An orthogonal similarity reduction of a matrix into semiseparable form

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

An algorithm to reduce a symmetric matrix to a similar semiseparable one of semiseparability rank 1, using orthogonal similarity transformations, is proposed in this paper.

It is shown that, while running to completion, the proposed algorithm gives information on the spectrum of the similar initial matrix. In fact, the proposed algorithm shares the same properties of the Lanczos method and the Householder reduction to tridiagonal form.

Furthermore, at each iteration, the proposed algorithm performs a step of the QR method without shift to a principal submatrix to retrieve the semiseparable structure. The latter step can be considered a kind of subspace-like iteration method, where the size of the subspace increases by one dimension at each step of the algorithm. Hence, when during the execution of the algorithm the Ritz values approximate the dominant eigenvalues closely enough, diagonal blocks will appear in the semiseparable part where the norm of the corresponding subdiagonal blocks goes to zero in the subsequent iteration steps, depending on the corresponding gap between the eigenvalues.

A numerical experiment is included, illustrating the properties of the new algorithm.

Article information

- Van Barel, Marc; Vandebril, Raf; Mastronardi, Nicola. *An orthogonal similarity reduction of a matrix into semiseparable form*, SIAM Journal on Matrix Analysis and Applications, volume 27, issue 1, pages 176-197, 2005.
- 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/S0895479803428229>
- KU Leuven's repository url: <https://lirias.kuleuven.be/handle/123456789/124558>

AN ORTHOGONAL SIMILARITY REDUCTION OF A MATRIX INTO SEMISEPARABLE FORM*

M. VAN BAREL[†], R. VANDEBRIL[†], AND N. MASTRONARDI[‡]

Abstract. An algorithm to reduce a symmetric matrix to a similar semiseparable one of semiseparability rank 1, using orthogonal similarity transformations, is proposed in this paper.

It is shown that, while running to completion, the proposed algorithm gives information on the spectrum of the similar initial matrix. In fact, the proposed algorithm shares the same properties of the Lanczos method and the Householder reduction to tridiagonal form.

Furthermore, at each iteration, the proposed algorithm performs a step of the QR method without shift to a principal submatrix to retrieve the semiseparable structure. The latter step can be considered a kind of subspace-like iteration method, where the size of the subspace increases by one dimension at each step of the algorithm. Hence, when during the execution of the algorithm the Ritz values approximate the dominant eigenvalues closely enough, diagonal blocks will appear in the semiseparable part where the norm of the corresponding subdiagonal blocks goes to zero in the subsequent iteration steps, depending on the corresponding gap between the eigenvalues.

A numerical experiment is included, illustrating the properties of the new algorithm.

Key words. similarity transformation, semiseparable matrix, Lanczos algorithm, Ritz values

AMS subject classifications. 15A18, 15A21, 65F15

DOI. 10.1137/S0895479803428229

1. Introduction. Numerical linear algebra problems involving semiseparable matrices have raised much interest in the last decade. In particular, some fast and stable algorithms to compute the eigendecomposition of symmetric diagonal plus semiseparable matrices have been developed [2, 3, 6, 11].

In this paper an algorithm that transforms a symmetric matrix $A \in \mathbb{R}^{n \times n}$ into a similar symmetric semiseparable one by means of orthogonal transformations with $\frac{4}{3}n^3 + \mathcal{O}(n^2)$ flops¹ is proposed. Hence, the latter algorithm, combined with one of those for computing the eigendecomposition of semiseparable matrices, could give an alternative way to compute the eigenvalue decomposition of symmetric matrices.

There are excellent algorithms that compute the complete eigendecomposition of a dense symmetric matrix in the literature. The standard procedure to compute

*Received by the editors May 22, 2003; accepted for publication (in revised form) by D. A. Bini October 8, 2004; published electronically July 13, 2005.

<http://www.siam.org/journals/simax/27-1/42822.html>

[†]Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Leuven (Heverlee), Belgium (Raf.Vandebril@cs.kuleuven.ac.be, Marc.VanBarel@cs.kuleuven.ac.be). The research of the first two authors 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 (AN-CILA: Asymptotic Analysis of the Convergence Behavior of Iterative Methods in Numerical Linear Algebra), and G.0184.02 (CORFU: Constructive Study of Orthogonal Rational Functions); by the K.U. Leuven (Bijzonder Onderzoeksfonds); 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 scientific responsibility rests with the authors.

[‡]Istituto per le Applicazioni del Calcolo “M. Picone,” sez. Bari, Consiglio Nazionale delle Ricerche, Via G. Amendola, 122/I, I-70126 Bari, Italy (irmanm21@area.ba.cnr.it). The research of this author was partially supported by MIUR, grant 2002014121.

¹In this paper the definition given in [7] is adopted: “A flop is a floating point operation.” The square root is considered a flop, too.

the complete eigendecomposition of a dense symmetric matrix first reduces it to a similar symmetric tridiagonal one by means of orthogonal transformations. This step is accomplished in $\frac{4}{3}n^3 + \mathcal{O}(n^2)$ flops. Once the similar symmetric tridiagonal matrix has been computed, many fast and stable algorithms can be considered to compute the spectral decomposition of the latter matrix (see, for instance, [7, 15, 13, 5, 9, 12] and the references therein).

The computation of the spectral decomposition of dense symmetric matrices is beyond the scope of this paper. However, the proposed algorithm for reducing a symmetric matrix into a semiseparable one, while running to completion, gives information on the spectrum of the similar initial matrix. In fact, similarly to the Lanczos method [5, 7] and the Householder reduction to tridiagonal form [10], [7, pp. 472–473], after k steps of the algorithm the eigenvalues of the partially reduced matrix are the Ritz values of the original matrix with respect to the Krylov subspace $\{e_n, Ae_n, \dots, A^{k-1}e_n\}$, where e_n is the last vector of the canonical basis of $\mathbb{R}^n$.

Furthermore, at the k th iteration, the proposed algorithm performs a step of the QR method without shift to the principal submatrix of order $k+1$ to retrieve the semiseparable structure. The latter step can be considered a kind of subspace-like iteration method, where the size of the subspace increases by one dimension at each step of the algorithm. Hence, when, during the execution of the algorithm, the dominant eigenvalues are approximated sufficiently well by the Lanczos–Ritz values, diagonal blocks will appear in the semiseparable part. The norm of the corresponding subdiagonal blocks will go to zero depending on the corresponding gap between the eigenvalues. Therefore it turns out that the proposed algorithm could be chosen when an approximation of the subspace associated to the largest eigenvalues of large dense matrices is required. Such problems arise in many different areas, such as principal component analysis, data mining, magnetic resonance spectroscopy, microarray data analysis, gene expression data, and computation of the eigensolutions of the Schrödinger equation on a grid (see, for instance, [1, 8, 20, 21, 24, 25] and the references therein).

The paper is organized as follows. In section 2 the definition and possible stable representations of semiseparable matrices are discussed. The algorithm for reducing symmetric matrices to similar symmetric semiseparable ones is described in section 3. The unsymmetric case is considered briefly in section 4. In section 5 the computational complexity is discussed. It also contains an algorithm for reducing symmetric tridiagonal matrices into similar symmetric semiseparable ones. The strong connection with the Lanczos–Ritz values is investigated in section 6. Convergence properties of the subspace iteration are studied in section 7. A numerical experiment is shown in section 8, followed by the conclusions.

2. Definition and representation of semiseparable matrices. In the literature the class of semiseparable matrices is defined in slightly different ways. For an overview of the various definitions, representations, and properties, we refer the interested reader to [19, 14]. In this paper, semiseparable matrices are defined in the following way.

DEFINITION 2.1. *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 .*

In the remainder of the paper, we will consider only symmetric semiseparable matrices of (lower- and upper-)semiseparability rank 1, referring to them as semiseparable matrices.

In [19], it is shown that under weak assumptions, the matrix S can be represented by two (column-)vectors u and v , called generators of the matrix S :

$$S = \text{triu}(uv^T) + \text{tril}(vu^T),$$

where $\text{triu}(A)$ and $\text{tril}(A)$ denote, respectively, the strictly upper triangular part and the lower triangular part of the matrix A .

THEOREM 2.2. *Suppose S is a symmetric semiseparable matrix of size n . Then we have the following equivalence:*

S can be represented by two generators u and v

$$(2.1) \quad \Updownarrow$$

for each $1 \leq j \leq i \leq n$ such that $S(i, j) = 0 \Rightarrow S(i, 1 : i) = 0$ or $S(j : n, j) = 0$.

Proof. The proof can be found in [19]. $\square$

Because the representation with the generators is cheap in terms of memory usage and it is easy to reconstruct arbitrary elements within the matrix, several algorithms for semiseparable matrices are based on this representation. For several of these algorithms, this representation can lead to a big loss of accuracy when performing the computations in finite precision. This can be seen from the following example.

