A quasiseparable approach to solve the definite generalized eigenvalue problem

Raf Vandebril, Marc Van Barel and Nicola Mastronardi

Raf Vandebril

Department of Computer Science
KU Leuven, Belgium
Raf.Vandebril@cs.kuleuven.be

Marc Van Barel

Department of Computer Science
KU Leuven, Belgium
Marc.VanBarel@cs.kuleuven.be

Nicola Mastronardi

Istituto per le Applicazioni del
Calcolo M. Picone, Bari, Italy
N.Mastronardi@ba.iac.cnr.it

Abstract

We present a new fast algorithm for solving the generalized eigenvalue problem $T\mathbf{x} = \lambda S\mathbf{x}$, in which both T and S are real symmetric tridiagonal matrices and S is positive definite. A method for solving this problem is to compute a Cholesky factorization $S = LL^T$ and solve the equivalent symmetric standard eigenvalue problem $L^{-1}TL^{-T}(L^T\mathbf{x}) = \lambda(L^T\mathbf{x})$. We prove that the matrix $L^{-1}TL^{-T}$ is quasi-separable; that is, all submatrices taken out of its strictly lower triangular part have rank at most 1. We show how to efficiently compute the $\mathcal{O}(n)$ parameters defining $L^{-1}TL^{-T}$ and review eigensolvers for quasi-separable matrices. Our approach shows that by fully exploiting the structure, the eigenvalues of $T\mathbf{x} = \lambda S\mathbf{x}$ can be computed in $\mathcal{O}(n^2)$ operations, as opposed to the $\mathcal{O}(n^3)$ operations for standard methods such as the so-called Cholesky- QR method. It will be shown that the computation of the representation of this quasi-separable matrix is only linear in time, and numerical experiments will illustrate the effectiveness of the presented approach.

Article information

- Vandebril, Raf; Van Barel, Marc; Mastronardi, Nicola. *A quasiseparable approach to solve the definite generalized eigenvalue problem*, SIAM Journal on Matrix Analysis and Applications, volume 31, issue 1, pages 154-174, 2009
- This article equals the final publisher's version. A link is found below.
- Journal's homepage:
<https://www.siam.org/journals/simax.php>
- Published version: <http://dx.doi.org/10.1137/070679491>
- KU Leuven's repository url: <https://lirias.kuleuven.be/handle/123456789/256358>

A QUASI-SEPARABLE APPROACH TO SOLVE THE SYMMETRIC DEFINITE TRIDIAGONAL GENERALIZED EIGENVALUE PROBLEM*

RAF VANDEBRIL[†], GENE GOLUB[‡], AND MARC VAN BAREL[†]

Abstract. We present a new fast algorithm for solving the generalized eigenvalue problem $T\mathbf{x} = \lambda S\mathbf{x}$, in which both T and S are real symmetric tridiagonal matrices and S is positive definite. A method for solving this problem is to compute a Cholesky factorization $S = LL^T$ and solve the equivalent symmetric standard eigenvalue problem $L^{-1}TL^{-T}(L^T\mathbf{x}) = \lambda(L^T\mathbf{x})$. We prove that the matrix $L^{-1}TL^{-T}$ is quasi-separable; that is, all submatrices taken out of its strictly lower triangular part have rank at most 1. We show how to efficiently compute the $\mathcal{O}(n)$ parameters defining $L^{-1}TL^{-T}$ and review eigensolvers for quasi-separable matrices. Our approach shows that by fully exploiting the structure, the eigenvalues of $T\mathbf{x} = \lambda S\mathbf{x}$ can be computed in $\mathcal{O}(n^2)$ operations, as opposed to the $\mathcal{O}(n^3)$ operations for standard methods such as the so-called Cholesky- QR method. It will be shown that the computation of the representation of this quasi-separable matrix is only linear in time, and numerical experiments will illustrate the effectiveness of the presented approach.

Key words. generalized eigenvalue problem, quasi-separable, tridiagonal matrices

AMS subject classifications. 65F15, 15A18

DOI. 10.1137/070679491

1. Introduction. In this paper we consider generalized eigenvalue problems of the form

$$T\mathbf{x} = \lambda S\mathbf{x},$$

where both T and S are symmetric tridiagonal matrices and S is positive definite. This problem arises in several applications such as the numerical solution of the radial Schrödinger and Sturm–Liouville equations and in vibrational analysis [4, 47]. More references to applications can be found in [38].

A method for solving this problem is the reduction to a standard eigenvalue problem in the following sense:

$$L^{-1}TL^{-T}(L^T\mathbf{x}) = \lambda(L^T\mathbf{x}),$$

where $S = LL^T$ is the Cholesky decomposition of the matrix S (see [33, 44]). This approach is considered less attractive since the generated matrix $L^{-1}TL^{-T}$ is dense.

*Received by the editors January 9, 2007; accepted for publication (in revised form) by Z. Drmač September 9, 2008; published electronically February 27, 2009.

<http://www.siam.org/journals/simax/31-1/67949.html>

[†]Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Leuven (Heverlee), Belgium (Raf.Vandebril@cs.kuleuven.be, Marc.VanBarel@cs.kuleuven.be). The research of these authors was partially supported by the Research Council K.U.Leuven, project OT/05/40 (Large rank structured matrix computations), Center of Excellence: Optimization in Engineering, by the Fund for Scientific Research–Flanders (Belgium), Iterative methods in numerical Linear Algebra, G.0455.0 (RHPH: Riemann–Hilbert problems, random matrices and Padé–Hermite approximation), G.0423.05 (RAM: Rational modelling: Optimal conditioning and stable algorithms), and by the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister’s Office for Science, Technology and Culture, project IUAP V-22 (Dynamical Systems and Control: Computation, Identification & Modelling). The first author also received a grant of “Postdoctoraal Onderzoeker” from the Funds of Scientific Research Flanders (FWO-Vlaanderen).

[‡]This author is deceased. Former address: Department of Computer Science, Stanford University, Stanford, CA 94305.

This leads hence to an $\mathcal{O}(n^3)$ method for computing the eigenvalues. Moreover, the accuracy of this method also depends on the condition number of S , as the inverse of its Cholesky factors is required. In [12] a detailed error analysis of the Cholesky- QR method is presented, providing error bounds potentially much smaller than $\kappa_2(S)u$, in which u is the unit round-off. Moreover, [12] suggests using for the symmetric definite generalized eigenvalue problem the Cholesky- QR method in which complete pivoting is used to compute the Cholesky factorization. In this paper we opt to use the Cholesky decomposition, but unfortunately we cannot use pivoting since this would destroy the structure and increase the computational complexity of the method.

Also other different techniques exist, working directly on the $T\mathbf{x} = \lambda S\mathbf{x}$ problem. These methods take advantage of both the tridiagonal and symmetric structure and lead to $\mathcal{O}(n^2)$ methods. In [38] one computes the eigenvalues by applying Laguerre's iteration on the associated characteristic polynomial of the pencil. In [6, 29] a divide-and-conquer method is presented. Also methods for the banded symmetric generalized matrix eigenvalue problem exist [36]. More related references can be found, e.g., in [38].

In this paper we prove that the dense matrix $L^{-1}TL^{-T}$ is quasi-separable; that is, all submatrices taken out of the strictly lower triangular part of this matrix are of rank at most one (see [26, 24]).

We show how to efficiently compute the representation of the quasi-separable matrix $L^{-1}TL^{-T}$. As the quasi-separable matrix is highly structured, only $\mathcal{O}(n)$ parameters are needed to define it. We will represent the quasi-separable matrix using the Givens-vector representation [55]. A fast $\mathcal{O}(n)$ method for transforming the generalized eigenvalue problem into an eigenvalue problem involving a quasi-separable matrix will be given.

Based on the $\mathcal{O}(n)$ representation of the quasi-separable matrix, alternative fast methods for computing the whole spectrum are reviewed. We refer to section 3 for an overview.

The manuscript is organized as follows. In the first section some definitions and a theoretical proof of the structure of the matrix $L^{-1}TL^{-T}$ are given. To conclude this section a description is given of the representation used for the quasi-separable matrix and a method for effectively computing the quasi-separable matrix representation of $L^{-1}TL^{-T}$. Section 3 discusses some methods for computing the eigenvalues of quasi-separable matrices. The final section of this manuscript presents the implementation and some numerical experiments related to the accuracy of the new technique for computing the eigenvalues of the generalized eigenvalue problem.

