

On the convergence properties of the orthogonal similarity transformations to tridiagonal and semiseparable (plus diagonal) form

Raf Vandebril, Ellen Van Camp, Marc Van Barel and Nicola Mastronardi

Raf Vandebril

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

Ellen Van Camp

Department of Computer Science
KU Leuven, Belgium
Ellen.VanCamp@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

In this paper, we will compare the convergence properties of three basic reduction methods, by placing them in a general framework. It covers the reduction to tridiagonal, semiseparable and semiseparable plus diagonal form. These reductions are often used as the first step in the computation of the eigenvalues and/or eigenvectors of arbitrary matrices. In this way, the calculation of the eigenvalues using, for example, the QR-algorithm reduces in complexity. First we will investigate the convergence properties of these three reduction algorithms. It will be shown that for the partially reduced matrices at step k of any of these reduction algorithms, the lower right $k \times k$ (already reduced) sub-block will have the Lanczos-Ritz values, w.r.t. a certain starting vector. It will also be shown that the reductions to semiseparable and to semiseparable plus diagonal form have an extra convergence behavior a special type of subspace iteration is performed on the lower right $k \times k$ submatrix, which contains these Ritz-values. Secondly we look in more detail at the behavior of the involved subspace iteration. It will be shown that the reduction method can be interpreted as a nested type of multi-shift iteration. Theoretical results will be presented, making it possible to predict the convergence behavior of these reduction algorithms. Also a theoretical bound on the convergence rate is presented. Finally we illustrate by means of numerical examples, how it is possible to tune the convergence behavior such that it can become a powerful tool for certain applications.

Article information

- Vandebril, Raf; Van Camp, Ellen; Van Barel, Marc; Mastronardi, Nicola. *On the convergence properties of the orthogonal similarity transformations to tridiagonal and semiseparable (plus diagonal) form*, Numerische Mathematik, volume 104, issue 2, pages 205-239, 2006.
- The content of this article is identical to the content of the published paper, but without the final typesetting by the publisher.
- Journal's homepage: <http://link.springer.com/journal/211>
- Published version: <http://dx.doi.org/10.1007/s00211-006-0017-2>
- KU Leuven's repository url: <https://lirias.kuleuven.be/handle/123456789/125170>

On the convergence properties of the orthogonal similarity transformations to tridiagonal and semiseparable (plus diagonal) form.*

Raf Vandebril¹, Ellen Van Camp¹, Marc Van Barel¹, Nicola Mastronardi²

¹ K.U.Leuven, Dept. of Computer Science, 3001 Leuven(Heverlee)

e-mail: {raf.vandebril,ellen.vancamp,marc.vanbarel}@cs.kuleuven.be

² Istituto per le Applicazioni del Calcolo M. Picone, sez. Bari,

e-mail: n.mastronardi@ba.iac.cnr.it

Received: June 7, 2006/ Revised version:

Summary In this paper, we will compare the convergence properties of three basic reduction methods, by placing them in a general framework. It covers the reduction to tridiagonal, semiseparable and semiseparable plus diagonal form. These reductions are often used as the first step in the computation of the eigenvalues and/or eigenvectors of arbitrary matrices. In this way, the calculation of the eigenvalues using, for example, the QR -algorithm reduces in complexity.

* The research was partially supported by the Research Council K.U.Leuven, projects OT/00/16 (SLAP: Structured Linear Algebra Package), OT/05/40 (Large rank structured matrix computations), 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), G.0184.02 (CORFU: Constructive study of Orthogonal Functions) and G.0455.0 (RHPH: Riemann-Hilbert problems, random matrices and Padé-Hermite approximation), 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 IUAPV-22 (Dynamical Systems and Control: Computation, Identification & Modelling) The research of the second author was partially supported by MIUR, grant number 2004015437, by the short term mobility program, Consiglio Nazionale delle Ricerche and by VII Programma Esecutivo di Collaborazione Scientifica Italia–Comunità Francese del Belgio, 2005–2006. The scientific responsibility rests with the authors.

First we will investigate the convergence properties of these three reduction algorithms. It will be shown that for the partially reduced matrices at step k of any of these reduction algorithms, the lower right $k \times k$ (already reduced) sub-block will have the Lanczos-Ritz values, w.r.t. a certain starting vector. It will also be shown that the reductions to semiseparable and to semiseparable plus diagonal form have an extra convergence behavior: a special type of subspace iteration is performed on the lower right $k \times k$ submatrix, which contains these Ritz-values.

Secondly we look in more detail at the behavior of the involved subspace iteration. It will be shown that the reduction method can be interpreted as a nested type of multi-shift iteration. Theoretical results will be presented, making it possible to predict the convergence behavior of these reduction algorithms. Also a theoretical bound on the convergence rate is presented.

Finally we illustrate by means of numerical examples, how it is possible to tune the convergence behavior such that it can become a powerful tool for certain applications.

Key words orthogonal similarity reductions – tridiagonal, semiseparable – semiseparable plus diagonal – Lanczos-Ritz values – multi-shift – subspace iteration

Mathematics Subject Classification (1991): 65F15,15A18

1 Introduction

Recently, two new reduction algorithms were proposed for reducing arbitrary symmetric matrices via orthogonal similarity transformations to semiseparable and semiseparable plus diagonal form [1,2]. These reductions are closely related with the tridiagonalization algorithm of a symmetric matrix [3,4]. In computational complexity, they only differ with a factor $O(n^2)$ w.r.t. the latter algorithm. This $O(n^2)$ factor ($9n^2$ and $10n^2$ for the reduction to semiseparable and semiseparable plus diagonal form respectively), is due to the fact that inside the reduction algorithm, some kind of chasing is performed to chase the semiseparable structure downwards.

It is proved in [1], that this chasing step can be interpreted as a nested QL -iteration without shift on the original matrix. This gives the advantage, that the reduction algorithm, might reveal the largest eigenvalues in the spectrum of the reduced matrix. This application, the so-called Rank Revealing property, is used for example in [5,6].

Combined with an effective implementation for computing the eigenvalues of semiseparable matrices [7–10], one can use this reduction algorithm to compute the spectrum of an arbitrary symmetric matrix.

In [2] a method was presented for reducing a symmetric matrix into a similar semiseparable plus diagonal one. This reduction offered the possibility to freely choose the diagonal used in the reduction scheme. It was shown that the choice of the diagonal heavily determined the convergence behavior of the reduction algorithm. Also for the class of semiseparable plus diagonal matrices, several algorithms exist to compute the eigendecomposition. For example divide and conquer methods [7,8], and QR -algorithms [11].

In this paper we will investigate in detail the reduction to semiseparable plus diagonal form. First we will show that the reduction to semiseparable, and the reduction to tridiagonal can be seen as special cases of the reduction to semiseparable plus diagonal form. We will prove that one can interpret the reduction to tridiagonal form as a reduction to a semiseparable plus diagonal matrix with the diagonal equal to $-\infty$. Secondly we will investigate the convergence behavior of all three reduction algorithms. We will prove that all three algorithms have as eigenvalues in the already reduced lower right block the Lanczos-Ritz values. Moreover the reduction to semiseparable plus diagonal form, has an extra convergence behavior, which we can interpret as a nested multi-shift iteration on the original (untransformed) matrix. Having some information on the clusters of the spectrum of the matrix, the diagonal can be chosen in order to enforce the convergence to different clusters. Finally we combine both convergence behaviors, and prove that the multi-shift subspace iteration will start converging as soon as the Lanczos-Ritz values approximate well enough the dominant eigenvalues w.r.t. the multi-shift iteration.

The paper is organized as follows. In Section 2, we repeat briefly the reduction algorithm to semiseparable plus diagonal form and we show that the other two orthogonal similarity transformations are special cases of this reduction. In Section 3, we briefly prove two convergence behaviors, namely the Lanczos-Ritz value convergence and the subspace iteration. As the subspace iteration as presented in this section does not explain the convergence properties of the reduction algorithm in an appealing manner, we investigate it in more detail in Section 4. We prove that the subspace iteration can be interpreted, such that a nested multi-shift iteration is performed on the original unreduced matrix. Section 5 investigates the combined behavior between the two convergence properties. The speed of convergence of

this nested multi-shift QL -iteration is examined in Section 6. In the numerical experiments of Section 7, the behavior of the reduction algorithm with respect to the theorems in this paper is investigated.

2 The reduction algorithms

As already mentioned in the abstract and the introduction of the paper, we will consider here three types of orthogonal similarity reductions, namely the reduction to tridiagonal, semiseparable and semiseparable plus diagonal form.

In this section we will show that the reduction to semiseparable plus diagonal form is the most general one. The reduction to semiseparable and the reduction to tridiagonal form can be seen as special cases of this reduction.

First we will briefly repeat the definition of a semiseparable matrix:

Definition 1 *A matrix S is called a semiseparable matrix if all submatrices which can be taken out of the lower and upper triangular part of the matrix S , including the diagonal, have rank ≤ 1 .*

Actually, this is the class of semiseparable matrices of semiseparability rank 1 (More information about higher order semiseparable matrices can be found for example in [12, 13]). The inverse of a nonsingular semiseparable matrix is a tridiagonal matrix. Hence, there is a close relation between all the presented reduction algorithms.

Theorem 1 *Let A be a real symmetric matrix and d_1, \dots, d_n n arbitrary real numbers. Then there exists an orthogonal matrix U such that*

$$U^T A U = S + D,$$

where S is a semiseparable matrix and D is a diagonal matrix containing the numbers d_1, \dots, d_n as diagonal elements.

Proof The proof is by finite induction and constructive. We illustrate it on a 5×5 example, as the more general case is completely similar.

Let us introduce some notation. In the proof, arbitrary elements in the matrix are denoted with the \times and $+$ sign. If an element changes, symbolically we change the symbol from \times to $+$ or vice versa. The elements surrounded by \circ denote the elements that will be annihilated by an orthogonal transformation and the elements surrounded by \square denote the elements which already satisfy the semiseparable structure.

Before starting the reduction procedure, to reduce the matrix A to the desired semiseparable plus diagonal form, we can always perform an initial orthogonal similarity transformation Q_0 : $A^{(1)} = Q_0^T A Q_0$. This transformation is not essential for obtaining the final semiseparable plus diagonal structure, but it does influence the convergence behavior as we will show later on.

Let us denote our matrices at the beginning of step k of the algorithm by $A^{(k)} = A_0^{(k)}$. If an orthogonal similarity transformation is performed on the matrix $A_l^{(k)}$ the subindex l is increased with 1: $A_{l+1}^{(k)}$. The idea is, to add in every step of the algorithm, one more row and column to the semiseparable plus diagonal structure.

- **Step 1:** In this first step we will make the last two rows and columns of the matrix of semiseparable plus diagonal form. In every subsequent step we will add one more row and column to this structure.