Example 1. Suppose we have a symmetric 5×5 matrix with the following eigenvalues: $(1, 2, 3, 100, 10^4)$. Constructing a semiseparable matrix from it (the algorithm used to do this will be explained later on) generates the following matrix:

$$S = \begin{pmatrix} 1.6254 & -5.0812 \cdot 10^{-1} & -8.0163 \cdot 10^{-2} & -6.3532 \cdot 10^{-5} & 5.2384 \cdot 10^{-11} \\ -5.0812 \cdot 10^{-1} & 1.4259 & 2.2496 \cdot 10^{-1} & 1.7829 \cdot 10^{-4} & -1.4597 \cdot 10^{-10} \\ -8.0163 \cdot 10^{-2} & 2.2496 \cdot 10^{-1} & 2.9485 & 2.3368 \cdot 10^{-3} & -1.9152 \cdot 10^{-9} \\ -6.3532 \cdot 10^{-5} & 1.7829 \cdot 10^{-4} & 2.3368 \cdot 10^{-3} & 9.9999 \cdot 10^1 & -8.1957 \cdot 10^{-5} \\ 5.2384 \cdot 10^{-11} & -1.4597 \cdot 10^{-10} & -1.9152 \cdot 10^{-9} & -8.1957 \cdot 10^{-5} & 1.0000 \cdot 10^4 \end{pmatrix}.$$

Although this matrix is semiseparable, it can clearly be seen that the lower right element has already converged to the largest eigenvalue. Representing this matrix now with the generators u and v gives us the following vectors: Even though this matrix is semiseparable, it can clearly be seen that the last entry of the diagonal already approximates the largest eigenvalue. Representing this matrix with the traditional generators u and v gives us the following vectors:

$$u = \begin{pmatrix} 3.1029 \cdot 10^{10} & -9.6998 \cdot 10^9 & -1.5302 \cdot 10^9 & -1.2128 \cdot 10^6 & 1.0000 \end{pmatrix}^T$$

and

$$v = \begin{pmatrix} 5.2384 \cdot 10^{-11} & -1.4597 \cdot 10^{-10} & -1.9152 \cdot 10^{-9} & -8.1957 \cdot 10^{-5} & 1.0000 \cdot 10^4 \end{pmatrix}^T.$$

Because the first element of v is of the order 10^{-11} and is constructed by summations of elements of order 1, we can expect that this element has a precision of only 5 significant decimal digits left. Using this number to reconstruct the elements within the matrix will give these elements with only a limited number of exact digits.

To overcome this problem we consider another representation. For a semiseparable matrix of dimension n , this representation consists of $n - 1$ Givens transformations and a vector of length n . The Givens transformations are denoted as $G = [G_1, \dots, G_{n-1}]$ and the vector as $d = [d_1, \dots, d_n]$.

The following figures denote how the semiseparable matrix can be reconstructed using this information. The elements denoted by $\boxtimes$ make up the semiseparable part of the matrix. The construction of the semiseparable matrix is for a matrix of size 5. Initially one starts on the first two 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.

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

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

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

This process can be repeated by applying the Givens transformation G_3 on the third and the fourth rows 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, the upper triangular part is also known.

Suppose the Givens and vector representation of a semiseparable matrix S is known. Let us denote the Givens transformations by

$$(2.2) \quad G_l = \begin{pmatrix} c_l & -s_l \\ s_l & c_l \end{pmatrix}.$$

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, due to the symmetry. The elements of the semiseparable matrix can therefore be calculated in a stable way.

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

$$G = \begin{pmatrix} 0.9534 & 0.9878 & 1.0000 & 1.0000 \\ -0.3017 & 0.1558 & 0.0008 & 0.0000 \end{pmatrix}$$

and

$$d = (1.7049 \quad 1.4435 \quad 2.9485 \quad 9.9999 \cdot 10 \quad 1.0000 \cdot 10^4).$$

Moreover this representation has another advantage connected to the QR factorization of this semiseparable matrix. Let $\hat{G}_j$ represent the Givens transformation G_j embedded in an identity matrix of size $n \times n$ such that the Givens transformation G_j is executed on rows j and $j + 1$. Then, as was proved in [17], the QR factorization of S can be written as

$$\begin{aligned} S &= \hat{G}_{n-1} \cdots \hat{G}_2 \hat{G}_1 R \\ &= QR \end{aligned}$$

with R upper triangular.

3. Orthogonal similarity transformation to reduce a symmetric matrix to a semiseparable one.

3.1. Theory. An algorithm to transform a symmetric matrix into a semiseparable one using an orthogonal similarity transformation is presented in this section. The constructive proof of the next theorem provides the algorithm.

THEOREM 3.1. *Let A be a symmetric matrix. Then there exists an orthogonal matrix U such that*

$$(3.1) \quad U^T A U = S,$$

where S is a semiseparable matrix.

Proof. The proof is a constructive one and is made by induction on the rows of matrix A . We will construct a similar symmetric semiseparable matrix from a symmetric one. Let $A_0^{(0)} \equiv A$. Let $G_i^{(k)}$ be the Givens transformation, such that the product $A_{i-1}^{(k)} G_i^{(k)}$ has the entry $(n-k, i)$ annihilated, modifying the i th and the $(i+1)$ th columns of $A_{i-1}^{(k)}$.

Step 1. We will start by constructing a similarity transformation which will make the last two rows (columns) linearly dependent in the lower (upper) triangular part. To this end, we multiply $A_0^{(0)}$ to the left by $G_1^{(0)T}$ and to the right by $G_1^{(0)}$ to annihilate the elements in positions $(1, n)$ and $(n, 1)$ in $A_0^{(0)}$, respectively:

$$\begin{array}{c} \downarrow \quad \downarrow \\ \rightarrow \left(\begin{array}{ccccc} \times & \times & \cdots & \times & \otimes \\ \times & \ddots & \vdots & \vdots & \times \\ \cdots & \cdots & \ddots & \vdots & \vdots \\ \times & \cdots & \cdots & \ddots & \times \\ \otimes & \times & \cdots & \times & \times \end{array} \right) \xrightarrow{G_1^{(0)T} A_0^{(0)} G_1^{(0)}} \left(\begin{array}{ccccc} \times & \times & \cdots & \times & 0 \\ \times & \ddots & \vdots & \vdots & \times \\ \cdots & \cdots & \ddots & \vdots & \vdots \\ \times & \cdots & \cdots & \ddots & \times \\ 0 & \times & \cdots & \times & \times \end{array} \right) \\ \Downarrow \\ A_0^{(0)} \xrightarrow{G_1^{(0)T} A_0^{(0)} G_1^{(0)}} A_1^{(0)}. \end{array}$$

Note that $\times$ denotes arbitrary elements of the matrix, while $\otimes$ denotes the elements which are annihilated by the orthogonal similarity transformation. Summarizing, we obtain that $A_1^{(0)} \equiv G_1^{(0)T} A_0^{(0)} G_1^{(0)}$.

Continuing this process of annihilating all the elements in the last row (column), except for the element in position $(n, n-1)$ ($(n-1, n)$), we get

$$A_{n-2}^{(0)} = \left(\begin{array}{ccccc} \times & \times & \cdots & \times & 0 \\ \times & \ddots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \ddots & \times & 0 \\ \times & \cdots & \times & \times & \times \\ 0 & \cdots & 0 & \times & \times \end{array} \right).$$

Multiplying $A_{n-2}^{(0)}$ to the left by $G_{n-1}^{(0)T}$, we have the following situation:

$$\begin{aligned} & \begin{matrix} \rightarrow \\ \rightarrow \end{matrix} \begin{pmatrix} \times & \times & \cdots & \times & 0 \\ \times & \ddots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \ddots & \vdots & 0 \\ \times & \cdots & \cdots & \times & \otimes \\ 0 & \cdots & 0 & \times & \times \end{pmatrix} \xrightarrow{G_{n-1}^{(0)T} A_{n-2}^{(0)}} \begin{pmatrix} \times & \times & \cdots & \times & 0 \\ \times & \ddots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \ddots & \times & 0 \\ \boxtimes & \cdots & \boxtimes & \times & 0 \\ \boxtimes & \cdots & \boxtimes & \times & \times \end{pmatrix} \\ & \quad \quad \quad \Updownarrow \\ & A_{n-2}^{(0)} \xrightarrow{G_{n-1}^{(0)T} A_{n-2}^{(0)}} \tilde{A}_{n-2}^{(0)}; \end{aligned}$$

i.e., the last two rows are proportional with the exception of the entries in the last two columns. (To emphasize the semiseparable structure among the rows (columns) we denote by $\boxtimes$ the elements of the matrix belonging to these rows (columns).) Let $\tilde{A}_{n-2}^{(0)} \equiv G_{n-1}^{(0)T} A_{n-2}^{(0)}$. Moreover, multiplying $\tilde{A}_{n-2}^{(0)}$ to the right by $G_{n-1}^{(0)}$, the last two columns become linearly dependent above and on the main diagonal; i.e., they form an uppersemiseparable part. Because of symmetry, the last two rows become linearly dependent below and on the main diagonal and form a lowersemiseparable part:

$$\begin{aligned} & \begin{matrix} \downarrow & \downarrow \\ \begin{pmatrix} \times & \times & \cdots & \times & 0 \\ \times & \ddots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \ddots & \vdots & 0 \\ \boxtimes & \cdots & \boxtimes & \times & 0 \\ \boxtimes & \cdots & \boxtimes & \times & \times \end{pmatrix} \end{matrix} \xrightarrow{\tilde{A}_{n-2}^{(0)} G_{n-1}^{(0)}} \begin{pmatrix} \times & \times & \cdots & \boxtimes & \boxtimes \\ \times & \ddots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \ddots & \boxtimes & \boxtimes \\ \boxtimes & \cdots & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \cdots & \boxtimes & \boxtimes & \boxtimes \end{pmatrix} \\ & \quad \quad \quad \Updownarrow \\ & \tilde{A}_{n-2}^{(0)} \xrightarrow{\tilde{A}_{n-2}^{(0)} G_{n-1}^{(0)}} A_{n-1}^{(0)}. \end{aligned}$$

To start the next step, $A_0^{(1)}$ is defined as $A_0^{(1)} \equiv A_{n-1}^{(0)}$.

Step 2. Let $m = n - j$, $1 \leq j < n$. Assume by induction that $A_0^{(m)}$ has the lower (upper) triangular part already semiseparable from row n up to row j (column j to n). We will now prove that we can make the lower (upper) triangular part semiseparable up to row $j - 1$ (column $j - 1$). Let us denote the lower (and upper) triangular elements which form a semiseparable part with $\boxtimes$. So matrix $A_0^{(m)}$ looks like

$$A_0^{(m)} = \begin{pmatrix} \times & \cdots & \times & \boxtimes & \cdots & \boxtimes \\ \vdots & \ddots & \cdots & \vdots & \vdots & \vdots \\ \times & \cdots & \times & \boxtimes & \cdots & \boxtimes \\ \boxtimes & \cdots & \boxtimes & \boxtimes & \cdots & \boxtimes \\ \vdots & \cdots & \cdots & \cdots & \ddots & \vdots \\ \boxtimes & \cdots & \boxtimes & \boxtimes & \cdots & \boxtimes \end{pmatrix} \begin{matrix} \leftarrow 1 \\ \vdots \\ \leftarrow j-1 \\ \leftarrow j \\ \vdots \\ \leftarrow n \end{matrix}.$$

Because the submatrix of $A_0^{(m)}$ consisting of the first $j - 1$ rows and the last $n - j + 1$ columns is already semiseparable, it has rank ≤ 1 . Hence, we can construct

$$\begin{array}{ccc}
\begin{array}{c} j-1 \rightarrow \\ j \rightarrow \end{array} \begin{pmatrix} \times & \times & \cdots & \times & \times & 0 & 0 & \cdots & 0 \\ \times & \times & \cdots & \times & \times & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \times & \times & \cdots & \times & \times & 0 & 0 & \cdots & 0 \\ \times & \times & \cdots & \times & \times & \boxtimes & \boxtimes & \cdots & \boxtimes \\ 0 & \cdots & 0 & 0 & \boxtimes & \boxtimes & \boxtimes & \cdots & \boxtimes \\ 0 & \cdots & 0 & 0 & \boxtimes & \boxtimes & \boxtimes & \cdots & \boxtimes \end{pmatrix} & \xrightarrow{G_{j-1}^{(m)T} A_{j-2}^{(m)}} & \begin{pmatrix} & & & j-1 & j & & & & \\ & & & \downarrow & \downarrow & & & & \\ \times & \times & \cdots & \times & \times & 0 & 0 & \cdots & 0 \\ \times & \times & \cdots & \times & \times & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \times & \times & \cdots & \times & \times & 0 & 0 & \cdots & 0 \\ \boxtimes & \cdots & \boxtimes & \boxtimes & \boxtimes & 0 & 0 & \cdots & 0 \\ \boxtimes & \cdots & \boxtimes & \boxtimes & \boxtimes & 0 & 0 & \cdots & 0 \\ \boxtimes & \cdots & \boxtimes & \boxtimes & \boxtimes & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \boxtimes & \boxtimes & \boxtimes & \cdots & \boxtimes \end{pmatrix} \\
& & \xrightarrow{G_{j-1}^{(m)T} A_{j-2}^{(m)} G_{j-1}^{(m)}} & \begin{pmatrix} \times & \times & \cdots & \times & \boxtimes & \boxtimes & 0 & \cdots & 0 \\ \times & \times & \cdots & \times & \boxtimes & \boxtimes & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \times & \times & \cdots & \times & \boxtimes & \boxtimes & 0 & \cdots & 0 \\ \boxtimes & \cdots & \boxtimes & \boxtimes & \boxtimes & \boxtimes & 0 & \cdots & 0 \\ \boxtimes & \cdots & \boxtimes & \boxtimes & \boxtimes & \boxtimes & 0 & \cdots & 0 \\ \boxtimes & \cdots & 0 & 0 & 0 & \boxtimes & \boxtimes & \cdots & \boxtimes \\ 0 & \cdots & 0 & 0 & 0 & \boxtimes & \boxtimes & \cdots & \boxtimes \end{pmatrix}.
\end{array}$$

FIG. 3.1. The transformation $A_{j-1}^{(m)} \equiv G_{j-1}^{(m)T} A_{j-2}^{(m)} G_{j-1}^{(m)}$ makes the first $j-1$ entries of rows (columns) $j-1$ and j proportional.

a Givens transformation $G_1^{(m)}$ such that when we multiply $A_0^{(m)}$ to the left by $G_1^{(m)T}$ all elements $(1, j)$, $(1, j+1)$ up to $(1, n)$ of the first row are annihilated. In a similar way, we can construct the Givens transformations $G_2^{(m)} \cdots G_{j-2}^{(m)}$ which make zero elements in columns $j, j+1, \dots, n$ in rows 2 up to $j-2$. Applying these similarity transformations, we obtain the following matrix:

$$A_{j-2}^{(m)} \equiv G_{j-2}^{(m)T} \cdots G_1^{(m)T} A_0^{(m)} G_1^{(m)} \cdots G_{j-2}^{(m)} = \begin{pmatrix} \times & \cdots & \times & \times & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \times & \cdots & \times & \times & 0 & \vdots & 0 \\ \times & \cdots & \times & \times & \boxtimes & \cdots & \boxtimes \\ 0 & \cdots & 0 & \boxtimes & \boxtimes & \cdots & \boxtimes \\ \vdots & \cdots & \vdots & \cdots & \cdots & \ddots & \vdots \\ 0 & \cdots & 0 & \boxtimes & \boxtimes & \cdots & \boxtimes \end{pmatrix} \begin{array}{l} \leftarrow 1 \\ \vdots \\ \vdots \\ \leftarrow j-1 \\ \leftarrow j \\ \vdots \\ \leftarrow n \end{array}.$$

Since rows $j-1$ and j are proportional for the indexes of columns greater than $j-1$, the Givens transformation $G_{j-1}^{(m)T}$ can also be constructed annihilating all the entries in row $j-1$ with column index greater than $j-1$. Furthermore, the product of $G_{j-1}^{(m)T} A_{j-2}^{(m)}$ to the right by $G_{j-1}^{(m)}$ makes columns $j-1$ and j proportional in the first $j-1$ entries. The latter similarity transformation is depicted in Figure 3.1.

To retrieve the semiseparable structure, below the j th row and to the right of the j th column, $n-j$ more Givens similarity transformations $G_k^{(m)}$, $k = j, j+1, \dots, n-1$, are considered. Each of these is chosen such that when the Givens transformation $G_k^{(m)T}$ is first applied to the left, the elements $k+1, k+2, \dots, n$ are annihilated in row k , using the corresponding elements of row $k+1$. To obtain a similar matrix, we apply $G_k^{(m)}$ also on the right, adding one more row (column) to the semiseparable part. The whole process is summarized in Figure 3.2. $\square$

FIG. 3.2. *Description of the similarity transformations used to retrieve the semiseparable structure.*

In the next subsection, details are given on how to implement this algorithm using the Givens-vector representation for the semiseparable part of the matrices involved. This will result in an $O(n^3)$ algorithm to compute the Givens-vector representation of a semiseparable matrix similar to the original symmetric matrix.

Once the semiseparable matrix has been computed, several algorithms can be considered to compute its eigenvalues (see, for instance, [2, 3, 6, 11]).

