Open access • Journal Article • DOI:10.1007/S00211-004-0550-9

A QR –method for computing the singular values via semiseparable matrices — Source link

Raf Vandebril, Marc Van Barel, Nicola Mastronardi

Institutions: Katholieke Universiteit Leuven

Published on: 01 Nov 2004 - Numerische Mathematik (Springer-Verlag)

Topics: Bidiagonal matrix, Symmetric matrix, Singular value decomposition, Band matrix and Matrix decomposition

An implicit QR algorithm for symmetric semiseparable matrices

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, sez. Bari, Italy
N.Mastronardi@ba.iac.cnr.it

Abstract

The QR algorithm is one of the classical methods to compute the eigendecomposition of a matrix. If it is applied on a dense n x n matrix, this algorithm requires $O(n^3)$ operations per iteration step. To reduce this complexity for a sytmmetric matrix to $O(n)$, the original matrix is first reduced to tridiagonal form using orthogonal similarity transformations.

Article information

- Vandebril, Raf; Van Barel, Marc; Mastronardi, Nicola. *An implicit QR algorithm for symmetric semiseparable matrices*, Numerical Linear Algebra With Applications, volume 12, issue 7, pages 625-658, 2005.
- The content of this article is pre-reviewing. The original published, reviewed, and corrected version can differ significantly from this one.
- Journal's homepage: http://onlinelibrary.wiley.com/journal/10.1002/(ISSN)1099-1506
- Published version: http://dx.doi.org/10.1002/nla.425
- KU Leuven's repository url: https://lirias.kuleuven.be/handle/123456789/124691

An implicit QR algorithm for semiseparable matrices to compute the eigendecomposition of symmetric matrices.

Raf Vandebril
Marc Van Barel
Nicola Mastronardi

Report TW 367, August 2003

Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

An implicit QR algorithm for semiseparable matrices to compute the eigendecomposition of symmetric matrices. [*]

Raf Vandebril [†], Marc Van Barel [‡], Nicola Mastronardi [§]

22nd August 2003

Abstract

The QR algorithm is one of the classical methods to compute the eigendecomposition of a matrix. If it is applied on a dense $n \times n$ matrix, this algorithm requires $O(n^3)$ operations per iteration step. To reduce this complexity for a symmetric matrix to $O(n)$, the original matrix is first reduced to tridiagonal form using orthogonal similarity transformations.

In the report [Van Barel, Vandebril, Mastronardi 2003] a reduction from a symmetric matrix into a similar semiseparable one is described. In this paper a QR algorithm to compute the eigenvalues of semiseparable matrices is designed where each iteration step requires $O(n)$ operations. Hence, combined with the reduction to semiseparable form, the eigenvalues of symmetric matrices can be computed via intermediate semiseparable matrices, instead of tridiagonal ones.

The eigenvectors of the intermediate semiseparable matrix will be computed by applying inverse iteration to this matrix. This will be achieved by using an $O(n)$ system solver, for semiseparable matrices.

A combination of the previous steps leads to an algorithm for computing the eigenvalue decompositions of semiseparable matrices. Combined with the reduction of a symmetric matrix towards semiseparable form, this algorithm can also be used to calculate the eigenvalue decomposition of symmetric matrices. The presented algorithm has the same order of complexity as the tridiagonal approach, but has

[*] The research of the first and second author was supported by the Research Council K.U.Leuven, project OT/00/16 (SLAP: Structured Linear Algebra Package), by the Fund for Scientific Research–Flanders (Belgium), projects G.0078.01 (SMA: Structured Matrices and their Applications), G.0176.02 (ANCILA: Asymptotic aNalysis of the Convergence behavior of Iterative methods in numerical Linear Algebra), and G.0184.02 (CORFU: Constructive study of Orthogonal Functions), 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 work of the third author was partially supported by MIUR, grant number 2002014121. The scientific responsibility rests with the authors.

[†] Email: raf.vandebril@cs.kuleuven.ac.be

[‡] Email: marc.vanbarel@cs.kuleuven.ac.be

[§] Email: nicola.mastronardi@cs.kuleuven.ac.be

larger lower order terms. Numerical experiments illustrate the complexity and the numerical accuracy of the proposed method.

Keywords: symmetric matrix, semiseparable matrix, similarity reduction to semiseparable form, implicit *QR* algorithm for semiseparable matrices, eigenvalues, eigenvectors

1 Introduction

Semiseparable matrices appear in various research fields. In the papers [18, 20, 21, 27], the matrices arising from the discretization of the integral equations are semiseparable matrices. They arise also in statistics as covariance, variance matrices [17]. Physical applications such as electromagnetic scattering theory [5, 6], mechanical systems [14] and time varying systems [9, 15], all have connections with these matrices. Also in the field of rational interpolation and approximation theory semiseparable matrices [11, 12] play an important role.

Within the structured matrix field, semiseparable matrices are well known as the inverses of strict band matrices [1, 19, 24, 25, 26].

In several applications symmetric matrices arise from which the complete, or part of the spectrum has to be computed. Traditional approaches for calculating the spectrum are based on a tridiagonalization of the symmetric matrix [16], and then eigenvalue solvers are applied on this tridiagonal matrix. For tridiagonal matrices there exists a huge class of eigenvalue solvers, for example the traditional *QR* algorithm [16] and divide and conquer algorithms [2, 3, 7].

The algorithm proposed in [29], performs a similarity transformation of a symmetric matrix towards semiseparable form. The algorithm as presented there, has two interesting properties. The first property of the reduction is the fact that while running the algorithm, semiseparable matrices of increasing dimensions are created. These intermediate semiseparable matrices have the Lanczos-Ritz-values as eigenvalues. A second convergence property of the reduction is the performance of subspace iteration, while transforming the matrix. This subspace iteration tends to make the matrix block diagonal. Both of the mentioned properties are advantaguous, when searching for the eigenvalues of this semiseparable matrix, coming from the reduction algorithm. These benifits can be seen rather clearly in the numerical examples.

For the class of semiseparable matrices already several eigenvalue solvers exists, which exploit the structure of the matrices [4, 11]. Another possible approach to calculate the eigenvalues of these semiseparable matrices is the reduction towards bi- or tridiagonal form, and then applying one of the techniques mentioned above [13, 22].

In this paper we propose a new approach to compute the eigenvalue decomposition of symmetric and semiseparable matrices. Instead of a reduction towards a tridiagonal matrix the matrix is reduced into a similar semiseparable one [29]. Afterwards an implicit *QR* algorithm for semiseparable matrices is applied. Both of the algorithms mentioned, have the same order of complexity as their corresponding algorithms for the tridiagonal approach.

Although the algorithms have the same order of complexity, they have larger lower order terms. Nevertheless this approach can be very useful as will be demonstrated in one of the numerical examples where it is shown that the average number of iteration steps to approximate each eigenvalue for a semiseparable matrix is less than the number of iteration steps required by the QR algorithm applied to the corresponding tridiagonal matrix. Other experiments are performed, comparing the semiseparable approach with the tridiagonal approach. The behavior of both algorithms applied to special problem matrices is investigated. The reduction towards semiseparable and tridiagonal form is compared and a final experiment is dedicated to the influence of the criterion how to cut off eigenvalues, on the accuracy of the results.

2 Different definitions and representations of semiseparable matrices

In the literature semiseparable matrices are defined in different ways. The most frequently used definition is given in the next subsection.

2.1 A first definition and a corresponding representation

Definition 1 *S is called a semiseparable matrix of semiseparability rank r if there exist two matrices R_1 and R_2, both of rank r, such that*

$$S = triu(R_1) + tril(R_2);$$

$triu(R_1)$ and $tril(R_2)$ denote respectively the upper triangular part of the matrix R_1 and the strictly lower triangular part of the matrix R_2. Suppose the semiseparability rank to be equal to 1, this means that R_1 and R_2 are two rank one matrices. Therefore they can both be written as the outer product of two vectors, respectively u and v for R_1 and s and t for R_2. These vectors are also called the generators of the semiseparable matrix S.

Theoretically this definition with u, v, s and t is very useful and many algorithms for semiseparable matrices are written in terms of the generators u, v, s and t of the semiseparable matrix S. Even though this representation is cheap in terms of memory usage, it is easy to reconstruct arbitrary elements within the matrix and several algorithms for semiseparable matrices are based on this representation, it lacks numerical stability for our purpose. Because our algorithm is based on applying implicit QR steps to semiseparable matrices, we know that the semiseparable matrix will tend to become more and more block diagonal. This will introduce very small elements in the lower left and the upper right corner and these small elements will cause the latter representation to fail, as can be seen in the following example.

Talking about semiseparable matrices throughout the remaining part of the paper, we consider symmetric semiseparable matrices of semiseparability rank 1. These matrices can be represented with two generators.

