as $A = L_s L_1 D R_1 R_s$, the new representation of $A^{(k+1)} = D R_1 R_s L_s L_1 - \sigma_{k+1} I$ in terms of the updated parameters, that is, $A^{(k+1)} = \bar{\bar{L}}_s \bar{\bar{L}}_1 \bar{\bar{D}} \bar{\bar{R}}_1 \bar{\bar{R}}_s$, given that $A^{(k+1)} \in \mathcal{S}$.

The entire procedure consists in updating the parameters x_i, y_i, d_i, a_i, b_i which define quasiseparable matrices in $\mathcal{S}$. The updating process starts with the computation of $\tilde{A} = D R_1 R_s L_s L_1$ and replaces products of the type RL with products of the type LDR, where D is diagonal. In particular, we have

$$\tilde{A} = D R_1 (R_s L_s) L_1 = D R_1 (\bar{L}_s E \bar{R}_s) L_1 \tag{4.4}$$

$$= (D R_1 \bar{L}_s) E \bar{R}_s L_1 = (\bar{\bar{L}}_s F \bar{R}_1) E \bar{R}_s L_1 \tag{4.5}$$

$$= \bar{\bar{L}}_s F \bar{R}_1 (E \bar{R}_s L_1) = \bar{\bar{L}}_s F \bar{R}_1 (\bar{L}_1 G \bar{\bar{R}}_s), \tag{4.6}$$

where $D = \mathrm{diag}(d_i)$, $E = \mathrm{diag}(e_i)$, $F = \mathrm{diag}(f_i)$, and $G = \mathrm{diag}(g_i)$ are diagonal matrices. Equations (4.4), (4.5), (4.6) make sense if the intermediate matrices $R_s L_s$, $D R_1 \bar{L}_s$, and $E \bar{R}_s L_1$ are all LU-factorizable. Anyway, if the procedure described by (4.4), (4.5), (4.6) can be carried on and $A^{(k+1)} = \tilde{A} - \sigma_{k+1} I$ is LU-factorizable, too, then

$$A^{(k+1)} = \bar{\bar{L}}_s F \bar{R}_1 \bar{L}_1 G \bar{\bar{R}}_s - \sigma_{k+1} I = \bar{\bar{L}}_s \bar{\bar{L}}_1 \bar{\bar{D}} \bar{\bar{R}}_1 \bar{\bar{R}}_s, \tag{4.7}$$

where $\bar{\bar{D}}$ is diagonal.

The assumptions required by the updating (4.4), (4.5), (4.6), besides the preliminary request for $A^{(k+1)}$ to be in $\mathcal{S}$, are satisfied for all TN matrices, with a suitable choice of the shift parameter (see section 4.3).

Let us describe the updating process of equality (4.4). Set

$$\bar{L}_s E \bar{R}_s = R_s L_s$$

for a suitable nonsingular diagonal matrix E. If we rewrite the above equation as $\bar{R}_s^{-1} E^{-1} \bar{L}_s^{-1} = L_s^{-1} R_s^{-1}$, where all the matrices involved are bidiagonal or diagonal, and we define the auxiliary variables

$$\begin{cases} \alpha_i = e_i^{-1} - x_{i-1} y_{i-1}, \quad i = 1, \dots, n-1, \\ \alpha_n = 1, \end{cases} \tag{4.8}$$

we obtain the following recurrences:

$$\begin{cases} e_i^{-1} = \alpha_i + x_{i-1} y_{i-1}, \quad i = n, \dots, 2\,, \\ \alpha_{i-1} = \alpha_i / e_i^{-1}, \quad i = n, \dots, 2, \\ e_1^{-1} = \alpha_1. \end{cases} \tag{4.9}$$

The first recurrence of (4.9) for determining the e_i^{-1} is obtained from the definition of the auxiliary variables α_i, while the recurrence for the computation of the α_i comes from equating the diagonal entries of the tridiagonal matrices $\bar{R}_s^{-1} E^{-1} \bar{L}_s^{-1}$ and $L_s^{-1} R_s^{-1}$. By equating the codiagonal entries we get that the entries of $\bar{L}_s^{-1}$ and $\bar{R}_s^{-1}$ can be computed as follows:

$$\bar{x}_i = x_i / e_{i+1}^{-1}, \quad i = 1, \dots, n-1,$$
$$\bar{y}_i = y_i / e_{i+1}^{-1}, \quad i = 1, \dots, n-1.$$

The updating described in (4.5) consists of

$$\bar{\bar{L}}_s F \bar{R}_1 = D R_1 \bar{L}_s$$

for a suitable nonsingular diagonal matrix F. Again we can use the fact that the inverse of matrices of type L_s is bidiagonal. We have $F \bar{R}_1 \bar{L}_s^{-1} = \bar{\bar{L}}_s^{-1} D R_1$, and setting

$$\beta_i = 1 - b_i \bar{x}_i, \qquad i = 1, \dots, n-1,$$

we obtain the following recurrences for the diagonal entries of F:

$$\begin{cases} f_1 = d_1 \beta_1, \\ f_i = d_i \beta_i / \beta_{i-1}, \quad i = 2, \dots, n-1, \\ f_n = d_n / \beta_{n-1}, \end{cases}$$

and the final $\bar{\bar{x}}_i$ describing $\bar{\bar{L}}_s$,

$$\bar{\bar{x}}_i = \bar{x}_i f_{i+1} / d_i, \qquad i = 1, \dots, n-1.$$

The intermediate entries of $\bar{R}_1$ are

$$\begin{cases} \bar{b}_1 = b_1/\beta_1, \\ \bar{b}_i = b_i\beta_i/\beta_{i-1}, \quad i = 2, \ldots, n-1. \end{cases}$$

Let us describe step (4.6), that is,

$$\bar{L}_1 G \bar{\bar{R}}_s = E \bar{R}_s L_1.$$

If we rewrite the above equation as $\bar{R}_s^{-1} E^{-1} \bar{L}_1 = L_1 \bar{\bar{R}}_s^{-1} G^{-1}$, and defining the auxiliary variables

$$\gamma_i = 1 - \bar{y}_i a_i, \qquad i = 1, \ldots, n-1,$$

we obtain the following recurrences for the diagonal entries of G:

$$\begin{cases} g_1 = \gamma_1/e_1^{-1}, \\ g_i = \gamma_i/(\gamma_{i-1} e_i^{-1}), \quad i = 2, \ldots, n-1, \\ g_n = 1/(\gamma_{n-1} e_n^{-1}). \end{cases}$$

The final entries of $\bar{\bar{R}}_s$ are computed as follows:

$$\bar{\bar{y}}_i = \bar{y}_i g_{i+1}/e_{i+1}^{-1}, \qquad i = 1, \ldots, n-1,$$

while the intermediate entries of $\bar{L}_1$ are

$$\bar{a}_i = a_i/(e_{i+1}^{-1} g_i), \qquad i = 1, \ldots, n-1.$$

It remains to describe the final updating which involves the terms in the brackets in (4.7). If no shift is chosen, then $A^{(k+1)} = A^{(1)}$, so $\bar{\bar{L}}_1$, $\bar{\bar{D}}$, and $\bar{\bar{R}}_1$ have to be found such that

$$\bar{\bar{L}}_1 \bar{\bar{D}} \bar{\bar{R}}_1 = F \bar{R}_1 \bar{L}_1 G. \tag{4.10}$$

The final matrices $\bar{\bar{L}}_1$ and $\bar{\bar{R}}_1$ can be computed introducing auxiliary variables δ_i defined as

$$\begin{cases} \delta_i = \bar{\bar{d}}_i - \bar{a}_i \bar{b}_i f_i g_i, \quad i = 1, \ldots, n-1, \\ \delta_n = \bar{\bar{d}}_n. \end{cases}$$