The proposed algorithm itself, however, while running to completion, already gives information on the spectrum of the initial matrix. In section 6 we will show that the eigenvalues of the subsequent bottom-right semiseparable matrices computed by the proposed algorithm are nothing else but the Ritz values obtained by the Lanczos method.

Moreover, the algorithm can be considered as a nested subspace iteration method, as shown in section 7, where in each step a QL step without shift is performed on the semiseparable part of the reduced matrix.

First, we look at a more detailed implementation of the algorithm in the next subsection. The computational complexity is discussed in section 5.

3.2. Effective implementation of the reduction. Now we describe how to implement the method above using the Givens-vector representation of a semiseparable matrix.

This representation gives information which can be used directly in the QR algorithm applied to the resulting semiseparable matrix [18].

The implementation involves detailed shuffling of indices; therefore only the mathematics behind the implementation is given. The MATLAB files can be downloaded from <http://www.cs.kuleuven.ac.be/~marc>. Following the mathematical details, one should be able to understand and/or reconstruct the algorithm.

Suppose rows $j, j+1, \dots, n$ (columns $j, j+1, \dots, n$) are already semiseparable. We can represent this semiseparable part by a row-vector $R^{(j)} \in \mathbb{R}^{1 \times (j-1)}$, Givens transformations

$$\begin{pmatrix} c_j & -s_j \\ s_j & c_j \end{pmatrix}, \quad k = j, j+1, \dots, n-1,$$

and a vector $[d_j, d_{j+1}, \dots, d_n]$. At this point, the matrix similar to the original symmetric matrix can be divided into four blocks:

$$(3.2) \quad A_0^{(m)} = \left(\begin{array}{cc|ccc} A^{(j)} & c_j R_j^T & c_{j+1} s_j R_j & \cdots & s_{n-1} \dots s_j R_j^T \\ c_j R_j & c_j d_j & c_{j+1} s_j d_j & \cdots & s_{n-1} \dots s_j d_j \\ \hline c_{j+1} s_j R_j & c_{j+1} s_j d_j & c_{j+1} d_{j+1} & & \\ \vdots & & & \ddots & \vdots \\ s_{n-1} \dots s_j R_j & & & \cdots & d_n \end{array} \right),$$

with $m = n - j$ and $A^{(j)} \in \mathbb{R}^{(j-1) \times (j-1)}$.

The first substeps in step m of the method described in the proof of Theorem 3.1 eliminate the elements $1, 2, \dots, j-2$ in row j by multiplying $A_0^{(m)}$ to the right by the Givens transformations $G_1^{(m)}, G_2^{(m)}, \dots, G_{j-2}^{(m)}$, i.e.,

$$R_j G_1^{(m)} G_2^{(m)} \dots G_{j-2}^{(m)} = [0, \dots, 0, \tilde{\alpha}_j] = \tilde{R}_j.$$

It is clear that we can obtain a similar result by applying a Householder transformation $H^{(m)}$ on the row R_j such that the following equation is obtained:

$$R_j H^{(m)} = [0, \dots, 0, \alpha_j] = \hat{R}_j,$$

with $|\alpha_j| = |\tilde{\alpha}_j|$. Performing the similarity Householder transformation on the matrix (3.2) transforms this matrix into the following one:

$$(3.3) \quad \left(\begin{array}{cc|ccc} H^{(m)T} A^{(j)} H^{(m)} & c_j \hat{R}_j^T & c_{j+1} s_j \hat{R}_j^T & \cdots & s_{n-1} \dots s_j \hat{R}_j^T \\ c_j \hat{R}_j & c_j d_j & c_{j+1} s_j d_j & \cdots & s_{n-1} \dots s_j d_j \\ \hline c_{j+1} s_j \hat{R}_j & c_{j+1} s_j d_j & c_{j+1} d_{j+1} & & \vdots \\ \vdots & & & \ddots & \\ s_{n-1} \dots s_j \hat{R}_j & \cdots & & & d_n \end{array} \right).$$

From this point the Givens transformation $G_{j-1}^{(m)}$ can be calculated such that the following equation is satisfied:

$$[\alpha_j, d_j] G_{j-1}^{(m)} = [0, \alpha_{j+1}] \quad \text{with} \quad G_{j-1}^{(m)} = \begin{pmatrix} c_{j-1} & s_{j-1} \\ -s_{j-1} & c_{j-1} \end{pmatrix}.$$

The upper left part $H^{(m)T} A^{(j)} H^{(m)}$ can be written as

$$H^{(m)T} A^{(j)} H^{(m)} = \begin{pmatrix} A^{(j-1)} & R_{j-1}^T \\ R_{j-1} & d_{j-1} \end{pmatrix}.$$

After applying the similarity Givens transformation $G_{j-1}^{(m)}$ on the matrix (3.3), we get the following matrix:

$$(3.4) \quad \begin{pmatrix} A^{(j-1)} & c_{j-1}R_{j-1}^T & s_{j-1}R_{j-1}^T & 0 & \dots & 0 \\ c_{j-1}R_{j-1} & c_{j-1}\hat{d}_{j-1} & s_{j-1}\hat{d}_{j-1} & 0 & \dots & 0 \\ s_{j-1}R_{j-1} & s_{j-1}\hat{d}_{j-1} & c_j\alpha_{j+1} & c_{j+1}s_j\alpha_{j+1} & \dots & s_{n-1}\dots s_j\alpha_{j+1} \\ 0 & 0 & c_{j+1}s_j\alpha_{j+1} & c_{j+1}d_{j+1} & & \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & s_{n-1}\dots s_j\alpha_{j+1} & & \dots & d_n \end{pmatrix},$$

with $\hat{d}_{j-1} = \tilde{d}_{j-1}c_{j-1} + s_{j-1}c_j\alpha_j$. The Givens transformation $G_{j-1}^{(m)T}$ and the element $\hat{d}_{j-1}$ can already be stored as part of the representation of the new semiseparable part, which has to be formed at the bottom of the matrix.

The process explained here can now be repeated to make all rows $j-1, j$ up to n semiseparable, because the matrix (3.4) has essentially the same structure as the matrix (3.3).

Note. The latter step, i.e., the reduction of the lower right part towards the semiseparable structure, can also be applied directly on a semiseparable matrix. This corresponds to performing a QR step (in fact, a QL step) without shift on this semiseparable matrix [18], as will be shown in section 7.

4. The reduction of an arbitrary matrix. It is well known that any matrix can be transformed into an upper Hessenberg one by using an orthogonal similarity transformation. To find all the eigenvalues of the original matrix, the QR algorithm can then be applied to this upper Hessenberg matrix. By using an algorithm similar to the one described in the previous section, it is possible to transform every matrix into the sum of a semiseparable and a strictly upper triangular part. This is proved in the next theorem.

THEOREM 4.1. *Let A be an $n \times n$ matrix. Then there exist a symmetric semiseparable matrix S of semiseparability rank 1, a strictly upper triangular matrix R , and an orthogonal matrix U such that the following equation is satisfied:*

$$A = U(S + R)U^T;$$

i.e., A is similar to the sum of a symmetric semiseparable and a strictly upper triangular matrix R by means of orthogonal transformations.

Proof. Following the same reduction as in the proof of Theorem 3.1, we can find an orthogonal matrix U such that the lower triangular part of $U^T AU$ is semiseparable.

Hence, the lower triangular part of A can be written as the lower triangular part of a symmetric semiseparable matrix S , i.e.,

$$\begin{aligned} U^T AU &= \text{tril}(S) + \bar{R} \\ &= S + (\bar{R} - \text{triu}(S)) \\ &= S + R. \quad \square \end{aligned}$$

Note that the algorithm to reduce an arbitrary matrix by means of orthogonal transformations into a similar matrix whose lower triangular part is semiseparable also requires $\mathcal{O}(n^3)$ operations.

5. The computational complexity of each of the algorithms.

5.1. Reduction of a symmetric matrix into a similar semiseparable one.

When reducing a symmetric matrix to a similar semiseparable one, the implementation can be done using only Givens transformations or using Householder and Givens transformations. One can easily see that the complexity can be divided into two parts: first, there is the application of a Householder transformation or a number of Givens transformations on the upper left block of the matrix, and second, there is the updating of the semiseparable part. Executing the first part of all the steps requires the same number of operations as reducing a symmetric matrix into a similar tridiagonal one either by Householders or by Givens transformations. Hence, the computational complexity of this part is

$$\begin{aligned} &2n^3 + O(n^2) \text{ when using Givens transformations,} \\ &\frac{4}{3}n^3 + O(n^2) \text{ when using Householder transformations.} \end{aligned}$$

Executing the second part of all the steps requires $O(n^2)$ operations. More details can be found in the next subsection.