Example 1 *Suppose a symmetric* 5×5 *matrix with eigenvalues:* $(1,2,3,100,10^5)$ *is given. Constructing a semiseparable matrix from it (see e.g. [29]) generates the following matrix:*

$$\left(\begin{array}{ccccc} 1.2738 & -5.7004 \cdot 10^{-1} & 1.2664 \cdot 10^{-1} & -1.6459 \cdot 10^{-4} & 1.5753 \cdot 10^{-12} \\ -5.7004 \cdot 10^{-1} & 2.2236 & -4.9398 \cdot 10^{-1} & 6.4202 \cdot 10^{-4} & -1.5858 \cdot 10^{-13} \\ 1.2664 \cdot 10^{-1} & -4.9398 \cdot 10^{-1} & 2.5026 & -3.2527 \cdot 10^{-3} & 1.5679 \cdot 10^{-12} \\ -1.6459 \cdot 10^{-4} & 6.4202 \cdot 10^{-4} & -3.2527 \cdot 10^{-3} & 1.0000 \cdot 10^{2} & 4.8030 \cdot 10^{-8} \\ 1.5753 \cdot 10^{-12} & -1.5858 \cdot 10^{-13} & 1.5679 \cdot 10^{-12} & 4.8030 \cdot 10^{-8} & 1.0000 \cdot 10^{5} \end{array}\right).$$

Although this matrix is semiseparable it can clearly be seen that the lower right element already converged to the largest eigenvalue. Representing this matrix now with the generators u and v (with the last component of u equal to 1*) gives us the following vectors:*

$$u = \left(\begin{array}{ccccc} 8.0861 \cdot 10^{11} & -3.6187 \cdot 10^{11} & 8.0391 \cdot 10^{10} & -1.0448 \cdot 10^{8} & 1.0000 \end{array}\right)^T$$

and

$$v = \left(\begin{array}{ccccc} 1.5753 \cdot 10^{-12} & -1.5858 \cdot 10^{-13} & 1.5679 \cdot 10^{-12} & 4.8030 \cdot 10^{-8} & 1.0000 \cdot 10^{5} \end{array}\right).$$

Because the second element of v is of the order 10^{-13} *and is constructed by summations of elements of order* 1*, we can expect that this element has a precision of only* 3 *signi£cant decimal digits left. Using this number to reconstruct the elements within the matrix will only give these elements with a limited number of exact digits.*

The explanation why this representation fails is rather straightforward. The limit of a sequence of semiseparable matrices generated by e.g. QR is a (block-)diagonal matrix. Only some special (block-)diagonal matrix can be represented by generators u and v.

To overcome this problem, semiseparable matrices are de£ned as a more general class in the following subsection.

2.2 An alternative de£nition

De£nition 2 *A matrix S is called a lower- (upper-)semiseparable matrix of semiseparability rank r if all submatrices which can be taken out of the lower (upper) triangular part of the matrix S have rank* $\leq r$ *and there exists at least one submatrix having exact rank r.*

Assume that a semiseparable matrix satisfying De£nition 2 is denoted as S, and a semiseparable matrix, representable with two generators u, v is denoted as $S(u,v)$. The next theorem shows how the class of semiseparable matrices represented with two generators can be embedded in the class of semiseparable matrices as de£ned in De£nition 2. To prove the theorem an interesting property is needed, revealing the close connection between the two de£nitions.

Proposition 2.1 *Suppose a symmetric semiseparable matrix S is given, which cannot be represented by two generators, then this matrix can be written as a block diagonal matrix, for which all the blocks are semiseparable matrices representable with two generators.*

PROOF: It can be seen that a matrix S cannot be represented by two generators (e.g. u and v), if and only if

$$\begin{array}{rll}
\exists k : 1 \leq k \leq n, \exists l : 1 \leq l \leq k & \text{such that} & S(k,l) = 0 \\
\exists i : l \leq i \leq n & \text{such that} & S(i,l) \neq 0 \\
\exists j : 1 \leq j \leq k & \text{such that} & S(k,j) \neq 0.
\end{array}$$

(In case it is representable with two generators, a u_k, v_l would exist for which one of the two has to be zero, this leads to a contradiction.)

$$\begin{array}{r|ccccc|}
 & & & l & j & k \\
 & & & \downarrow & \downarrow & \downarrow \\
 & \ddots & & \vdots & & \vdots \\
 & & \ddots & \vdots & & \vdots \\
l \rightarrow & \cdots & \cdots & \ddots & & \vdots \\
i \rightarrow & & & \times & \ddots & \vdots \\
k \rightarrow & \cdots & \cdots & 0 & \times & \ddots \\
 & & & \vdots & &
\end{array}$$

Because of the rank 1 assumption following from De£nition 2, extra conditions can be placed on the indices, namely: $i < k$ and $j > l$. Suppose now, that the element $S(\hat{i}, l) \neq 0$, with $l \leq \hat{i} < k$ and all $S(i,l) = 0$ for $\hat{i} < i < k$. The rank one assumption on the blocks implies that $S(i,j) = 0$, for all $\hat{i} < i \leq n$ and $1 \leq j < \hat{i} + 1$. This means that our matrix can be divided into two blocks. This procedure can be repeated until all the diagonalblocks are representable by two generators. ■

The following theorem justi£es the new de£nition of semiseparable matrices. Also clearly seen in the following proof, is the case when problems arise with the de£nition in terms of the generators. First the pointwise limit of a sequence of matrices will be de£ned.

De£nition 3 *The pointwise limit of a collection of matrices $A_\varepsilon \in \mathbb{R}^{n \times n}$ for $\varepsilon \in I$ (if it exists). With the matrices A_ε as*

$$A_\varepsilon = \begin{pmatrix} (a_{1,1})_\varepsilon & \cdots & (a_{1,n})_\varepsilon \\ \vdots & \ddots & \vdots \\ (a_{n,1})_\varepsilon & \cdots & (a_{n,n})_\varepsilon \end{pmatrix}$$

is de£ned as:

$$\lim_{\varepsilon \to \varepsilon_0} A_\varepsilon = \begin{pmatrix} \lim_{\varepsilon \to \varepsilon_0} (a_{1,1})_\varepsilon & \cdots & \lim_{\varepsilon \to \varepsilon_0} (a_{1,n})_\varepsilon \\ \vdots & \ddots & \vdots \\ \lim_{\varepsilon \to \varepsilon_0} (a_{n,1})_\varepsilon & \cdots & \lim_{\varepsilon \to \varepsilon_0} (a_{n,n})_\varepsilon \end{pmatrix}$$

Theorem 3.1 *The pointwise closure of the class of semiseparable matrices representable by two generators is the class of semiseparable matrices according to De£nition 2.*

PROOF:

⇒ Suppose a sequence of semiseparable matrices representable with two generators is given:

$$S(u(\varepsilon), v(\varepsilon)) \in \mathbb{R}^n \text{ for } \varepsilon \in I, \tag{1}$$

such that the pointwise limit exists:

$$\lim_{\varepsilon \to \varepsilon_0} S(u(\varepsilon), v(\varepsilon)) = S \in \mathbb{R}^n.$$

It will be shown that this matrix belongs to the class of semiseparable matrices from De£nition 2.

It is known that $\lim_{\varepsilon \to \varepsilon_0} (u_i(\varepsilon) v_j(\varepsilon)) \in \mathbb{R}$. (Note that this last demand does not imply that $\lim_{\varepsilon \to \varepsilon_0} u_i(\varepsilon), \lim_{\varepsilon \to \varepsilon_0} v_j(\varepsilon) \in \mathbb{R}$, which can lead to numerical unsound problems when representing these semiseparable matrices with two generators u, v.) It remains to prove that, $\forall i \in \{2, \ldots, n\}$:

$$rank\left(\lim_{\varepsilon \to \varepsilon_0} (S(u(\varepsilon), v(\varepsilon))(i:n, 1:i))\right) = 1.$$

All the limits within the above equation are well de£ned, and therefore the limit can be placed outside the $rank()$. We get:

$$\lim_{\varepsilon \to \varepsilon_0} (rank((S(u(\varepsilon), v(\varepsilon))(i:n, 1:i)))) = \lim_{\varepsilon \to \varepsilon_0} 1 = 1.$$

Which proves one direction of the theorem.

⇐ Suppose a semiseparable matrix S is given such that it cannot be represented by two generators. Then there exists a sequence $S(u(\varepsilon), v(\varepsilon))$ with $\varepsilon \longrightarrow \varepsilon_0$ such that

$$\lim_{\varepsilon \to \varepsilon_0} S(u(\varepsilon), v(\varepsilon)) = S \tag{2}$$

According to Proposition 2.1 the matrix can be written as a block diagonal matrix, consisting of 2 diagonal blocks (more diagonal blocks can be dealt with in an analogous way), which can both be represented by two generators, i.e., S has the following structure:

$$S = \begin{pmatrix} S(u,v) & 0 \\ 0 & S(s,t) \end{pmatrix}. \tag{3}$$

In a straightforward way we can de£ne the generators $u(\varepsilon), v(\varepsilon)$:

$$\begin{aligned} u(\varepsilon) &= \left[\frac{u_1}{\varepsilon}, \ldots, \frac{u_k}{\varepsilon}, s_1, \ldots, s_l\right] \\ v(\varepsilon) &= [\varepsilon v_1, \ldots, \varepsilon v_k, t_1, \ldots, t_n]. \end{aligned}$$

It is clearly seen that the limit:

$$\lim_{\varepsilon \to 0} S(u(\varepsilon), v(\varepsilon)) = S. \tag{4}$$

This proves the theorem. ■

The proof shows that the limit

$$\lim_{\varepsilon \to 0} S(u(\varepsilon), v(\varepsilon)) = S \tag{5}$$

exists, but the limits of the generating vectors

$$\begin{array}{rcl} u(\varepsilon) & = & \left[\frac{u_1}{\varepsilon}, \ldots, \frac{u_k}{\varepsilon}, s_1, \ldots, s_l\right] \\ v(\varepsilon) & = & [\varepsilon v_1, \ldots, \varepsilon v_k, t_1, \ldots, t_n] \end{array}$$

do not necessarily exist. In fact for $\varepsilon \to 0$ some elements of $u(\varepsilon)$ will become extremely large, while some elements of $v(\varepsilon)$ will become extremely small. This is the behaviour observed in Example 1.