We have the following recurrences for the final $\bar{\bar{D}}$:

$$\begin{cases} \delta_1 = f_1 g_1, \\ \bar{\bar{d}}_i = \delta_i + \bar{a}_i \bar{b}_i f_i g_i, \quad i = 1, \ldots, n-1, \\ \delta_{i+1} = \delta_i f_{i+1} g_{i+1}/\bar{\bar{d}}_i, \quad i = 1, \ldots, n-1, \\ \bar{\bar{d}}_n = \delta_n. \end{cases} \tag{4.11}$$

The entries of $\bar{\bar{L}}_1$ and $\bar{\bar{R}}_1$ can then be obtained as follows:

$$\bar{\bar{a}}_i = \bar{a}_i f_{i+1} g_i/\bar{\bar{d}}_i, \qquad i = 1, \ldots, n-1. \tag{4.12}$$

$$\bar{\bar{b}}_i = \bar{b}_i f_i g_{i+1}/\bar{\bar{d}}_i, \qquad i = 1, \ldots, n-1. \tag{4.13}$$

In case of shift $\sigma = \sigma_{k+1}$, note that $A^{(k+1)} = A^{(1)} - \sigma I$ can be expressed as

$$\bar{\bar{L}}_s F \bar{R}_1 \bar{L}_1 G \bar{\bar{R}}_s - \sigma I = \bar{\bar{L}}_s (F \bar{R}_1 \bar{L}_1 G - \sigma \bar{\bar{L}}_s^{-1} \bar{\bar{R}}_s^{-1}) \bar{\bar{R}}_s.$$

So we must modify only the updating of the variables $\bar{\bar{a}}_i$, $\bar{\bar{b}}_i$ and $\bar{\bar{d}}_i$ described by (4.12), (4.13), and (4.11). In particular, (4.10) becomes

$$\bar{\bar{L}}_1 \bar{\bar{D}} \bar{\bar{R}}_1 = F \bar{R}_1 \bar{L}_1 G - \sigma \bar{\bar{L}}_s^{-1} \bar{\bar{R}}_s^{-1},$$

and $\bar{\bar{L}}_1$, $\bar{\bar{D}}$, and $\bar{\bar{R}}_1$ can be computed by means of the following recurrences, where δ_i is defined as before as

$$\begin{cases} \delta_i = \bar{\bar{d}}_i - \bar{a}_i \bar{b}_i f_i g_i, & i = 1, \dots, n-1, \\ \delta_n = \bar{\bar{d}}_n. \end{cases}$$

The recurrences replacing (4.11) are

$$\begin{cases} \delta_1 = f_1 g_1 - \sigma, \\ \bar{\bar{d}}_i = \delta_i + \bar{a}_i \bar{b}_i f_i g_i, \quad i = 1, \dots, n-1, \\ \delta_{i+1} = f_{i+1} g_{i+1} \delta_i / \bar{\bar{d}}_i - \sigma(1 + \bar{\bar{x}}_i(\bar{\bar{y}}_i - \bar{\bar{b}}_i) - f_{i+1} g_i \bar{a}_i \bar{\bar{y}}_i / \bar{\bar{d}}_i), \quad i = 1, \dots, n-1, \\ \bar{\bar{d}}_n = \delta_n. \end{cases} \tag{4.14}$$

The new $\bar{\bar{a}}_i$ and $\bar{\bar{b}}_i$ can be obtained as follows:

$$\bar{\bar{a}}_i = (\bar{a}_i f_i g_{i+1} - \sigma \bar{\bar{x}}_i) / \bar{\bar{d}}_i, \qquad i = 1, \dots, n-1. \tag{4.15}$$

$$\bar{\bar{b}}_i = (\bar{b}_i f_{i+1} g_i - \sigma \bar{\bar{y}}_i) / \bar{\bar{d}}_i, \qquad i = 1, \dots, n-1. \tag{4.16}$$

Alternatively, one can first compute $\bar{\bar{a}}_i$ and $\bar{\bar{b}}_i$ with (4.15) and (4.16), and then express δ_{i+1} as

$$\delta_{i+1} = f_{i+1} g_{i+1} \delta_i / \bar{\bar{d}}_i - \sigma(1 + \bar{\bar{y}}_i(\bar{\bar{x}}_i - \bar{\bar{a}}_i) - f_i g_{i+1} \bar{b}_i \bar{\bar{x}}_i / \bar{\bar{d}}_i), \qquad i = 1, \dots, n-1.$$

Remark 3. We denoted the method described in this section as belonging to the family of qd algorithms [20], [10], [17]. The reason is that it has similar characteristics, since the definition of the auxiliary variables α_i, δ_i makes it possible, with some sign hypotheses (see Theorem 13), to get rid of subtractions which are hidden in the auxiliary parameters; only divisions, multiplications, and sums are needed (except for the shift).

4.2. Working with invariants: Another qd-type method. In Corollary 11 we proved that if $x_i \neq 0$ and $y_i \neq 0$ for all $i = 1, \dots, n-1$, then the quantities h_i and k_i defined as in (4.2) are invariant under LR steps. This observation allows us to rewrite the previous recurrences using these quantities. The new algorithm, described by Algorithm 1, although applicable only to a subclass of matrices in $\mathcal{S}$, has a lower computational cost, as we will see in section 5.

The new recurrences consider the aggregated quantities $m_i = x_i y_i$, the invariants k_i, h_i, and the diagonal entries d_i. Note that the use of invariants allows us to simplify the recurrences since we have only to update the values m_i and the new diagonal entries d_i.

This procedure should be combined with an effective shift strategy. A particularly convenient choice for the shift is described in section 4.3.

As in the recurrences presented in section 4.1, we see that we still need auxiliary variables α_i and δ_i, but the aggregation of parameters x_i and y_i into m_i allows us to reduce the number of recurrences. This method does not have a natural matrix formulation, but it is easy to verify its correctness by merging the recurrences for the updating of x_i and y_i to get the updating formula for m_i.

Algorithm 1. $[\bar{\bar{m}},\bar{\bar{h}},\bar{\bar{k}},\bar{\bar{d}}]\leftarrow$`LRstep`$(m,h,k,d,\sigma)$.

```
$\alpha_n \leftarrow 1$;
for $i = n$ to 2 do
  $e_i^{-1} \leftarrow \alpha_i - m_{i-1}$; {Auxiliary variables};
  $\alpha_{i-1} \leftarrow \alpha_i / e_i^{-1}$
end for
$e_1^{-1} \leftarrow \alpha_1$;
for $i = 1$ to $n-1$ do
  $p_i \leftarrow (1 - \frac{k_i m_i}{d_i e_{i+1}^{-1}})(1 - \frac{h_i m_i}{d_i e_{i+1}^{-1}})$; {Auxiliary variables}
end for
for $i = 1$ to $n-1$ do
  $\bar{\bar{m}}_i \leftarrow m_i p_{i+1} d_{i+1} / (p_i d_i e_{i+1}^{-2})$;  {Updating of $m_i$}
end for
$\delta_1 \leftarrow p_1 d_1 / e_1^{-1} - \sigma$;
for $i = 1$ to $n-1$ do
  $\bar{\bar{d}}_i \leftarrow \delta_i + h_i k_i m_i / (d_i e_{i+1}^{-1})$; {Updating of $d_i$}
  $\delta_{i+1} \leftarrow \delta_i p_{i+1} d_{i+1} / (p_i d_i e_{i+1}^{-1}) - \sigma(1 + \bar{\bar{m}}_i (1 + (\sigma - (h_i + k_i)) / \bar{\bar{d}}_i))$;
end for
$\bar{\bar{d}}_n \leftarrow \delta_n$;  {Updating of the last $d_n$}
for $i = 1$ to $n-1$ do
  $\bar{\bar{h}}_i \leftarrow h_i - \sigma$;
  $\bar{\bar{k}}_i \leftarrow k_i - \sigma$.
end for
```

4.3. The Laguerre shift. The choice of an adequate shift strategy is always of crucial importance for the convergence. Various shift strategies have been proposed, ranging from the classical Rayleigh shift defined as the entry in position $(n,\ n)$, to the Wilkinson shift in the case the matrix might have complex eigenvalues [27]. In this section we describe in detail a shift technique known as Laguerre shift [27], [18] which has shown to be particularly well suited in the case of quasiseparable matrices with real positive eigenvalues.