2. Transforming the generalized eigenvalue problem. We begin with a couple of definitions.

DEFINITION 2.1. *A matrix $A \in \mathbb{R}^{n \times n}$ is called quasi-separable if any submatrix taken out of the strictly lower and strictly upper triangular part has rank at most one. More precisely this means that¹*

$$\begin{aligned} \text{rank } A(i : n, 1 : i - 1) &\leq 1 & \text{for all } i &= 2, \dots, n, \\ \text{rank } A(1 : i - 1, i : n) &\leq 1 & \text{for all } i &= 2, \dots, n. \end{aligned}$$

We will also need to refer to lower/upper triangular semiseparable matrices, whose definition is given below.

¹We use the colon notation.

DEFINITION 2.2. A matrix $A \in \mathbb{R}^{n \times n}$ is called semiseparable if any submatrix taken out of the lower and upper triangular part has rank at most 1; that is, semiseparable matrices satisfy

$$\begin{aligned} \text{rank } A(i : n, 1 : i) &\leq 1 \quad \text{for all } i = 1, \dots, n, \\ \text{rank } A(1 : i, i : n) &\leq 1 \quad \text{for all } i = 1, \dots, n. \end{aligned}$$

A matrix is called lower (resp., upper) semiseparable or quasi-separable if only the lower (resp., upper) triangular part satisfies the rank constraints.

The only difference between quasi-separable and semiseparable is the fact that a quasi-separable matrix does not have the diagonal included in the low rank structure, whereas a semiseparable matrix does have this diagonal included in the structure. Even though these definitions seem little different from each other, there are quite significant differences. For example, the inverse of a semiseparable matrix is of tridiagonal form, whereas the inverse of a quasi-separable matrix is again a quasi-separable matrix. The quasi-separable class of matrices contains also the semiseparable and the tridiagonal matrices.

In this paper some of the discussed references deal with semiseparable plus diagonal matrices instead of quasi-separable matrices. The techniques presented can, however, be adapted easily to be applicable on quasi-separable matrices.

2.1. Theoretical proof of the structure. Let us reconsider now the matrix product $L^{-1}TL^{-T}$, with L satisfying $S = LL^T$. Due to the fact that the matrix S is tridiagonal, the matrix L will be of lower bidiagonal form, and the matrix $L^{-1}TL^{-T}$ will be of quasi-separable form. We will formulate this as a theorem.

THEOREM 2.3. For a symmetric tridiagonal matrix T and a nonsingular lower bidiagonal matrix L , the product

$$A = L^{-1}TL^{-T}$$

is a symmetric quasi-separable matrix.

Proof. We assume the considered matrices to be of size n . It is well known that the inverse of a lower bidiagonal matrix is a lower triangular semiseparable matrix (see, e.g., [31]).

The QR -factorization of such a lower semiseparable matrix can be computed by performing a bottom-up sequence of Givens transformations, $n - 1$ in total. More precisely the first Givens transformation is performed on the bottom two rows of the matrix L^{-1} to remove the complete last row up to the diagonal. Note that it is possible to remove this complete row with one transformation as the last and second-to-last rows are dependent on each other due to the semiseparable structure. The second Givens transformation acts on rows $n - 2$ and $n - 1$ and removes the whole row $n - 1$ up to the diagonal. This procedure can easily be repeated and gives us the following factorization:

$$G_1^T G_2^T \dots G_{n-2}^T G_{n-1}^T L^{-1} = R,$$

with R an upper triangular matrix. This means that $L^{-1} = G_{n-1}G_{n-2} \dots G_2G_1R$, which is the QR -factorization of the considered matrix.

Let us now look closer at the structure of the matrix

$$L^{-1}T = G_{n-1}G_{n-2} \dots G_2G_1RT.$$

The matrix RT is upper Hessenberg.

The remainder of the proof proceeds by finite induction. Let $\tilde{A}^{(k)} = G_k \dots G_1 RT$. The Givens transformations G_i equal the identity matrix except for the diagonal block $G_i(i : i+1, i : i+1)$, which is of the following form:

$$G_i(i : i+1, i : i+1) = \begin{bmatrix} c_i & -s_i \\ s_i & c_i \end{bmatrix}, \quad \text{where } c_i^2 + s_i^2 = 1.$$

For every k (with $1 \leq k \leq n-1$) we will prove that the constraints

$$(2.1) \quad \text{rank } \tilde{A}^{(k)}(i : n, 1 : i-1) \leq 1 \quad \text{for all } i = 2, \dots, n$$

are satisfied. Therefore, every intermediate matrix $\tilde{A}^{(k)}$ as well as $\tilde{A}^{(n-1)} = L^{-1}T$ will be of lower quasi-separable form.

- Consider $k = 1$. The matrix $\tilde{A}^{(1)} = G_1 RT$ is a Hessenberg matrix, in which we denote the first subdiagonal element as v_1 :

$$\tilde{A}^{(1)} = \begin{bmatrix} \times & \times & \times & \times & \cdots \\ \times & \times & \times & \times & \cdots \\ & \times & \times & \times & \\ & & \times & \times & \\ & & & \ddots & \ddots \end{bmatrix} = \begin{bmatrix} \times & \times & \times & \times & \cdots \\ v_1 & \times & \times & \times & \cdots \\ & \times & \times & \times & \\ & & \times & \times & \\ & & & \ddots & \ddots \end{bmatrix}.$$

The constraints (2.1) are clearly satisfied. For simplicity we include the case $k = 2$.

- Assume $k = 2$. The matrix $\tilde{A}^{(2)} = G_2 G_1 RT$ will be of the following form (in the right-hand matrix we denote the element in position $(3, 2)$ by v_2):

$$\tilde{A}^{(2)} = \begin{bmatrix} \times & \times & \times & \times & \cdots \\ c_2 v_1 & \times & \times & \times & \cdots \\ s_2 v_1 & \times & \times & \times & \\ & \times & \times & & \\ & & \ddots & \ddots & \end{bmatrix} = \begin{bmatrix} \times & \times & \times & \times & \cdots \\ c_2 v_1 & \times & \times & \times & \cdots \\ s_2 v_1 & v_2 & \times & \times & \\ & \times & \times & & \\ & & \ddots & \ddots & \end{bmatrix}.$$

- By induction we assume that the statement holds for $k-1$. Let us show rows $k-2, k-1, k$, and $k+1$ of the matrix $\tilde{A}^{(k-1)}$:

$$\begin{bmatrix} & & & & \ddots & & & & \\ c_{k-2} s_{k-3} \cdots s_3 s_2 v_1 & c_{k-2} s_{k-3} \cdots s_3 v_2 & \cdots & c_{k-2} v_{k-3} & \times & & & & \\ c_{k-1} s_{k-2} \cdots s_3 s_2 v_1 & c_{k-1} s_{k-2} \cdots s_3 v_2 & \cdots & c_{k-1} s_{k-2} v_{k-3} & c_{k-1} v_{k-2} & \times & & & \\ s_{k-1} s_{k-2} \cdots s_3 s_2 v_1 & s_{k-1} s_{k-2} \cdots s_3 v_2 & \cdots & s_{k-1} s_{k-2} v_{k-3} & s_{k-1} v_{k-2} & v_{k-1} & \times & & \\ 0 & 0 & \cdots & 0 & 0 & 0 & \times & \ddots & \end{bmatrix}$$

Performing the Givens transformation G_k on $\tilde{A}^{(k-1)}$ gives us the matrix $\tilde{A}^{(k)}$ which is of the following form (only rows $k-1, k, k+1$, and $k+1$ are depicted):

$$\begin{bmatrix} & & & & \ddots & & & & \\ c_{k-1} s_{k-2} \cdots s_3 s_2 v_1 & c_{k-1} s_{k-2} \cdots s_3 v_2 & \cdots & c_{k-1} v_{k-2} & \times & & & & \\ c_k s_{k-1} s_{k-2} \cdots s_3 s_2 v_1 & c_k s_{k-1} s_{k-2} \cdots s_3 v_2 & \cdots & c_k s_{k-1} v_{k-2} & c_k v_{k-1} & \times & & & \\ s_k s_{k-1} s_{k-2} \cdots s_3 s_2 v_1 & s_k s_{k-1} s_{k-2} \cdots s_3 v_2 & \cdots & s_k s_{k-1} v_{k-2} & s_k v_{k-1} & v_k & \times & & \\ 0 & 0 & \cdots & 0 & 0 & 0 & \times & \ddots & \end{bmatrix}$$

One can check that the resulting matrix is lower quasi-separable. This concludes the induction procedure.