2.3 Another representation

Still one question remains: can these matrices be represented such that they preserve the interesting properties of the representation with the generators? The answer is affirmative: the new representation consists of a sequence of Givens transformations and a vector. Suppose the semiseparable matrix is of dimension n, then $n-1$ Givens transformations and a vector of length n are needed. The following figures denote how a semiseparable matrix can be constructed, using this information. From the rows and columns built up by the elements denoted by $\boxtimes$ only matrices having maximum rank 1 can be constructed. The new representation is introduced considering, as an example, the construction of a semiseparable matrix of order 5. The Givens transformations are denoted as $G = [G_1, \ldots, G_4]$ and the vector as $d = [d_1, \ldots, d_5]$. Initially one starts on the first 2 rows of the matrix. The element d_1 is placed in the upper left position, then a Givens transformation is applied, and finally to complete the first step element d_2 is added in position $(2,1)$. Only the first two columns and rows are shown here.

$$\begin{pmatrix} d_1 & 0 \\ 0 & 0 \end{pmatrix} \to G_1 \begin{pmatrix} d_1 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & d_2 \end{pmatrix} \to \begin{pmatrix} \boxtimes & 0 \\ \boxtimes & d_2 \end{pmatrix}.$$

The second step consists of applying the Givens transformation G_2 to the second and the third row, furthermore d_3 is added in position $(3,3)$. Here only the first three columns are shown and the second and third row. This leads to:

$$\begin{pmatrix} \boxtimes & d_2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \to G_2 \begin{pmatrix} \boxtimes & d_2 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & d_3 \end{pmatrix} \to \begin{pmatrix} \boxtimes & \boxtimes & 0 \\ \boxtimes & \boxtimes & d_3 \end{pmatrix}.$$

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 d_4. After applying all the Givens transformations and adding all the diagonal elements, the lower triangular part of a symmetric semiseparable matrix is constructed. Because of the symmetry also the upper triangular part is known.

Suppose the Givens and vector representation of a semiseparable matrix S is known. When denoting the Givens transformations as:

$$G_l = \begin{pmatrix} c_l & -s_l \\ s_l & c_l \end{pmatrix}. \tag{6}$$

The elements $S(i,j)$ with $n > i \geq j$ are calculated in the following way:
$S(i,j) = c_i s_{i-1} s_{i-2} \cdots s_j d_j$. When $n = i$ we have $S(i,j) = s_{n-1} s_{n-2} \cdots s_j d_j$. When $n \geq j > i$, $S(i,j)$ can be calculated in a similar way, because of the symmetry. The elements of the semiseparable matrix can therefore be calculated in a stable way. In this way, the matrix presented above in Example 1 can be represented also with this Givens-vector representation.

Example 2 (Example 1 continued) *The Givens-vector representation of the matrix in Example 1 is the following: (In the £rst row of G the elements $c_1, \ldots, c_4$ are stored and in the second row the elements $s_1, \ldots, s_4$.)*

$$G = \begin{pmatrix} 9.0903 \cdot 10^{-1} & 9.7620 \cdot 10^{-1} & 9.9999 \cdot 10^{-1} & 1.0000 \\ -4.1672 \cdot 10^{-1} & -2.1686 \cdot 10^{-1} & -1.2997 \cdot 10^{-3} & 4.8030 \cdot 10^{-10} \end{pmatrix}$$

and

$$d = \begin{pmatrix} 1.4012 & 2.2778 & 2.5026 & 1.0000 \cdot 10^2 & 1.0000 \cdot 10^5 \end{pmatrix} \tag{7}$$

When the elements of G and d are known with high relative precision, also all the elements of the semiseparable matrix can be reconstructed now with high relative precision.

It is a well known fact that the representation with the generators leads to numerical instabilities, e.g [10], and sometimes there are also theoretical problems involved. A profound study of the possible representations, advantages, and drawbacks can be found in [32].

3 The similarity reduction of a symmetric matrix towards a semiseparable one

In this section a brief overview is given of the properties of the algorithm, which transforms a symmetric matrix into a similar semiseparable one. A more elaborate discussion of the properties of this reduction can be found in [29].

The algorithm creates a sequence of semiseparable matrices which increase 1 dimension at each step. The overall cost of the algorithm is the same as the reduction of a symmetric matrix into a tridiagonal one namely $O(n^3)$. As already stated in the introduction, the generated sequence of semiseparable matrices have as eigenvalues the Ritz-values as generated by the Lanczos algorithm, choosing e_1, the £rst vector of the canonical basis, as initial vector. Even though we will not investigate this behaviour more thouroughly in this paper, one can see that it is worth calculating the eigenvalues of the intermediate semiseparable matrices. This can for example be done by the implicit QR algorithm for semiseparable matrices, as will be explained further on.

A second feature of the reduction algorithm, is the performance of subspace iteration. While reducing the symmetric matrix A to semiseparable form, a type of nested subspace iteration is performed on the matrix A. Under certain assumptions this subspace iteration, will tend to make the matrix block diagonal. This means that, under some mild assumptions, the resulting semiseparable matrix, can already be approximated by a block diagonal. This block diagonal structure is extremely useful, when computing the eigenvalues of this semiseparable matrix. The eigenvalues of the complete semiseparable matrix correspond to the eigenvalues of the separate diagonal blocks. This division into blocks can reduce the complexity.

The next secion is dedicated to calculating the eigenvalues of an arbitrary semiseparable matrix. In Section 5 we will show that the eigenvectors can be calculated by applying inverse iteration to the semiseparable matrices.

4 An implicit QR algorithm for semiseparable matrices

4.1 The semiseparable form

In the following parts of this section we assume that the semiseparable matrix we are working with is nonsingular, and representable with two generators. If it is not representable with two generators the matrix can be split up in several blocks, all representable with two generators (see Proposition 2.1). Remark once more that this representation with the generators u and v is only needed for theoretical purposes. In the practical implementation the representation with a vector and a sequence of Givens transformations is used to overcome numerical instabilities.

For the nonsingularity property, we have to divide the matrices we are working with in two different classes. The £rst class of matrices, consists of the semiseparable matrices coming from the reduction of a symmetric matrix into semiseparable form. If the symmetric matrix we want to reduce is singular, the similar semiseparable matrix will have zero rows and columns. These can be extracted such that the remaining semiseparable matrix is nonsingular.

The second class of semiseparable matrices, are arbitrary semiseparable matrices, not constructed from a symmetric matrix. For this class of matrices one has to check for singularities during execution of the algorithm. The algorithm consists of two parts, twice a sequence of $n-1$ Givens transformations is performed on the matrices. The £rst sequence of Givens transformations corresponds to performing a QR step without shift on the semiseparable matrix. The combination of the £rst and the second step correspond to performing a QR step on the semiseparable matrix with a shift. After having performed the £rst $n-1$ Givens transformations, singularities will be revealed, because a QR step without shift is performed. This means that after the £rst step we have a similar semiseparable matrix, from which zero rows and or columns can be extracted.

This short explanation assures that we can assume that the semiseparable matrices we are working with are indeed nonsingular and representable with two generators.

4.2 Theoretical proof of the implicit QR algorithm

In [29] the attention was restricted to the reduction of a symmetric matrix to semiseparable one. Here we design an implicit QR algorithm for £nding the eigenvalues of semiseparable matrices. A lot of attention is paid to an effective implementation of this implicit QR algorithm, such that we get an $O(n)$ complexity for every iteration step of the algorithm.

The implicit QR algorithm is based on the QR factorization of semiseparable plus diagonal matrices as described in [30]. The Q factor of this reduction consists of $2n-2$ Givens transformations. The £rst $n-1$ Givens transformations are performed on the rows of the semiseparable matrix from bottom to top. These Givens transformations, transform the semiseparable structure into an upper triangular one. When taking into consideration the diagonal, one can see that these $n-1$ Givens transformations, transform the diagonal part into an upper Hessenberg one. Recombining the upper triangular matrix and the upper Hessenberg matrix gives an upper Hessenberg matrix. This matrix will then be made upper triangular by $n-1$ Givens transformations from top to bottom.

In the following part a method is explained to calculate directly the matrix QSQ^T, when S is a semiseparable matrix, i.e., instead of £rst calculating the QR factorization of $S-\kappa I$ and then multiplying R on the right with Q. (κ is a shift chosen in order to speed up the convergence of the implicit QR algorithm, e.g., the Rayleigh shift, the Wilkinson shift).

Because the QR factorization in fact consists of two parts, a £rst reduction to upper Hessenberg and a second reduction to make the matrix upper triangular, the implicit QR algorithm will also be divided into two parts.