In particular, the shift $\sigma = \sigma_{k+1}$ in (4.14) is chosen in accordance with the following formula:

$$\sigma = \frac{n}{s_1^{(k)} + \sqrt{(n-1)(n s_2^{(k)} - (s_1^{(k)})^2)}}, \tag{4.17}$$

where $s_1^{(k)} = \mathrm{trace}((A^{(k)})^{-1})$ and $s_2 = \mathrm{trace}((A^{(k)})^{-2})$. The choice of the Laguerre shift guarantees that the eigenvalues are computed in an ordered way since $0 < \sigma \le \mu_n$, where μ_n is the smallest eigenvalue of $A^{(k)}$. If μ_n is simple, then $\sigma < \mu_n$.

We start from $A^{(k)}$ factorized as

$$A^{(k)} = L_s L_1 D R_1 R_s,$$

so we need the diagonal entries c_{ii} of

$$C = (A^{(k)})^{-1} = R_s^{-1} R_1^{-1} D^{-1} L_1^{-1} L_s^{-1}.$$

In the case x_i, $y_i \neq 0$, using the invariants[4] by direct computation, we have

$$\begin{cases} c_{nn} = d_n^{-1}, \\ c_{n-1,n-1} = d_{n-1}^{-1} + d_n^{-1} m_{n-1}(h_{n-1} d_{n-1}^{-1} - 1)(k_{n-1} d_{n-1}^{-1} - 1), \\ c_{ii} = d_i^{-1} + (d_{i+1}^{-1} + \sum_{r=i+1}^{n-1} (\prod_{j=i+1}^{r} m_j h_j k_j d_j^{-2}) d_{r+1}^{-1}) m_i (h_i d_i^{-1} - 1)(k_i d_i^{-1} - 1), \\ \qquad i = n-2, \dots, 1. \end{cases}$$

Setting

$$\begin{cases} t_{n-1} = d_n^{-1} m_{n-1}, \\ t_i = (d_{i+1}^{-1} + \sum_{r=i+1}^{n-1} (\prod_{j=i+1}^{r} m_j h_j k_j d_j^{-2}) d_{r+1}^{-1}) m_i, \quad i = n-2, \dots, 1, \end{cases}$$

the requested trace s_1 can be computed with about $14n$ flops in this way:

$$\begin{cases} c_{nn} = d_n^{-1}, \\ t_{n-1} = d_n^{-1} m_{n-1}, \\ t_i = (t_{i+1} h_{i+1} k_{i+1} d_{i+1}^{-1} + 1) m_i d_{i+1}^{-1}, \quad i = n-2, \dots, 1, \\ c_{ii} = d_i^{-1} + t_i (h_i d_i^{-1} - 1)(k_i d_i^{-1} - 1), \quad i = n-1, \dots, 1, \\ s_1 = \mathrm{trace}(C) = \sum_i^n c_{ii}. \end{cases}$$

Since $s_2 = \mathrm{trace}(C^2) = \sum_{i=1}^n c_{ii}^2 + 2 \sum_{j=1}^{n-1} \sum_{i=j+1}^{n} c_{ij} c_{ji}$, the sum of all the products $c_{ij} c_{ji}$ has to be computed. Let us start from those involving codiagonal entries $c_{i,i-1} c_{i-1,i}$, which have the form

$$\begin{cases} c_{n,n-1} c_{n-1,n} = m_{n-1}(h_{n-1} d_{n-1}^{-1} - 1)(k_{n-1} d_{n-1}^{-1} - 1) d_n^{-2}, \\ c_{n-1,n-2} c_{n-2,n-1} = (1 + h_{n-1} m_{n-1} d_n^{-1}(k_{n-1} d_{n-1}^{-1} - 1))(1 + k_{n-1} m_{n-1} d_n^{-1}(h_{n-1} d_{n-1}^{-1} - 1)) \\ \qquad\qquad m_{n-2}(h_{n-2} d_{n-2}^{-1} - 1)(k_{n-2} d_{n-2}^{-1} - 1) d_{n-1}^{-2}, \\ c_{i,i-1} c_{i-1,i} = (1 + h_i m_i (k_i d_i^{-1} - 1)(d_{i+1}^{-1} + \sum_{r=i+1}^{n-1} d_{r+1}^{-1} \prod_{j=i+1}^{r} m_j h_j k_j d_j^{-1})) \\ \qquad\qquad (1 + k_i m_i (h_i d_i^{-1} - 1)(d_{i+1}^{-1} + \sum_{r=i+1}^{n-1} d_{r+1}^{-1} \prod_{j=i+1}^{r} m_j h_j k_j d_j^{-1})) \\ \qquad\qquad m_{i-1}(h_{i-1} d_{i-1}^{-1} - 1)(k_{i-1} d_{i-1}^{-1} - 1) d_i^{-2}, \quad i = n-2, \dots, 2. \end{cases}$$

If we set

$$\begin{cases} t'_{n-1} = (1 + h_{n-1} m_{n-1} d_n^{-1}(k_{n-1} d_{n-1}^{-1} - 1)), \\ t''_{n-1} = (1 + k_{n-1} m_{n-1} d_n^{-1}(h_{n-1} d_{n-1}^{-1} - 1)), \\ t'_i = (1 + h_i m_i (k_i d_i^{-1} - 1)(d_{i+1}^{-1} + \sum_{r=i+1}^{n-1} d_{r+1}^{-1} \prod_{j=i+1}^{r} m_j h_j k_j d_j^{-1})), \\ \qquad i = n-2, \dots, 2, \\ t''_i = (1 + k_i m_i (h_i d_i^{-1} - 1)(d_{i+1}^{-1} + \sum_{r=i+1}^{n-1} d_{r+1}^{-1} \prod_{j=i+1}^{r} m_j h_j k_j d_j^{-1})), \\ \qquad i = n-2, \dots, 2, \end{cases}$$

[4] When some of the x_i or y_i are zero, we need to work with the full set of parameters since the invariants h_i and k_i cannot be computed. In that case we have different formulas, with a higher computational cost, not described here because they are too complicated. However, new formulas can be obtained following the same reasoning and using similar techniques as those employed in this section.

all products $c_{i,i-1}c_{i-1,i}$ can be computed according to the following scheme with $18n$ flops:

$$\begin{cases} t'_{n-1} = (1 + h_{n-1}m_{n-1}d_n^{-1}(k_{n-1}d_{n-1}^{-1} - 1)), \\ t''_{n-1} = (1 + k_{n-1}m_{n-1}d_n^{-1}(h_{n-1}d_{n-1}^{-1} - 1)), \\ t'_i = h_{i+1}m_{i+1}d_{i+2}(1 + t'_{i+1}k_{i+1}d_{i+2}), \quad i = n-2, \dots, 2, \\ t''_i = k_{i+1}m_{i+1}d_{i+2}(1 + t''_{i+1}h_{i+1}d_{i+2}), \quad i = n-2, \dots, 2, \\ z_{n-1} = d_n^{-2}, \\ z_i = (1 + t'_{i+1}(k_{i+1}d_{i+1}^{-1} - 1))(1 + t''_{i+1}(h_{i+1}d_{i+1}^{-1} - 1)), \quad i = n-2, \dots, 1, \\ c_{i,i-1}c_{i-1,i} = z_{i-1}m_{i-1}(h_{i-1}d_{i-1}^{-1} - 1)(k_{i-1}d_{i-1}^{-1} - 1), \quad i = n, \dots, 2. \end{cases}$$