5.2. Reduction to a semiseparable matrix via a tridiagonal matrix.

The third approach to reduce a symmetric matrix to semiseparable form is the following. First, we reduce the matrix to a similar tridiagonal one by standard algorithms. (This can be achieved with Householder or Givens transformations; see, e.g., [7].) The latter tridiagonal matrix can then be reduced into semiseparable form. In this case each iteration step starts with a matrix already of the form (3.3). Hence, the first Householder or Givens transformations do not have to be performed. Note that this algorithm involves essentially the same orthogonal similarity transformations as the algorithms described before. Only the order in which they are applied is different. The results of sections 6 and 7 state, however, that it is worth intertwining the two reductions, instead of first reducing towards tridiagonal form and thereafter to a semiseparable matrix. Hence, the number of operations needed to reduce a tridiagonal into a semiseparable matrix is the same as for the second step in the reduction from a symmetric matrix to semiseparable form.

The calculation of the Givens requires 7 flops, as described in [7]. The multiplication on the left- and right-hand sides takes 11 operations, when everything is optimized. Therefore, the computational complexity is

$$\sum_{j=n:-1:2} \left(\sum_{i=j:1:n} 18 \right) = 9n^2 + O(n).$$

5.3. Reduction of a nonsymmetric matrix. As already mentioned, the algorithm applied to nonsymmetric matrices reduces the matrix to the sum of a semiseparable and a strictly upper triangular matrix. To implement this, similar algorithms to those mentioned above can be used. We now take a look at the complexity of this reduction. Let us assume that we first reduce the matrix to an upper Hessenberg one, and then apply the algorithm to make the lower triangular part semiseparable. (This is a similar algorithm to those applied to tridiagonal matrices.) Then we first have the reduction to Hessenberg form, which costs $5/3n^3 + O(n^2)$ flops. The reduction from Hessenberg to lower semiseparable form costs an extra $3n^3 + O(n^2)$ flops.

5.4. Comparison with the classical reduction algorithms reducing a symmetric matrix to tridiagonal form. When we compare the algorithm which tridiagonalizes a symmetric matrix to the algorithm which constructs a similar semiseparable matrix, we find that the latter algorithm performs only $9n^2 + O(n)$ more flops than the former algorithm, while both algorithms have an overall complexity of $O(n^3)$. It is worthwhile to perform these additional operations as is shown by the convergence properties as studied in the next sections.

6. Connection with the Lanczos–Ritz values. In this section, we show that at each step of the algorithm the eigenvalues of the semiseparable matrix at the bottom right are the Ritz values of the matrix A with respect to the Krylov subspace $\mathcal{K}_m$ (by $\langle x, y, z, \dots \rangle$ we denote the subspace spanned by the vectors $x, y, z, \dots$):

$$\mathcal{K}_m = \langle e_n, Ae_n, \dots, A^m e_n \rangle,$$

with e_n the n th vector of the standard basis in $\mathbb{R}^n$. Hence the convergence behavior of these eigenvalues is the same as the convergence behavior of the Lanczos algorithm with initial vector e_n .

As indicated in subsection 5.2, for theoretical purposes one can consider the similarity reduction to semiseparable form as consisting of two parts. First, the orthogonal similarity reduction into tridiagonal form is performed, which requires $O(n^3)$ flops. Then the tridiagonal matrix is transformed into a semiseparable one, requiring $O(n^2)$ flops. More specifically, at the bottom right of the matrix, semiseparable submatrices are formed which are orthogonally similar to the original tridiagonal submatrix at that position. Hence, because we know that the eigenvalues of these tridiagonal submatrices are the Ritz values as described above, we also have that the eigenvalues of the corresponding semiseparable submatrices are equal to these Ritz values. However, besides this property, the part of the method reducing the tridiagonal to a similar semiseparable matrix has an additional advantage, as is shown in the following section.

7. Subspace iteration while reducing the matrix. The intermediate matrices do not only contain the Lanczos–Ritz values in the already semiseparable part. During the construction, also a kind of nested subspace iteration is performed on the matrix which corresponds to executing a QL step on the already semiseparable part. Depending on the gaps and the approximation of the Ritz values, the subspace iteration will transform the semiseparable part into a block-diagonal matrix.

7.1. Theoretical proof. At each step of the algorithm introduced in Theorem 3.1, one more row (column) is added by means of orthogonal transformations to the set of the rows (columns) of the matrix already forming a semiseparable part. In this section, using arguments similar to those considered in [22, 23], we show that this algorithm can be interpreted as a kind of nested subspace iteration method [7], where the size of the vector subspace is increased by one and a change of coordinate system is made at each step of the algorithm. As a consequence, depending on the gap between the eigenvalues, the semiseparable part of the matrix will converge to a block-diagonal matrix, and the eigenvalues of these blocks converge to the largest eigenvalues in absolute value of the original symmetric matrix.

Given a matrix A and an initial subspace $\mathcal{S}^{(0)}$, the subspace iteration method [7] can be described as follows:

$$\mathcal{S}^{(i)} = A\mathcal{S}^{(i-1)}, \quad i = 1, 2, 3, \dots$$

Under weak assumptions on A and $\mathcal{S}^{(0)}$, the $\mathcal{S}^{(i)}$ converge to an invariant subspace. We will see that the reduction algorithm from symmetric to semiseparable matrix can be seen as such a subspace iteration, where the size of the subspace grows by one dimension at each step of the algorithm. Let $A_0^{(0)} = A$. Suppose we have only performed the first orthogonal similarity transformations such that the rows (columns) n and $n - 1$ are already proportional:

$$(7.1) \quad A_0^{(1)} = Q_1^T A_0^{(0)} Q_1,$$

where $A_0^{(1)}$ has the semiseparable structure in the rows (columns) n and $n - 1$ and $Q_1 = [q_1^{(1)}, \dots, q_n^{(1)}]$. From (7.1), we can write

$$(7.2) \quad A_0^{(0)} = Q_1 \begin{pmatrix} \times & \cdots & \times & 0 \\ \vdots & & \vdots & \vdots \\ \times & \cdots & \times & 0 \\ \times & \cdots & \times & \times \end{pmatrix} \equiv Q_1 L_1.$$

Let $e_1, \dots, e_n$ be the standard basis vectors of $\mathbb{R}^n$. From (7.2), because of the structure of L_1 , it can clearly be seen that

$$A_0^{(0)} \langle e_n \rangle = \langle q_n^{(1)} \rangle.$$

This means that the last column of $A_0^{(0)}$ and $q_n^{(1)}$ span the same one-dimensional space. In fact one subspace iteration step is performed on column e_n . The first step of the algorithm is completed when the following orthogonal transformation is performed:

$$A_0^{(1)} = Q_1^T A_0^{(0)} Q_1.$$

The latter transformation can be interpreted as a change of coordinate system: $A_0^{(0)}$ and $A_0^{(1)}$ represent the same linear transformation with respect to different coordinate systems. Let $y \in \mathbb{R}^n$. Then y is represented in the new system by $Q_1^T y$. This means that for the vector $q_n^{(1)}$ we get $Q_1^T q_n^{(1)} = e_n$. Summarizing, this means that one step of subspace iteration on the subspace $\langle e_n \rangle$ is performed, resulting in a new subspace $\langle q_n^{(1)} \rangle$, and then, by means of a coordinate transformation, it is transformed back into the subspace $\langle e_n \rangle$. So, instead of working with a fixed matrix and changing subspaces, we work with fixed subspaces and changing matrices.

The second step can be interpreted in a completely analogous way. Suppose we already have the semiseparable structure in the last two rows (columns). Then, in order to make the rows (columns) n up to $n - 2$ dependent, we perform the following similarity transformation on $A_0^{(1)}$:

$$(7.3) \quad A_0^{(2)} = Q_2^T A_0^{(1)} Q_2.$$

Using (7.3), $A_0^{(1)}$ can be written as follows:

$$A_0^{(1)} = Q_2 \begin{pmatrix} \times & \cdots & \times & 0 & 0 \\ \vdots & & & \vdots & \vdots \\ \times & \cdots & \times & 0 & 0 \\ \times & \cdots & \times & \times & 0 \\ \times & \cdots & \times & \times & \times \end{pmatrix} = Q_2 L_2.$$

Considering the subspace $\langle e_{n-1}, e_n \rangle$ and using the same notation as above, we have

$$A_0^{(1)} \langle e_{n-1}, e_n \rangle = \langle q_{n-1}^{(2)}, q_n^{(2)} \rangle.$$

This means that the second step of the algorithm is a step of subspace iteration performed on a slightly grown subspace. For every new dependency that is created in the symmetric matrix $A_0^{(m)}$, the dimension of the subspace is increased by one.