The proof of the correctness of the approach followed is based on the fact that applying a QR step to a semiseparable matrix gives again a semiseparable matrix. The theorem is based on the representation of the semiseparable matrices in terms of the generators u and v. To prove this for an arbitrary semiseparable matrix S, we split the matrix up into several diagonal blocks, representable with two generators.

To prove the theorem for u, v representable semiseparable matrices a small theorem is needed:

Proposition 3.1 *Suppose S is a nonsingular symmetric semiseparable matrix, representable with two generators. Then S can be written as the sum of a rank 1 matrix and a strictly upper triangular matrix. The upper triangular matrix has nonzero superdiagonal elements.*

$$S = uv^T + R_u \tag{8}$$

PROOF: De£ne R_u as $S-uv^T$. The nonsingularity demand corresponds to the fact that the superdiagonal elements have to be nonzero. (proof can be found in [31]) ■

The following theorem states that applying one step of QR to a semiseparable matrix representable by u, v is again a semiseparable matrix.

Theorem 3.2 *Suppose a nonsingular symmetric semiseparable matrix S, representable with two generators, and a real number κ are given. Suppose the following equalities, where Q is an orthogonal matrix, R is upper triangular, and I is the identity matrix are satis£ed:*

$$\begin{aligned} QR &= S-\kappa I && (9)\\ \hat{S} &= RQ+\kappa I. && (10)\end{aligned}$$

Then the matrix $\hat{S}$ is semiseparable.

PROOF: Following from equation (10) we have:

$$\hat{S}R = RQR+\kappa IR.$$

Substituting equation (9) gives

$$\hat{S}R = R(S-\kappa I)+\kappa IR = RS.$$

This means that $\hat{S}R = RS$. Using Proposition 3.1 and the fact that the matrix S is nonsingular, we can write the following equation

$$\begin{aligned} \hat{S} &= RSR^{-1} = R\left(uv^T+R_u\right)R^{-1}\\ &= (Ru)\left(v^TR^{-1}\right)+RR_uR^{-1}\end{aligned}$$

Because RR_uR^{-1} is strictly upper triangular and the matrix $\hat{S}$ is symmetric, the last equation gives the desired result. ■

The next two subsections are dedicated to the construction of the implicit QR algorithm. The £rst $n-1$ Givens transformations of the factorization are written as $G_1,\ldots,G_{n-1}$ and the second $n-1$ Givens transformations are denoted as $G_n,\ldots,G_{2n-2}$.

4.3 Applying the £rst $n-1$ Givens transformations

The £rst $n-1$ Givens transformations are in fact completely determined by the semiseparable matrix. To perform these transformations, the shift κ is not yet needed. When applying these Givens transformations to the left from the bottom to the top of the semiseparable matrix S, this matrix becomes upper triangular:

$$G_{n-1}\ldots G_1S = S_u.$$

S_u denotes an upper triangular matrix. Directly applying the Givens transformations $G_1^T\ldots G_{n-1}^T$ on the right of the matrix S_u, will construct a matrix $S_uG_1^T\ldots G_{n-1}^T$ whose lower triangular part is semiseparable. Because of symmetry reasons the resulting matrix $S_uG_1^T\ldots G_{n-1}^T$ is a symmetric semiseparable matrix. It can be seen that the application of the different Givens transformations can be done at the same time, i.e. instead of £rst applying all the transformations to the left, we apply them left and right at the same time (more details can be found in Section 4.6):

$$S^{(1)} = G_1SG_1^T$$

followed by

$$S^{(2)} = G_2 S^{(1)} G_2^T.$$

This process has to be repeated to achieve the desired result.

As stated before, this step corresponds to applying a *QR* step without shift to the semiseparable matrix.

4.4 Applying the second $n-1$ Givens transformations

This step is the hardest of the two and requires some theoretical results. To initialize this step the knowledge of the Givens transformation G_n is crucial. G_n is the Givens transformation which will start to reduce the upper Hessenberg matrix $G_{n-1} \dots G_1(S - \kappa I)$ to upper triangular form. The algorithm however did not calculate $G_{n-1} \dots G_1(S - \kappa I)$ but a semiseparable matrix $S^{(n-1)} = G_{n-1} \dots G_1 S G_1^T \dots G_{n-1}^T$. Because however we use the special matrix representation as mentioned in the £rst section, we know that the upper left element of the matrix $G_{n-1} \dots G_1 S$ is the £rst element in the vector d from the representation of the matrix S. It can be seen that the elements in the upper left positions $(1,1)$ and $(2,1)$ of the matrix $G_{n-1} \dots G_1 \kappa I$ equal

$$G_{n-1} \begin{pmatrix} \kappa \\ 0 \end{pmatrix}$$

This means that the Givens transformation G_n already can be applied to the matrix $S^{(n-1)}$, i.e., $S^{(n)} = G_n S^{(n-1)} G_n^T$. From this point we will work directly on the matrix $S^{(n)}$ and therefore we switch to the implicit approach.

The matrix $S^{(n-1)}$ is a semiseparable matrix and the output of one step of the implicit *QR* algorithm also has to be a semiseparable matrix. However after having applied the similarity transformation $G_n S^{(n-1)} G_n^T$ the semiseparable structure is disturbed. The following sequence of Givens transformations which will be applied to the matrix $S^{(n)}$ will restore the semiseparable structure. Even more: the resulting matrix will essentially be the same as the matrix coming from the *QR* algorithm. We will show that it is possible to rebuild a semiseparable matrix out of $S^{(n)}$ without changing the £rst row and column. The prove the two main Theorems, two properties are needed.

Proposition 3.2 *Suppose the following symmetric* 3×3 *matrix is given,*

$$A = \begin{pmatrix} x & a & d \\ a & b & e \\ d & e & f \end{pmatrix}, \tag{11}$$

which is not yet semiseparable. Then there exists a Givens transformation

$$G = \frac{1}{\sqrt{1+t^2}} \begin{pmatrix} 1 & 0 & 0 \\ 0 & t & 1 \\ 0 & -1 & t \end{pmatrix}$$

such that the following matrix

$$GAG^T$$

is a symmetric semiseparable matrix.

Remark that diagonal matrices are also considered to be semiseparable according to De£nition 2.

PROOF: We can assume that $ae-bd$ is different from zero, otherwise, the matrix would already be semiseparable. The proof is constructive, i.e. the matrix G will be constructed such that the matrix GAG^T indeed is a semiseparable matrix. Calculating explicitly the product GAG^T gives the following matrix:

$$\frac{1}{1+t^2}\left(\begin{array}{ccc} x & at+d & -a+dt \\ ta+d & (tb+e)t+(te+f) & (-1)(tb+e)+(te+f)t \\ -a+td & (-b+te)t+(-e+tf) & (-1)(-b+te)+(-e+tf)t \end{array}\right).$$

To be semiseparable, the lower left 2×2 block has to be of rank 1, this means that the following equation should be satis£ed:

$$\frac{ta+d}{-a+td} = \frac{(tb+e)t+(te+f)}{(-b+te)t+(-e+tf)}$$

Solving this equation towards t gives the following result:

$$t = -\frac{-de+af}{ae-db}$$

■

Proposition 3.3 *Suppose the following symmetric* 4×4 *matrix is given,*

$$A = \left(\begin{array}{cccc} x & a & 0 & d \\ a & b & 0 & e \\ 0 & 0 & 0 & f \\ d & e & f & y \end{array}\right), \qquad (12)$$

which is not yet semiseparable. Then there exists a Givens transformation

$$G = \frac{1}{\sqrt{1+t^2}}\left(\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & t & 1 & 0 \\ 0 & -1 & t & 0 \\ 0 & 0 & 0 & 1 \end{array}\right)$$

such that the upper 3×3 *block of the following matrix*

$$GAG^T$$

is a symmetric semiseparable matrix. And the lower left 2×2 *block is of rank one.*

PROOF: The proof is straightforward, calculating the product GAG^T, shows that the block

$$\left(\begin{array}{cc} -a & -tb \\ d & te+f \end{array}\right)$$

has to be of rank 1. This corresponds with $t = -af/(ae-bd)$. $(ae-bd)$ is different from zero because the matrix is not yet semiseparable. ■

One might wonder how both these theorems can be used for larger matrices, this is proven in the following theorem:

Theorem 3.3 *Suppose a symmetric nonsingular matrix A is given, such that the upper left* 2×2 *block of the matrix does not satisfy the semiseparable structure of the remaing part of the matrix.*

Then there exists a Givens transformation G:

$$G = \frac{1}{\sqrt{1+t^2}} \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots \\ 0 & t & 1 & 0 & \cdots \\ 0 & -1 & t & 0 & \cdots \\ 0 & 0 & 0 & 1 & \\ \vdots & \vdots & \vdots & & \ddots \end{pmatrix}$$

such that the upperleft 3×3 *block of the matrix* GAG^T *is semiseparable, and the £rst two columns are dependent, except for the £rst element.*

PROOF: The proof is divided in different cases, thereby using both of the above propositions.

Case 1 Suppose the matrix A is the same as in equation 11, with d and e different from zero. In this case Proposition 3.2 is applied.