The sums $\epsilon_j = \sum_{i=j+2}^{n} c_{ij}c_{ji}$, $j = n-2, \dots, 1$, can be computed in this way:

$$\begin{cases} w_{n-1} = 0, \\ w_j = (w_{j+1} + z_{j+1})m_{j+1}h_{j+1}k_{j+1}d_{j+1}^{-2}, \quad j = n-2, \dots, 1, \\ \epsilon_j = w_j m_j (h_j d_j^{-1} - 1)(k_j d_j^{-1} - 1), \quad j = n-2, \dots, 1. \end{cases}$$

Finally, the sum $\sum_{j=1}^{n-1}\sum_{i=j+1}^{n} c_{ij}c_{ji}$, which is required to complete the computation of s_2, can be expressed as

$$\begin{aligned} \sum_{j=1}^{n-1}\sum_{i=j+1}^{n} c_{ij}c_{ji} &= c_{n,n-1}c_{n-1,n} + \sum_{j=1}^{n-2}(\epsilon_j + c_{j+1,j}c_{j,j+1}) \\ &= \sum_{j=1}^{n-1}(w_j + z_j)m_j(h_j d_j^{-1} - 1)(k_j d_j^{-1} - 1), \end{aligned}$$

and it costs $9n$ flops more.

In the case where we deal with a restricted class of matrices, for example, when dealing with tridiagonal or semiseparable matrices, we can simplify this procedure and compute the shift with a lower number of operations. In section 7, for example, we simplified the computation of the shift when our method is applied to tridiagonal matrices, obtaining the Laguerre shift with a lower number of flops.

5. Stability and computational cost. In sections 4.1 and 4.2 we described the implicit LR algorithm acting on the Neville representation of the matrix, assuming that the algorithm proceeds without incurring in situations requiring the algorithm to stop. However, it is well known [27] that the LR process can break down if, at step k, the matrix $A^{(k)}$ is no more LU-factorizable. In practical cases, to overcome this situation and resume the iterative process, one can change the value of the shift σ_k, and hopefully the problem is not present in the new matrix. When our method is applied, however, as observed in section 4.1, we can have that the process halts also because one of the quantities appearing in the denominator of the recurrences in sections 4.1 or 4.2 becomes zero. We will refer to this anomalous situation as a breakdown, too.

In this section we first show that the class of Neville-representable quasiseparable matrices is closed under shifted LR steps. Then we analyze stability and structure preservation of the method when applied to TN matrices. Moreover, we show that both breakdown situations do not occur when the algorithm is applied to TN matrices.

In Theorem 10 it is proven that the quasiseparable structure is preserved under LR steps. The next corollary proves that, if no anomalous situations happen, also the subclass $\mathcal{S}$ is closed by LR steps.

Corollary 12. *Let $A \in \mathcal{S}$; if breakdown does not occur at each LR step, then each $A^{(k)}$ in* (4.1) *is still in $\mathcal{S}$.*

The proof is based on the observation that if breakdowns do not occur at each step, then we can construct a matrix $A^{(k+1)}$ similar to $A^{(k)} - \sigma_{k+1} I$, by means of updating the parameters involved in the Neville representation as described in section 4.1 or 4.2.

The natural class to which the proposed algorithm can be applied is the class of TN matrices.

Theorem 13. *Let $A^{(k)}$ be a TN quasiseparable matrix represented as in Definition* 4 *with $d_i > 0$, then the qd-type methods described in sections* 4.1 *and* 4.2 *without shift or with a shift which preserves the positivity of the eigenvalues have the following properties:*

1. *They are breakdown-free, and the updated parameters involved in the representation are all nonnegative.*
2. *The new matrix $A^{(k+1)}$ produced is still quasiseparable TN, with $d_i > 0$.*
3. *They do not contain subtractions (except a subtraction for the shift).*

Proof. For a TN matrix $A^{(k)}$, from Theorem 8 we know that x_i, $y_i \geq 0$, a_i, $b_i \leq 0$. If we assume x_i, $y_i \neq 0$, we have equivalently $m_i > 0$, h_i, $k_i \leq 0$. The shift σ_{k+1} is such that $0 < \sigma_{k+1} < \mu_n$, where μ_n is the smallest eigenvalue of $A^{(k)}$.

1. Consider the recurrences (4.9); we see immediately, by induction on i, that $e_i^{-1} \geq \alpha_i > 0$, where α_i are the auxiliary variables defined by (4.8). As a consequence, considering the other recurrences in section 4.1, we have $\bar{x}_i$, $\bar{y}_i \geq 0$, $\beta_i, \gamma_i \geq 1, f_i, g_i > 0$, $\bar{\bar{x}}_i, \bar{\bar{y}}_i \geq 0, \bar{a}_i, \bar{b}_i \leq 0$. Similarly, if we assume x_i, $y_i \neq 0$ and we refer to the formulation introduced in section 4.2 and used in Algorithm 1, we find that $p_i \geq 1$ and $\bar{\bar{m}}_i \geq 0$.
   If there is no shift, then $\bar{\bar{d}}_i \geq \delta_i > 0$, $i = 1, \ldots, n-1$, again by induction on i, and, as a consequence, $\bar{\bar{a}}_i$, $\bar{\bar{b}}_i \leq 0$.
   In the case of a shift, we have by hypothesis that at each step σ_{k+1} is such that $0 < \sigma_{k+1} < \mu_n$. We have to show also that in this case $\bar{\bar{d}}_i > 0$. We know that
   $$A^{(k+1)} = L^{(k)} A^{(k)} L^{(k)-1} - \sigma_{k+1} I = \bar{\bar{L}}_s T \bar{\bar{R}}_s,$$
   where $T = F \bar{R}_1 \bar{L}_1 G - \sigma_{k+1} \bar{\bar{L}}_s^{-1} \bar{\bar{R}}_s^{-1}$ is tridiagonal. Now, let T_j be the jth leading principal minor of T, observe that for each j, $\det(T_j) > 0$, since T_j is similar to $A_j^{(k)} - \sigma_{k+1} I_j$, where $A_j^{(k)}$ is the jth leading principal minor of $A^{(k)}$. From the interlacing property for TN matrices (Theorem 9), we have that $0 < \mu_n - \sigma_{k+1} \leq \mu_i^{(j)} - \sigma_{k+1}$, where $\mu_i^{(j)}$ is an eigenvalue of $A_j^{(k)}$. This means that $\det(T_j) > 0$, so T is strongly nonsingular, and in its LU factorization $T = \bar{\bar{L}}_1 \bar{\bar{D}} \bar{\bar{R}}_1$ the entries $\bar{\bar{d}}_i$ are all positive. Moreover, $\bar{\bar{a}}_i$, $\bar{\bar{b}}_i \leq 0$, as shown by (4.15) and (4.16).
   No breakdown happens, because no division by zero can occur in computing the quantities α_i, f_i, g_i, p_i, δ_i.
2. $A^{(k+1)}$ is still TN, because it is Neville-representable with nonnegative parameters. In particular, all $\bar{\bar{d}}_i$ are positive.
3. It is easy to check that the updating formulas in section 4.1 and Algorithm 1 described in section 4.2 do not contain subtractions when applied to TN matrices, with the exception of a subtraction for the computation of δ_{i+1}. In fact, e_i^{-1}, β_i, γ_i are sums of nonnegative quantities, and p_i in the second loop of Algorithm 1 is the product of sums of nonnegative quantities, since k_i, $h_i \leq 0$, and the other factors are positive. Moreover, by induction we can prove

that, in case no shift is applied, $\delta_i > 0$, and hence each $\bar{\bar{d}}_i$ is obtained (see (4.14)) as the sum of two nonnegative quantities, δ_i and $\bar{a}_i\bar{b}_i f_i g_i$, or $k_i h_i m_i/(d_i e_{i+1}^{-1})$ in Algorithm 1.