Based on the induction procedure we can conclude that the matrix $L^{-1}T$ is lower quasi-separable.

Due to the fact that the matrix L^{-T} is upper triangular, it is obvious that a multiplication of $L^{-1}T$ on the right with the matrix L^{-T} does not change the low rank structure below the diagonal. Hence we have proved that our matrix $A = L^{-1}TL^{-T}$ has the lower triangular part of quasi-separable form. Due to symmetry also the upper triangular part satisfies these constraints, and hence the complete matrix is quasi-separable. $\square$

Let us now see how we can effectively represent a quasi-separable matrix and how to compute this representation.

2.2. The Givens-vector representation. We proved in the previous theorem that the resulting matrix is of quasi-separable form. To be able to work with the matrix an effective representation of the low rank part is necessary. A straightforward choice might be to represent the low rank part as coming from a rank-one matrix. This means representing the lower triangular part as coming from $\mathbf{u}\mathbf{v}^T$, with $\mathbf{u}$ and $\mathbf{v}$ two vectors. This is, however, a bad choice. First, this representation does not cover all kinds of quasi-separable matrices (consider, e.g., a tridiagonal matrix), and second, it suffers heavily from numerical instabilities, when computing, e.g., the spectrum via a QR -method for quasi-separable matrices. More information on the problems with this representation can be found in [55]. There exist various kinds of other suitable representations, such as the quasi-separable [26, 19], diagonal-subdiagonal [46, 32], and Givens-vector representation [55, 16].

In this manuscript we will focus on the Givens-vector representation. There are several reasons to prefer this representation above the other ones. First, it uses $3n - 3$ parameters for representing a symmetric quasi-separable matrix, whereas the quasi-separable representation uses $4n - 2$ parameters. These extra $n + 1$ parameters used in the quasi-separable representation are therefore not uniquely determined. A good choice of these $n + 1$ (almost free) parameters is important for numerical stability reasons. Second, it can be considered as an extension of the straightforward $\mathbf{u}\mathbf{v}^T$ representation of a rank-one matrix. A third reason for using the Givens-vector representation is that it is based on unitary transformations, which are stable, and a single vector. Possible numerical instabilities occur only in the presence of small or large elements in the vector and are therefore easily recognized. In the numerical experiments we will see that the representation provides accurate results, even for some very ill-conditioned problems.

It is important to note that the Givens-vector representation is strongly related to the QR -factorization of the considered low rank part. We will come back to this at the end of this section. A final reason for choosing the Givens-vector representation is the ease of constructing the representation in this case. This will be shown further in the text.

Let us briefly recapitulate some of the results for this representation. We will first show how to reconstruct a low rank part based on Givens transformations and a vector. Secondly we will show how to retrieve the representation given a matrix of, e.g., quasi-separable form.

To represent the strictly lower triangular part of the quasi-separable matrix, we will use a representation consisting of $n - 2$ Givens transformations $G = [G_1, \dots, G_{n-2}]$ and a vector $\mathbf{v} = [v_1, \dots, v_{n-1}]$ of length $n - 1$. The diagonal of the quasi-separable matrix is stored separately, leading to a global storage of $2n - 1$ elements and $n - 2$ Givens transformations. To represent the matrices T and S , $4n - 2$ elements are

needed; to represent the quasi-separable matrix $L^{-1}TL^{-T}$, essentially only $3n - 3$ parameters are used. For actual implementations, however, we will not store the Givens transformations by a single parameter, but both the sine and cosine will be stored. So, in the practical implementation $4n - 5$ parameters are used.

The following figures denote how the strictly lower triangular part of the matrix can be reconstructed. We show here only the strictly lower triangular part of the quasi-separable matrix. Initially one starts on the first two rows of the strictly lower triangular part. The element v_1 is placed in the upper left position, then a Givens transformation is applied, and finally, to complete the first step, element v_2 is added in position $(2, 1)$. Only the first two columns and rows are shown here:

$$\begin{bmatrix} v_1 & 0 \\ 0 & 0 \end{bmatrix} \rightarrow G_1 \begin{bmatrix} v_1 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & v_2 \end{bmatrix} \rightarrow \begin{bmatrix} c_1 v_1 & 0 \\ s_1 v_1 & v_2 \end{bmatrix}.$$

The second step consists of applying the Givens transformation G_2 to the second and the third row; furthermore, v_3 is added in position $(3, 3)$. Here only the first three columns are shown and the second and third row. This leads to

$$\begin{bmatrix} s_1 v_1 & v_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \rightarrow G_2 \begin{bmatrix} s_1 v_1 & v_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & v_3 \end{bmatrix} \rightarrow \begin{bmatrix} c_2 s_1 v_1 & c_2 v_2 & 0 \\ s_2 s_1 v_1 & s_2 v_2 & v_3 \end{bmatrix}.$$

This process can be repeated by applying the Givens transformation G_3 to the third and the fourth row of the matrix, and afterwards adding the diagonal element v_4 . After applying all the Givens transformations and adding all the diagonal elements, the strictly lower triangular part of a quasi-separable matrix has been constructed. Because of the symmetry also the strictly upper triangular part is known. Finally one obtains a lower triangular matrix which equals the strictly lower triangular part of the matrix $L^{-1}TL^{-T}$:

$$(2.2) \quad \begin{bmatrix} c_1 v_1 & & & \\ c_2 s_1 v_1 & c_2 v_2 & & \\ c_3 s_2 s_1 v_1 & c_3 s_2 v_2 & c_3 v_3 & \\ \vdots & \vdots & & \ddots \end{bmatrix}.$$

We remark that the construction of the quasi-separable part of the matrix resembles the application of the Givens transformations to the Hessenberg matrix in the proof of Theorem 2.3.

Once the Givens-vector representation of, e.g., quasi-separable or semiseparable matrices is known, many techniques exist for solving systems of equations and computing eigenvalues [53]. How to retrieve the representation is, however, not always trivial. A general technique for retrieving a low rank representation, of undetermined rank, is based on cross-approximation. This is a relatively cheap technique and suitable for a wide variety of structured rank matrices [49, 48]. In many cases, however, the Givens-vector, the $\mathbf{u}\mathbf{v}^T$, or the quasi-separable representation is already known, e.g., as the result of inverting a band matrix [43] or applying an orthogonal similarity reduction to a semiseparable form [50]. We will not go into the details (see [57] for a complete exposition on several representations and how to compute them), but for semiseparable matrices as in Definition 2.2 there is a close relation between the QR -factorization and the Givens-vector representation. More precisely if one computes the QR -factorization of the semiseparable matrix by a sequence of bottom-up Givens transformations (exactly $n - 1$ are needed), then these Givens transformations correspond to the Givens transformations of the Givens-vector representation. The vector

can then be computed by some minor calculations. The fact that the Givens transformations of the QR -factorization coincide with the Givens transformations of the Givens-vector representation will come in handy when computing the representation of the quasi-separable matrix. For a quasi-separable matrix A also such a relation holds, but in this case one needs to compute the QR -factorization of $A(2:n, 1:n-1)$.

2.3. Computing the Givens-vector representation. Let us now show how to compute the representation of the quasi-separable matrix. Several things need to be computed. To obtain the Givens-vector representation of the matrix $L^{-1}TL^{-T}$, we need to compute intermediate representations of the matrices L^{-1} and TL^{-T} . These representations are used for computing the Givens-vector representation of $L^{-1}TL^{-T}$.

We will depict now all consecutive steps for computing the representation. Pseudocode for the implementation can be found in subsection 4.1.

- **Compute the Givens-vector representation for L^{-1} .** The matrix L is lower bidiagonal; hence its inverse is a lower semiseparable matrix. Therefore, this matrix can be represented by a Givens-vector representation.
 - *The Givens transformations.* ($8n - 8$ operations.²) There exists a sequence of Givens transformations $G_{n-1} \dots G_1$ applied on the right of the matrix L such that $LG_{n-1} \dots G_1 = B$ is an upper bidiagonal matrix. This corresponds to computing the RQ -factorization of the matrix³ $L = B(G_1^T \dots G_{n-1}^T)$. The Givens transformation G_{n-1} works on the last two columns, the transformation G_{n-2} on columns $n-2$ and $n-1$ and so forth. Inverting L leads to

$$L^{-1} = G_{n-1} \dots G_2 G_1 B^{-1}.$$

Since the right-hand side is in fact a QR -factorization of the matrix L^{-1} , the Givens transformations $G_1, \dots, G_{n-1}$ coincide with the Givens transformations in the Givens-vector representation of L^{-1} . In the remaining calculations often the tangent of the corresponding Givens transformations is desired. Let us denote s_i as the sine, c_i as the cosine, and $\tau_i = s_i/c_i$ as the tangent corresponding to Givens transformation G_i .