Case 2 Suppose the matrix A is the same as in equation 12. It is clear that d, e and f have to be different from zero, otherwise there are contradictions with the facts that the matrix is semiseparable below the upper left 2×2 block or the fact that the matrix is nonsingular or the fact that the matrix is not yet semiseparable. For example assume $f = 0$ and d and e different from zero. Because the complete lower part is semiseparable, the rank 1 assumption assumes that the complete columns in which f appears is zero. This is in contradiction with the fact the the matrix is nonsingular. Or another example, assume all d,e,f equal to zero, this means that row 3 and row 4 are dependent, what is in contradiction with the nonsingularity assumption.

In this case Proposition 3.3 is used.

■

Now we are ready to reduce a semiseparable matrix which is disturbed in the upper left 2×2 block, back to semiseparable form.

Theorem 3.4 *Suppose we have an* $n\times n$ *nonsingular symmetric semiseparable matrix A, representable with two generators, which will be disturbed in the £rst two rows and columns by means of a similarity Givens transformation G. Then there exists an orthogonal transformation U with* $Ue_1 = e_1$ *such that* $UGAG^TU^T$ *is again a symmetric semiseparable matrix.*

PROOF: In this theorem a 5×5 matrix is considered, and we will use the special Givens transformation from Theorem 3.3. Some more notation is needed to make the construction of U more easy: Denote with $G^{(i)}$ the orthogonal transformation which performs a Givens transformation on the rows i and $i+1$ of the matrix A. To prove that the algorithm gives the desired result, several £gures are included. Starting with the matrix A each £gure shows all the dependencies in the matrix. In the following £gure, the blocks grouped by the full lines represent semiseparable parts in the matrix, and the elements grouped by the dashed lines represent rank 1 parts. These rank 1 parts are very important for the progress of the algorithm. The arrows point out the rows and columns on which there will be performed the Givens transformations to come to the desired result. The £rst £gure, shows what happens with the matrix A after the disturbing Givens transformation is applied.

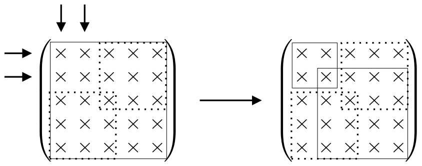

Figure 1: Applying the Givens transformation which disturbs the semiseparable structure.

The following step consists of calculating the Givens transformation from Theorem 3.3. For case 1 of the theorem, Proposition 3.2 is used and a little explanation is needed why we get Figure 4. For case 2, we can skip the following comments, and we immediately arrive at Figure 4.

When applying the Givens transformation of the £rst kind, we take a closer look to see how the dependencies will change. First we apply the transformation $G^{(2)}$ only to the left of the matrix, to see how the dependencies will change. (See Figure 2.) There are now two rank 1 parts, the small 2×2 matrix and the larger 3×3 matrix. The small block has to be of rank 1 because the next Givens transformation $G^{(2)}$ applied to the right will not change the rank of this block, and after this Givens transformation that block is part of the semiseparable matrix, and therefore of rank 1. The 3×3 block is of rank 1 because the transformation involved the large 3×3 rank 1 block of the preceding matrix. This means that after applying the Givens transformation to the left we have a 2×4 matrix of rank 1 (See Figure 3).

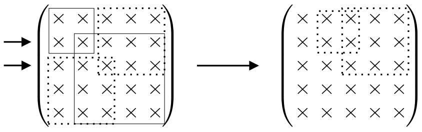

Figure 2: Applying the transformation $G^{(2)}$ to the left.

Applying the transformation $G^{(2)}$ to the right of the matrix, gives the matrix $A^{(2)} =$

$G^{(2)}A(G^{(2)})^T$ which, because of symmetry reasons has the following structure:

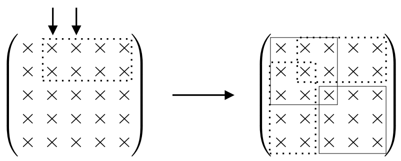

Figure 3: Applying the transformation $G^{(2)}$ to the right

The figure shows clearly that the upper semiseparable part has increased, while the lower semiseparable part is reduced in dimension. Very important are the remaining rank 1 parts. The remaining dependency in these blocks will make sure that the next Givens transformation will indeed create a semiseparable matrix of dimension 4. The next Givens transformation $G^{(3)}$ is calculated by using Theorem 3.3 applied to the matrix $A^{(2)}$ without the first row and column. Applying the transformation $G^{(3)}$, this means calculating $A^{(3)} = G^{(3)}A(G^{(3)})^T$ will create a semiseparable block in the middle of the matrix, however because of the rank 1 parts, the complete upper left 4×4 block will become dependent. This is shown in Figure 4:

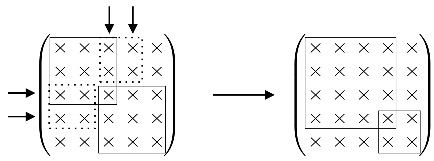

Figure 4: Applying the similarity transformation $G^{(3)}$

Before performing the final Givens transformation, one can also search the rank 1 blocks such that the last Givens transformation will transform the matrix into complete semiseparable one. ■

This final theorem produces an algorithm to transform the semiseparable matrix with a disturbance in the upper left part back to an orthogonal similar semiseparable matrix. In the next subsection it will be proven that the constructed semiseparable matrix will be essentially the same as the semiseparable matrix coming directly from the *QR* algorithm.

4.5 Proof of the correctness of the approach

We will prove that the matrix constructed with the approach above is essentially the same as the semiseparable matrix, constructed directly from the *QR* algorithm.

Definition 4 *Two matrices $S^{(1)}$ and $S^{(2)}$ are called essentially the same if there exists* *a matrix* $W = diag(\pm 1, \pm 1, \ldots, \pm 1)$ *such that the following equation holds:*

$$S^{(1)} = WS^{(2)}W^T.$$

The following theorem which is proven in [31] states the fact that the approach mentioned above is correct. This theorem can be seen as an analogue of the implicit Q theorem for semiseparable matrices.

Theorem 4.1 *Suppose* A *is a nonsingular symmetric matrix and we have the following two equations:*

$$\begin{array}{rcl} Q_1^T A Q_1 & = & S^{(1)} \\ Q_2^T A Q_2 & = & S^{(2)} \end{array}$$

with $Q_1 e_1 = Q_2 e_2$*, where* $S^{(1)}$ *and* $S^{(2)}$ *are two semiseparable matrices and* Q_1 *and* Q_2 *are orthogonal matrices. Then we can assert that the matrices* $S^{(1)}$ *and* $S^{(2)}$ *are essentially the same.*

PROOF: The proof can be found in [31]. ■

We can assert that our matrices are nonsingular, because the procedure, which constructs the semiseparable matrix, swaps dependent rows to zero, and then it chases the zeros to the upper diagonal positions.

4.6 Effective $O(n)$ implementation of the implicit QR algorithm

The implementation can be downloaded at http://www.cs.kuleuven.ac.be/∼marc/. Not all the details of the implementation are given, but the mathematical ideas, behind the algorithm are included, and they should make it possible for the reader to implement this algorithm.

The following function in Matlab[1]-style notation

```
[Gnew,dnew]=CRep(G,d)
```

will perform the first sequence of Givens transformations. The input consists of the Givens vector representation of the matrix, the output is the representation of the new matrix, but from bottom to top.

Suppose our semiseparable matrix A is built up with the Givens transformations G and the vector elements d. We will now perform the first $n-1$ Givens transformations on both sides of the matrix, and we will retrieve the representation of the resulting semiseparable matrix. The matrix A has the following structure:

$$\begin{pmatrix} A^{(n-1)} & c_{n-1}R_{n-1}^T & s_{n-1}R_{n-1}^T \\ c_{n-1}R_{n-1} & c_{n-1}d_{n-1} & s_{n-1}d_{n-1} \\ s_{n-1}R_{n-1} & s_{n-1}d_{n-1} & d_n \end{pmatrix} \qquad (13)$$

[1] Matlab is a registered trademark of the Mathworks Inc.

And the Givens transformations are denoted in the same way as in (6). Remark that the Givens transformations needed for the £rst step of the *QR* algorithm are exactly the Givens transformations G_j^T from the representation. This is a huge advantage, because these Givens transformations do not need to be calculated anymore.

Applying the £rst transformation G_{n-1}^T to the left of matrix (13) gives us the following equations

$$\hat{d}_1 \quad = \quad s_{n-1}^2 d_{n-1} + c_{n-1} d_n$$

and the matrix looks like:

$$\left(\begin{array}{ccc} A & c_{n-1} R_{n-1}^T & s_{n-1} R_{n-1}^T \\ R_{n-1} & d_{n-1} & s_{n-1}(c_{n-1} d_{n-1} - d_n) \\ 0 & 0 & \hat{d}_1 \end{array} \right).$$

Applying the transformation to the right gives the following equations:

$$\left(\begin{array}{ccc} A & R_{n-1}^T & 0 \\ R_{n-1} & \tilde{d}_{n-1} & s_{n-1} \hat{d}_1 \\ 0 & s_{n-1} \hat{d}_1 & c_{n-1} \hat{d}_1 \end{array} \right),$$

with $\tilde{d}_{n-1} = \left(1 + s_{n-1}^2\right) c_{n-1} d_{n-1} - d_n$. When denoting now the new representation with $\hat{G}$ and $\hat{d}$, we get:

$$\hat{G}_1 = \left(\begin{array}{cc} c_{n-1} & -s_{n-1} \\ s_{n-1} & c_{n-1} \end{array} \right)$$

and $\hat{d}_1$. This procedure can now be continued to £nd the full new matrix $\hat{G}$ and the vector $\hat{d}$. Remark once more that this representation is constructed from bottom to top.