So let us start with the matrix $A = A^{(1)} = A_0^{(1)}$, and let us annihilate the elements marked in the last row and column, by performing a similarity Householder transformation $H_j^{(i)}$. For the orthogonal similarity transformations we use the same notation, the superscript i denotes in which step of the algorithm we are, and the subscript j denotes that this is the j th orthogonal similarity transformation performed on the matrix $A^{(i)}$. The Householder transformations are denoted by H , while the Givens transformations are denoted by G .

$$\begin{array}{ccc} \begin{pmatrix} \times & \times & \times & \times & \otimes \\ \times & \times & \times & \times & \otimes \\ \times & \times & \times & \times & \otimes \\ \times & \times & \times & \times & \times \\ \otimes & \otimes & \otimes & \times & \times \end{pmatrix} & \longrightarrow & \begin{pmatrix} \times & \times & \times & \times & 0 \\ \times & \times & \times & \times & 0 \\ \times & \times & \times & \times & 0 \\ \times & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{pmatrix} \\ & & \Downarrow \\ & & A_0^{(1)} \xrightarrow{H_1^{(1)T} A_0^{(1)} H_1^{(1)}} A_1^{(1)}. \end{array}$$

Before performing the next similarity transformation, which will make the last two rows and columns dependent, we have to extract some diagonal elements out of the matrix $A_1^{(1)}$. The matrix $A_1^{(1)}$ is written now as the sum of matrix and a diagonal matrix (it is essential that both the diagonal elements are equal to d_1):

$$\begin{aligned}
A_1^{(1)} &= \begin{pmatrix} \times & \times & \times & \times & 0 \\ \times & \times & \times & \times & 0 \\ \times & \times & \times & \times & 0 \\ \times & \times & \times & + & \times \\ 0 & 0 & 0 & \times & + \end{pmatrix} + \begin{pmatrix} 0 & & & & \\ & 0 & & & \\ & & 0 & & \\ & & & d_1 & \\ & & & & d_1 \end{pmatrix} \\
&= \hat{A}_1^{(1)} + D_1^{(1)}.
\end{aligned}$$

Determining now the Givens transformation $G_2^{(1)}$, such that applying it on the right to $\hat{A}_1^{(1)}$ (without application on the left) annihilates the element in position (5, 4) of the matrix $\hat{A}_1^{(1)}$. Applying $G_2^{(1)}$ as a similarity transformation on the matrix $\hat{A}_1^{(1)}$ gives us the following transformation (More information on this type of transformations can be found in [2, 1]).

$$\begin{aligned}
\begin{pmatrix} \times & \times & \times & \times & 0 \\ \times & \times & \times & \times & 0 \\ \times & \times & \times & \times & 0 \\ \times & \times & \times & + & \times \\ 0 & 0 & 0 & \times & + \end{pmatrix} &\xrightarrow{G_2^{(1)T} \hat{A}_1^{(1)} G_2^{(1)}} \begin{pmatrix} \times & \times & \times & \boxtimes & \boxtimes \\ \times & \times & \times & \boxtimes & \boxtimes \\ \times & \times & \times & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{pmatrix} \\
\hat{A}_1^{(1)} &\xrightarrow{G_2^{(1)T} \hat{A}_1^{(1)} G_2^{(1)}} \hat{A}_2^{(1)}.
\end{aligned}$$

Applying now this similarity transformation (with $G_2^{(1)}$) to the matrix $A_1^{(1)}$ instead of $\hat{A}_1^{(1)}$ we get the following matrix

$$G_2^{(1)T} A_1^{(1)} G_2^{(1)} = A_2^{(1)} = A_0^{(2)} = \begin{pmatrix} \times & \times & \times & \boxtimes & \boxtimes \\ \times & \times & \times & \boxtimes & \boxtimes \\ \times & \times & \times & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{pmatrix} + \begin{pmatrix} 0 & & & & \\ & 0 & & & \\ & & 0 & & \\ & & & d_1 & \\ & & & & d_1 \end{pmatrix}.$$

One can see that the Givens transformations can be shifted through the diagonal matrix, as the Givens is performed on the last two rows and columns and the diagonal is a multiple of the identity in

this block. We rewrite the matrix $A_0^{(2)}$ now in the following form :

$$A_0^{(2)} = \begin{pmatrix} \times & \times & \times & \boxtimes & \boxtimes \\ \times & \times & \times & \boxtimes & \boxtimes \\ \times & \times & \times & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxplus \end{pmatrix} + \begin{pmatrix} 0 & & & & \\ & 0 & & & \\ & & 0 & & \\ & & & d_1 & \\ & & & & d_2 \end{pmatrix}$$

$$= \hat{A}_0^{(2)} + D_0^{(2)}.$$

This completes the first step in the proof. We have now two dependent rows (in the lower triangular part) and the diagonal elements corresponding to these two rows are d_1 and d_2 . This means that the last two rows of this matrix satisfy already the semiseparable plus diagonal structure. The remainder of the proof proceeds by induction.

- **Step k:** Assume that the last k rows of the matrix $A^{(k)} = A_0^{(k)}$ are already in semiseparable plus diagonal form and the corresponding diagonal elements are d_1, \dots, d_k . We will add now one row to this structure such that $A^{(k+1)}$ has $k+1$ rows satisfying the structure and the corresponding diagonal elements are d_1, \dots, d_{k+1} . For simplicity we assume here $k = 3$ in our example. Our matrix is therefore of the following form:

$$A_0^{(3)} = \begin{pmatrix} \times & \times & \boxtimes & \boxtimes & \boxtimes \\ \times & \times & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{pmatrix} + \begin{pmatrix} 0 & & & & \\ & 0 & & & \\ & & d_1 & & \\ & & & d_2 & \\ & & & & d_3 \end{pmatrix}$$

$$= \hat{A}_0^{(3)} + D_0^{(3)}.$$

In a similar way as in step 1 we introduce some zeros in the structure by applying a similarity Householder transformation $H_1^{(3)}$ to the first two rows and columns. We remark that applying this transformation does not affect the diagonal matrix as the first two rows and columns of the diagonal matrix $D_0^{(3)}$ equal zero. Therefore we only demonstrate this similarity transformation on the matrix $\hat{A}_0^{(3)}$. (Because of the semiseparable structure, zeros are created in the complete first column.)

$$\begin{pmatrix} \times & \times & \otimes & \otimes & \otimes \\ \times & \times & \boxtimes & \boxtimes & \boxtimes \\ \otimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \otimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \otimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{pmatrix} \xrightarrow{H_1^{(3)T} \hat{A}_0^{(3)} H_1^{(3)}} \begin{pmatrix} \times & \times & 0 & 0 & 0 \\ \times & \times & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{pmatrix}$$

This transformation is applied to the matrix $A_0^{(3)}$ which gives us: $H_1^{(3)T} A_0^{(3)} H_1^{(3)} = A_1^{(3)}$ and which is rewritten as the sum of a new semiseparable matrix $\hat{A}_1^{(3)}$ and a new diagonal $D_1^{(3)}$.

$$\begin{aligned} A_1^{(3)} &= \begin{pmatrix} \times & \times & 0 & 0 & 0 \\ \times & \times & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{pmatrix} + \begin{pmatrix} 0 & & & & \\ & 0 & & & \\ & & d_1 & & \\ & & & d_2 & \\ & & & & d_3 \end{pmatrix} \\ &= \begin{pmatrix} \times & \times & 0 & 0 & 0 \\ \times & + & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{pmatrix} + \begin{pmatrix} 0 & & & & \\ & d_1 & & & \\ & & d_1 & & \\ & & & d_2 & \\ & & & & d_3 \end{pmatrix} \\ &= \hat{A}_1^{(3)} + D_1^{(3)}. \end{aligned}$$

The first upper left two nonzero elements of the diagonal matrix are chosen equal to each other. In this way we can shift again the next Givens transformation through this matrix. Consecutively we use the matrix $\hat{A}_1^{(3)}$ to determine the next Givens transformation $G_2^{(3)}$. The Givens $G_2^{(3)}$ is constructed in such a way that performing it on the right of $\hat{A}_1^{(3)}$ annihilates the element in position (3, 2). The similarity transformation, will therefore transform the matrix $\hat{A}_1^{(3)}$ in the following way:

$$\begin{pmatrix} \times & \times & 0 & 0 & 0 \\ \times & + & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{pmatrix} \xrightarrow{G_2^{(3)T} \hat{A}_1^{(3)} G_2^{(3)}} \begin{pmatrix} \times & \boxtimes & \boxtimes & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \end{pmatrix}.$$

More information on this procedure can be found in [1,2]. Applying this transformation to the matrix $A_1^{(3)}$ and rewriting the

matrix, results into:

$$\begin{aligned}
A_2^{(3)} &= \begin{pmatrix} \times & \boxtimes & \boxtimes & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \end{pmatrix} + \begin{pmatrix} 0 & & & & \\ & d_1 & & & \\ & & d_1 & & \\ & & & d_2 & \\ & & & & d_3 \end{pmatrix} \\
&= \begin{pmatrix} \times & \boxtimes & \boxtimes & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & 0 & 0 \\ \boxtimes & \boxtimes & \boxplus & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \end{pmatrix} + \begin{pmatrix} 0 & & & & \\ & d_1 & & & \\ & & d_2 & & \\ & & & d_2 & \\ & & & & d_3 \end{pmatrix} \\
&= \hat{A}_2^{(3)} + D_2^{(3)}.
\end{aligned}$$

In a similar way transformation $G_3^{(3)}$ is determined such that we have the following equations for the matrix $A_3^{(3)}$, which is rewritten such that the diagonal elements subject to the next Givens transformation are equal to each other.

$$\begin{aligned}
A_3^{(3)} &= \begin{pmatrix} \times & \boxtimes & \boxtimes & \boxtimes & 0 \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & 0 \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & 0 \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{pmatrix} + \begin{pmatrix} 0 & & & & \\ & d_1 & & & \\ & & d_2 & & \\ & & & d_2 & \\ & & & & d_3 \end{pmatrix} \\
&= \begin{pmatrix} \times & \boxtimes & \boxtimes & \boxtimes & 0 \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & 0 \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & 0 \\ \boxtimes & \boxtimes & \boxtimes & \boxplus & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{pmatrix} + \begin{pmatrix} 0 & & & & \\ & d_1 & & & \\ & & d_2 & & \\ & & & d_3 & \\ & & & & d_3 \end{pmatrix} \\
&= \hat{A}_3^{(3)} + D_3^{(3)}.
\end{aligned}$$

Applying the last transformation $G_4^{(3)}$ gives us the desired result $A_4^{(3)}$ which completes step k in the iterative procedure.

$$\begin{aligned}
A_4^{(3)} &= \begin{pmatrix} \times & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{pmatrix} + \begin{pmatrix} 0 & & & & \\ & d_1 & & & \\ & & d_2 & & \\ & & & d_3 & \\ & & & & d_3 \end{pmatrix} \\
&= \begin{pmatrix} \times & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxplus \end{pmatrix} + \begin{pmatrix} 0 & & & & \\ & d_1 & & & \\ & & d_2 & & \\ & & & d_3 & \\ & & & & d_4 \end{pmatrix} \\
&= \hat{A}_4^{(3)} + D_4^{(3)}. \tag{1}
\end{aligned}$$

To obtain the complete semiseparable matrix for this example one has to perform one extra chasing step.

More information concerning accuracy and the number of involved operations can be found in [2]. We will now prove the existence of the reductions to semiseparable and to tridiagonal form as being two special cases of the previous reduction.

Theorem 2 *Let A be a real symmetric matrix. Then there exists an orthogonal matrix U such that*

$$U^T A U = S,$$

where S is a semiseparable matrix.

Proof The proof can be found in [1]. We will prove it here based on the previous theorem.

Using Theorem 1, we know that we can reduce any symmetric matrix into a similar semiseparable plus diagonal one, with a free choice of the diagonal. Taking the diagonal equal to zero, states that we can reduce any symmetric matrix into a similar semiseparable one.

The orthogonal similarity transformation of a symmetric matrix to tridiagonal form is a basic tool and can be found in several books [3, 14, 4].