- *The vector in the Givens-vector representation.* As the diagonal elements of L^{-1} are the inverses of the diagonal elements of L , one has as representation the Givens transformations from above, with the corresponding vector:

$$(2.3) \quad \mathbf{v}_L = \left[\frac{1}{l_{11}c_1}, \frac{1}{l_{22}c_2}, \dots, \frac{1}{l_{n-1,n-1}c_{n-1}}, \frac{1}{l_{nn}} \right],$$

with $L = (l_{ij})$. We remark that these computations are well defined, as one can easily verify that all cosines in the different Givens transformations are different from zero. A cosine equal to zero translates to the fact that a diagonal element of L needed to be zero, which is not possible due to the positive definiteness of S .

However, in the next bullets we will notice that the division by the cosines c_i is not necessary. It creates extra operations and can cause numerical instabilities in case of small cosines. In fact, in most of the

²An operation consists of performing one of the following operations: $+$, $-$, $\times$, $/$.

³One can also consider $L(G_{n-1} \dots G_1) = B$ as the LQ -factorization of the matrix B .

computations only the diagonal of L^{-1} is involved. Hence, we do not compute the vector from the Givens-vector representation. We keep this in mind and compute these elements only when absolutely necessary.

- **Compute the representation of the matrix TL^{-T} .** The matrix TL^{-T} is a Hessenberg matrix having the upper triangular part of quasi-separable form. Hence we have to compute the Givens-vector representation for the strictly upper triangular part, the diagonal and subdiagonal elements of the Hessenberg-matrix $H = TL^{-T}$. Define $T = (t_{ij})$ and $H = (h_{ij})$. Some intermediate values for reducing the computations are stored in $\alpha, \beta, \gamma, \delta$. These extra vectors require an additional $4n$ memory but can reduce the computational cost from $37n - 49$ operations to $30n - 49$, an improvement of almost 20%.

- *Its diagonal elements.* ($5n - 5$ operations.) Straightforward computations lead to

$$\begin{aligned} h_{11} &= \beta_1, \quad \text{with} \quad \beta_1 = \frac{t_{11}}{l_{11}}, \\ h_{ii} &= \alpha_i c_i + \beta_i, \quad \text{with} \quad \alpha_i = \frac{t_{i,i-1} \tau_{i-1}}{l_{i-1,i-1}}, \quad \beta_i = \frac{t_{ii}}{l_{ii}}, \\ &\quad (\text{for all } i = 2, \dots, n-1) \\ h_{nn} &= \alpha_n + \beta_n, \quad \text{with} \quad \alpha_n = \frac{t_{n,n-1} \tau_{n-1}}{l_{n-1,n-1}}, \quad \beta_n = \frac{t_{nn}}{l_{nn}}. \end{aligned}$$

The above equations show that using the vector $\mathbf{v}_L$ in (2.3) from the Givens-vector representation of L^{-1} would only complicate matters.

- *Its subdiagonal elements.* (Not computed.) The subdiagonal elements can be computed as follows:

$$h_{i+1,i} = \frac{t_{i+1,i}}{l_{ii}} \quad (\text{for all } i = 1, \dots, n-1).$$

However, they are not essential for the construction of the quasi-separable matrix, due to symmetry of the final matrix $L^{-1}TL^{-T}$.

- *The Givens-vector representation of the strictly upper triangular part.* ($7n - 12$ operations.) The Givens transformations are exactly the same as the one used above, only one fewer is required: $G_2, G_3, \dots, G_{n-1}$. The vector of the representation of this upper triangular part can be obtained by computing the superdiagonal elements of the matrix TL^{-T} and dividing them by the corresponding cosines of the Givens transformations. As before, we will not, however, divide these elements by the cosines, but continue working with the superdiagonal elements and perform the division only when absolutely necessary.

The following formulas compute the superdiagonal elements:

$$\begin{aligned} h_{1,2} &= \beta_1 \tau_1 c_2 + \frac{t_{12}}{l_{22}}, \\ h_{i,i+1} &= \alpha_i \gamma_i + \beta_i \tau_i c_{i+1} + \frac{t_{i,i+1}}{l_{i+1,i+1}}, \quad \text{with } \gamma_i = s_i c_{i+1} \\ &\quad (\text{for } i = 2, \dots, n-2), \\ h_{n-1,n} &= \alpha_{n-1} s_{n-1} + \beta_{n-1} \tau_{n-1} + \frac{t_{n-1,n}}{l_{n,n}}. \end{aligned}$$

- **Compute the representation of the matrix $A = L^{-1}TL^{-T}$.** It was proven before that the matrix is a symmetric quasi-separable matrix. Hence we need the Givens-vector representation of the strictly lower triangular part as well as the diagonal elements.
 - *Compute the diagonal elements.* ($7n - 11$ operations.) To compute the diagonal elements, one can use the following loop (with t a temporary variable):

$$a_{11} = \frac{h_{11}}{l_{11}},$$

$$t = \delta_1, \quad \text{with} \quad \delta_1 = \frac{\tau_1 h_{12}}{l_{11}}.$$

Iterate now for $i = 2, \dots, n - 2$ the following equations:

$$a_{ii} = \frac{h_{ii}}{l_{i,i}} + t c_i,$$

$$t = (t \gamma_i + \delta_i) \tau_i, \quad \text{with} \quad \delta_i = \frac{h_{i,i+1}}{l_{i,i}}.$$

Finally we have the last two diagonal elements:

$$a_{n-1,n-1} = \frac{h_{n-1,n-1}}{l_{n-1,n-1}} + t c_{n-1},$$

$$t = (t s_{n-1} + \delta_{n-1}) \tau_{n-1}, \quad \text{with} \quad \delta_{n-1} = \frac{h_{n-1,n}}{l_{n-1,n-1}},$$

$$a_{nn} = \frac{h_{nn}}{l_{nn}} + t.$$

- *The Givens-vector representation of the strictly lower triangular part.* ($3n - 3$ operations.) Based on the relations discussed before we obtain

$$L^{-1}TL^{-T} = (BG_1^T G_2^T \dots G_{n-1}^T)^{-1} TL^{-T}$$

$$= G_{n-1} \dots G_2 (G_1 B^{-1} TL^{-T}).$$

Combining the factors $G_1 B^{-1} TL^{-T} = H$, we get an upper Hessenberg matrix. Hence it is clear due to construction that the Givens transformations G_2 up to G_{n-1} are the Givens transformations needed for the Givens-vector representation of the quasi-separable part in the matrix. The multiplication between the lower semiseparable matrix L^{-1} and the strictly upper triangular part of (TL^{-T}) , which is of quasi-separable form, can be done in $\mathcal{O}(n)$ operations. Writing down the lower semiseparable matrix and the strictly upper triangular part of the matrix TL^{-T} as in (2.2), one can easily deduce a simple loop which computes the sub-diagonal elements of the new quasi-separable matrix. Based on these subdiagonal elements and on the fact that the cosines of the Givens transformations are different from zero, one can easily obtain the representation of the strictly lower triangular part.

The sub- or superdiagonal elements can be computed as follows:

$$a_{12} = \delta_1,$$

$$a_{i+1,i} = a_{i,i-1} \tau_{i-1} \gamma_i + \delta_i \quad (\text{for } i = 2, \dots, n-1),$$

$$a_{n,n-1} = a_{n-1,n-2} \tau_{n-2} s_{n-1} + \delta_{n-1}.$$

Summarizing, the complexity of computing the Cholesky decomposition of the positive definite tridiagonal matrix takes $5n - 3$ operations. Computing the representation of the quasi-separable matrix takes $30n - 49$ operations. Hence the total cost for computing the representation of the quasi-separable matrix is $35n - 52$ operations.

Traditionally, one assumed that the approach of computing the eigenvalues via $L^{-1}TL^{-T}$ was too expensive because the reduction to tridiagonal form of a dense matrix already took $\mathcal{O}(n^3)$ operations. This reduction was essential before being able to compute the spectrum in $\mathcal{O}(n^2)$ operations, via, for example, a divide-and-conquer technique or a QR -algorithm. Using the method presented above, however, we see that it takes $\mathcal{O}(n)$ operations to obtain the representation of the quasi-separable matrix. For this quasi-separable matrix there exist techniques $\mathcal{O}(n^2)$ for computing the whole spectrum. In the next section we will briefly discuss some of these methods.