The following function

```
[Gnew,dnew]=PerSeqGiv(G,d,specgiv)
```

in the algorithm performs in fact the next $n-1$ Givens transformations on the matrix. This function starts with one special Givens transformation $\hat{G} = specgiv$. Because the following sequence of Givens transformations will divide the matrix into two semiseparable parts, we have to store twice a semiseparable matrix. The decreasing lower right semiseparable matrix will be stored in the Givens and vector representation. While the growing upper left part will also be stored in a the Givens vector representation $Gnew, dnew$.

Suppose we want to perform the £rst special Givens transformation $\hat{G}$ on the matrix A, which looks like (this is different from above, because the representation is from bottom to top now):

$$\left(\begin{array}{ccc} d_n & s_{n-1} d_{n-1} & s_{n-1} R_{n-1} \\ s_{n-1} d_{n-1} & c_{n-1} d_{n-1} & c_{n-1} R_{n-1} \\ s_{n-1} R_{n-1}^T & c_{n-1} R_{n-1}^T & A^{(n-1)} \end{array} \right) \qquad (14)$$

Applying now the £rst special Givens transformation $\hat{G}$ to the left of the matrix (14), we get the following matrix:

$$\begin{pmatrix} \hat{c}d_n - \hat{s}s_{n-1}d_{n-1} & (\hat{c}s_{n-1} - \hat{s}c_{n-1})\,d_{n-1} & (\hat{c}s_{n-1} - \hat{s}c_{n-1})\,R_{n-1} \\ \hat{s}d_n - \hat{c}s_{n-1}d_{n-1} & (\hat{s}s_{n-1} + \hat{c}c_{n-1})\,d_{n-1} & (\hat{s}s_{n-1} + \hat{c}c_{n-1})\,R_{n-1} \\ s_{n-1}R_{n-1}^T & c_{n-1}R_{n-1}^T & A^{(n-1)} \end{pmatrix}$$

Applying the Givens transformation $\hat{G}$ on the right gives, for

$$\begin{aligned} \hat{d}_1 &= \hat{c}^2 d_n + \hat{s}^2 c_{n-1}d_{n-1} - 2\hat{c}\hat{s}c_{n-1}d_{n-1}, \\ f_1 &= (\hat{c}s_{n-1} - \hat{s}c_{n-1})\,/s_{n-1}, \\ f_2 &= (\hat{s}s_{n-1} + \hat{c}c_{n-1})\,/c_{n-1}, \\ \alpha_1 &= \hat{c}\hat{s}d_n + \left((\hat{c}^2 - \hat{s}^2)\,s_{n-1} - \hat{s}\hat{c}s_{n-1}\right)d_{n-1}, \\ \tilde{d}_{n-1} &= \hat{s}^2 d_n + \left(2\hat{c}\hat{s}s_{n-1} + \hat{c}^2 c_{n-1}\right)d_{n-1}, \end{aligned}$$

the following matrix:

$$\begin{pmatrix} \hat{d}_1 & \alpha_1 & f_1 s_{n-1}R_{n-1} \\ \alpha_1 & \tilde{d}_{n-1} & f_2 c_{n-1}R_{n-1} \\ f_1 s_{n-1}R_{n-1}^T & f_2 c_{n-1}R_{n-1}^T & A^{(n-1)} \end{pmatrix}$$

The element $\hat{d}_1$ can already be stored in the Givens vector representation of the new semiseparable matrix. The lower right reduced semiseparable matrix can still be constructed by the old representation and the knowledge of $\tilde{d}_{n-1}$ and the factor f_2. The upper left 3×3 block can now be used to construct the next Givens transformation according to Theorem 3.2. One can clearly see that this procedure, can be repeated to £nd the new diagonal element $\hat{d}_2$ and the £rst subdiagonal element $\hat{d}_1^{(s)}$, and so on.

In the practical implementation the factor f_2 is not used, instead the Givens $\hat{G}$ transformation is applied directly on the vector $[s_{n-1}, c_{n-1}]$ from which the value of the £rst element of $f_2 c_{n-1}R_{n-1}^T$ can be calculated and also the other elements in the same column.

Also the new representation is built up at the same time, by storing extra information concerning the values of α_1 and $\tilde{d}_{n-1}$. For a full understanding the software can be downloaded and investigated.

4.7 The shift

The chosen shift in the QR-algorithm can increase the convergence of the algorithm. In our implementation the rayleigh shift [23] was choosen.

4.8 The de¤ation criterion

An important, yet uncovered topic, is the de¤ation or cutting criterion. When should we divide the semiseparable matrix into smaller blocks, without losing too much information? For semiseparable matrices two things have to be taken into consideration.

The first point of difference with the tridiagonal approach, is the fact that an off diagonal element in the tridiagonal matrix, has all the information corresponding to the non-diagonal block in which the element appears. This is straightforward, because all the other elements are zero. This is however not the case for semiseparable matrices; in fact they are dense matrices. This means that we should derive a way to calculate the norms of the off-diagonal blocks in a fast way. Moreover comparing the norms of all the off-diagonal blocks towards the cutting criterion should in total cost not more than $O(n)$, otherwise this would be the slowest step in the algorithm, which is unacceptable.

The second issue: whether the norm of the block is small enough to divide the problem into two subproblems or not. This is a difficult problem and in fact we will test two different cutting criterions, and see what the difference in accuracy is. The two cutting criterions which will be compared in the numerical experiments section are the aggressive and the normal cutting criterion [16]. The aggressive criterion allows deflation when the norm of the block is relatively smaller than the square root of the machine precision. The normal criterion allows deflation when the norm is relatively smaller then the machine precision. Denoting the machine precision with ε_M, we consider the following two deflation criteria: The aggressive:

$$\|S(i+1:n,1:i)\|_F \leq \sqrt{|d_i d_{i+1}|}\sqrt{\varepsilon_M} \tag{15}$$

or the normal deflation criterion:

$$\|S(i+1:n,1:i)\|_F \leq \sqrt{|d_i d_{i+1}|}\varepsilon_M. \tag{16}$$

When the deflation criterion is satisfied, deflation is allowed and the matrix S is divided into two matrices $S(1:i,1:i)$ and $S(i+1:n,i+1:n)$, thereby neglecting the block $S(i+1:n,1:i)$.

In the remaining part of this section we will derive an order n algorithm to compute the norms of the nondiagonal blocks and to compare them to the current cutting criterion. The semiseparable structure should be exploited when calculating these norms. An easy calculation shows, that for a semiseparable matrix S with the Givens-vector representation the following equations are satisfied:

$$\begin{aligned}
\|S(2:n,1:1)\|_F &= \sqrt{(s_1 d_1)^2} \\
\|S(3:n,1:2)\|_F &= \sqrt{(s_2 s_1 d_1)^2 + (s_2 d_2)^2} \\
&= |s_2|\sqrt{(s_1 d_1)^2 + d_2^2} \\
&= |s_2|\sqrt{\|S(2:n,1:1)\|_F^2 + d_2^2}
\end{aligned}$$

This can be continued and in general we get:

$$\|SS(i+1:n,1:i)\|_F = |s_i|\sqrt{\|S(i:n,1:i-1)\|_F^2 + d_i^2}. \tag{17}$$

This formula allows us to derive an $O(n)$ algorithm to compute and compare the norms of these blocks with the actual cutting criterion.

5 Computing the eigenvectors

In this section the computation of the eigenvectors will be discussed. We divided this into two different subsections. In the first subsection the eigenvectors of the semiseparable matrix S will be approximated by using inverse iteration. In the second subsection these eigenvectors will be transformed into the eigenvectors of the original matrix A by performing several orthogonal transformations.

5.1 Inverse Iteration on semiseparable matrices

Suppose we have already a good approximation of the eigenvalues. With inverse iteration applied to the semiseparable matrix and these approximations, the eigenvectors can be computed rather cheap.

In fact the following problem needs to be solved:

$$(S-\lambda_i I)x = b, \tag{18}$$

with λ_i as an approximate eigenvalue.

We will not go into details on how to solve this system in $O(n)$ operations, and in a stable way. There will only be an outline on how the different steps in the algorithm work. The system will be solved by computing the QR factorization of the semiseparable plus diagonal matrix $S-\lambda_i = QR$. In [30] theoretical results are proven, stating that the strict upper triangular part of the matrix R is of rank two. The algorithm is implemented using the Givens-vector representation of the matrix S, the consecutive Givens transformations to transform the matrix S into upper triangular form are performed on the righthandside b. This gives us the following equation:

$$Q^T(S-\lambda_i)x = Rx = Q^T b \tag{19}$$

The upper triangular matrix R is represented by two sequences of Givens-vector elements, representing the strict upper triangular part of the matrix R. An extra diagonal is kept corresponding to the diagonal elements of R. The final equation $Rx = Q^T b$ is solved by applying backward substitution, because of the special structure of R and the representation, this system can be solved in $O(n)$ operations.