Theorem 3 *Let A be a real symmetric matrix. Then there exists an orthogonal matrix U such that*

$$U^T A U = T,$$

where T is a tridiagonal matrix.

Proof Considering the proof of Theorem 1, we see that after each Householder transformation in step k a chasing technique involving k Givens transformations is applied. Replacing now all these Givens transformations by a special Givens transformation, namely the identity, we see that the resulting matrix is not semiseparable but tridiagonal.

It is clear that the reduction to semiseparable plus diagonal is the most general one, while the two other transformations, are special cases. In fact we have even more, we can see the reduction to tridiagonal form as the reduction to semiseparable form with all the diagonal elements equal to $-\infty$.

Theorem 4 *The orthogonal similarity reduction to tridiagonal form, can be seen as the orthogonal similarity reduction to semiseparable plus diagonal form, where we choose the diagonal equal to $-\infty$.*

Proof If we prove that the performed Givens transformations in this case equal the identity matrices, we know that the resulting matrix will be of tridiagonal form.

Let us define a Givens transformation as follows:

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ r \end{pmatrix}$$

Where c , s and r are defined as:

$$\begin{aligned} r &= \sqrt{x^2 + y^2}, \\ c &= y/r, \\ s &= -x/r. \end{aligned}$$

The c and s are respectively the cosine and sine of a specific angle.

Assume we would like to obtain a semiseparable plus diagonal matrix with diagonal $-\epsilon$. This means that we have to perform the following Givens transformations, in the chasing technique, on the right to annihilate the element x , where $y + \epsilon$ is the diagonal element:

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} x \\ y + \epsilon \end{pmatrix} = \begin{pmatrix} 0 \\ r \end{pmatrix}$$

With the elements defined as

$$\begin{aligned} r &= \sqrt{x^2 + (y + \epsilon)^2}, \\ c &= (y + \epsilon)/r, \\ s &= -x/r. \end{aligned}$$

Taking the limit now for $\epsilon \rightarrow \infty$ leads to the Givens transformation equal to the identity.

$$\lim_{\epsilon \rightarrow \infty} \left(\frac{1}{\sqrt{x^2 + (y + \epsilon)^2}} \begin{pmatrix} (y + \epsilon) & -x \\ x & (y + \epsilon) \end{pmatrix} \right) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

We remark, that this previous theorem implies that we can construct for every tridiagonal matrix a sequence of semiseparable plus diagonal matrices converging to this tridiagonal matrix. This implies that the class of semiseparable plus diagonal matrices is not point-wise closed. We briefly illustrate this with a symmetric three by three matrix.

$$\lim_{\epsilon \rightarrow \infty} \left(\begin{pmatrix} a & b & \frac{1}{\epsilon} \\ b & \epsilon bd & d \\ \frac{1}{\epsilon} & d & e \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & c - \epsilon bd & 0 \\ 0 & 0 & 0 \end{pmatrix} \right) = \begin{pmatrix} a & b & \\ b & c & d \\ & d & e \end{pmatrix}$$

The sequence of matrices on the left are all semiseparable plus diagonal matrices, and their limit is tridiagonal.

3 The convergence properties

In this section we will investigate two types of convergence properties, related to the orthogonal similarity transformations: the Lanczos convergence behavior and the subspace iteration.

3.1 The Lanczos-Ritz values

It is well-known that the reduction to tridiagonal form has the Lanczos-Ritz values in the lower right $k \times k$ block of the tridiagonal matrix $T^{(k)}$. Moreover we know by [15], that also the intermediate semiseparable and the intermediate semiseparable plus diagonal matrices inherit this behavior. We will however briefly state the results.

Suppose, we have a matrix $A^{(0)} = A$, which is transformed via an initial orthogonal similarity transformation into the matrix $A^{(1)} = Q_0^T A^{(0)} Q_0$. Denote by Q_m the product of all orthogonal transformations used at the m th step of the reduction algorithm to either tridiagonal, semiseparable or semiseparable plus diagonal form. This means: $A^{(2)} = Q_1^T A^{(1)} Q_1$, $A^{(3)} = Q_2^T A^{(2)} Q_2, \dots$. Hence, the orthogonal transformation to go from $A^{(m)}$ to $A^{(m+1)}$ is Q_m . With $Q_{0:m}$ the

orthogonal matrix equal to the product $Q_0 Q_1 \dots Q_m$ is denoted. This means that

$$\begin{aligned} A^{(m+1)} &= Q_m^T A^{(m)} Q_m \\ &= Q_m^T Q_{m-1}^T \dots Q_1^T Q_0^T A Q_0 Q_1 \dots Q_{m-1} Q_m \\ &= Q_{0:m}^T A Q_{0:m}. \end{aligned}$$

The matrix $A^{(m+1)}$ is of the following form:

$$\left(\begin{array}{c|c} A_{m+1} & \times \\ \times & R_{m+1} \end{array} \right)$$

where R_{m+1} stands for that part of the matrix of dimension $(m + 1) \times (m + 1)$ which is already transformed to the appropriate form, e.g., tridiagonal, semiseparable or semiseparable plus diagonal. The matrix A_{m+1} is of dimension $(n - m - 1) \times (n - m - 1)$ matrices.

The eigenvalues of R_{m+1} are called the Ritz values of A with respect to the subspace spanned by the columns of $Q_{0:m} \langle e_{n-m}, \dots, e_n \rangle^1$ (see e.g. [4]).

Suppose we have now the Krylov subspace of order $m + 1$ with initial vector v :

$$\mathcal{K}_{m+1}(A, v) = \langle v, Av, \dots, A^m v \rangle$$

Remark 1 For simplicity, we assume in this paper that the Krylov subspaces we are working with are not invariant, i.e. that for every m : $\mathcal{K}_m(A, v) \neq \mathcal{K}_{m+1}(A, v)$, where $m = 1, 2, \dots, n - 1$. The special case of invariant subspaces can be dealt with in a completely similar way (see [15]).

If the columns of the matrix $Q_{0:m} \langle e_{n-m}, \dots, e_n \rangle$ form an orthonormal basis of the Krylov subspace $\mathcal{K}_{m+1}(A, v)$, then we say that the eigenvalues of R_{m+1} are the Lanczos-Ritz (Arnoldi in the nonsymmetric case) values of A with respect to the initial vector v .

We can now formulate the following theorem:

Theorem 5 *Let A be a symmetric matrix and U is the orthogonal matrix (from Theorem 1,2 or 3) such that*

$$U^T A U = B,$$

where B is either a tridiagonal, semiseparable or semiseparable plus diagonal matrix.

If we consider the reduction algorithms like in the proofs of Theorem 1,2 or 3, the intermediate matrices at step m of the reduction

¹ With $\langle a, b, c \rangle$ we denote the subspace spanned by the vectors a, b and c .

have as eigenvalues of the lower right $m \times m$ block the Lanczos-Ritz values w.r.t. $Q_0 e_n$.

Proof We will not prove this statement for the tridiagonal matrices, as this is a well-known, and classical results (see e.g. [3]). Taking a closer look at the algorithms from the proofs of Theorems 1 and 2, we see that these latter two orthogonal similarity transformations, perform always a chasing step after performing the Householder transformation. This chasing step is applied on the lower right $k \times k$ block. In fact an orthogonal similarity transformation is performed on the lower right block, and hence it does not change the eigenvalues of this block. The eigenvalues of this block are in fact essentially the same eigenvalues as the ones coming from the reduction to tridiagonal form, as all three reduction methods use exactly the same (up to the sign) Householder transformations.

3.2 A nested subspace iteration

The reduction to semiseparable and to semiseparable plus diagonal form performs more operations than the corresponding reduction to tridiagonal form. More precisely, at every step m of the reduction algorithm m extra Givens transformations are performed. These extra Givens transformations create the “nested subspace iteration” behavior. In this section we will investigate more in detail what is meant by this behavior.

The nested subspace iteration, connected to the orthogonal similarity transformation of a matrix to semiseparable form was investigated in [1]. As proved before, the reduction to semiseparable form is a special case of the reduction to semiseparable plus diagonal form, therefore its convergence behavior is also a special case of the more general convergence behavior. Hence, we will derive the nested subspace iteration, related to the transformation into semiseparable plus diagonal form and afterwards translate this to the reduction to semiseparable and the reduction to tridiagonal form.

Denote by $D^{(m)}$ the diagonal matrix of dimension n , with the lower right m diagonal elements equal to $[d_1, d_2, \dots, d_m]$.

When looking at the first step of the reduction algorithm, we can state that:

$$\begin{aligned} Q_0^T A Q_0 &= A^{(1)} = Q_1(Q_1^T A^{(1)}) \\ &= Q_1(Q_1^T(D^{(1)} + S^{(1)})) \\ &= Q_1 \left(Q_1^T \left(\begin{array}{c|c} 0 & \\ \vdots & \\ \hline 0 & \\ \hline & d_1 \end{array} \right) + \left(\begin{array}{c|c} \times \dots \times & 0 \\ \vdots & \vdots \\ \times \dots \times & 0 \\ \times \dots \times & \times \end{array} \right) \right). \end{aligned}$$

Multiplying both sides of the former equality to the right by $\langle e_n \rangle$ leads to

$$\begin{aligned} A^{(1)}\langle e_n \rangle &= d_1 I_n \langle e_n \rangle + Q_1 \langle e_n \rangle \\ \Rightarrow (A^{(1)} - d_1 I_n)\langle e_n \rangle &= Q_1 \langle e_n \rangle = \langle q_n^{(1)} \rangle \end{aligned} \quad (2)$$

with $q_n^{(1)}$ the last column of Q_1 . We assume that the lower-right element in the matrix $Q_1^T S^{(1)}$ is different from zero, otherwise d_1 would be an eigenvalue, and e_n would be an eigenvector. This brings us to the case of invariant subspaces, which is in fact good news. We will however not go into these details and assume, throughout the remainder of the text, that the subspaces we work with are not invariant. The invariant case naturally splits up in blocks, and the blocks can be dealt with completely similar as in the remainder of the paper.

To complete the first step of the algorithm, the following transformation is performed:

$$A^{(2)} = Q_1^T A^{(1)} Q_1.$$

This can be interpreted as a transformation of the basis used: when transforming a vector y in the old basis into $Q_1^T y$ in the new basis, then $A^{(1)}$ will become $A^{(2)} = Q_1^T A^{(1)} Q_1$ in the new basis. The vector $q_n^{(1)}$ becomes $Q_1^T q_n^{(1)} = e_n$ and hence, because of Equation (2), $(A^{(1)} - d_1 I_n)\langle e_n \rangle$ becomes $\langle e_n \rangle$. This means that the subspace $\langle e_n \rangle$ we are working on stays the same, only the matrix we use changes.