In the case of a shift, assume by contradiction that, for a given i, $\delta_i \leq 0$. It is easy to see that, if this is the case, $\delta_j \leq 0$ for all $j > i$, since δ_{i+1} is computed as the sum of two negative quantities in (4.14) and in the third loop of Algorithm 1 as well. But this is a contradiction since $\bar{\bar{d}}_n = \delta_n > 0$, as already proved. The updating of the invariants does not involve subtractions since $h_i,\ k_i \leq 0$ and $\sigma_{k+1} > 0$. □

The approximation of distinct eigenvalues in increasing order is a well-known property of LR convergence in the real positive case if the shift strategy preserves the ordering, as for Laguerre shift. In more detail, for nonsingular quasiseparable TN matrices the Laguerre shift σ_{k+1} is such that $0 < \sigma_{k+1} \leq \mu_n$. Furthermore, the shift computed in finite precision arithmetics is generally multiplied by a factor slightly less than one, also to prevent the effect of rounding errors which could destroy the TN structure.

As a consequence of the shift strategy the eigenvalues are computed in increasing order, so the deflation criterion is based only on the magnitude of the off-diagonal entries in the last row and column of $A^{(k)}$.

The cost of Algorithm 1 is about $17n$ flops without shift and about $27n$ in the case a shift strategy is applied. The cost doubles if one works without invariants, and it decreases if one deals, for example, with tridiagonal matrices, because the parameters x_i and y_i are not needed.

6. Reduction to tridiagonal form.

The Neville representation of quasiseparable matrices makes it easy to describe also an $O(n^2)$ algorithm for the reduction into tridiagonal form of a matrix in $\mathcal{S}$. A tridiagonalization procedure for TN matrices with the generalized quasiseparable structure has been described by Gemignani in [14] and is inspired by the algorithm of Koev designed for a generic TN matrix [15].

Algorithm 2. Tridiagonalization procedure.

```
$\hat{x}$←$x$; $\hat{y}$←$y$, $\hat{a}$←$a$; $\hat{b}$←$b$; $\hat{d}$←$d$;
for $j = 1$ to $n-1$ do
  for $i = n-1$ to $j$ do
    [$\hat{x}$, $\hat{y}$, $\hat{a}$, $\hat{b}$, $\hat{d}$]←swap ($\hat{x}$, $\hat{y}$, $\hat{a}$, $\hat{b}$, $\hat{d}$, $i$); {Annihilates row $i$}
    [$\hat{x}$, $\hat{y}$, $\hat{a}$, $\hat{b}$, $\hat{d}$]←swap ($\hat{y}$, $\hat{x}$, $\hat{b}$, $\hat{a}$, $\hat{d}$, $i$); {Annihilates column $i$}
  end for
end for
```

However, it is possible to see that the algorithm can be extended also to matrices not TN but belonging to the class $\mathcal{S}$. In the case where the tridiagonalization process is applied to TN matrices, the stability of the process is guaranteed since it is subtraction-free, and breakdown cannot occur since there are no divisions by zero. Let us describe the process of tridiagonalization of a matrix $A \in \mathcal{S}$ using the representation in terms of parameters $x_i,\ y_i,\ a_i,\ b_i,\ d_i$. The algorithm can be described as follows.

To understand the tridiagonalization procedure, note that each time we apply the `swap` procedure with parameter i, we annihilate the entry x_i or y_i appearing in the representation of the quasiseparable matrix. We need to annihilate each x_i several times, since the swap procedure creates a bulge that has to be removed with more Gauss transformations. In particular, we have to apply `swap` on row i for each column, and then we have a total of $n(n-1)$ calls to `swap`. The `swap` function, with parameter i, acts on rows i, $i+1$ and annihilates x_i, multiplying by a Gauss elementary matrix G_i on the right and by its inverse on the left. We have $G_i^{-1}AG_i = G_i^{-1}L_sL_1DR_1R_sG_i$, where $G_i = I_{i-1}\bigoplus E \bigoplus I_{n-i-1}$ and $E = \left[\begin{smallmatrix} 1 & 0 \\ x(i) & 1 \end{smallmatrix}\right]$. Reasoning similarly to what is done for deriving the recurrences in section 4.1, the Gauss transformation G_i acting on

Algorithm 3. $[\hat{x},\hat{y},\hat{a},\hat{b},\hat{d}]$←swap$(x,y,a,b,d,i)$ **.**

```
Require: $i <= n-1$
  $\alpha \leftarrow x(i)$    $\{E = \left[\begin{smallmatrix} 1 & 0 \\ \alpha & 1 \end{smallmatrix}\right]$ is the transformation on rows $i$ and $i+1\}$
  $\hat{x} \leftarrow x$, $\hat{y} \leftarrow y$, $\hat{a} \leftarrow a$, $\hat{b} \leftarrow b$, $\hat{d} \leftarrow d$;    $\{$Parameter initialization$\}$
  $\hat{x}(i) \leftarrow 0$;
  $w \leftarrow 1 + \alpha y(i)$;    $\{E^{(1)} = \left[\begin{smallmatrix} w & 0 \\ \alpha & 1/w \end{smallmatrix}\right]$ is the transformation acting on the left of $R_s\}$
  $\hat{y}(i-1) \leftarrow w y(i-1)$, $\hat{y}(i) \leftarrow y(i)/w$;
  $k \leftarrow w - \alpha b(i)$    $\{E^{(2)} = \left[\begin{smallmatrix} k & 0 \\ \alpha & 1/k \end{smallmatrix}\right]$ is the transformation acting on the left of $R_1\}$
  $\hat{b}(i-1) \leftarrow w b(i-1)$; $\hat{b}(i) \leftarrow k b(i)/w$;
  if $i \neq n-1$ then
    $\hat{b}(i+1) \leftarrow k b(i+1)$
  end if
  $\hat{d}(i) \leftarrow k d(i)$; $\hat{d}(i+1) \leftarrow d(i+1)/k$;
  $\hat{a}(i) \leftarrow a(i) - (\alpha d(i+1)/d(i))$;
  if $i \neq n-1$ then
    $\delta \leftarrow (\alpha d(i+1)/d(i)) a(i+1)/a(i)$,    $\{E^{(3)} = \left[\begin{smallmatrix} 1 & 0 \\ \delta & 1 \end{smallmatrix}\right]$ is the transformation acting
                                                             on rows $i+1$, $i+2$ on the left of $L_1\}$
    $\hat{a}(i+1) \leftarrow a(i+1) + \delta$;
  end if
  if $i \neq n-1$ then
    $\hat{x}(i+1) \leftarrow \delta - x(i+1)$
  end if
```

the right is moved inside as follows:

$$\begin{aligned} G_i^{-1}AG_i &= G_i^{-1}L_sL_1DR_1(R_sG_i) = L_s^{(1)}L_1DR_1G_i^{(1)}\hat{R}_s \\ &= L_s^{(1)}L_1D(R_1G_i^{(1)})\hat{R}_s = L_s^{(1)}L_1DG_i^{(2)}\hat{R}_1\hat{R}_s \\ &= L_s^{(1)}G_{i+1}^{(3)}\hat{L}_1\hat{D}\hat{R}_1\hat{R}_s = \hat{L}_s\hat{L}_1\hat{D}\hat{R}_1\hat{R}_s. \end{aligned}$$

In particular, with the same notation of the pseudocode in Algorithm 3, we have $G_i^{(1)} = I_{i-1}\bigoplus E^{(1)}\bigoplus I_{n-i-1}$ with $E^{(1)} = \left[\begin{smallmatrix} w & 0 \\ x(i) & 1/w \end{smallmatrix}\right]$, $G_i^{(2)} = I_{i-1}\bigoplus E^{(2)}\bigoplus I_{n-i-1}$ with $E^{(2)} = \left[\begin{smallmatrix} k & 0 \\ x(i) & 1/k \end{smallmatrix}\right]$, and finally $G_{i+1}^{(3)} = I_i\bigoplus E^{(3)}\bigoplus I_{n-i-2}$ with $E^{(3)} = \left[\begin{smallmatrix} 1 & 0 \\ \delta & 1 \end{smallmatrix}\right]$. Since $G_i^{(1)}L_s$ has only the effect of annihilating the ith row, when we multiply on the right by $G_{i+1}^{(3)}$, this will only update the entry $i+1$ of the vector describing $L_s^{(1)}$.