A more detailed overview of the involved transformations, and tests concerning the stability and speed of this implementation can be found in [30].

We can solve the system mentioned above now in a stable and accurate way, and therefore we are able to calculate the eigenvectors of the semiseparable matrix via inverse iteration. In the numerical tests the following algorithm from [28] is used.

```
While (true)
        Solve the system $(SS-\kappa I)y = x$ ;
        $\hat{x} = y/\|y\|_2$;
        $w = x/\|y\|_2$;
        $\rho = \hat{x}^T w$; (Rayleigh quotient)
        $\mu = \kappa + \rho$;
        $\kappa = \mu$;
```

```
    $r = w - \rho\hat{x}$;
    $x = \hat{x}$;
    $if(\|r\|_2/\|A\|_F \leq \varepsilon)$ leave while;
end
```

Starting with a random vector x and the shift κ equal to an approximation of an eigenvalue coming from the implicit QR algorithm, the eigenvectors can be calculated fast and accurate.

5.2 The eigenvectors of an arbitrary matrix A

As mentioned before there can still be the need to transform these eigenvectors, corresponding to the semiseparable matrix S back to the eigenvectors corresponding to the original matrix A.

The most natural way to achieve this goal, is to store the orthogonal transformations performed while reducing the matrix A into the semiseparable matrix S. This corresponds to keeping either a sequence of $n-1$ Householder transformations plus $n(n-1)/2$ Givens transformations, or keeping (when using Givens transformations instead of Householder transformations) $n(n-1)$ Givens transformations.

Once these transformations are stored, one can compute all or part of the eigenvectors, corresponding to the demands.

6 Numerical Experiments

6.1 The experiments

In this section several numerical tests are performed to compare the traditional algorithm for finding all the eigenvalues with the new semiseparable approach. The algorithm is based on the QR step as described in the algorithm section, and implemented in a recursive way, if division in blocks is possible because of the convergence behaviour, then these blocks are dealt with separately.

Before starting the numerical tests some remarks have to be made: first of all the complexity of the reduction of a symmetric matrix into a similar semiseparable one costs $9n^2 + O(n)$ flops more than the reduction of a matrix to tridiagonal form. But as an advantage we get already a quite good ordering of the eigenvalues and already some good approximations, as can be seen from the numerical experiments. An implicit QR step applied to a symmetric tridiagonal matrix costs $31n$ flops while it costs $\approx 10n$ flops more for a symmetric semiseparable matrix. However, this increased complexity is compensated when comparing the number of iteration steps the traditional algorithm needs with the number of steps the semiseparable algorithm needs. Figures about these results can be found in [29] and in the following tests.

6.2 The block experiment

This experiment is taken from [23, pag. 153]. Suppose we have a symmetric matrix $A^{(0)}$ of dimension $n = 10$ and we construct the following matrices $T(m, \delta)$ uxing $A^{(0)}$, where m denotes the number of blocks, and δ are the small subdiagonal elements, between the blocks. For example:

$$T(3,\delta) = \left(\begin{array}{c|c|c} A^{(0)} & \blacksquare & \\ \hline \blacksquare & A^{(0)} & \blacksquare \\ \hline & \blacksquare & A^{(0)} \end{array}\right) \qquad \blacksquare = \delta.$$

We get the following results: For $m = 10$ we compare for the semiseparable and the tridiagonal approach the maximum number of iterations for any eigenvalue to converge. This is done for a varying size of δ, $A^{(0)}$ has eigenvalues $1:10$. Both of the algorithms use the same normal de¤ation criterion.

δ	10^{-13}	10^{-12}	10^{-11}	10^{-10}	10^{-9}	10^{-8}	10^{-7}
semiseparable QR	3	3	2	2	2	3	3
tridiagonal QR	4	4	4	4	4	4	3

For $m = 25$:

δ	10^{-15}	10^{-14}	10^{-13}	10^{-12}	10^{-11}
semiseparable QR	4	4	4	3	3
tridiagonal QR	4	5	5	4	4
δ	10^{-10}	10^{-9}	10^{-8}	10^{-7}	
semiseparable QR	3	2	2	2	
tridiagonal QR	3	4	2	3	

For $m = 40$

δ	10^{-19}	10^{-18}	10^{-17}	10^{-16}	10^{-15}	10^{-14}	10^{-13}
semiseparable QR	4	5	4	4	4	4	4
tridiagonal QR	4	5	4	5	5	4	4
δ	10^{-12}	10^{-11}	10^{-10}	10^{-9}	10^{-8}	10^{-7}	10^{-6}
semiseparable QR	3	1	1	1	3	3	
tridiagonal QR	4	5	4	4	4	5	

It can be seen clearly that the tridiagonal approach has more dif£culties in £nding particular eigenvalues. Figure 5 gives a comparison in the complete number of QR steps for the last experiment ($m = 40$).

The £gure shows clearly that the semiseparable approach needs less iterations than the traditional approach. The matrices involved are of size 400. It can be seen that for δ in the neighbourhood of 10^{-10} the number of QR steps with the semiseparable approach are even less than 400. This can also be seen in the table for $m = 40$.

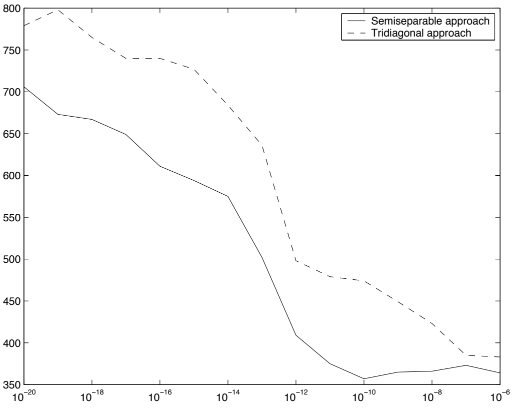

Figure 5: Total number of steps compared to several values of δ.

6.3 Stewart's devil's stairs

In the following example, we do not apply the *QR* algorithm, but we only take a look at the diagonal elements of the semiseparable matrix and the tridiagonal one, after the reduction step. We will compare these diagonal elements with the real eigenvalues, which are Stewart's devil's stairs. In the following £gure one can see 10 stairs, with gaps of order 50 between the stairs, all the blocks are of size 10.

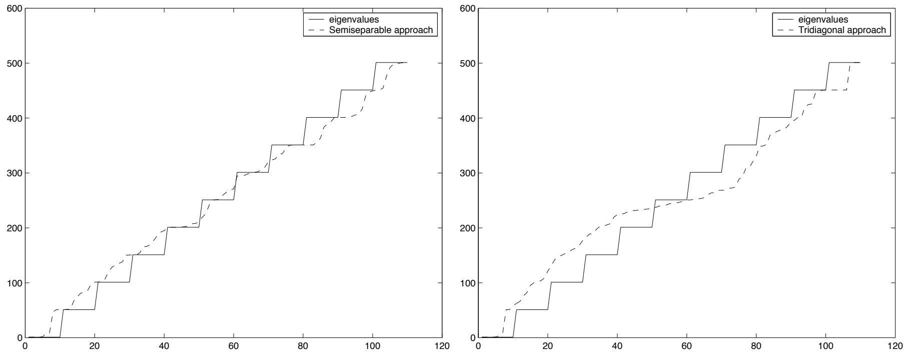

Figure 6: Stewart's Devil's stairs

As explained in [29] it can be seen that the devil's stairs are approximated much better by the diagonal elements of the semiseparable than by the diagonal elements of the tridiagonal matrix.

6.4 Problem matrices

The following diﬃcult matrices can be found in [8]. We take some of the eigenvalue problems which the traditional *QR* algorithm cannot solve. We consider the following matrix: $A = DPD$, where $D = diag(10^{20}, 10^{10}, 1)$,

$$A = \begin{pmatrix} 10^{40} & 10^{29} & 10^{19} \\ 10^{29} & 10^{20} & 10^{9} \\ 10^{19} & 10^{9} & 1 \end{pmatrix} \text{ and } P = \begin{pmatrix} 1 & 0.1 & 0.1 \\ 0.1 & 1 & 0.1 \\ 0.1 & 0.1 & 1 \end{pmatrix}.$$

The eigenvalues of the matrix A are the following:

$$\Lambda = [1.00000 \cdot 10^{40}, 9.90000 \cdot 10^{19}, 9.81818 \cdot 10^{-1}].$$

The eigenvalues computed by the routine $eig(\cdot)$ in matlab gives the following results:

$$[-3.85544 \cdot 10^{23}, 9.90002 \cdot 10^{-1}, 1.00000 \cdot 10^{40}].$$

One can see that the eigenvalue solver of Matlab, was only able to calculate one eigenvalue correctly. The eigenvalues computed by the semiseparable procedure are the following:

$$[1.00000 \cdot 10^{40}, 9.81818 \cdot 10^{-1}, 9.90000 \cdot 10^{19}]$$

all these eigenvalues are correct at least up to six digits.