Looking now at the m th step, $m = 2, \dots, n - 1$, of the reduction algorithm, we get:

$$\begin{aligned} A^{(m)} &= Q_m(Q_m^T A^{(m)}) \\ &= Q_m(Q_m^T(D^{(m)} + S^{(m)})) \\ &= Q_m \left(Q_m^T \left(\begin{array}{c|c} 0 & \\ \vdots & \\ \hline & d_1 \\ & \\ & \vdots \\ & d_m \end{array} \right) + \left(\begin{array}{c|c} \times \dots \times & 0 \dots 0 \\ \vdots & \vdots \\ \times \dots \times & 0 \dots 0 \\ \times \dots \times & \times \dots 0 \\ \vdots & \vdots \\ \times \dots \times & \times \dots \times \end{array} \right) \right). \end{aligned}$$

Multiplying both sides of the latter equality to the right by the subspace $\langle e_{n-(m-1)}, e_{n-(m-2)}, \dots, e_{n-1}, e_n \rangle$ leads to

$$\begin{aligned} A^{(m)} \langle e_{n-(m-1)}, e_{n-(m-2)}, \dots, e_n \rangle &= \\ D^{(m)} \langle e_{n-(m-1)}, e_{n-(m-2)}, \dots, e_n \rangle &+ Q_m \langle e_{n-(m-1)}, e_{n-(m-2)}, \dots, e_n \rangle. \end{aligned}$$

This implies that

$$(A^{(m)} - D^{(m)}) \langle e_{n-(m-1)}, \dots, e_n \rangle = \langle q_{n-(m-1)}^{(m)}, \dots, q_n^{(m)} \rangle, \quad (3)$$

with $q_i^{(m)}$ the i th column of Q_m . The left-hand side can be rewritten as

$$\begin{aligned} (A^{(m)} - D^{(m)}) \langle e_{n-(m-1)}, e_{n-(m-2)}, \dots, e_n \rangle &= \\ \langle (A^{(m)} - D^{(m)})e_{n-(m-1)}, \dots, (A^{(m)} - D^{(m)})e_n \rangle &= \\ \langle (A^{(m)} - d_1 I_n)e_{n-(m-1)}, \dots, (A^{(m)} - d_m I_n)e_n \rangle. \end{aligned}$$

Hence, the completion of each m th step ($m = 2, \dots, n - 1$):

$$A^{(m+1)} = Q_m^T A^{(m)} Q_m,$$

can also be considered as a change of coordinate system: transform any vector y of the old system into $Q_m^T y$ for the new system. Then $A^{(m)}$ will be transformed into $A^{(m+1)}$ and the subspace of Equation (3): $\langle q_{n-(m-1)}^{(m)}, q_{n-(m-2)}^{(m)}, \dots, q_n^{(m)} \rangle$ will become

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

Therefore, at each step the basis remains the same but the matrix used changes. It is called a nested subspace iteration because the subspace involved increases in each step of the algorithm.

The subspace iteration involved for the semiseparable reduction is just a regular subspace iteration without the shift (see e.g. [1]), as the chasing technique is not involved in the tridiagonal case, no subspace iteration is involved for the tridiagonal matrices. In the next section, we will investigate in more detail the resulting convergence behavior, subject to this subspace iteration.

4 More on the subspace iteration

In this section we will give some theoretical results which might help us to choose the diagonal in the reduction to semiseparable plus diagonal form in such a way that we can tune the convergence behavior. In the last section numerical experiments are given to illustrate these results. The results we will give here are based on the convergence properties of a generic *GR*-algorithm as derived in [16,17].

4.1 The reduction as a nested multi-shift iteration

In this section we will rewrite the subspace iteration as presented in the previous section, such that it can be interpreted as a nested multi-shift iteration.

Related to the diagonal elements d_i , used in the reduction algorithm we define the following monic polynomials $p_i(\lambda) = \lambda - d_i$. The monic polynomial $\hat{p}_i(\lambda)$ of degree i represents a multiplication of all the polynomials p_i, \dots, p_1 , i.e.

$$\begin{aligned} \hat{p}_i(\lambda) &= p_i(\lambda)p_{i-1}(\lambda) \dots p_1(\lambda) \\ &= (\lambda - d_i)(\lambda - d_{i-1}) \dots (\lambda - d_1). \end{aligned}$$

Moreover we also need partial combinations of these polynomials. Define the polynomials $\hat{p}_{j:i}(\lambda)$ with $(j \geq i)$ in the following way:

$$\hat{p}_{j:i}(\lambda) = p_j(\lambda)p_{j-1}(\lambda) \dots p_i(\lambda).$$

Note that $\hat{p}_i(\lambda) = \hat{p}_{i:1}(\lambda)$ and assume $\hat{p}_0 = \hat{p}_{0:0} = 1$.

Let us prove now the following theorem, which rewrites the subspace iteration behavior in terms of the original matrix. This theorem is an extension of Lemma 3.1 in [12].

Theorem 6 *Let us use the notation as defined before. At step $m = 1, 2, \dots, n - 1$ of the algorithm we have for every $n \geq k \geq n - m$, that (denote $\eta = k - n + m$)*

$$\begin{aligned} Q_{0:m} \langle e_k, \dots, e_n \rangle &= \hat{p}_\eta(A) \langle f_k^{(m)}, \hat{p}_{\eta+1:\eta+1}(A) f_{k+1}^{(m)}, \dots \\ &\quad \dots, \hat{p}_{j-n+m:\eta+1}(A) f_j^{(m)}, \dots, \hat{p}_{m:\eta+1}(A) f_n^{(m)} \rangle, \end{aligned}$$

where the vectors $f_k^{(m)}$ are defined as follows. For every m , $f_n^{(m)} = e_n$. For $n \geq k > n - m$, with $\eta = k - n + m$:

$$f_k^{(m)} \in \langle f_k^{(m-1)}, \hat{p}_{\eta:\eta}(A)f_{k+1}^{(m-1)}, \dots, \hat{p}_{m-1:\eta}(A)f_n^{(m-1)} \rangle,$$

and the vector $f_{n-m}^{(m)}$ equals $Q_{0:m}e_{n-m}$ and is hence orthogonal to the subspace

$$\hat{p}_1(A)\langle f_{n-m+1}^{(m)}, \hat{p}_{\eta+1:2}(A)f_{k+1}^{(m)}, \dots, \hat{p}_{j-n+m:2}(A)f_j^{(m)}, \dots, \hat{p}_{m:2}(A)f_n^{(m)} \rangle.$$

Proof We will prove the theorem by induction on m .

- **Step $m = 0$:** We will prove that for every $n \geq k \geq n - 0$ (i.e. $k = n$) the following holds:

$$Q_{0:0}\langle e_n \rangle = \langle f_n^{(0)} \rangle.$$

This is straightforward, by choosing $f_n^{(0)}$ equal to the last column of $Q_{0:0}$. We remark that $Q_{0:0}$ is an initial transformation, which in fact does not explicitly needs to be applied on the matrix A , to reduce it to semiseparable plus diagonal form. In the reduction method we proposed $Q_{0:0} = I$ and hence $f_n^{(0)} = e_n$.

- **Step $m = 1$:** Before starting the induction procedure on m , we will demonstrate the case $m = 1$. We have to prove two things: for $k = n$:

$$\begin{aligned} Q_{0:1}\langle e_n \rangle &= (A - d_1 I)\langle f_n^{(1)} \rangle \\ &= \hat{p}_1(A)\langle f_n^{(1)} \rangle \end{aligned}$$

and for $k = n - 1$:

$$\begin{aligned} Q_{0:1}\langle e_{n-1}, e_n \rangle &= \langle f_{n-1}^{(1)}, (A - d_1 I)f_n^{(1)} \rangle \\ &= \langle f_{n-1}^{(1)}, \hat{p}_1(A)f_n^{(1)} \rangle. \end{aligned}$$

We will first prove that the equation holds for $k = n$.

When the transformation Q_1 used at the first step, is only applied on the rows, the matrix $A^{(1)} = Q_{0:0}^T A Q_{0:0}$ is transformed into:

$$\begin{aligned} Q_1^T A^{(1)} &= Q_1^T (D^{(1)} + S^{(1)}) = Q_1^T D^{(1)} + L^{(1)} \\ &= Q_1^T \left(\begin{array}{c|c} 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ \hline 0 \dots 0 & d_1 \end{array} \right) + \left(\begin{array}{c|c} \times \dots \times & 0 \\ \vdots & \vdots \\ \times \dots \times & 0 \\ \hline \times \dots \times & \times \end{array} \right), \quad (4) \end{aligned}$$

with all the elements of the strictly upper-triangular part of the last column of $L^{(1)}$ zero.

Hence, combining Equation (4) with

$$Q_1^T A^{(1)} = Q_1^T (Q_{0:0}^T A Q_{0:0}) = Q_{0:1}^T A Q_{0:0}$$

implies that

$$A Q_{0:0} - Q_{0:0} D^{(1)} = Q_{0:1} L^{(1)}. \quad (5)$$

Multiplying both sides of Equation (5) to the right by $\langle e_n \rangle$ and using the knowledge for $m = 0$, leads to:

$$\begin{aligned} (A Q_{0:0} - Q_{0:0} D^{(1)}) \langle e_n \rangle &= (Q_{0:1} L^{(1)}) \langle e_n \rangle = Q_{0:1} \langle e_n \rangle \\ &= (A Q_{0:0} - Q_{0:0} d_1 I) \langle e_n \rangle \\ &= (A - d_1 I) Q_{0:0} \langle e_n \rangle \\ &= (A - d_1 I) \langle f_n^{(0)} \rangle \\ &= \hat{p}_1(A) \langle f_n^{(0)} \rangle = \hat{p}_1(A) \langle f_n^{(1)} \rangle, \end{aligned}$$

with $f_n^{(0)} = f_n^{(1)}$. This completes the case $k = n$. Using this equation, the case $k = n-1$ is straightforward. Taking $f_{n-1}^{(1)} = Q_{0:1} e_{n-1}$ immediately leads to:

$$Q_{0:1} \langle e_{n-1}, e_n \rangle = \langle f_{n-1}^{(1)}, \hat{p}_1(A) f_n^{(1)} \rangle$$

Moreover, we also have that $f_{n-1}^{(1)}$ is orthogonal to $\hat{p}_1(A) f_n^{(1)}$.

- **Step m :** We will prove now the general formulation, assuming that the case $m-1$ holds for every $n \geq k \geq n-m-1$. So we know that for every $n \geq k \geq n-m-1$, the following equation is true (denote $\eta = k-n+m$)²:

$$\begin{aligned} &Q_{0:m-1} \langle e_k, \dots, e_n \rangle \\ &= \hat{p}_{\eta-1}(A) \langle f_k^{(m-1)}, \hat{p}_{\eta:\eta}(A) f_{k+1}^{(m-1)}, \dots, \hat{p}_{m-1:\eta}(A) f_n^{(m-1)} \rangle. \end{aligned}$$

To prove the case m , we will distinguish two cases, namely $n \geq k > n-m$ and $k = n-m$.

² Remark that the definition of η is slightly different than the one from the theorem, this is done to obtain the final formulation in the correct form.

We start with the case $n \geq k > n - m$, in a similar way as for $m = 1$. When the transformation Q_m used at the m th step, is only applied on the rows, the matrix $A^{(m)}$ is transformed into:

$$\begin{aligned} Q_m^T A^{(m)} &= Q_m^T (D^{(m)} + S^{(m)}) = Q_m^T D^{(m)} + L^{(m)} \\ &= Q_m^T \left(\begin{array}{c|ccc} 0 & & & \\ & \ddots & & \\ & & 0 & \\ \hline & & d_1 & \\ & & & \ddots \\ & & & & d_m \end{array} \right) + \left(\begin{array}{c|ccc} \times \dots \times & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ \times \dots \times & 0 & \dots & 0 \\ \hline \times \dots \times & \times \dots & & 0 \\ \vdots & \vdots & \vdots & \ddots \\ \times \dots \times & \times \dots & & \times \end{array} \right), \end{aligned} \quad (6)$$

with all the elements of the strictly upper-triangular part of the last m columns of $L^{(m)}$ zero.