One can also see the algorithm as performing in each iteration step a QL step without shift on the semiseparable bottom-right submatrix. Let us partition the matrix $A_0^{(m-1)}$ (from (3.2)) as follows:

$$(7.4) \quad A_0^{(m-1)} = \left(\begin{array}{c|c} A_{m-1} & u_{m-1}v_{m-1}^T \\ \hline v_{m-1}u_{m-1}^T & S_{m-1} \end{array} \right),$$

with $A^{(j+1)} = A_{m-1} \in \mathbb{R}^{(n-m) \times (n-m)}$, $m = n - j$, $u_{m-1} \in \mathbb{R}^{(n-m) \times 1}$, $v_{m-1} \in \mathbb{R}^{m \times 1}$, and $S_{m-1} \in \mathbb{R}^{m \times m}$. From the semiseparable structure, we know that $[v_{m-1}, S_{m-1}e_1]$ has rank 1. In the first part of the m th iteration step, an $(n-m) \times (n-m)$ orthogonal matrix $Q_{m,1}$ is constructed such that

$$Q_{m,1}^T u_{m-1} = [0, 0, \dots, 0, \eta]^T,$$

with $\eta = \pm \|u_{m-1}\|$. Hence, our matrix $A_0^{(m-1)}$ is transformed into the following orthogonally similar matrix:

$$(7.5) \quad \begin{aligned} & \left(\begin{array}{c|c} Q_{m,1}^T & 0 \\ \hline 0 & I_m \end{array} \right) \left(\begin{array}{c|c} A_{m-1} & u_{m-1}v_{m-1}^T \\ \hline v_{m-1}u_{m-1}^T & S_{m-1} \end{array} \right) \left(\begin{array}{c|c} Q_{m,1} & 0 \\ \hline 0 & I_m \end{array} \right) \\ &= \left(\begin{array}{c|c} \tilde{A}_{m-1} & \eta e_{n-m}v_{m-1}^T \\ \hline \eta v_{m-1}e_{n-m}^T & S_{m-1} \end{array} \right). \end{aligned}$$

Let us partition the matrix a little differently now, adding one more column and row to the semiseparable part. Let

$$\tilde{A}_{m-1} = \begin{pmatrix} A_m & a \\ a^T & \alpha \end{pmatrix}.$$

Then we can define the semiseparable matrix $\tilde{S}_m$ as follows:

$$\tilde{S}_m = \begin{pmatrix} \alpha & \eta v_{m-1}^T \\ \eta v_{m-1} & S_{m-1} \end{pmatrix}.$$

In the second part of the m th iteration step, we perform a QL step on the semiseparable matrix $\tilde{S}_m$ of order $m+1$; i.e., we construct an orthogonal matrix $Q_{m,2}$ such that

$$\tilde{S}_m = Q_{m,2}L_m.$$

Hence, the matrix of (7.5) is transformed into the orthogonal similar matrix

$$\left(\begin{array}{c|c} I_{n-m-1} & 0 \\ \hline 0 & Q_{m,2}^T \end{array} \right) \left(\begin{array}{c|c} A_m & ae_1^T \\ \hline e_1a^T & \tilde{S}_m \end{array} \right) \left(\begin{array}{c|c} I_{n-m-1} & 0 \\ \hline 0 & Q_{m,2} \end{array} \right) = \left(\begin{array}{c|c} A_m & u_mv_m^T \\ \hline v_mu_m^T & S_m \end{array} \right).$$

FIG. 7.1. Description of QL step without shift applied at each iteration of the semiseparable reduction.

The execution of the two steps is illustrated in Figure 7.1. At the beginning of the iteration the last four rows and columns already have a semiseparable structure. In step (1), a similarity transformation (either an Householder matrix or a product of Givens transformations) is considered in order to annihilate the entries indicated by $\otimes$. Due to the semiseparable structure, all the entries in the first four rows (columns) with column (row) indexes greater than 5 are annihilated, too. In the second step, consisting of substeps (2), (3), (4), and (5), the Givens rotations are applied only to the left, transforming the (5×5) principal submatrix in the bottom-right corner to lower triangular form. To retrieve the semiseparable structure in the last 5 rows and columns, the transpose of the latter Givens rotation must be applied to the right.

Summarizing, this means that from step m , $m = 0, 1, \dots, n$ (so there is dependency between rows n up to $n - m$), all the consecutive steps perform subspace iterations on the subspace of dimension $m + 1$. From [23], we know that these consecutive iterations on subspaces tend to create block upper triangular matrices. Hence, for a symmetric matrix these are block diagonal. Furthermore, the process works for all the nested subspaces at the same time, and so the semiseparable part of the matrices $A_0^{(m)}$ generated by the proposed algorithm becomes more and more block diagonal, and the blocks contain the eigenvalues having a comparable absolute value. This explains why the lower-right block already gives a good estimate of the largest eigenvalues, since they are connected to a subspace on which the subspace iteration step is already performed several times. Hence as a special case, denoting by $z^{(m)}$ the eigenvector corresponding to the largest eigenvalue in absolute value λ of $A_0^{(m)}$, $m = 0, 1, 2, \dots$, we can say that if $z^{(0)}$ has a nonzero last component and if the largest eigenvalue is unique, the sequence $\{z^{(m)}\}$ converges to e_n , and, consequently, the lower-right element of $A_0^{(m)}$ converges to λ .

This insight also opens several new perspectives. In many problems, only the few largest eigenvalues (in absolute value) need to be computed [1, 8, 20, 21, 24, 25]. In such cases, the proposed algorithm gives the required information after only a few steps, without running the algorithm to completion. Moreover, because the sequence of similar matrices generated at each step of the algorithm converges to a block-diagonal matrix, the original problem can be divided into smaller independent subproblems (deflation). This is illustrated in the first numerical experiment of section 8.

Furthermore, if the second step of the algorithm is “iterated,” the sequence of semiseparable matrices generated converges to a block-diagonal matrix (see the remark at the end of section 3.2).

We finish this section with a theorem from [23] concerning the speed of convergence of subspace iterations [23, Theorem 5.4].

DEFINITION 7.1. *Denote by $\mathcal{S}$ and $\mathcal{T}$ two subspaces. Then the distance $d(\mathcal{S}, \mathcal{T})$ between these two subspaces is defined in the following way:*

$$d(\mathcal{S}, \mathcal{T}) = \sup_{s \in \mathcal{S}, \|s\|_2=1} \inf_{t \in \mathcal{T}} \|s - t\|_2.$$

Using this definition, we can state the following convergence theorem.

THEOREM 7.2. *Let $A \in \mathbb{C}^{n \times n}$, and let p be a polynomial of degree $\leq n$. Let $\lambda_1, \dots, \lambda_n$ denote the eigenvalues of A , ordered so that $|p(\lambda_1)| \geq |p(\lambda_2)| \geq \dots \geq |p(\lambda_n)|$. Suppose k is a positive integer less than n for which $|p(\lambda_k)| > |p(\lambda_{k+1})|$. Let (p_i) be a sequence of polynomials of degree $\leq n$ such that $p_i \rightarrow p$ as $i \rightarrow \infty$ and $p_i(\lambda_j) \neq 0$ for $j = 1, \dots, k$ and all i . Let $\rho = |p(\lambda_k)|/|p(\lambda_{k+1})|$. Let $\mathcal{T}$ and $\mathcal{U}$ be the invariant subspaces of A associated with $\lambda_1, \dots, \lambda_k$ and $\lambda_{k+1}, \dots, \lambda_n$, respectively. Consider the nonstationary subspace iteration*

$$\mathcal{S}_i = p_i(A)\mathcal{S}_{i-1},$$

where $\mathcal{S}_0$ is a k -dimensional subspace of $\mathbb{C}^n$ satisfying $\mathcal{S} \cap \mathcal{U} = \{0\}$. Then for every $\hat{\rho}$ satisfying $\rho < \hat{\rho} < 1$ there exists a constant $\hat{C}$ such that

$$d(\mathcal{S}_i, \mathcal{T}) \leq \hat{C}\hat{\rho}^i, \quad i = 1, 2, 3, \dots$$

In our case Theorem 7.2 can be applied with the polynomials $p_i(z)$ and $p(z)$ chosen in the following sense: $p_i(z) = p(z) = z$.