The next matrix we will investigate is constructed in a similar way as the previous one, suppose we have $D = diag(10^{20}, 10^{10}, 1), A = DPD, \mu = 10^{(-6)}$,

$$A = \begin{pmatrix} 10^{40} & 9.99 \cdot 10^{29} & 9.99 \cdot 10^{19} \\ 9.99 \cdot 10^{29} & 10^{20} & 9.99 \cdot 10^{9} \\ 9.99 \cdot 10^{19} & 9.99 \cdot 10^{9} & 1 \end{pmatrix} \text{ and } P = \begin{pmatrix} 1 & 1-\mu & 1-\mu \\ 1-\mu & 1 & 1-\mu \\ 1-\mu & 1-\mu & 1 \end{pmatrix}.$$

The eigenvalues of this matrix are the following: $\Lambda = [10^{40}, 2 \cdot 10^{14}, 1.5 \cdot 10^{-6}]$. The eigenvalue solver of matlab cannot solve this problem. The implicit *QR* algorithm for semiseparable matrices gives the following eigenvalues.

$$[1.000000000000000 \cdot 10^{40}, 1.999999000102929 \cdot 10^{14}, 1.499999749837038 \cdot 10^{-06}]$$

these results are correct up to six digits.

These results show immediately, that although the algorithm is slower, it performs in several cases much better than the traditional *QR* with tridiagonal matrices. In some cases it can be seen that the *QR* steps do not need to be performed anymore, because the reduction already reveals all the information.

6.5 Cutting off the last eigenvalue

In the next £gure (7), we compared the accuracy of the eigenvalues, depending on the cutting off criterion. A sequence of matrices of varying size was generated, with equal spaced eigenvalues in the interval $[0, 1]$. The eigenvalues for these matrices were calculated by using a cutting off level at $10^{(-8)}$ and $10^{(-16)}$. For both these tests the absolute error of the residuals was computed and plotted in the next £gure. One can clearly see, that applying the aggressive de¤ation criterion is almost as good as the normal de¤ation criterion.

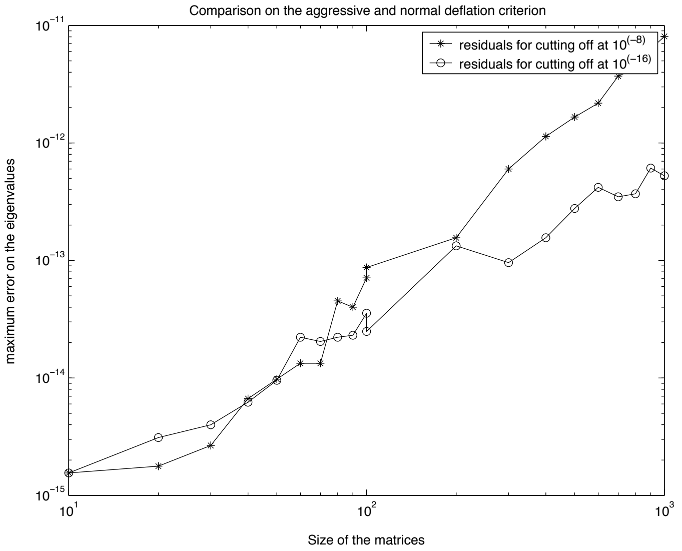

Figure 7: Comparing different de¤ation criteria

7 Conclusions

In this paper, we have designed an implicit *QR* algorithm to compute the eigendecomposition of symmetric semiseparable matrices. To get accurate results in £nite precision arithmetic, we have indicated the importance of using a suitable de£nition and corresponding representation of the class of semiseparable matrices. We have explained the main ideas behind our Matlab implementation which can be downloaded from http://www.cs.kuleuven.ac.be/∼marc/. In the near future we plan to make an implementation in C++.

References

[1] W. Barrett and P. Feinsilver. Inverses of banded matrices. *Linear Algebra and Its Applications*, 41:111–130, 1981.

[2] Dario Bini and Luca Gemignani. Iteration schemes for the divide-and-conquer eigenvalue solver. *Numerische Mathematik*, 67:403–425, 1994.

[3] C.F. Borges and W.B. Gragg. A parallel divide and conquer algorithm for the generalized real symmetric de£nite tridiagonal eigenproblem, 1992.

[4] S. Chandrasekaran and M. Gu. A divide and conquer algorithm for the eigendecomposition of symmetric block-diagonal plus semi-separable matrices. *submitted for publication*, 2001.

[5] D. Colton and R. Kress. *Integral equation methods in scattering theory.* Wiley, New York, 1983.

[6] D. Colton and R. Kress. *Inverse acoustic and electromagnetic scattering theory*. Springer-Verlag, second edition, 1998.

[7] J. J. M. Cuppen. A divide and conquer method for the symmetric tridiagonal eigenproblem. *Numerische Mathematik*, 36:177–195, 1981.

[8] J. Demmel and K. Veselić. Jacobi's method is more accurate than *QR*. *SIAM Journal on Matrix Analysis and its Applications*, 13(4):1204–1245, 1992.

[9] P. Dewilde and A.-J. van der Veen. *Time-varying systems and computations*. Kluwer academic publishers, Boston, June 1998.

[10] I. S. Dhillon. Reliable computation of the condition number of a tridiagonal matrix in $o(n)$ time. *SIAM Journal on Matrix Analysis and its Applications*, 19(3):776–796, 1998.

[11] D. Fasino and L. Gemignani. Direct and inverse eigenvalue problems, for diagonal-plus-semiseparable matrices, 2001. Submitted for publication.

[12] D. Fasino and L. Gemignani. A lanczos type algorithm for the *QR* factorization of regular cauchy matrices. *Numerical Linear Algebra with Applications*, 9:305–319, 2002.

[13] D. Fasino, N. Mastronardi, and M. Van Barel. Fast and stable algorithms for reducing diagonal plus semi-separable matrices to tridiagonal and bidiagonal form. *Contemporary Mathematics*, 323:105–119, 2003.

[14] F. P. Gantmacher and M. G. Krein. *Oscillation matrices and kernels and small vibrations of mechanical systems*. AMS Chelsea Publishing. AMS, revised edition, July 2002.

[15] I. Gohberg and M. A. Kaashoek. Time varying linear systems with boundary conditions and integral operators, I. the transfer operator and its properties. *Integral equations and Operator theory*, 7:325–391, 1984.

[16] G. H. Golub and C. F. Van Loan. *Matrix Computations, third edition*. The Johns Hopkins University Press, 1996.

[17] F. A. Graybill. *Matrices with applications in statistics*. Thomson Brooks/Cole, Colorado State University, second edition, 2002.

[18] L. Greengard and V. Rokhlin. On the numerical solution of two-point boundary value problems. *Comm. Pure Appl. Math*, 44:419–452, 1991.

[19] A. Kavció and J. M. F. Moura. Matrices with banded inverses: inversion algorithms and factorization of Gauss-Markov processes. *IEEE Transactions on Information Theory*, 46:1495–1509, 2000.

[20] I. Koltracht, R. A. Gonzales, J. Eisert, M. Neumann, and G. Rawitscher. Integral equation method for the continuous spectrum radial Schroedinger equation. *J. Comput. Phys.*, 134:134–139, 1997.

[21] J.-Y. Lee and L. Greengard. A fast adaptive numerical method for stiff two-point boundary value problems. *SIAM Journal on Scienti£c Computing*, 18(2):403–429, 1997.

[22] N. Mastronardi, S. Chandrasekaran, and S. Van Huffel. Fast and stable algorithms for reducing diagonal plus semiseparable matrices to tridiagonal and bidiagonal form. *BIT*, 41(1):149–157, 2003.

[23] B. N. Parlett. *The symmetric eigenvalue problem*. Prentice-Hall, Englewood Cliffs, N.J. 07632, 1980.

[24] F. Romani, P. Rozsa, R. Bevilacqua, and P. Favati. On band matrices and their inverses. *Linear Algebra and Its Applications*, 150:287–296, 1991.

[25] P. Rozsa. Band matrices and semi-separable matrices. *Colloquia Mathematica Societatis Janos Bolyai*, 50:229–237, 1986.

[26] P. Rozsa. On the inverse of band matrices. *Integral equations and operator theory*, 10:82–95, 1987.

[27] H. P. Jr. Starr. *On the numerical solution of one-dimensional integral and differential equations*. PhD thesis, Yale university, Department of Computer Science, May 1992.

[28] G. W. Stewart. *Matrix Algorithms, Vol II Eigensystems*. SIAM, 1999.

[29] Marc Van Barel, Raf Vandebril, and Nicola Mastronardi. The Lanczos-Ritz values appearing in an orthogonal similarity reduction of a matrix into semiseparable form. Report TW 360, Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Leuven (Heverlee), May 2003.

[30] E. Van Camp, N. Mastronardi, and M. Van Barel. A fast QR solver for diagonal plus semiseparable linear systems. Technical Report 7 (5/2003), Istituto per le Applicazioni del Calcolo "M. Picone", Consiglio Nazionale delle Ricerche, Bari, Italy, 2002.

[31] R. Vandebril, M. Van Barel, and N. Mastronardi. An algorithm for computing the singular values, via semiseparable matrices. Technical report, Dept. Computer Science, Celestijnenlaan 300A, 3001 Heverlee (Leuven), 2003.

[32] R. Vandebril, M. Van Barel, and N. Mastronardi. A note on the representation of semiseparable matrices. In preparation, 2003.