Hence, combining Equation (6) with

$$Q_m^T A^{(m)} = Q_m^T (Q_{0:m-1}^T A Q_{0:m-1}) = Q_{0:m}^T A Q_{0:m-1}$$

implies that

$$A Q_{0:m-1} - Q_{0:m-1} D^{(m)} = Q_{0:m} L^{(m)}. \quad (7)$$

Multiplying now Equation (7) on the right by $\langle e_k, \dots, e_n \rangle$ leads to:

$$\begin{aligned} &Q_{0:m} \langle e_k, \dots, e_n \rangle \\ &= (Q_{0:m} L^{(m)}) \langle e_k, \dots, e_n \rangle \\ &= (A Q_{0:m-1} - Q_{0:m-1} D^{(m)}) \langle e_k, \dots, e_n \rangle \\ &= \langle (A Q_{0:m-1} - Q_{0:m-1} D^{(m)}) e_k, \dots, (A Q_{0:m-1} - Q_{0:m-1} D^{(m)}) e_n \rangle \\ &= \langle (A - d_1 I) Q_{0:m-1} e_k, \dots, (A - d_m I) Q_{0:m-1} e_n \rangle. \end{aligned} \quad (8)$$

We know by induction, that for every k , with $k \leq j \leq n$ the following equation is true (denote $\eta_j = j - n + m$):

$$\begin{aligned} &Q_{0:m-1} \langle e_j, \dots, e_n \rangle \\ &= \hat{p}_{\eta_j-1}(A) \langle f_j^{(m-1)}, \hat{p}_{\eta_j:\eta_j}(A) f_{j+1}^{(m-1)}, \dots, \hat{p}_{m-1:\eta_j}(A) f_n^{(m-1)} \rangle. \end{aligned}$$

So we can write:

$$Q_{0:m-1} \langle e_j \rangle = \hat{p}_{\eta_j-1}(A) \langle f_j^{(m)} \rangle$$

where $f_j^{(m)}$ is a suitably chosen vector such that

$$f_j^{(m)} \in \langle f_j^{(m-1)}, \hat{p}_{\eta_j:\eta_j}(A) f_{j+1}^{(m-1)}, \dots, \hat{p}_{m-1:\eta_j}(A) f_n^{(m-1)} \rangle.$$

Using now this relation for every vector $Q_{0:m-1}e_j$ in Equation (8), we get the following relations:

$$\begin{aligned} & \langle (A - d_\eta I)Q_{0:m-1}e_k, \dots, (A - d_m I)Q_{0:m-1}e_n \rangle \\ &= \langle (A - d_\eta I)\hat{p}_{\eta_k-1}(A)f_k^{(m)}, \dots, (A - d_m I)\hat{p}_{\eta_n-1}(A)f_n^{(m)} \rangle, \\ &= \langle \hat{p}_{\eta_k}(A)f_k^{(m)}, \dots, \hat{p}_{\eta_n}(A)f_n^{(m)} \rangle, \\ &= \hat{p}_\eta(A)\langle f_k^{(m)}, \hat{p}_{\eta+1:\eta+1}(A)f_{k+1}^{(m)}, \dots, \hat{p}_{m:\eta+1}(A)f_n^{(m)} \rangle. \end{aligned}$$

Proving thereby the theorem for $k > n - m$. The case $k = n - m$ is again straightforward by defining $f_{n-m}^{(m)}$ as $Q_{0:m}e_{n-m}$.

This means that at step m of the reduction algorithm we perform for every $n \geq k \geq n - m$ a multi-shift iteration on the subspace $\langle f_k^{(m)}, \dots, f_n^{(m)} \rangle$. This is called a nested type of multi-shift iteration. Under mild assumptions we will therefore get a similar convergence behavior as in the multi-shift case. Before giving some reformulations of Theorem 6. We will give a first intuitive interpretation to this convergence behavior. Let us write down for some k , the different formulas

$$Q_{0:m}\langle e_n \rangle = \langle \hat{p}_m(A)f_n^{(m)} \rangle \tag{9}$$

$$Q_{0:m}\langle e_{n-1}, e_n \rangle = \langle \hat{p}_{m-1}(A)f_{n-1}^{(m)}, \hat{p}_m(A)f_n^{(m)} \rangle \tag{10}$$

$$Q_{0:m}\langle e_{n-2}, e_{n-1}, e_n \rangle = \langle \hat{p}_{m-2}(A)f_{n-2}^{(m)}, \hat{p}_{m-1}(A)f_{n-1}^{(m)}, \hat{p}_m(A)f_n^{(m)} \rangle$$

We will assume, for simplicity reasons, that the vectors $f_k^{(m)}$ do not have a significant influence on the convergence behavior. This means that they do not have a small or zero component in the direction we want them to converge too. Further on in the text we will investigate in more detail the influence of these vectors $f_k^{(m)}$, w.r.t. the convergence speed. Under this assumption, according to Equation (9), the last vector of $Q_{0:m}$ will converge (for increasing m) towards the eigenvector of the matrix $\hat{p}_m(A)$ corresponding to the dominant eigenvalue of $\hat{p}_m(A)$. Combining this with Equation (10) shows us that $Q_{0:m}e_{n-1}$ will converge to the eigenvector perpendicular to the vector $Q_{0:m}e_n$ and corresponding to the dominant eigenvalue for $\hat{p}_{m-1}(A)$. Similarly a combination of all above equations reveals that $Q_{0:m}e_{n-2}$ converges to an eigenvector perpendicular to the above two and corresponding to the dominant eigenvalue for $\hat{p}_{m-2}(A)$. More details on the convergence behavior will be given in Section 6.

Below, we have rewritten Theorem 6 in different forms, to illustrate more clearly what happens, and to make different interpretations of the method possible:

- **Formulation 1: The shift through form.** We drag the polynomial in front of the subspace completely through the subspace.

Corollary 1 *Let us use the notation as defined before. At step m of the algorithm we have for every $n \geq k \geq n - m$: (denote $\eta = k - n + m$):*

$$Q_{0:m}\langle e_k, \dots, e_n \rangle = \langle \hat{p}_\eta(A)f_k^{(m)}, \hat{p}_{\eta+1}(A)f_{k+1}^{(m)}, \dots, \hat{p}_m(A)f_n^{(m)} \rangle.$$

This means that on every vector, a form of the shifted power method is applied and the vectors are re-orthogonalized w.r.t. each other.

- **Formulation 2: The nested shift formulation.** We can also reformulate the theorem such that we apply on each nested subspace an iteration with shift.

Corollary 2 *Let us use the notation as defined before. At step m of the algorithm we have for every $n \geq k \geq n - m$, (denote $\eta = k - n + m$)*

$$\begin{aligned} Q_{0:m}\langle e_k, \dots, e_n \rangle \\ = \hat{p}_\eta(A)\langle f_k^{(m)}, p_{\eta+1}(A)\langle f_{k+1}^{(m)}, p_{\eta+2}(A)\langle f_{k+2}^{(m)}, \dots, p_m(A)f_n^{(m)} \rangle \dots \rangle \rangle, \end{aligned}$$

which can be rewritten as:

$$\begin{aligned} Q_{0:m}\langle e_k, \dots, e_n \rangle \\ = \hat{p}_\eta(A)\langle f_k^{(m)}, (A - d_{\eta+1}I)\langle f_{k+1}^{(m)}, (A - d_{\eta+2}I)\langle f_{k+2}^{(m)}, \dots \\ \dots, (A - d_mI)f_n^{(m)} \rangle \dots \rangle \rangle. \end{aligned}$$

So we can see, that on each nested subspace an iteration with shift is performed.

- **Formulation 3: The nested QL -iteration with shift.** Theorem 6 as presented above incorporates all the transformations in one orthogonal matrix $Q_{0:m}$. If we perform, however, after each step in the reduction algorithm the basis transformation, this corresponds to performing a similarity transformation, leading to a different form of the theorem. This formulation corresponds to a shifted QL -iteration on the matrix A .

We already know from the results in Section 3.2, that we can interpret the reduction method as a nested subspace iteration, as follows:

$$(A^{(m)} - D^{(m)})\langle e_{n-(m-1)}, \dots, e_n \rangle = \langle q_{n-(m-1)}^{(m)}, \dots, q_n^{(m)} \rangle.$$

The interpretation of this type of iteration is not straightforward as we are not subtracting a shift matrix from the matrix $A^{(m)}$, but a diagonal matrix. This interpretation says that at every step m a subspace iteration of the matrix $(A^{(m)} - D^{(m)})$ is performed on the space $\langle e_{n-(m-1)}, \dots, e_n \rangle$, which gives us part of the orthogonal matrix Q_m . If a column of this matrix Q_m is close enough to an eigenvector of the matrix $A^{(m)}$, this will be visible after performing the basis transformation $Q_m^T A^{(m)} Q_m$. (For more information see [16].)

Using Theorem 6, we can reformulate this nested iteration, towards an iteration with a shift and another subspace on which one iterates.

We know that $Q_{0:m-1}^T A Q_{0:m-1} = A^{(m)}$. One can also easily prove the following relations:

$$\begin{aligned} Q_{0:m-1}^T p_i(A) Q_{0:m-1} &= p_i(A^{(m)}), \\ Q_{0:m-1}^T \hat{p}_i(A) Q_{0:m-1} &= \hat{p}_i(A^{(m)}), \\ Q_{0:m-1}^T \hat{p}_{j:i}(A) Q_{0:m-1} &= \hat{p}_{j:i}(A^{(m)}). \end{aligned}$$

If we perform now the basis transformation corresponding to the orthogonal matrix $Q_{0:m-1}$ on the subspace $Q_{0:m} \langle e_k, \dots, e_n \rangle$, we get the following relations:

$$\begin{aligned} Q_m \langle e_k, \dots, e_n \rangle &= Q_{0:m-1}^T Q_{0:m} \langle e_k, \dots, e_n \rangle \\ &= Q_{0:m-1}^T \hat{p}_\eta(A) \langle f_k^{(m)}, \hat{p}_{\eta+1:\eta+1}(A) f_{k+1}^{(m)}, \dots, \hat{p}_{m:\eta}(A) f_n^{(m)} \rangle \\ &= \hat{p}_\eta(A^{(m)}) \langle Q_{0:m-1}^T f_k^{(m)}, \hat{p}_{\eta+1:\eta+1}(A^{(m)}) Q_{0:m-1}^T f_{k+1}^{(m)}, \dots \\ &\quad \dots, \hat{p}_{m:\eta}(A^{(m)}) Q_{0:m}^T f_n^{(m)} \rangle \end{aligned}$$

We can formulate the following equivalent corollary with $\hat{f}_j^{(m)} = Q_{0:m-1}^T f_j^{(m)}$.

Corollary 3 *Let us use the notation as defined before. At step m of the algorithm we have for every $n \geq k \geq n - m$, (denote $\eta = k - n + m$)*

$$\begin{aligned} Q_m \langle e_k, \dots, e_n \rangle &= \hat{p}_\eta(A^{(m)}) \langle \hat{f}_k^{(m)}, \hat{p}_{\eta+1:\eta+1}(A^{(m)}) \hat{f}_{k+1}^{(m)}, \dots \\ &\quad \dots, \hat{p}_{j-n+m:\eta+1}(A^{(m)}) \hat{f}_j^{(m)}, \dots, \hat{p}_{m:\eta}(A^{(m)}) \hat{f}_n^{(m)} \rangle \end{aligned}$$

In this way we know that a partial QL -iteration, with partial we mean, a subspace of dimension less than n , is performed on the subspace defined by the vectors $\hat{f}_j^{(m)}$ and the last columns of Q_m span this space.