3. Computing the eigenvalues of a quasi-separable matrix. We do not go into the details of how to compute the eigenvalues of quasi-separable matrices. Pointers to manuscripts in which all the essential information can be found will be given. For solving the overall problem, in fact, only the eigenvalues of the quasi-separable matrix are necessary. The eigenvectors can be computed afterwards, based on the equation $T\mathbf{x} = \lambda S\mathbf{x}$, once the eigenvalues are known. This can be done in $\mathcal{O}(n^2)$ operations.

Alternatively, one can also compute the eigenvectors of the quasi-separable matrix and then transform them back to the original problem at a cost of $\mathcal{O}(n^2)$, exploiting the bidiagonal structure of L . The cost of this second approach is dependent on how efficiently the eigenvectors of the quasi-separable matrix are computed. This can lead to $\mathcal{O}(n^3)$ as well as $\mathcal{O}(n^2)$ methods; in the next subsection some complexities for different approaches are given.

3.1. Reduction to tridiagonal form. Due to the specific rank structure of the matrix A , we can reduce this matrix to tridiagonal form in $\mathcal{O}(n^2)$ operations instead of the traditional reduction, which needs $\mathcal{O}(n^3)$ operations. There exist several variants to reduce a quasi-separable matrix to tridiagonal form [30, 40]. In fact, the traditional algorithms are adapted to fully exploit the rank structure in the involved matrices. Also a parallel method to reduce a quasi-separable matrix to tridiagonal form was developed in [39], starting the reduction to tridiagonal form simultaneously at two sides of the matrix. Recently also more general reduction schemes, to reduce arbitrary structured rank matrices to tridiagonal (Hessenberg in the nonsymmetric case), were proposed [15, 25]. All presented algorithms are of order $\mathcal{O}(rn^2)$, where r is a factor related to the rank of the structured rank parts. Remember that in our case we consider a quasi-separable matrix of quasi separability rank $r = 1$.

The benefit of reducing the quasi-separable matrix to tridiagonal form is that one can use all available solvers for tridiagonal matrices. The reduction to tridiagonal form costs, however, $\mathcal{O}(n^2)$ operations, which is of the same order as computing the full spectrum. This reduction cost is not needed, however, if one applies methods which are directly applicable to the quasi-separable matrix.

On the other hand, plenty of robust and efficient methods for tridiagonal matrices exist (LAPACK/Matlab implementations are available); e.g., the QR -method for tridiagonal matrices is discussed in many textbooks [33, 44, 17, 58] ($\mathcal{O}(n^2)$ for computing the eigenvalues, $\mathcal{O}(n^3)$ when computing the eigenvectors by accumulating the unitary transformations performed as is done in the current LAPACK implementation), divide-and-conquer methods [5, 6, 35, 11, 8] ($\mathcal{O}(n^2)$ if only eigenvalues are desired, in the worst case $\mathcal{O}(n^3)$ for the eigenvectors; in practice, however, the ex-

ponent is less than 3), bisection and inverse iteration [42, 13] ($\mathcal{O}(n^2)$ for the whole spectrum; in case of well separated eigenvalues it takes $\mathcal{O}(n^2)$ operations for all eigenvectors; otherwise, in case of clusters it might result in $\mathcal{O}(n^3)$ operations), the MRRR algorithm, which is an adapted version of inverse iteration, not using Gram–Schmidt orthogonalization [23, 20, 21, 22, 45] (this method requires $\mathcal{O}(n^2)$ for computing both eigenvalues and eigenvectors), etc. A recent overview of these methods, comparing them in LAPACK, was given in [18]. When applying the QR -algorithm to the resulting tridiagonal matrix, one applies in a certain sense a tuned Cholesky- QR method, exploiting the quasi-separable structure for reducing the matrix to tridiagonal form. This tuned Cholesky- QR method also involves only $\mathcal{O}(n^2)$ operations for computing the whole spectrum as well as the eigenvectors, when using, e.g., MRRR.

3.2. Applying the QR -algorithm directly on the quasi-separable matrix. The last few years people have intensively studied QR -algorithms for structured rank matrices. Let us present some of these results. There is an implicit QR -algorithm for semiseparable matrices [54] and an implicit one for semiseparable plus diagonal matrices [51]. Explicit QR -algorithms for higher order quasi-separable matrices can be found in [28, 14]. The algorithm for semiseparable plus diagonal matrices is after minor modifications also suitable for quasi-separable matrices. Recently more general types of QR -algorithms exist for low rank perturbations of unitary matrices and so forth [1, 2, 3]. Computing the eigenvalues using the QR -method for quasi-separable matrices results in an order $\mathcal{O}(n^2)$ method, whereas computing the eigenvectors by accumulating the transformations performed imposes a cubical complexity $\mathcal{O}(n^3)$.

One might be in favor of the QR -method because it is widely spread and these methods are directly applicable on the quasi-separable matrices. Unfortunately QR -methods for quasi-separable matrices have a higher computational complexity (50%) w.r.t. the QR -method than do tridiagonal matrices. Nevertheless they provide accurate results. Recently, however, also a faster algorithm for computing the spectrum of structured rank matrices was proposed [56]. The method is based on a QR -iteration driven by a rational function. The resulting method is much faster than the traditional QR -methods for rank structured matrices.

3.3. Other methods. As mentioned in the introduction also different techniques for computing the eigenvalues of quasi-separable matrices exist. For example, the bisection method and a method based on Sturm sequences can be found in [27]. With the bisection method one can compute a single eigenvalue in an interval, whereas the Sturm sequences method is an adaptation of the bisection method to compute the k th largest eigenvalue. Computing a single eigenvalue involves $\mathcal{O}(n)$ operations, whereas computing the full spectrum requires $\mathcal{O}(n^2)$ flops.

Another technique is based on halving the problem size at every step of the algorithm. These so called divide-and-conquer methods are based on solving the secular equation [41, 10].

Both methods mentioned above need $\mathcal{O}(n^2)$ operations for computing the whole spectrum. Up till now the divide-and-conquer method for quasi-separable and semiseparable plus diagonal matrices is the fastest and most accurate available method for computing the spectrum of quasi-separable matrices. Hence we choose to use this method for computing the eigenvalues of the corresponding quasi-separable matrix in the upcoming numerical experiments. Computing all eigenvectors increases the computational complexity and can lead to $\mathcal{O}(n^3)$. Using specialized techniques based on the FMM method [9, 34] as discussed in [10] can reduce the complexity of computing the eigenvectors to $\mathcal{O}(n^2 \log(n))$.

4. Numerical experiments. In this section we will present results concerning the timings and accuracy of the presented approach. We chose to use the divide-and-conquer method for computing the eigenvalues and eigenvectors of the considered quasi-separable matrices. To our knowledge this algorithm is among the fastest available methods and provides accurate results when computing the eigenvalues of quasi-separable matrices. The software was implemented in MATLAB and executed on a Linux platform. The divide-and-conquer approach used is the one from [41], where we needed to adapt the software, as the implementation as presented in [41] (the straightforward solver) was based on the generator representation whereas the resulting quasi-separable matrix in our approach is represented using the Givens-vector representation.

In the following experiments the relative forward error and relative backward error are computed. First the pseudocode for generating the quasi-separable matrix is given.

Recall that the presented method and Laguerre's iteration take $\mathcal{O}(n^2)$ operations for computing the eigenvalues. The standard Cholesky- QR and QZ -approach take $\mathcal{O}(n^3)$ operations for computing the eigenvalues. We use the MATLAB built-in functions for performing the QZ and Cholesky- QR approach, which are of the order $\mathcal{O}(n^3)$. We remark, however, that one can reduce the complexity count of the Cholesky- QR method from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$ when using an adapted reduction of the quasi-separable matrix to tridiagonal form. However, in this case also one has to exploit the quasi-separable structure. Even though one can reduce the Cholesky- QR complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$, the constant will be larger than the one in the quasi-separable approach combined with the divide-and-conquer method. Once the eigenvalues are known various methods exist ($\mathcal{O}(n^2)$ and $\mathcal{O}(n^3)$) for computing the eigenvectors; hence we will not go into the details related to eigenvectors.