7.2. Practical interpretation. Taking a look at the numerical examples, sometimes it seems that initially there is no sign of subspace iteration. The proofs stated above are correct, but there is an interaction with the Lanczos behavior of the algorithm, which will sometimes lead to a delay in the convergence behavior of the subspace iteration. This can happen only when all the vectors $\langle e_{n-i}, e_{n-i+1}, \dots, e_n \rangle$ have a small component in one or more directions of the eigenspace connected to the dominant eigenvalues. Before we show by an example that this happens in practice, we first give a condition under which we are sure that the subspace iteration lets the matrix converge to one having a diagonal block containing the dominant eigenvalues. As soon as some of the Ritz values approximate all of the dominant eigenvalues, this convergence behavior appears. To show this we assume that we can split up the eigenvalues of the initial matrix A into two subsets, $\Lambda_1 = \{\lambda_{1,j} | j \in J\}$ and $\Lambda_2 = \{\lambda_{2,i} | i \in I\}$, with a gap between cluster 2 and cluster 1, $\min_{i \in I} |\lambda_{2,i}| \gg \max_{j \in J} |\lambda_{1,j}|$. Using the notation $\Delta_1 = \text{diag}(\Lambda_1)$ and $\Delta_2 = \text{diag}(\Lambda_2)$, we can write the matrix $A_0^{(m)}$ in the following way:

$$(7.6) \quad \begin{pmatrix} \tilde{A} & R \\ R & S \end{pmatrix} = \begin{pmatrix} V_1 & W_1 \\ V_2 & W_2 \end{pmatrix} \begin{pmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{pmatrix} \begin{pmatrix} V_1^T & V_2^T \\ W_1^T & W_2^T \end{pmatrix},$$

where $\tilde{A} = A^{(j)} = A_m$, $R = u_m v_m^T$, and $S = S_m$ in correspondence with (3.2). From (7.6), we have the following equation for S :

$$S = V_2 \Delta_1 V_2^T + W_2 \Delta_2 W_2^T.$$

Let the eigenvalue decomposition of S be denoted as

$$(7.7) \quad S = \begin{pmatrix} V_S & W_S \end{pmatrix} \begin{pmatrix} \Delta_{S,1} & 0 \\ 0 & \Delta_{S,2} \end{pmatrix} \begin{pmatrix} V_S^T \\ W_S^T \end{pmatrix},$$

where we assume that some of the eigenvalues of S (in fact, the Ritz values) approximate already the real dominant eigenvalues, meaning we assume $|\lambda_{S,2,i}| \approx |\lambda_{2,i}|$, $i \in I$.

If W_2 has full rank and is far from being not of full rank, there is no delay in the convergence behavior of the subspace iteration. This corresponds to the condition that the last vectors $\{e_{n-m}, e_{n-m+1}, \dots, e_n\}$ projected onto the invariant subspace connected to the dominant eigenvalues $\lambda_{2,i}$ have a large component in every direction of this subspace.

Via a Householder similarity transformation we can transform (7.6) into

$$\begin{pmatrix} \bar{A} & \begin{pmatrix} 0 \\ v^T \end{pmatrix} \\ \begin{pmatrix} 0 & v \end{pmatrix} & S \end{pmatrix} = \begin{pmatrix} \bar{V}_1 & \bar{W}_1 \\ V_2 & W_2 \end{pmatrix} \begin{pmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{pmatrix} \begin{pmatrix} \bar{V}_1^T & V_2^T \\ \bar{W}_1^T & W_2^T \end{pmatrix}.$$

Then we can substitute S by its eigenvalue decomposition (7.7).

$$\begin{pmatrix} \bar{A} & \begin{pmatrix} 0 \\ \bar{v}_1^T & \bar{v}_2^T \end{pmatrix} \\ \begin{pmatrix} 0 & \bar{v}_1 \\ & \bar{v}_2 \end{pmatrix} & \begin{pmatrix} \Delta_{S,1} & 0 \\ 0 & \Delta_{S,2} \end{pmatrix} \end{pmatrix} = \begin{pmatrix} \bar{V}_1 & \bar{W}_1 \\ V_2 & W_2 \end{pmatrix} \begin{pmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{pmatrix} \begin{pmatrix} \bar{V}_1^T & \bar{V}_2^T \\ \bar{W}_1^T & \bar{W}_2^T \end{pmatrix}.$$

Note that W_2 is of full rank if and only if $\bar{W}_2$ is of full rank. We get

$$\Delta_{2,S} = \begin{pmatrix} P\bar{V}_2\Delta_1 & P\bar{W}_2\Delta_2 \end{pmatrix} \begin{pmatrix} \bar{V}_2^T P^T \\ \bar{W}_2^T P^T \end{pmatrix},$$

where P is the projection matrix $[0, I]$ of size $n_{S,2} \times n$, where I is the identity matrix of size $n_{S,2} \times n_{S,2}$, with $n_{S,2}$ the dimension of $\Delta_{S,2}$. This means that

$$(7.8) \quad \Delta_{2,S} - P\bar{V}_2\Delta_1\bar{V}_2^T P^T = P\bar{W}_2\Delta_2\bar{W}_2^T P^T.$$

Note that $P\bar{W}_2$ is a square matrix.

Because there is a gap between the eigenvalues $\lambda_{1,j}$, $j \in J$, and $\lambda_{2,i}$, $i \in I$, there will be a comparable gap between $\lambda_{S,2,i}$, $i \in I$, and $\lambda_{1,j}$, $j \in J$. Hence the matrix at the left-hand side of (7.8) is far from singular. Therefore this is also true for the right-hand side, and $\bar{W}_2$ is of full rank.

In the remaining part of this paragraph, a brief explanation is given why, in certain situations, the subspace iteration behavior does not show up immediately from the start. Suppose the size of the block Δ_2 is 2. We can write the matrix A in decomposed form:

$$(7.9) \quad A_0^{(0)} = A = \begin{pmatrix} \bar{A} & a_n \\ a_n^T & \bar{\alpha} \end{pmatrix} = \begin{pmatrix} V & W \\ v^T & w^T \end{pmatrix} \begin{pmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{pmatrix} \begin{pmatrix} V^T & v \\ W^T & w \end{pmatrix}.$$

Applying already the Householder similarity transformation on the matrix A gives us the following decomposition:

$$\begin{aligned} & \begin{pmatrix} \tilde{H} & 0 \\ q^T & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} V & W \\ v^T & w^T \end{pmatrix} \begin{pmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{pmatrix} \begin{pmatrix} V^T & v \\ W^T & w \end{pmatrix} \begin{pmatrix} \tilde{H}^T & q & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} \tilde{A} & a^T & 0 \\ a & \gamma & \beta \\ 0 & \beta & \alpha \end{pmatrix} = \tilde{A}^{(1)}. \end{aligned}$$

The Ritz values are the eigenvalues of the 2×2 matrix:

$$\begin{pmatrix} \gamma & \beta \\ \beta & \alpha \end{pmatrix}.$$

In fact, to complete the previous step, a Givens transformation should still be performed on matrix $\tilde{A}^{(1)}$ to make the last two rows and columns dependent, and thereby transforming the matrix into $A_0^{(1)}$. Note that this last Givens transformation does not change the eigenvalues of the bottom-right submatrix.

We show now that the assumptions for an immediate convergence behavior of the subspace iteration are not satisfied. We prove that

$$[e_{n-1}, e_n]$$

does not have two large linearly independent components in the span of

$$\begin{pmatrix} \tilde{H} & 0 \\ q^T & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} W \\ w^T \end{pmatrix}.$$

We can write $[e_{n-1}, e_n]$ as a linear combination of the eigenvectors:

$$[e_{n-1}, e_n] = \begin{pmatrix} \tilde{H} & 0 \\ q^T & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} V & W \\ v^T & w^T \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}.$$

The coordinates C_2 of $[e_{n-1}, e_n]$ with respect to the dominant eigenvectors are the following:

$$\begin{aligned} C_2 &= \begin{pmatrix} W^T & w \end{pmatrix} \begin{pmatrix} \tilde{H}^T & q & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} e_{n-1}, & e_n \end{bmatrix} \\ &= \begin{pmatrix} W^T & w \end{pmatrix} \begin{pmatrix} q & 0 \\ 0 & 1 \end{pmatrix} = \begin{bmatrix} W^T q, & w \end{bmatrix}. \end{aligned}$$

Using (7.9), we get that

$$\begin{pmatrix} V^T & v \\ W^T & w \end{pmatrix} \begin{pmatrix} a_n \\ \bar{\alpha} \end{pmatrix} = \begin{pmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix}.$$

Hence,

$$(7.10) \quad W^T a_n + w \bar{\alpha} = \Delta_2 w.$$

Because of the Householder transformation we know that $q = a_n / \|a_n\|$, hence

$$(7.11) \quad W^T q = (\Delta_2 - \bar{\alpha} I) \frac{w}{\|a_n\|}.$$

Therefore,

$$\begin{aligned} C_2 &= \begin{bmatrix} W^T q, & w \end{bmatrix} \\ &= \begin{bmatrix} (\Delta_2 - \bar{\alpha} I) \frac{w}{\|a_n\|}, & w \end{bmatrix}. \end{aligned}$$

This means that if $\Delta_2 w$ lies almost in the same direction as w that C_2 is almost singular. If the two largest eigenvalues $\lambda_{2,1} \approx \lambda_{2,2}$, this is the case. Note, however, that when $\lambda_{2,1} \approx -\lambda_{2,2}$, the subspace iteration converges immediately, because they are both extreme and immediately located by the Lanczos procedure.

8. A numerical experiment. In this experiment we want to illustrate the interplay between the Lanczos convergence behavior and the nested subspace iteration of the new method. More numerical experiments can be found in [16].