The swap procedure costs 19 flops; hence the cost of the tridiagonalization procedure is $19n^2$. Once the tridiagonal matrix is available, one can apply one of the well-known techniques for tridiagonal matrices. For example, since we already have the LU factorization, we can proceed by applying our method described in section 4.1 pruned of the unnecessary recurrences related to the updating of the parameters x_i

and y_i that now are zero. Similarly, one can apply the dqds algorithm [17]. Another possible way is to symmetrize the tridiagonal matrix first, and then apply the QR or the LL^H method. The cost of the dqds algorithm is about $6n$ flops plus the cost of the shift, that is, $31n$ if the Laguerre shift is applied using a simplified version of the formulas proposed in section 4.3.

7. Numerical experiments.

In this section we report some numerical results obtained using our methods for the computation of all eigenvalues of a possible unsymmetric quasiseparable matrix. The experiments were done using MATLAB 2006B on a Mac Powerbook, running OS X.5.

We denote by $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_n]$ the vector containing the exact eigenvalues and by $\tilde{\lambda} = [\tilde{\lambda}_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_n]$ the vector of the computed ones, sorted in such a way that $\tilde{\lambda}_i$ is an approximation of λ_i. For the purpose of estimating the relative error, we will consider the results of MATLAB's eig as the exact eigenvalues, that is, $\lambda = \mathtt{eig}(A)$. Accuracy is measured considering the absolute and relative error, that is,

$$E^{(\text{abs})} = \|\lambda - \tilde{\lambda}\|_\infty = \max_i \{|\lambda_i - \tilde{\lambda}_i|\}, \qquad E^{(\text{rel})} = \max_i \left\{ \frac{|\lambda_i - \tilde{\lambda}_i|}{|\lambda_i|} \right\}.$$

By QS-qd (for quasiseparable-qd) we denote the implementation of Algorithm 1 endowed by Laguerre shift as described in section 4.3. TridLR denotes the implementation of the tridiagonalization procedure described in Algorithm 2 followed by steps of the dqds method, as described in [17].

In our experiments we compared the eigenvalues computed with QS-qd with MATLAB eig, using a cutting criterion of 10^{-16} as deflation tolerance. It is well known that, for rank-structured matrices, it is often more convenient to compute the eigenvalues, by applying directly an iterative method without the preliminary reduction to tridiagonal or Hessenberg form [25]. In fact, the possibility of representing the matrices with a low number of parameters, and the closure under *GR*-type [26] steps, makes it convenient to apply directly an implicit method acting on the representation of the rank-structured matrix. The overhead of the computation of the tridiagonal structure is, in fact, not rewarded by the efficiency of the tridiagonal eigensolver. The purpose of our experimentation is to show that good accuracy and efficiency can be obtained by applying QS-qd directly on the representation of a quasiseparable matrix. The experimentation has three main goals. The first goal is a comparison with the specialized tools for eigenvalues computation of symmetric semiseparable plus diagonal (SSPD) matrices, both TN and not. To this extent, the results obtained with our solver both in terms of accuracy and CPU time are compared with the results obtained by using the QR solver implemented by the routine EIGSSD[5] which represents the state of the art of the QR implementation for SSPD matrices.

In the case of non-TN matrices (Table 7.1), our algorithm performs a little worse from the point of view of accuracy, but it is faster, requiring almost half the time required by the QR routine. To make a fair comparison, the time for the conversion from our representation to the Givens-vector representation used by the EIGSSD routine is not accounted for. In the case the matrix is TN, our algorithm performs better also in terms of accuracy, and we gain a digit over the QR implementation (see Table 7.2).

The second goal is to show the effectiveness of our method on (possibly) unsymmetric TN matrices. Table 7.3 reports the results obtained by our method on instances

[5] EIGSSD is a function included in the MATLAB package SSPACK accompanying the book [25], and it is available online at http://www.cs.kuleuven.be/~mase/.

Table 7.1
Random SSPD matrix. Comparison with `QS-qd` *and* `EIGSSD`.

n	$E^{(\mathrm{abs})}$	$E^{(\mathrm{rel})}$	`QS-qd` (sec)	$E^{(\mathrm{abs})}$	$E^{(\mathrm{rel})}$	`EIGSSD` (sec)
10	2.7732e-15	1.0333e-15	0.08	1.0151e-14	3.7824e-15	0.14
50	4.6343e-13	7.5474e-14	0.39	7.4737e-14	1.2172e-14	0.94
100	2.7996e-13	3.1850e-14	1.36	2.7323e-13	3.1085e-14	3.29
200	3.0727e-12	2.3750e-13	5.25	1.0149e-12	7.8444e-14	12.59
300	2.4134e-12	1.4828e-13	14.80	8.9760e-12	5.5147e-13	32.64
500	6.9054e-12	3.5405e-13	35.29	2.6940e-12	1.3812e-13	78.63

Table 7.2
Comparison for TN SSPD matrices between `QS-qd` *and* `EIGSSD`.

n	$E^{(\mathrm{abs})}$	$E^{(\mathrm{rel})}$	`QS-qd`(sec)	$E^{(\mathrm{abs})}$	$E^{(\mathrm{rel})}$	`EIGSSD` (sec)
10	1.8438e-14	1.8196e-15	0.10	2.4170e-14	2.3852e-15	0.17
50	7.5067e-14	4.2194e-15	1.98	1.4532e-13	1.2172e-14	4.39
100	6.4859e-14	3.0592e-15	6.40	2.0281e-12	9.5658e-14	16.66
200	3.5978e-13	9.3222e-15	10.05	1.5309e-12	3.9668e-14	31.09
300	4.4886e-13	8.2383e-15	10.71	1.2081e-12	2.2174e-14	22.60
400	4.6996e-13	8.9675e-15	21.38	1.6238e-12	3.0985e-14	48.55
500	6.4709e-13	1.0402e-14	39.42	6.0125e-10	9.6647e-12	76.97

Table 7.3
Results for TN random matrices.

n	niter	$E^{(\mathrm{abs})}$	$E^{(\mathrm{rel})}$	`QS-qd` (sec)
10	37	3.3526e-15	4.8898e-16	0.38
50	239	5.6901e-14	3.5140e-15	0.56
100	488	1.3217e-13	6.0148e-15	1.65
200	993	2.2589e-13	7.3909e-15	4.62
300	1546	3.1303e-13	8.7152e-15	10.61
400	2080	3.0187e-13	7.6370e-15	19.29
500	2576	3.8896e-13	8.6375e-15	33.69
700	3812	6.1653e-13	1.1881e-14	68.83
1000	5689	9.5412e-13	1.4728e-14	135.19

of random generated TN matrices of different sizes. The time in seconds of our implementation is reported as well. We see that the absolute and relative errors are quite good, and the increase of time with respect to the size shows, as expected, a quadratic behavior.