This means that if a column of Q_m is close to an eigenvector of $A^{(m)}$ the basis transformation will reveal it. But of course the convergence behavior is heavily dominated by the vectors $\hat{f}^{(m)}$. This will be investigated further on in the text. For a traditional convergence behavior, related to QR and QL -iterations, these subspaces are always equal to $\langle e_1, \dots, e_n \rangle$, and one can assume (in most cases), that these vectors do not heavily influence the convergence speed. Here however these vectors of the subspace are constructed in a specific way, and do have an important impact on the convergence behavior.

Of course we can also reformulate this last theorem, w.r.t. the first two formulations in this list.

Before investigating in more detail the convergence speed, and the interaction between the Ritz-value convergence behavior and the subspace iteration, we will translate the theorem towards the semiseparable and tridiagonal case.

1. In the tridiagonal case, there is no chasing step involved, as all the performed Givens transformations are equal to the identity. Hence the lower right already reduced part of the matrix, contains the Lanczos-Ritz values, and no subspace iteration is performed.
2. In a previous publication it was stated that on the lower right part of the semiseparable matrix always a step of non-shifted subspace iteration was performed. We get exactly this behavior if we take the shift equal to zero: Theorem 6 is therefore an extension of Lemma 3.1 in [12].

Theorem 7 *Let us use the notation as defined before. At step $m = 1, 2, \dots, n - 1$ of the algorithm we have for every $n \geq k \geq n - m$, that (denote $\eta = k - n + m$)*

$$\begin{aligned} & Q_{0:m} \langle e_k, \dots, e_n \rangle \\ &= A^{k-1} \langle f_k^{(m)}, Af_{k+1}^{(m)}, \dots, A^{j-n+m} f_j^{(m)}, \dots, A^{(m)} f_n^{(m)} \rangle \end{aligned}$$

Using Corollary 3 one can easily explain the convergence behavior as observed in [2]. In this paper, it was observed, that the reduction of a symmetric matrix into a similar semiseparable plus diagonal one, with the top left elements of the diagonal equal to eigenvalues of this matrix, revealed these eigenvalues. More precisely the transformed

semiseparable plus diagonal matrix was in the upper left $k \times k$ block diagonal, where k is the number of top left diagonal elements equal to eigenvalues of the original matrix. This is natural, as after the complete reduction method on the complete matrix a step of the QL -iteration with shift d_1 is performed. If d_1 equals an eigenvalue, we have a perfect shift, and this will be revealed in the upper left position. If we continue now with the trailing $(n - 1) \times (n - 1)$ block of this matrix, we know that this matrix has as eigenvalues, the same eigenvalues as the original matrix without the eigenvalue d_1 . As on this matrix a QL -iteration with shift d_2 is performed, with d_2 and eigenvalue, the procedure, will again reveal this eigenvalue. This process continues as long as the first d_1, \dots, d_k diagonal elements are equal to the eigenvalues of the original matrix. As soon as one diagonal element does not correspond anymore to an eigenvalue, the procedure stops.

Let us investigate in more detail now the relation between the subspace iteration and the Lanczos-Ritz values.

5 The interaction between the subspace iteration and the Lanczos-Ritz values

In the previous two sections, we investigated two convergence behaviors of the reduction to semiseparable plus diagonal form. In this section we will prove the following behavior:

The nested multi-shift iteration will start converging as soon as the Lanczos-Ritz values approximate well enough the dominant eigenvalues with respect to the multi-shift iteration.

Let us use the notation as defined before. At step $m = 1, 2, \dots, n - 1$ of the algorithm we have for every $n \geq k \geq n - m$, that (denote $\eta = k - n + m$)

$$Q_{0:m} \langle e_k, \dots, e_n \rangle = \hat{p}_\eta(A) \langle f_k^{(m)}, \hat{p}_{\eta+1:\eta+1}(A) f_{k+1}^{(m)}, \dots, \hat{p}_{m:\eta+1}(A) f_n^{(m)} \rangle$$

Moreover, we also have that, due to the Lanczos-Ritz value convergence

$$Q_{0:m} \langle e_{n-m}, \dots, e_n \rangle = \mathcal{K}_{m+1}(A, Q_0 e_n).$$

Clearly the following relation holds between the two above presented subspaces, for all k :

$$Q_{0:m} \langle e_k, \dots, e_n \rangle \subset Q_{0:m} \langle e_{n-m}, \dots, e_n \rangle.$$

These relations do exactly explain the behavior as presented above. The multi-shift subspace iteration works on the vectors $f_j^{(i)}$, but they are constructed in such a way that after the subspace iteration we get a subspace which is part of the Krylov subspace $\mathcal{K}_{m+1}(A, Q_0 e_n)$. As long as this Krylov subspace is not large enough, to contain the eigenvectors corresponding to the dominant eigenvalues of the matrix polynomials $\hat{p}_{j-n+m:1}(A)$, the subspace iteration can simply not converge to these eigenvectors.

As soon as the dominant eigenvectors, w.r.t. the multishift polynomial, will be present in the Krylov subspace, the Ritz-values will approximate the corresponding eigenvalues and this means that the multi-shift iteration can start converging to these eigenvalues / eigenvectors. This behavior will be illustrated in the numerical experiments.

6 The convergence speed of the nested multi-shift iteration

In this section we will present some theorems concerning the speed of convergence, using the nested multi-shift QL -iteration as presented in this paper. In a first part we present some theorems from [16,17], which are useful for traditional GR -algorithms. In the second part, we apply these theorems to our nested subspace formulation.

6.1 General subspace iteration theory

First we will reconsider some general results concerning the distances between subspaces. A more elaborate study can be found in [16,3]. Given two subspaces \mathcal{S} and \mathcal{T} in \mathbb{R}^n and denote with $P_{\mathcal{S}}$ and $P_{\mathcal{T}}$ the orthonormal projector onto the subspace \mathcal{S} and \mathcal{T} respectively. The standard metric between subspaces (see [3]) is defined as

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

if $\dim(\mathcal{S}) = \dim(\mathcal{T})$ and $d(\mathcal{S}, \mathcal{T}) = 1$ otherwise.

The next theorem states how the distance between subspaces changes, when performing subspace iteration with shifted polynomials.

Theorem 8 (Theorem 5.1 from [16]) *Given a simple³ matrix $A \in \mathbb{R}^{n \times n}$ with eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$ and associated linearly*

³ A matrix is called simple if it has n linearly independent eigenvectors.

independent eigenvectors v_1, v_2, \dots, v_n . Let $V = [v_1, v_2, \dots, v_n]$ and κ_V is the condition number of V , w.r.t. to the spectral⁴ norm. Let l be an integer $1 \leq l \leq n - 1$, and define the invariant subspaces $\mathcal{U} = \langle v_1, \dots, v_{l-1} \rangle$ and $\mathcal{T} = \langle v_l, \dots, v_n \rangle$. Denote with (p_i) a sequence of polynomials and let $\hat{p}_i = p_i \dots p_2 p_1$. Suppose that

$$\hat{p}_i(\lambda_j) \neq 0 \quad j = l, \dots, n$$

for all i , and let

$$r_i = \frac{\max_{1 \leq j \leq l-1} |\hat{p}_i(\lambda_j)|}{\min_{l \leq j \leq n} |\hat{p}_i(\lambda_j)|}.$$

Let \mathcal{S} be a k -dimensional subspace of \mathbb{R}^n , satisfying

$$\mathcal{S} \cap \mathcal{U} = \{0\}.$$

Let $\mathcal{S}_i = \hat{p}_i(A)\mathcal{S}, i = 1, 2, \dots$. Then there exists a constant C (depending on \mathcal{S}) such that for all i ,

$$d(\mathcal{S}_i, \mathcal{T}) \leq C \kappa_V r_i.$$

In particular $\mathcal{S}_i \rightarrow \mathcal{T}$ if $r_i \rightarrow 0$. More precisely we have that

$$C = \frac{d(V^{-1}\mathcal{S}, V^{-1}\mathcal{T})}{\sqrt{1 - d(V^{-1}\mathcal{S}, V^{-1}\mathcal{T})}}$$

We remark, that similar theorems exist for defective matrices. Also more information concerning the conditions put on the matrices in Theorem 8, can be found in [16]. We will however not go into these details.

The following lemma relates the subspace convergence, towards the vanishing of certain subblocks in a matrix.

Lemma 1 (Lemma 6.1 from [16]) *Suppose $A \in \mathbb{R}^{n \times n}$ is given, and let \mathcal{T} be a subspace, which is invariant under A . Assume G to be a nonsingular matrix and assume \mathcal{S} to be the subspace spanned by the last k columns of G . (The subspace \mathcal{S} can be seen as an approximation of the subspace \mathcal{T} .) Assume $B = G^{-1}AG$, and consider the matrix B , partitioned in the following way:*

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix},$$

where $B_{21} \in \mathbb{R}^{k \times (n-k)}$. Then we have:

$$\|B_{21}\|_2 \leq 2 \sqrt{2} \kappa_G \|A\|_2 d(\mathcal{S}, \mathcal{T}),$$

where κ_G denotes the condition number of the matrix G .

⁴ The spectral norm is naturally induced by the $\|\cdot\|_2$ norm on vectors.

We are now ready to use these theorems, to derive an upper bound on the norm of the subblocks, while reducing a matrix to semiseparable plus diagonal form.

6.2 Application to the nested multi-shift iteration

Let us apply the above theorems to our specific case, and see how we can derive convergence results for the reduction to semiseparable plus diagonal form.

Let us assume we are working with a symmetric matrix A (which is naturally simple), with eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$. The associated linear independent eigenvectors are denoted by v_1, v_2, \dots, v_n . As we proved before, in Section 5, the subspace iteration will only start working, as soon as the Lanczos-Ritz values, approximate well enough the dominant eigenvalues, w.r.t. the multi-shift polynomial. In this section however, we do not need to worry about the Lanczos convergence behavior. Our theoretical upper bound for the convergence speed, will naturally incorporate this Lanczos influence on the convergence.

Let $\mathcal{T} = \langle v_l, v_{l+1}, \dots, v_n \rangle$ and $\mathcal{U} = \langle v_1, v_2, \dots, v_{l-1} \rangle$. In this section we will derive an upper bound for the convergence towards the subspace \mathcal{T} .

The outcome of step m in the reduction algorithm is the matrix $A^{(m+1)} = Q_{0:m}^T A^{(0)} Q_{0:m}$, and we are interested in small subblocks of this matrix. (Assume $m \geq n - l + 1$, otherwise there are not yet subspace iteration steps performed on the lower right $(n - l + 1) \times (n - l + 1)$ block.) Using Lemma 1, we know that this is related to the orthogonal transformation matrix $Q_{0:m}$. Partition the matrix $A^{(m+1)}$ in the following way:

$$A^{(m+1)} = \begin{bmatrix} A_{11}^{(m+1)} & A_{12}^{(m+1)} \\ A_{21}^{(m+1)} & A_{22}^{(m+1)} \end{bmatrix},$$

where $A_{22}^{(m+1)}$ is of size $(n - l + 1) \times (n - l + 1)$. Denote with $\hat{\mathcal{S}}$ the space spanned by the last $n - l + 1$ components of $Q_{0:m}$. (Hence $\hat{\mathcal{S}} = Q_{0:m} \langle e_l, \dots, e_n \rangle$.) Then we have by Lemma 1 that

$$\|A_{21}^{(m+1)}\|_2 \leq 2 \sqrt{2} \|A^{(0)}\|_2 d(\hat{\mathcal{S}}, \mathcal{T}).$$

as $\kappa_2 = 1$, because $Q_{0:m}$ is an orthogonal matrix.