We can, therefore, already conclude that the quasi-separable approach is as fast as Laguerre's iteration and faster than the Cholesky- QR and the QZ by a factor n (from $\mathcal{O}(n^2)$ to $\mathcal{O}(n^3)$). Keep in mind that quasi-separable techniques can be used for tuning the Cholesky- QR method in order to obtain an $\mathcal{O}(n^2)$ approach.

We will therefore not focus on the complexity of the methods, but on a comparison of the accuracy of the four methods.

4.1. The pseudocode. We will present the pseudocode for computing the representation of the quasi-separable matrix $L^{-1}TL^{-T}$. The case where $n = 1$ and $n = 2$ is trivial and not covered.

Explanation of some variables used in the pseudocode:

- **dX** stands for the diagonal of the matrix X ;
- **sdX** stands for super(sub)diagonal of the matrix X ;
- **uh** stands for the upper Hessenberg matrix TL^{-T} ;
- Combinations lead to, for example, **sbduh**, which means the subdiagonal of the upper Hessenberg matrix TL^{-T} .
- **G** will contain the Givens transformations of the Givens-vector representation. This is a $2 \times (n - 1)$ matrix. Each column corresponds to a Givens transformation. The first element in each column contains the sine, the second element the cosine.
- **v** will contain the vector from the Givens-vector representation.

1. % Compute the Cholesky decomposition $S=LL^T$
Store the diagonal in **dL**, the subdiagonal in **sdL**
2. % Compute the RQ-factorization of the matrix **L**
Initialize: **G(1,i-1)=1**;

```

for i=n,n-1,n-2,...,2
    [cosine,sine,tau(i-1)]=givens(Giv(1,i-1)*dL(i),sdL(i-1));
    Store the cosine and the sine in column i-1 of G;
end for;
3. % Construct the representation of the upper Hessenberg matrix  $T L^{-T}$ 
% Compute the diagonal elements of  $T L^{-T}$ 
Initialize: beta(1)=dT(1)/dL(1); duh(1)=beta(1);
for i=2,3,...,n-1
    alpha(i)=sdT(i-1)/dL(i-1)*tau(i-1)
    beta(i)=dT(i)/dL(i);
    duh(i)=alpha(i)*G(1,i)+beta(i);
end for
duh(n)=sdT(n-1)/dL(n-1)*tau(n-1)+dT(n)/dL(n);
4. % Compute the superdiagonal elements of  $T L^{-T}$ 
Initialize: sduh(1)=beta(1)*tau(1)*G(1,2)+sdT(1)/dL(2);
for i=2,3,...,n-2
    gamma(i)=G(2,i)*G(1,i+1);
    sduh(i)=alpha(i)*gamma(i)+beta(i)*tau(i)*G(1,i+1)+sdT(i)/dL(i+1);
end for
sduh(n-1)=alpha(n-1)*G(2,n-1)+beta(n-1)*tau(n-1)+sdT(n-1)/dL(n);
5. % Compute the representation for  $L^{-1} T L^{-T}$  and the diagonal
Initialize: d(1)=duh(1)/dL(1); tmp=tau(1)*delta(1);
for i=2,3,...,n-2
    d(i)=duh(i)/dL(i)+tmp*G(1,i);
    tmp=(tmp*gamma(i)+delta(i))*tau(i);
end for
delta(n-1)=sduh(n-1)/dL(n-1);
d(n-1)=duh(n-1)/dL(n-1)+tmp*G(1,n-1);
tmp=(tmp*G(2,n-1)+delta(n-1))*tau(n-1);
d(n)=duh(n)/dL(n)+tmp;
6. % Compute the subdiagonal
Initialize: sd(1)=delta(1);
for i=2,3,...,n-2
    sd(i)=sd(i-1)*tau(i-1)*gamma(i)+delta(i);
end for
sd(n-1)=sd(n-2)*tau(n-2)*G(2,n-1)+delta(i);
7. % Assign the Givens-vector representation
G=G(:,2:n-1); v(1:n-2)=sd(1:n-2)./G(1,:);

```

NOTE 4.1. We remark that in several cases a scaled Givens-vector representation is enough. This means that as input for several structured rank based algorithms one can pass the variable sd instead of v . If this holds (e.g., in the case of the divide-and-conquer method), we can omit the last two lines of the routine, in which G and v are defined. Essentially only G as it is and the vector sd (scaled version of v) are needed to be able to compute the spectrum via the divide-and-conquer method. Not performing the division saves us $2n - 2$ operations in the overall implementation for computing the spectrum.

4.2. Relative forward error. In the first experiment we compared the computed eigenvalues with known eigenvalues of the problem. We solved the definite symmetric generalized eigenvalue problem with two tridiagonal Toeplitz matrices. The matrix T is constructed with a random element defining the diagonal and a random element defining the subdiagonal (generated using `rand` from MATLAB). The second matrix S has a random subdiagonal element, whereas the diagonal element is chosen $s_{ii} = 2 \max(s_{i,i-1}, s_{i,i+1}) + 1$. In this way we know that the matrix S is positive definite and moreover is well conditioned.

As both Toeplitz matrices commute, we can explicitly compute the spectrum of the generalized eigenvalue problem as it equals the ratios of the eigenvalues of T and

S . To compute the eigenvalues of both Toeplitz matrices we used the explicit formulas derived in [7]. For a tridiagonal Toeplitz matrix having subdiagonal t_{-1} , diagonal t_0 , and superdiagonal t_1 the eigenvalues are given by

$$t_0 + \sqrt{t_1 t_{-1}} \cos \frac{\pi i}{n+1} \quad (\text{for } i = 1, \dots, n).$$

Based on these “correct” eigenvalues of the generalized eigenvalue problem, we performed experiments for sizes ranging from 100 to 1500, and for each experiment we performed fifteen random tests (with the constraints on S as mentioned above). Moreover, the eigenvalues of the Toeplitz matrix were computed via variable precision arithmetic in MATLAB to ensure they are correct up to machine precision (16 digits). The relative forward errors

$$\max_i \frac{|\lambda_i - \tilde{\lambda}_i|}{|\lambda_i|},$$

where λ_i denote the “correct” eigenvalues and $\tilde{\lambda}_i$ the computed eigenvalues, are plotted in Figure 4.1.

Figure 4.1 shows that the forward error can be pretty large, up to 10^{-11} . When comparing, however, against the condition number of the symmetric quasi-separable matrix, we see that the loss of digits is due to the condition number, which sometimes reaches approximately 10^5 .

Comparing the quasi-separable approach with both other approaches, we see that overall it has the best forward error. In the upcoming experiment we see that the bad results are due to the conditioning of the problem, since all backward errors will be relatively good. The experiment was run for three approaches: the quasi-separable approach, the QZ -method, and Laguerre’s iteration from [38].

4.3. Relative backward error. In the following set of experiments we compute a relative error involving the eigenvalues and eigenvectors. We first solve the eigenproblem involving the quasi-separable matrix A :

$$A\mathbf{y} = \lambda\mathbf{y},$$

where $A = L^{-1}TL^{-T}$ and $\mathbf{y} = L^T\mathbf{x}$.

We hence compute the eigenvalues λ_i corresponding to the eigenvectors $\mathbf{y}_i$. To obtain the eigenvectors $\mathbf{x}_i$ of the generalized eigenvalue problem, we need to compute the following:

$$\mathbf{x}_i = L^{-T}\mathbf{y}_i.$$

As we know from the theoretical results, the matrix L^{-T} is an upper triangular semiseparable matrix. Moreover, the representation in terms of Givens transformations and a vector is known, since it was computed in the algorithm explained in subsection 2.3. The multiplication between the matrix $L^{-T}\mathbf{y}$ can easily be performed in $\mathcal{O}(n)$ operations (see, e.g., [52]).

For the following set of experiments we took matrix sizes ranging from 100 to 2000, and 15 experiments for each size were considered. The tridiagonal matrix T has random diagonal and subdiagonal elements. The matrix S has random subdiagonal elements, and the diagonal elements were taken $s_{ii} = 2 \max(s_{i,i-1}, s_{i,i+1}) + 1$, in order to make the matrix positive definite and well conditioned.

FIG. 4.1. *Relative forward errors on the eigenvalues.*

We considered the following relative backward error [37, 12]:

$$\max_i \left(\frac{\|T\mathbf{x}_i - \lambda_i S\mathbf{x}_i\|_2}{(\|T\|_2 + |\lambda_i| \|S\|_2) \|\mathbf{x}_i\|_2} \right),$$

where the eigenvectors $\mathbf{x}_i = L^{-T}\mathbf{y}_i$, eigenvalues λ_i , and eigenvectors $\mathbf{y}_i$ were computed using the presented method. The norm of the matrices T and S was estimated using the MATLAB built-in function `normest`.