The third goal consists of testing our `QS-qd` implementations on some "difficult" problems to see how accuracy is affected.[6] In Table 7.4 the results on a TN matrix with a small eigenvalue equal to α are reported. We see that, when the eigenvalue α becomes of the order of the machine precision, the results are meaningless because $E^{(rel)}$ is of order 10^{-1}. Looking at the plot in logarithmic scale in Figure 7.1 of relative errors of each

[6] We wish to thank one of the anonymous referees for suggesting we perform this group of experiments.

TABLE 7.4
Relative errors for random TN matrices with the smallest eigenvalue equal to α.

n	$\alpha = 10^{-3}$	$\alpha = 10^{-4}$	$\alpha = 10^{-5}$	$\alpha = 10^{-6}$	$\alpha = 10^{-8}$	$\alpha = 10^{-16}$
100	3.0184e-13	2.1717e-12	2.4724e-11	8.8302e-11	8.1036e-09	8.9569e-02
200	4.1395e-13	8.3456e-13	5.5461e-11	1.6362e-10	1.2799e-08	2.3694e-01
300	6.8436e-13	7.4789e-12	3.8750e-11	2.7403e-10	7.9448e-08	4.9603e-01
400	1.7029e-13	2.1902e-12	2.3919e-11	1.4864e-09	1.7421e-08	2.7570e-01
500	4.5175e-13	6.0512e-12	1.0577e-11	3.3654e-10	2.6841e-08	7.9474e-01

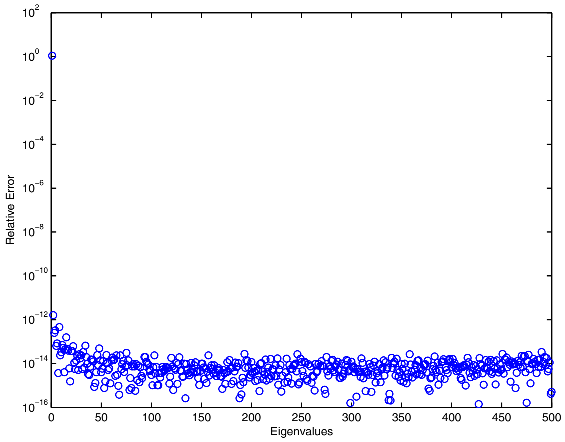

FIGURE 7.1. *Plot of the relative errors (in a logarithmic scale) of the computed eigenvalues of a randomly generated TN matrix of order* 500 *with an eigenvalue equal to* 10^{-16}.

eigenvalue approximated, we see, however, that the other eigenvalues are approximated very well. As we pointed out in section 5, the stability of `QS-qd` is not guaranteed for a generic symmetric matrix. However, the potentially dangerous subtraction of Algorithm 1 can only occur in the computation of the p_i at early steps of the process, since at convergence m_i goes to zero. In Table 7.5 the results obtained on a particular arrowhead matrix, with well-distributed eigenvalues, are reported. The algorithm in this case is fast and accurate despite the fact that the matrix is not TN.

Table 7.6 reports the behavior of the `QS-qd` method on a matrix with a couple of ill-conditioned eigenvalues. As test matrix we consider the inverse of the tridiagonal with

TABLE 7.5
An example of a matrix which is not TN. $A = (\mathrm{ones}(n) + \mathrm{diag}(0\colon n-1))^{-1}$. *A is an arrowhead symmetric matrix.*

n	niter	$E^{(\mathrm{abs})}$	$E^{(\mathrm{rel})}$	`QS-qd` (sec)
10	39	7.6328e-16	1.7383e-16	0.18
50	222	3.8659e-15	6.5444e-16	0.34
100	444	1.0880e-14	1.6580e-15	1.28
200	883	1.3769e-14	1.9065e-15	4.32
300	1314	1.1747e-14	1.5437e-15	9.56
500	2162	1.3649e-14	1.6849e-15	27.06

Table 7.6

$A = (\text{diag}(\text{ones}[(n-2,1);\alpha],-1) + \text{diag}(\text{ones}[(n-1,1),1) + \text{eye}(n))^{-1}$. *For different values of the parameter* α*, the behavior of the relative error and of the maximum conditioning of the eigenvalues is reported.*

n	$\alpha = 10^3$	$\alpha = 10^4$	$\alpha = 10^6$	$\alpha = 10^8$	$\alpha = 10^9$
50	6.8360e-13	4.7427e-12	8.6504e-10	1.2536e-07	2.4598e-06
100	1.6239e-11	6.7904e-12	5.1840e-09	1.4151e-07	1.2406e-06
200	3.7919e-07	3.6445e-07	3.5502e-07	5.8907e-07	1.1434e-06
300	3.8272e-07	3.8317e-07	3.8699e-07	4.3830e-07	1.0376e-05
max(condeig)	15.8272	50.0050	5.0000e+02	5.0000e+03	1.5811e+04

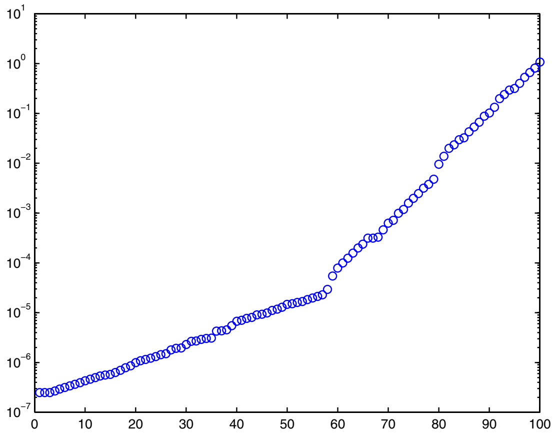

Figure 7.2. *Logarithmic plot of the distribution of the eigenvalues of a semiseparable matrix of size* 100.

all entries equal to one, except for the entry $(n, n-1)$ equal to α. For this problem, the conditioning of the eigenvalues increases with α, but it does not depend on the size of the matrix. We see that accuracy slightly deteriorates as α increases.

Another interesting numerical experiment concerns the behavior of the method on a semiseparable matrix with clustered eigenvalues.[7] We constructed a 100×100 symmetric semiseparable matrix whose eigenvalues are distributed according to Figure 7.2. Such a matrix has been constructed first generating a random orthogonal matrix Q and then constructing $A = QDQ^T$, where D is a diagonal matrix whose eigenvalues are distributed as follows: 20 eigenvalues geometrically distributed between 1 and 10^{-2}, 20 eigenvalues between $5 \cdot 10^{-3}$ and $5 \cdot 10^{-5}$, and 60 eigenvalues between $2.5 \cdot 10^{-5}$ and $2.5 \cdot 10^{-7}$. We then reduced A to semiseparable form employing the algorithm proposed in [3] and available online at http://www.cs.kuleuven.be/~mase/. We then computed, with MATLAB's `eig`, the eigenvalues of the matrix obtained through this procedure, obtaining the values plotted in Figure 7.2, which differ slightly from the theoretical values but are still clustered. Applying our method, we obtain a maximum relative error of

[7] This example has been taken from [22].

Table 7.7

For TN matrices, comparison between the two approaches proposed in this paper. The TridLR *first applies the tridiagonalization procedure of section* 6 *and the steps of the* dqds *algorithm.*

n	$E^{(\mathrm{abs})}$	$E^{(\mathrm{rel})}$	TriLR(sec)	$E^{(\mathrm{abs})}$	$E^{(\mathrm{rel})}$	QS-qd (sec)
10	3.3562e-15	7.2967e-16	0.17	6.2538e-15	1.3596e-15	0.10
50	3.7408e-14	3.1866e-15	0.78	3.6114e-14	3.0764e-15	0.38
100	9.8449e-14	4.4702e-15	2.26	1.1809e-13	5.3618e-15	1.35
200	2.1553e-13	7.0867e-15	8.74	2.5090e-13	8.2494e-15	4.69
400	2.4226e-13	6.3467e-15	39.96	3.1101e-13	8.1479e-15	21.79
600	3.3020e-13	6.5145e-15	80.22	4.8100e-13	9.4897e-15	49.92
800	3.9049e-13	7.1229e-15	149.61	7.0710e-13	1.2898e-14	91.31
1000	4.9826e-13	7.8901e-15	263.43	1.0501e-12	1.6629e-14	136.11

2.9099e-10, while the mean error for all the eigenvalues was 6.5959e-12, suggesting a better behavior on the average.