To determine the distance between $\hat{\mathcal{S}}$ and \mathcal{T} one can apply Theorem 8. As we are in step m of the reduction algorithm, we can apply

Theorem 6 for $k = l$, this means that (with $\eta = l - n + m$):

$$Q_{0:m} \langle e_l, \dots, e_n \rangle = \hat{p}_\eta(A) \langle f_l^{(m)}, \hat{p}_{\eta+1:\eta+1}(A) f_{l+1}^{(m)}, \dots, \hat{p}_{m:\eta+1}(A) f_n^{(m)} \rangle.$$

This

$$\hat{\mathcal{S}} = \hat{p}_\eta(A) \mathcal{S},$$

with

$$\mathcal{S} = \langle f_l^{(m)}, \hat{p}_{\eta+1:\eta+1}(A) f_{l+1}^{(m)}, \dots, \hat{p}_{m:\eta+1}(A) f_n^{(m)} \rangle.$$

Applying Theorem 8 gives us the following upper bound for the distance between $\hat{\mathcal{S}}$ and \mathcal{T} . For

$$r = \frac{\max_{1 \leq j \leq l-1} |\hat{p}_\eta(\lambda_j)|}{\min_{l \leq j \leq n} |\hat{p}_\eta(\lambda_j)|},$$

the following upper bound is obtained

$$d(\hat{\mathcal{S}}, \mathcal{T}) \leq C r,$$

where

$$C = \frac{d(V^{-1}\mathcal{S}, V^{-1}\mathcal{T})}{\sqrt{1 - d(V^{-1}\mathcal{S}, V^{-1}\mathcal{T})}}.$$

Summarizing this deduction we get that the norm $\|A_{21}^{(m+1)}\|_2$ is bounded as follows:

$$\begin{aligned} & \|A_{21}^{(m+1)}\|_2 & (11) \\ & \leq 2\sqrt{2} \|A^{(0)}\|_2 \left(\frac{d(V^{-1}\mathcal{S}, V^{-1}\mathcal{T})}{\sqrt{1 - d(V^{-1}\mathcal{S}, V^{-1}\mathcal{T})}} \right) \left(\frac{\max_{1 \leq j \leq l-1} |\hat{p}_\eta(\lambda_j)|}{\min_{l \leq j \leq n} |\hat{p}_\eta(\lambda_j)|} \right). \end{aligned}$$

If one is interested in the bound for the next iterate $m + 1$, one has to use in fact another subspace $\tilde{\mathcal{S}}$. But, due to the specific structure of the vectors $f_j^{(m)}$ (see Theorem 6), the subspaces \mathcal{S} and $\tilde{\mathcal{S}}$ span the same space. Hence, the distance remains the same, and only the polynomials in Formula 11 determine the change in norm of the subblock. This means that once the subspace iteration starts working on a specific part, one can calculate the constant C , and it will not change anymore.

In practice, the constant C , can be very large as long as the dominant eigenvectors, w.r.t. the polynomial p_η , are not present in the Krylov subspace, and hence the Lanczos-Ritz values are not close enough to the dominant eigenvalues, w.r.t. the polynomial p_η . This

constant can create a delay in the convergence of the subspace iteration behavior. The influence of the Lanczos convergence behavior on the subspace iteration is therefore captured in the constant C .

Let us give a traditional example on the convergence speed. We will only present the results, more information can be found in [16]. The polynomial considered here, namely $\hat{p}_\eta(\lambda)$, is in fact a multiplication between several polynomials:

$$\hat{p}_\eta(\lambda) = p_\eta(\lambda)p_{\eta-1}(\lambda) \cdots p_2(\lambda)p_1(\lambda).$$

Assume all $p_i(\lambda) = p(\lambda) = \lambda - d$, this means, that we always consider the same shift d . If $d = 0$, we get the power method. Ordering the eigenvalues such that: $|p(\lambda_1)| \leq |p(\lambda_2)| \leq \dots \leq |p(\lambda_n)|$. Assume l to be chosen such that

$$\rho = \frac{\max_{1 \leq j \leq l-1} |p(\lambda_j)|}{\min_{l \leq j \leq n} |p(\lambda_j)|} = \frac{|p(\lambda_{l-1})|}{|p(\lambda_l)|} < 1,$$

then we get that $r = \rho^\eta$, and hence we get linear convergence.

In the last section on the numerical experiments, we calculate some of these bounds in real experiments and we observe that this is a valuable and useful upper bound in practice. Moreover, we will see that one can use Formula 11 to predict possible convergence behavior to eigenvalues.

7 Numerical experiments

In this section, numerical experiments are given, illustrating the theoretical results presented in this paper. Several types of experiments will be performed. We will investigate the delay of convergence caused by the Lanczos-Ritz values behavior, we will experimentally check the convergence speed of the subspace iteration and we will present some experiments in which the diagonal is chosen in such a way to reveal a specific part of the spectrum. Moreover, we will start the numerical experiments with a section, in which we compare the complexity of the different reduction algorithms.

All the experiments are performed in Matlab⁵. We use Matlab-style notation. With $\mathit{zeros}(i, j)$, we denote a zero matrix with i rows, and j columns, with $\mathit{ones}(i, j)$, we denote a matrix with all entries equal to 1 of dimension $i \times j$, with $\mathit{rand}(i, j)$ we denote a matrix of dimension $i \times j$, with entries random chosen from a uniform distribution between 0 and 1.

⁵ Matlab is a registered trademark of the Mathworks inc.

7.1 Complexity analysis

In this section we will briefly compare the complexity of computing the eigenvalues of a symmetric matrix, via the reduction methods covered in this manuscript. Let us start first with comparing the complexities of the reduction methods.

To reduce a symmetric matrix to tridiagonal form, we only need to perform Householder transformations. The cost of performing a symmetric Householder transformation on a matrix of size n equals $4n^2 + 8n + 7$ operations, leading to a global reduction cost of $4/3n^3 + \mathcal{O}(n^2)$. This cost is shared by the reduction to semiseparable form, and one cannot get rid of this n^3 term.

The cost of the chasing steps performed in the reduction to semiseparable and semiseparable plus diagonal form equals (for a block of size n):

$$\alpha(n - 1) + \beta,$$

with $\alpha = 18, \beta = 9$ for the reduction to semiseparable form, and $\alpha = 20, \beta = 13$, for the reduction to semiseparable plus diagonal form. We chose $\alpha = 0 = \beta$ in case the matrix is reduced to tridiagonal form.

During the reduction, deflation of blocks may occur. Deflating these blocks has its influence on the complexity of the chasing, but it also heavily influences the complexity of the algorithms for computing the eigenvalues, as they work on smaller blocks. Let us firstly investigate the reduction algorithms.

Suppose we reduce a matrix A to semiseparable (plus diagonal) form. During the reduction we obtain convergence to k blocks, which we can deflate. Suppose the k blocks have the following sizes, in order of deflation: $n_1, n_2, n_3, \dots, n_k$. Each of the blocks is deflated after $l_1, l_1 + l_2, l_1 + l_2 + l_3, \dots$ steps.

This means that the global cost of reducing a matrix to one of the three forms is the following one. In order to deflate the first block, we need to perform

$$\sum_{i=1}^{l_1} (\alpha(i - 1) + \beta + 4(n - i)^2 + 8(n - i) + 7)$$

operations. To deflate the second block, an extra

$$\sum_{i=1}^{l_2} (\alpha(i - 1 + l_1 - n_1) + \beta + 4(n - i - l_1)^2 + 8(n - i - l_1) + 7)$$

operations are needed. Globally, in order to deflate block q with $1 \leq q \leq k$ we need

$$\sum_{j=1}^q \left(\sum_{i=1}^{l_j} \left(\alpha(i-1 + \sum_{p=1}^{q-1} (l_p - n_p)) + \beta \right. \right. \\ \left. \left. + 4 \left(n - i - \sum_{p=1}^{q-1} l_1 \right)^2 + 8 \left(n - i - \sum_{p=1}^{q-1} l_1 \right) + 7 \right) \right)$$

operations in total. The terms in the first line are related to the chasing, and the terms in the second line are related to the Householder tridiagonalization.

We know now, the number of operations needed in order to deflate blocks inside the reduction algorithms. Remark, that in general, the reduction to tridiagonal form does not reveal blocks. In order to compare globally the eigenvalue solvers based on the tridiagonal reduction, and the reduction to semiseparable (plus diagonal) form, we need to have estimates on the complexity of the computation of eigenvalues of semiseparable (plus diagonal) and tridiagonal matrices.

We list here some of the complexities of algorithms for computing the eigenvalues of a matrix of size n .

- Standard QR -algorithms for tridiagonal matrices cost approximately $30n$ for one step of the QR -method, and it takes globally 2 steps to converge to an eigenvalue.
- Standard QR -algorithms for semiseparable (plus diagonal) matrices (see [18,19]) cost approximately $40n$ for one step, but they converge slightly faster, at approximately 1.7 steps.
- Divide and conquer for tridiagonal matrices (see [20,21]) takes $n^2 + \mathcal{O}(n)$ operations
- A Divide and conquer method for semiseparable (plus diagonal matrices) (see [7,8]), takes also approximately $n^2 + \mathcal{O}(n)$ operations.

Assume now that after a block has been separated in the reduction method, that we immediately compute its eigenvalues via one of the above methods. Once we have computed these eigenvalues, we continue the reduction until another block separates, of which we compute then the eigenvalues.

In the following figures we compare the speed of convergence of the above approaches for the computation of the eigenvalues based on the semiseparable plus diagonal and on the tridiagonal approach. On the vertical axis the number of eigenvalues computed is depicted,

and on horizontal axis, the number of flops which were needed to compute this amount of eigenvalues is presented. The algorithm used for computing the eigenvalues of both approaches is the standard QR -algorithms. We considered also the complexity w.r.t. the divide and conquer methods, but they were rather similar, and hence not included.