FIG. 4.2. *Relative backward error.* FIG. 4.3. *Comparison of three approaches.*

Figure 4.2 clearly illustrates that the quasi-separable method returns a relatively good backward error for all experiments.

4.4. Comparison of different approaches. In this experiment we will perform more specific tests for comparing the accuracy of the proposed method with some methods available in MATLAB. The matrices were generated similarly as in the previous experiment. For every matrix the eigenvalues and eigenvectors were computed via the quasi-separable method, the Cholesky- QR approach, and the QZ -method. A comparison of the relative backward error is depicted in Figure 4.3. It can be seen that the quasi-separable method has an error comparable to the QZ -method but performs worse than the Cholesky- QR method. Keep in mind that for this experiment all three methods are $\mathcal{O}(n^3)$ methods, since both the eigenvalues and eigenvectors are required. When only the eigenvalues are desired, however, the complexity of the

FIG. 4.4. *Comparison with another $\mathcal{O}(n^2)$ method.*

divide-and-conquer reduces to $\mathcal{O}(n^2)$, whereas both other methods remain cubical in the number of operations.

The experiments show that for this type of well-conditioned problems the Cholesky- QR method is the most accurate one. The quasi-separable method is equally accurate as the QZ -method.

4.5. Comparison with another $\mathcal{O}(n^2)$ method. In this section we will compare the method from [38] with the quasi-separable method presented in this paper. The implementation from [38] was written in Fortran, whereas the code from this article was written in MATLAB. In order to compare the accuracy of both methods we choose to embed the Fortran code in MATLAB via a MEX-file. Since the routine from [38] only computes the eigenvalues, it is impossible to use the backward error measure norm from above. The experiments from subsection 4.2 were redone, and the relative forward error for both approaches is plotted in Figure 4.4.

We see that both methods are equally accurate for the chosen experiments. Approach 1 corresponds to the quasi-separable method combined with divide-and-conquer, and Approach 2 is based on an evaluation of the determinant via Laguerre's iteration as presented in [38].

4.6. The matrix S is ill conditioned. Let us reconsider in this section Experiment 2 from [38]. We compute the eigenvalues of the generalized definite eigenproblem in which T is a tridiagonal Toeplitz matrix having 4 on the diagonal and 1 on the subdiagonal. The matrix S is almost Toeplitz, having the diagonal elements equal to $2 \cdot 10^{-10}$, the subdiagonal elements equal to 10^{-10} , and $s_{11} = s_{nn} = 1$. The condition number of the matrix S is approximately 10^{12} . First we run tests for several sizes of the matrices, ranging from 100 to 2000 via steps of size 50. The relative backward error was computed and plotted for the divide-and-conquer based on quasi-separable matrices, the QZ -method, and the Cholesky- QR method. Results are presented in Figure 4.5.

The QZ -method performs the best in this case. The Cholesky- QR method seems to have a lot of difficulties in finding approximations of the eigenvalues and seems to provide less accurate results. The quasi-separable method performs significantly better than the Cholesky- QR method.

We reconsidered this type of experiment, but now we compared the quasi-separable

FIG. 4.5. Comparison for ill-conditioned S (subdiagonal size 10^{-10}). FIG. 4.6. Comparison for ill-conditioned S .

approach with the one based on Laguerre's iteration from [38]. Based on the previous experiment, we consider the eigenvalues computed by the QZ -method to be the correct ones λ_i . In Figure 4.6 you see the relative forward error.

When comparing the forward error, we see that the quasi-separable method performs the best. Moreover, when the dimension of the problem increases, Laguerre's iteration method seems to get into problems and no longer provides accurate results.

5. Conclusions. In this manuscript we showed that one can solve the definite generalized tridiagonal symmetric eigenvalue problem by transforming it into a standard symmetric eigenvalue problem and exploiting the quasi-separable structure of the coefficient matrix. The eigenvalues and eigenvectors can be computed using various methods. In this manuscript we used the divide-and-conquer method for computing the eigenvalues in $\mathcal{O}(n^2)$ operations.

It was shown that the quasi-separable method involves $\mathcal{O}(n^2)$ operations, the traditional Cholesky- QR method as well as the QZ -method $\mathcal{O}(n^3)$ operations, and the tuned Cholesky- QR method exploiting the quasi-separable structure uses $\mathcal{O}(n^2)$ operations. Among all these methods the quasi-separable approach is the fastest one and, moreover, more accurate than the Cholesky- QR approach in case of an

ill-conditioned matrix S .

A final comparison between the quasi-separable approach and another $\mathcal{O}(n^2)$ method was given, showing that both methods are of a comparable accuracy in the well-conditioned case, but the quasi-separable method performs better in the ill-conditioned case.

Acknowledgments. The authors wish to thank Françoise Tisseur for fruitful discussions at the ILAS meeting in The Netherlands, 2006, and for the useful comments which improved the overall presentation of the manuscript. We also thank Prof. Zhonggang Zeng for providing us the Fortran code of [38].

REFERENCES

- [1] D. A. BINI, F. DADDI, AND L. GEMIGNANI, *On the shifted QR iteration applied to companion matrices*, Electron. Trans. Numer. Anal., 18 (2004), pp. 137–152.
- [2] D. A. BINI, Y. EIDELMAN, L. GEMIGNANI, AND I. GOHBERG, *Fast QR eigenvalue algorithms for Hessenberg matrices which are rank-one perturbations of unitary matrices*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 566–585.
- [3] D. A. BINI, L. GEMIGNANI, AND V. Y. PAN, *Fast and stable QR eigenvalue algorithms for generalized companion matrices and secular equations*, Numer. Math., 100 (2005), pp. 373–408.
- [4] R. F. BOISVERT, *Families of high order accurate discretizations of some elliptic problems*, SIAM J. Sci. Stat. Comput., 2 (1981), pp. 268–284.
- [5] C. F. BORGES AND W. B. GRAGG, *Divide and conquer for generalized real symmetric definite tridiagonal eigenproblems*, in Proceedings of the '92 Shanghai International Numerical Algebra and Its Applications Conference, E.-X. Jiang, ed., China Science and Technology Press, Beijing, China, 1992, pp. 70–76.
- [6] C. F. BORGES AND W. B. GRAGG, *A parallel divide and conquer algorithm for the generalized real symmetric definite tridiagonal eigenproblem*, in Numerical Linear Algebra and Scientific Computing, L. Reichel, A. Ruttan, and R. S. Varga, eds., De Gruyter, Berlin, Germany, 1993, pp. 11–29.
- [7] A. BÖTTCHER AND S. M. GRUDSKY, *Spectral Properties of Banded Toeplitz Matrices*, SIAM, Philadelphia, PA, 2005.
- [8] J. R. BUNCH, C. P. NIELSEN, AND D. C. SORESENSEN, *Rank-one modification of the symmetric eigenvalue problem*, Numer. Math., 31 (1978), pp. 31–48.
- [9] J. CARRIER, L. GREENGARD, AND V. ROKHLIN, *A fast adaptive multipole algorithm for particle simulations*, SIAM J. Sci. Stat. Comput., 9 (1988), pp. 669–686.
- [10] S. CHANDRASEKARAN AND M. GU, *A divide-and-conquer algorithm for the eigendecomposition of symmetric block-diagonal plus semiseparable matrices*, Numer. Math., 96 (2004), pp. 723–731.
- [11] J. J. M. CUPPEN, *A divide and conquer method for the symmetric tridiagonal eigenproblem*, Numer. Math., 36 (1981), pp. 177–195.
- [12] P. I. DAVIES, N. J. HIGHAM, AND F. TISSEUR, *Analysis of the Cholesky method with iterative refinement for solving the symmetric definite generalized eigenproblem*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 472–493.
- [13] J.-M. DELOSME AND I. C. F. IPSEN, *From Bareiss' algorithm to the stable computation of partial correlations*, J. Comput. Appl. Math., 27 (1989), pp. 53–91.
- [14] S. DELVAUX AND M. VAN BAREL, *The Explicit QR-algorithm for Rank Structured Matrices*, Technical Report TW459, Department of Computer Science, Katholieke Universiteit Leuven, Leuven, Belgium, 2006.
- [15] S. DELVAUX AND M. VAN BAREL, *A Hessenberg reduction algorithm for rank structured matrices*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 895–926.
- [16] S. DELVAUX AND M. VAN BAREL, *A Givens-weight representation for rank structured matrices*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 1147–1170.
- [17] J. W. DEMMEL, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, PA, 1997.
- [18] J. W. DEMMEL, O. A. MARQUES, B. N. PARLETT, AND C. VÖMEL, *Performance and accuracy of LAPACK's symmetric tridiagonal eigensolvers*, SIAM J. Sci. Comput., 30 (2008), pp. 1508–1526.
- [19] P. DEWILDE AND A.-J. VAN DER VEEN, *Time-Varying Systems and Computations*, Kluwer Academic Publishers, Boston, MA, 1998.