On TN quasiseparable matrices, a comparison with the method (denoted by TridLR) described in section 6, which first reduces the matrix in tridiagonal form and then applies the method dqds [17] for tridiagonal matrices, is reported in Table 7.7. The comparison between the flops required by the tridiagonalization procedure followed by standard qd technique for tridiagonals (denoted, as mentioned before, as TridLR) and QS-qd seems to suggest that it is more convenient to first reduce the matrix into tridiagonal form. However, if one compares the time required by the two algorithms as reported in Table 7.7, we note that the times obtained are not those expected, while the accuracy is comparable. This suggests that the comparison of the times is more appropriate than flop count since the interpreter or compiler—depending on the language the code is written in—can optimize the code to get faster execution times when the flop count would suggest a different behavior.[8]

8. Conclusions.

In this paper two approaches to the computation of the eigenvalues of a quasiseparable Neville-representable matrix have been proposed. The first one is a qd-type algorithm inspired by the qd methods for tridiagonal matrices. The second idea is to reduce the matrix to tridiagonal form and then apply a method for tridiagonal matrices, for instance, the same qd algorithm.

We have presented several theoretical results showing that the class of Neville-representable matrices is closed under LR steps. For TN matrices we proved also that the proposed algorithms are subtraction-free and breakdown-free, and, if a shift-preserving positivity is adopted, then each LR step produces a new matrix still TN.

Extensive numerical testing has been performed showing the effectiveness of this approach.

Acknowledgment. The authors are indebted to the two anonymous referees whose comments and suggestions helped improve the quality of the presentation.

[8] This has been verified executing the same code, using different MATLAB versions and different machines. In particular, with MATLAB Release 12, the algorithm TridLR is a bit faster than QS-qd, but for all the subsequent versions we tested, namely, MATLAB R2006B and MATLAB R2011B, the QS-qd outperformed TridLR.

REFERENCES

[1] T. Ando, *Totally positive matrices*, Linear Algebra Appl., 90 (1987), pp. 165–219.

[2] M. Van Barel, R. Vandebril, and N. Mastronardi, *A note on the representation and definition of semiseparable matrices*, Numer. Linear Algebra Appl., 12 (2005), pp. 839–858.

[3] M. Van Barel, R. Vandebril, and N. Mastronardi, *An orthogonal similarity reduction of a matrix to a semiseparable form*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 176–197.

[4] T. Bella, Y. Eidelman, I. Gohberg, and V. Olshevsky, *Computations with quasiseparable polynomials and matrices*, Theoret. Comput. Sci., 409 (2008), pp. 158–179.

[5] D. A. Bini, L. Gemignani, and V. Y. Pan, *QR-like algorithms for generalized semiseparable matrices*, Technical report 1470, Department of Mathematics, University of Pisa, Pisa, Italy, 2003.

[6] F. Dopico, T. Bella, and V. Olshevsky, *Fast and accurate computations with some classes of quasiseparable matrices*, in SIAM Conference on Applied Linear Algebra, Monterey, CA, 2009.

[7] F. Dopico, T. Bella, and V. Olshevsky, *Fast and accurate computations with totally nonnegative quasiseparable matrices*, in Communication at Positive Systems: Multidisciplinary International Symposium on Positive Systems: Theory and Applications (POSTA09), Valencia, Spain, 2009.

[8] Y. Eidelman and I. Gohberg, *On a new class of structured matrices*, Integral Equations Operator Theory, 34 (1999), pp. 293–324.

[9] Y. Eidelman, I. Gohberg, and V. Olshevsky, *The QR iteration method for Hermitian quasiseparable matrices of an arbitrary order*, Linear Algebra Appl., 404 (2005), pp. 305–324.

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

[11] F. R. Gantmacher and M. G. Krein, *Oszillationsmatrizen, Oskillationskerne und Kleine Schwingungen Mechanischer Systeme*, Akademie-Verlag, Berlin, 1960.

[12] M. Gasca and J. M. Peñfa, *A matricial description of Neville elimination with applications to total positivity*, Linear Algebra Appl., 202 (1994), pp. 33–53.

[13] M. Gasca and J. M. Peña, *On factorization of totally positive matrices*, in Total Positivity and Its Applications, M. Gasca and C. A. Micchelli, eds., Kluwer, Norwell, MA, 1996, pp. 109–132.

[14] L. Gemignani, *Neville elimination for rank-structured matrices*, Linear Algebra Appl., 428 (2008), pp. 978–991.

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

[16] N. Mastronardi, S. Chandrasekaran, and S. Van Huffel, *Fast and stable two-way algorithm for diagonal plus semi-separable systems of linear equations*, Numer. Linear Algebra Appl., 8 (2001), pp. 7–12.

[17] B. N. Parlett, *The new qd algorithms*, Acta Numer., 4 (1995), pp. 459–491.

[18] B. Plestenjak, M. Van Barel, and E. Van Camp, *A Cholesky LR algorithm for the positive definite symmetric diagonal-plus-semiseparable eigenproblem*, Linear Algebra Appl., 428 (2008), pp. 586–599.

[19] P. Rózsa, R. Bevilacqua, F. Romani, and P. Favati, *On the inverse of block tridiagonal matrices with application to the inverses of band matrices and block band matrices*, in The Calgary Conference and Matrix Theory Papers, Oper. Theory Adv. Appl. 40, S. Goldberg, M. Kaashoeck, and P. Lancaster, eds., Bikhaüser, Basel, Switzerland, 1989, pp. 447–469.

[20] H. Rutishauser, *Solution of eigenvalue problem with the LR-transformation*, Nat. Bur. Standards Appl. Math. Ser. N. 49, 49 (1958), pp. 47–81.

[21] E. E. Tyrtyshnikov, *Mosaic ranks for weakly semiseparable matrices*, in Large-Scale Scientific Computations of Engineering and Enviromental Problems II, M. Griebel, S. Margenov, and P. Y. Yalamov, eds., Notes on Numerical Fluid Mechanics 73, Vieweg, Wiesbaden, Germany, 2000, pp. 36–41.

[22] R. Vandebril, M. Van Barel, and N. Mastronardi, *A comparison between the semiseparable and the tridiagonal eigenvalue solvers*, in Sixteenth International Symposium on Mathematical Theory of Networks and Systems, B. De Moor, et al., eds., 2004, pp. 1–10.

[23] R. Vandebril, M. Van Barel, and N. Mastronardi, *An implicit QR-algorithm for symmetric semiseprable matrices*, Numer. Linear Algebra Appl., 12 (2005), pp. 625–658.

[24] R. Vandebril, M. Van Barel, and N. Mastronardi, *Matrix Computations and Semiseparable Matrices, Volume* 1: *Linear Systems*, Johns Hopkins University Press, Baltimore, MD, 2008.

[25] R. Vandebril, M. Van Barel, and N. Mastronardi, *Matrix Computations and Semiseparable Matrices, Volume* 2: *Eigenvalue and Singular Value methods*, Johns Hopkins University Press, Baltimore, MD, 2008.

[26] D. S. Watkins, *The Matrix Eigenvalue Problem: GR and Krylov Subspace Methods*, SIAM, Philadelphia, 2007.

[27] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, London, 1965.