Let us construct a symmetric matrix having the eigenvalues as shown in Figure 8.1. There are three clusters of eigenvalues where the relative gap between the first and second clusters is equal to 0.5 and the same as the relative gap between the second and third clusters. The first cluster has 20 eigenvalues geometrically distributed between 1 and 10^{-2} , the second cluster 20 eigenvalues between $5 \cdot 10^{-3}$ and $5 \cdot 10^{-5}$, and the third 60 eigenvalues between $2.5 \cdot 10^{-5}$ and $2.5 \cdot 10^{-7}$. The symmetric matrix A is obtained by transforming the diagonal matrix Δ containing the prescribed eigenvalues by an orthogonal similarity transformation $A = Q^T \Delta Q$. The orthogonal matrix used is taken as the Q -factor in the QR factorization of a random matrix, built by the MATLAB command `rand(n)`, where n is the dimension of the matrix.

FIG. 8.1. *Eigenvalues of the symmetric matrix of experiment 1: 3 clusters.*

Let us first look at Figure 8.2, which shows the behavior of the Ritz values for this example. In each step of the algorithm (x -axis), a cross is placed if a Ritz value approximates a real eigenvalue up to eight correct digits. Note that the same behavior is also obtained when using the classical tridiagonalization approach (starting at the bottom right of the initial symmetric matrix). However, the tridiagonal matrix obtained using this approach does not give a clear indication of the possible clusters or of possible deflation possibilities; i.e., nowhere on the subdiagonal does the magnitude of the element become very small. The magnitude of the subdiagonal elements is plotted in Figure 8.3. During the execution of the algorithm to obtain a similar semiseparable matrix, there will be a clear point where a diagonal block corresponding to the first cluster can be deflated and another point where a diagonal block corresponding to the second cluster can be deflated. Looking at Figure 8.2, we see that the eigenvalues of the first cluster are approximated well by the Lanczos–Ritz values from step 25 on. This means that at that moment the subspace iteration where the subspace has dimension 20 will converge towards the subspace corresponding to the first cluster. The convergence factor will be 0.5. This can clearly be seen in Figure 8.4. This figure shows for each step $j = 1, 2, \dots, n - 1$ in the algorithm the norms of the subdiagonal

FIG. 8.2. Behavior of the Ritz values.

FIG. 8.3. Magnitude of subdiagonal elements of similar tridiagonal matrix.

blocks $S(i : n, 1 : i - 1)$ for $i = n - j : n$. Each line in the plot indicates the change of the norm at each step of the algorithm. The line indicated by “o” corresponds to the norm of the subdiagonal block $S(81 : 100, 1 : 80)$. For the eigenvalues of the second cluster a similar convergence behavior occurs around step 45. The line indicated by “*” corresponds to the norm of the subdiagonal block $S(61 : 100, 1 : 60)$. The parallel lines in the figure having a smaller slope correspond to the subspace convergence behavior inside each of the clusters. Hence, at step 68 a first diagonal 20×20 block can be deflated, while at step 80 the next diagonal 20×20 block can be deflated. This example shows clearly the influence of the convergence of the Lanczos–Ritz values on the convergence behavior of the nested subspace iteration.

FIG. 8.4. Norms of subdiagonal matrices during the semiseparable reduction.

9. Conclusions. In this paper, we have designed a new algorithm to transform any symmetric matrix into a similar semiseparable one by orthogonal similarity transformations. We have shown that during execution of the algorithm semiseparable matrices are generated whose eigenvalues are nothing else but the Ritz values obtained by the Lanczos process. Moreover, the algorithm can be interpreted as a nested subspace iteration method where a QL step without shift is performed in each iteration step. In [18] an implicit QR algorithm is described for semiseparable matrices. This algorithm can be used to compute the eigenvalues of the resulting matrix as well as of the subblocks which are cut off by the nested subspace iteration.

REFERENCES

- [1] M. BRAUN, S. A. SOFIANOS, D. G. PAPAGEORGIOU, AND I. E. LAGARIS, *An efficient Chebyshev Lanczos method for obtaining eigensolutions of the Schrödinger equation on a grid*, J. Comput. Phys., 126 (1996), pp. 315–327.
- [2] S. CHANDRASEKARAN AND M. GU, *Fast and stable eigendecomposition of symmetric banded plus semi-separable matrices*, Linear Algebra Appl., 313 (2000), pp. 107–114.
- [3] S. CHANDRASEKARAN AND M. GU, *A divide and conquer algorithm for the eigendecomposition of symmetric block-diagonal plus semi-separable matrices*, Numer. Math., 96 (2004), pp. 723–731.
- [4] J. DEMMEL AND K. VESELIĆ, *Jacobi's method is more accurate than QR*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1204–1245.
- [5] J. W. DEMMEL, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
- [6] 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, V. Olshevsky, ed., Contemp. Math. 323, AMS, Providence, RI, SIAM, Philadelphia, 2003, pp. 105–118.
- [7] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
- [8] I. T. JOLLIFFE, *Principal Component Analysis*, Springer-Verlag, New York, 1986.
- [9] R. B. LEHOUCQ AND D. C. SORESENSEN, *Deflation techniques for an implicitly restarted Arnoldi iteration*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 789–821.
- [10] R. S. MARTIN AND J. H. WILKINSON, *Householder's tridiagonalization of a symmetric matrix*, Numer. Math., 11 (1968), pp. 181–195.
- [11] N. MASTRONARDI, E. VAN CAMP, AND M. VAN BAREL, *Divide and conquer algorithms for com-*

- puting the eigendecomposition of symmetric diagonal-plus-semiseparable matrices, *Numer. Algorithms*, 39 (2005), pp. 379–398.
- [12] B. N. PARLETT AND I. S. DHILLON, *Relatively robust representations of symmetric tridiagonals*, *Linear Algebra Appl.*, 309 (2000), pp. 121–151.
 - [13] G. W. STEWART, *Matrix Algorithms, Vol. II. Eigensystems*, SIAM, Philadelphia, 2001.
 - [14] G. STRANG AND T. NGUYEN, *The Interplay of Ranks of Submatrices*, *SIAM Rev.*, 46 (2004), pp. 637–646.
 - [15] L. N. TREFETHEN AND D. BAU, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
 - [16] M. VAN BAREL, R. VANDEBRIL, AND N. MASTRONARDI, *The Lanczos-Ritz Values Appearing in an Orthogonal Similarity Reduction of a Matrix into Semiseparable Form*, Report TW 360, Department of Computer Science, K. U. Leuven, Leuven, Belgium, 2003. Available online from <http://www.cs.kuleuven.ac.be/publicaties/rapporten/tw/TW360.abs.html>.
 - [17] E. VAN CAMP, N. MASTRONARDI, AND M. VAN BAREL, *Two fast algorithms for solving diagonal-plus-semiseparable linear systems*, *J. Comput. Appl. Math.*, 164–165 (2004), pp. 731–747.
 - [18] R. VANDEBRIL, M. VAN BAREL, AND N. MASTRONARDI, *An implicit QR algorithm for symmetric semiseparable matrices*, *Numer. Linear Algebra Appl.*, to appear.
 - [19] R. VANDEBRIL, M. VAN BAREL, AND N. MASTRONARDI, *A note on the representation and definition of semiseparable matrices*, *Numer. Linear Algebra Appl.*, to appear.
 - [20] M. E. WALL, P. A. DYCK, AND T. S. BRETIN, *Svdman—singular value decomposition analysis of microarray data*, *Bioinform.*, 17 (2001), pp. 566–568.
 - [21] M. E. WALL, A. RECHTSTEINER, AND L. M. ROCHA, *Singular value decomposition and principal component analysis*, in *A Practical Approach to Microarray Data Analysis*, D. P. Berrar, W. Dubitzky, and M. Granzow, eds., Kluwer Academic, Dordrecht, The Netherlands, 2003, pp. 91–109.
 - [22] D. S. WATKINS, *QR-like algorithms: An overview of convergence theory and practice*, in *The Mathematics of Numerical Analysis*, *Lectures in Appl. Math.* 32, AMS, Providence, RI, 1996, pp. 879–893.
 - [23] D. S. WATKINS AND L. ELSNER, *Convergence of algorithms of decomposition type for the eigenvalue problem*, *Linear Algebra Appl.*, 143 (1991), pp. 19–47.
 - [24] M. K. S. YEUNG AND W. L. RUZZO, *Principal component analysis for clustering gene expression data*, *Bioinform.*, 17 (2001), pp. 763–774.
 - [25] M. K. S. YEUNG, J. TEGNÉR, AND J. J. COLLINS, EDS., *Reverse engineering gene networks using singular value decomposition and robust regression*, *Proc. Natl. Acad. Sci.*, 99 (2002), pp. 6163–6168.