- [20] I. S. DHILLON, *A New $O(n^2)$ Algorithm for the Symmetric Tridiagonal Eigenvalue/Eigenvector Problem*, Ph.D. thesis, Department of Computer Science, University of California, Berkeley, 1989.
- [21] I. S. DHILLON AND B. N. PARLETT, *Multiple representations to compute orthogonal eigenvectors of symmetric tridiagonal matrices*, Linear Algebra Appl., 387 (2004), pp. 1–28.
- [22] I. S. DHILLON AND B. N. PARLETT, *Orthogonal eigenvectors and relative gaps*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 858–899.
- [23] I. S. DHILLON, B. N. PARLETT, AND C. VÖMEL, *The design and implementation of the MRRR algorithm*, ACM Trans. Math. Software, 32 (2006), pp. 533–560.
- [24] Y. EIDELMAN, *Fast recursive algorithm for a class of structured matrices*, Appl. Math. Lett., 13 (2000), pp. 57–62.
- [25] Y. EIDELMAN, L. GEMIGNANI, AND I. C. GOHBERG, *On the fast reduction of a quasi-separable matrix to Hessenberg and tridiagonal forms*, Linear Algebra Appl., 420 (2007), pp. 86–101.
- [26] Y. EIDELMAN AND I. C. GOHBERG, *On a new class of structured matrices*, Integral Equations Operator Theory, 34 (1999), pp. 293–324.
- [27] Y. EIDELMAN, I. C. GOHBERG, AND V. OLSHEVSKY, *Eigenstructure of order-one-quasi-separable matrices. Three-term and two-term recurrence relations*, Linear Algebra Appl., 405 (2005), pp. 1–40.
- [28] Y. EIDELMAN, I. C. GOHBERG, AND V. OLSHEVSKY, *The QR iteration method for Hermitian quasi-separable matrices of an arbitrary order*, Linear Algebra Appl., 404 (2005), pp. 305–324.
- [29] L. ELSNER, A. FASSE, AND E. LANGMANN, *A divide-and-conquer method for the tridiagonal generalized eigenvalue problem*, J. Comput. Appl. Math., 86 (1997), pp. 141–148.
- [30] D. FASINO, N. MASTRONARDI, AND M. VAN BAREL, *Fast and stable algorithms for reducing diagonal plus semiseparable matrices to tridiagonal and bidiagonal form*, in Fast Algorithms for Structured Matrices: Theory and Applications, Contemp. Math. 323, American Mathematical Society, Providence, RI, 2003, pp. 105–118.
- [31] M. FIEDLER AND T. L. MARKHAM, *Completing a matrix when certain entries of its inverse are specified*, Linear Algebra Appl., 74 (1986), pp. 225–237.
- [32] M. FIEDLER AND Z. VAVŘÍN, *Generalized Hessenberg matrices*, Linear Algebra Appl., 380 (2004), pp. 95–105.
- [33] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
- [34] L. GREENGARD AND V. ROKHLIN, *A fast algorithm for particle simulations*, J. Comput. Phys., 73 (1987), pp. 325–348.
- [35] M. GU AND S. C. EISENSTAT, *A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 172–191.
- [36] L. KAUFMAN, *An algorithm for the banded symmetric generalized matrix eigenvalue problem*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 372–389.
- [37] S.-L. LEI AND X.-Q. JIN, *BCCB preconditioners for systems of BVM-based numerical integrators*, Numer. Linear Algebra Appl., 11 (2004), pp. 25–40.
- [38] K. LI, T.-Y. LI, AND Z. ZENG, *An algorithm for the generalized symmetric tridiagonal eigenvalue problem*, Numer. Algorithms, 8 (1994), pp. 269–291.
- [39] 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.
- [40] N. MASTRONARDI, S. CHANDRASEKARAN, AND S. VAN HUFFEL, *Fast and stable reduction of diagonal plus semi-separable matrices to tridiagonal and bidiagonal form*, BIT, 41 (2003), pp. 149–157.
- [41] N. MASTRONARDI, M. VAN BAREL, AND E. VAN CAMP, *Divide-and-conquer algorithms for computing the eigendecomposition of symmetric diagonal-plus-semiseparable matrices*, Numer. Algorithms, 39 (2005), pp. 379–398.
- [42] N. MASTRONARDI, M. VAN BAREL, E. VAN CAMP, AND R. VANDEBRIL, *On computing the eigenvectors of a class of structured matrices*, J. Comput. Appl. Math., 189 (2006), pp. 580–591.
- [43] G. MEURANT, *A review of the inverse of symmetric tridiagonal and block tridiagonal matrices*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 707–728.
- [44] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, Classics in Appl. Math. 20, SIAM, Philadelphia, PA, 1998.
- [45] B. N. PARLETT AND I. S. DHILLON, *Relatively robust representations of symmetric tridiagonals*, Linear Algebra Appl., 309 (2000), pp. 121–151.
- [46] S. PRÖSSDORF AND B. SILBERMANN, *Numerical Analysis for Integral and Related Operator Equations*, Akademie-Verlag, Berlin, Germany, 1991.

- [47] Y. M. RAM AND G. M. L. GLADWELL, *Constructing a finite element model of a vibratory rod from eigendata*, J. Sound Vibration, 169 (1994), pp. 229–237.
- [48] E. E. TYRTYSHNIKOV, *Incomplete cross approximation in the mosaic-skeleton method*, Computing, 64 (2000), pp. 367–380.
- [49] E. E. TYRTYSHNIKOV, *Mosaic ranks for weakly semiseparable matrices*, in Large-Scale Scientific Computations of Engineering and Environmental Problems II, M. Griebel, S. Margenov, and P. Y. Yalamov, eds., Notes on Numer. Fluid Mech. 73, Vieweg, Braunschweig, Germany, 2000, pp. 36–41.
- [50] M. VAN BAREL, R. VANDEBRIL, AND N. MASTRONARDI, *An orthogonal similarity reduction of a matrix into semiseparable form*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 176–197.
- [51] E. VAN CAMP, M. VAN BAREL, R. VANDEBRIL, AND N. MASTRONARDI, *An Implicit QR-Algorithm for Symmetric Diagonal-Plus-Semiseparable Matrices*, Technical Report TW419, Department of Computer Science, Katholieke Universiteit Leuven, Leuven, Belgium, 2005.
- [52] R. VANDEBRIL, *Semiseparable Matrices and the Symmetric Eigenvalue Problem*, Ph.D. thesis, Department of Computer Science, Katholieke Universiteit Leuven, Leuven, Belgium, 2004.
- [53] R. VANDEBRIL, M. VAN BAREL, G. H. GOLUB, AND N. MASTRONARDI, *A bibliography on semiseparable matrices*, Calcolo, 42 (2005), pp. 249–270.
- [54] R. VANDEBRIL, M. VAN BAREL, AND N. MASTRONARDI, *An implicit QR-algorithm for symmetric semiseparable matrices*, Numer. Linear Algebra Appl., 12 (2005), pp. 625–658.
- [55] R. VANDEBRIL, M. VAN BAREL, AND N. MASTRONARDI, *A note on the representation and definition of semiseparable matrices*, Numer. Linear Algebra Appl., 12 (2005), pp. 839–858.
- [56] R. VANDEBRIL, M. VAN BAREL, AND N. MASTRONARDI, *A New Iteration for Computing the Eigenvalues of Semiseparable (Plus Diagonal) Matrices*, Technical Report TW507, Department of Computer Science, Katholieke Universiteit Leuven, Leuven, Belgium, 2007.
- [57] R. VANDEBRIL, M. VAN BAREL, AND N. MASTRONARDI, *Matrix Computations and Semiseparable Matrices, Volume I: Linear Systems*, Johns Hopkins University Press, Baltimore, MD, 2008.
- [58] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Oxford University Press, New York, 1965.