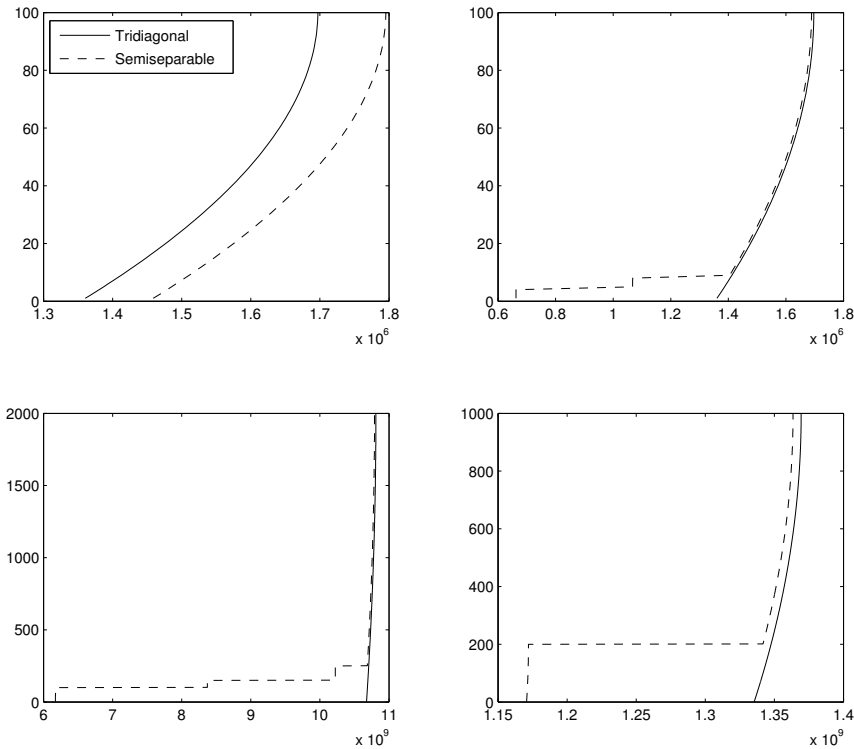


Fig. 1. Comparison in speed

In the first figure, we compared the speed, in case no blocks are separated in the semiseparable plus diagonal case. In this case one can clearly see that the semiseparable case is slightly slower than the tridiagonal case. The second figure was for a matrix of size 100, separating two blocks of sizes 4 each after an extra 20 steps in the reduction procedure. The third figure is for a problem size 2000 and $l_1 = 500, l_2 = 300$ and $l_3 = 500$. The separated blocks are of sizes 100, 50 and 100 respectively. In the second and third figure, the final

complexity is almost equal, but in the two cases, the semiseparable approach already revealed eigenvalues, a long time before the tridiagonal approach did. In case one is interested in only these largest eigenvalues, the semiseparable approach performs much better. In the last example only one large block is separated in the reduction of a matrix of size 1000. After 500 steps in the reduction procedure, a matrix of size 200 is separated. We see that in this case the semiseparable case outperforms the tridiagonal case and moreover, it finds much faster these 200 dominant eigenvalues.

To conclude we can say, that in case we can tune the convergence behavior, such that initially blocks are separated, the semiseparable approach reveals information faster concerning the separated blocks. Moreover it can outperform the tridiagonal approach.

Future work will be dedicated to choosing the diagonal in such a manner that deflation of as many blocks as possible is obtained. In this way we can speed up the computation of the eigenvalues and we can gain faster information concerning clustered eigenvalues.

7.2 Tuning the multi-shift convergence behavior

In these first experiments we construct several matrices, with specific eigenvalues, and we choose the diagonal for the reduction in such a way that it will reveal parts of the spectrum. In the following examples the eigenvalues $\Lambda = [\lambda_1, \dots, \lambda_n]$ of the matrix are given and the matrix itself is constructed as $A = Q^T \text{diag}(\Lambda)Q$, where Q is the orthogonal matrix coming from the QR -factorization of a random matrix. For every example we give the eigenvalues, the diagonal and the number of Householder and Givens transformations performed before the reduction algorithm separated a block containing the desired eigenvalues. A block is separated if the norm of the off-diagonal block is relatively less than 10^{-10} . Also the maximum absolute error between the real and the computed eigenvalues is given.

1. $A = [\text{rand}(10, 1); 100]$ and $d = \text{zeros}(11, 1)$.
 Number of Householder transformations: 6
 Number of Givens transformations: 21
 Separated eigenvalue: 100
 Maximum absolute error: $4.2633e - 14$
2. $A = [\text{rand}(100, 1); 100]$ and $d = \text{zeros}(101, 1)$
 Number of Householder transformations: 6
 Number of Givens transformations: 21
 Separated eigenvalue: 100
 Maximum absolute error: $1.4211e - 14$

3. $A = [\text{rand}(100, 1); 100; 101; 102]$ and $d = \text{zeros}(103, 1)$
 Number of Householder transformations: 10
 Number of Givens transformations: 55
 Separated eigenvalues: 100, 101, 102
 Maximum absolute error: $5.6843e - 14$
4. $A = [1; 100 + \text{rand}(10, 1)]$ and $d = 100 * \text{ones}(11, 1)$
 Number of Householder transformations: 6
 Number of Givens transformations: 21
 Separated eigenvalue: 1
 Maximum absolute error: $1.4211e - 14$
5. $A = [1; 100 + \text{rand}(100, 1)]$ and $d = 100 * \text{ones}(101, 1)$
 Number of Householder transformations: 6
 Number of Givens transformations: 21
 Separated eigenvalues: 1
 Maximum absolute error: $1.4211e - 14$
6. $A = [1; 2; 3; 100 + \text{rand}(100, 1)]$ and $d = 100 * \text{ones}(103, 1)$
 Number of Householder transformations: 11
 Number of Givens transformations: 66
 Separated eigenvalue: 1, 2, 3
 Maximum absolute error: $6.7502e - 14$
7. $A = [\text{ones}(50, 1) + \text{rand}(50, 1); 100; 10000 * \text{ones}(50, 1) + \text{rand}(50, 1)]$
 and
 $d = [10000, 1, 10000, 1, \dots, 10000, 1, 10000]$
 Number of Householder transformations: 12
 Number of Givens transformations: 78
 Separated eigenvalue: 100
 Maximum absolute error: $1.8190e - 12$
8. $A = [1; 2; 3; 100 + \text{rand}(100, 1); 10000; 10001; 10002]$ and
 $d = [\text{zeros}(6, 1); \text{ones}(96, 1) * 100]$
 First there is convergence to the cluster with the largest eigenvalues:
 Number of Householder transformations: 10
 Number of Givens transformations: 55
 Separated eigenvalues: 1001, 1002, 1003
 Maximum absolute error: $3.6380e - 12$
 Secondly there is convergence to the cluster with the smallest eigenvalues
 Extra number of Householder transformations: 15
 Extra number of Givens transformations: 170
 Separated eigenvalues: 1, 2, 3
 Maximum absolute error: $1.5099e - 14$

The examples illustrate clearly, that the convergence behavior can be tuned, by choosing different diagonal values, for reducing the matrix to semiseparable plus diagonal form.

7.3 The interaction between both convergence behaviors

In the following experiments, the interaction between the Lanczos and the multi-shift convergence behavior is shown. For each experiment two figures are given. The left figure shows the Lanczos-Ritz values behavior and the right figure shows the subspace iteration convergence.

The left figure depicts on the x-axis the iteration step of the reduction algorithm and on the y-axis the eigenvalues of the original matrix. If at step k of the reduction algorithm a Ritz-value of the lower right block approximates well-enough (closer than $10^{(-5)}$) an eigenvalue of the original matrix, a cross is placed on the intersection of this step (x-axis) and this eigenvalue (y-axis).

The right figure, shows for all off-diagonal blocks the norm (y-axis), w.r.t. the iteration step (x-axis).

According to the theory, one should observe decreasing norms, as soon as the Ritz-values approximate well enough the dominant eigenvalues w.r.t. the multi-shift polynomial.

In the first example, we generated an example with $A = [1; 2; 3; 10 + rand(22, 1)]$, and the diagonal used for the reduction is $d = 10 * ones(25, 1)$. In the left figure, we see that after six steps in the reduction algorithm, three eigenvalues are approximated up to 5 digits by the Lanczos convergence behavior. In the right figure, we see that after step 6, the norm of 1 subblock, starts to decrease. This means that the subspace iteration starts separating a block with these three eigenvalues.

In the second example, a matrix with three clusters in its eigenvalues, was generated: $A = [1; 2; 10 + rand(21, 1); 100; 101]$, and the diagonal used for the reduction $d = 10 * ones(25, 1)$. As the eigenvalues are separated in three clusters and two clusters are both dominant with respect to the multi-shift polynomial, we would expect two clusters to be separated by the reduction to semiseparable plus diagonal form. This is exactly what we observe in Figure 3. The Lanczos Ritz values approximate both clusters (see the left figure), and as soon as these clusters are well enough approximated, the subspace iteration starts converging. The subspace iteration converges to two clusters, and hence two subblocks show a decreasing norm.

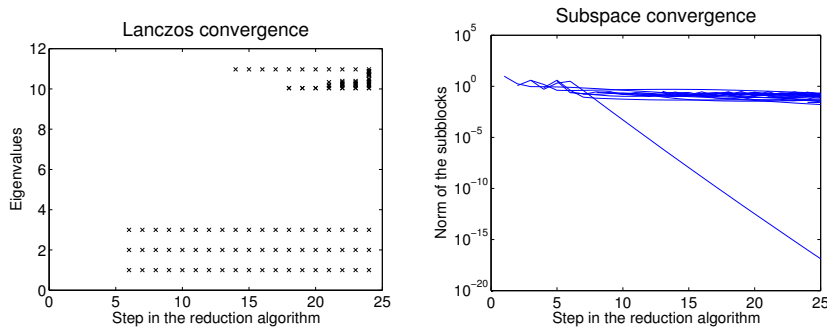


Fig. 2. Example 1

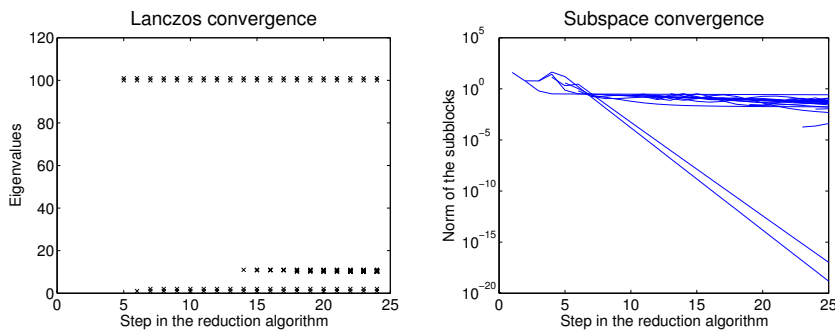


Fig. 3. Example 2

Continuing with this last example, but changing the diagonal d , should only influence the subspace convergence. This can clearly be seen in Figure 4. In the left figure we chose $d = [100; 101; 10 * ones(23, 1)]$ and in the right figure $d = [100; 101; 100; 101; 100; 101; 10 * ones(21, 1)]$. For the left figure, we see that the convergence towards the small eigenvalues starts, but then the norm starts to increase again, and finally we get only convergence towards the eigenvalues 100, 101. For the second polynomial however, we do get convergence to the smaller eigenvalues. (More information on this behavior can be found in the next section.)

In the last example (see Figure 5) convergence is forced into the middle of the spectrum of the matrix. The considered matrix has eigenvalues $A = [ones(50, 1) + rand(50, 1); 100; 10000 * ones(50, 1) + rand(50, 1)]$ and $d = [10000, 1, 10000, 1, \dots]$. This forces convergence towards the middle of the spectrum. As soon as there is convergence of the Ritz values towards the eigenvalue 100, the subspace iteration starts working. We can see that the convergence rate is not as

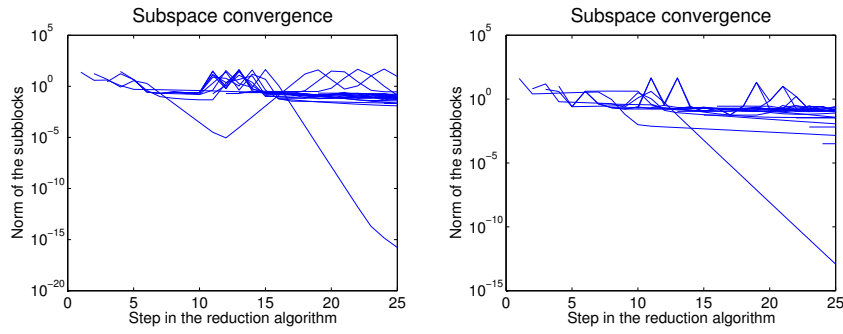


Fig. 4. Example 3

smooth as in the other cases, this is due to the changing roots in the polynomials $p_i(\lambda)$.

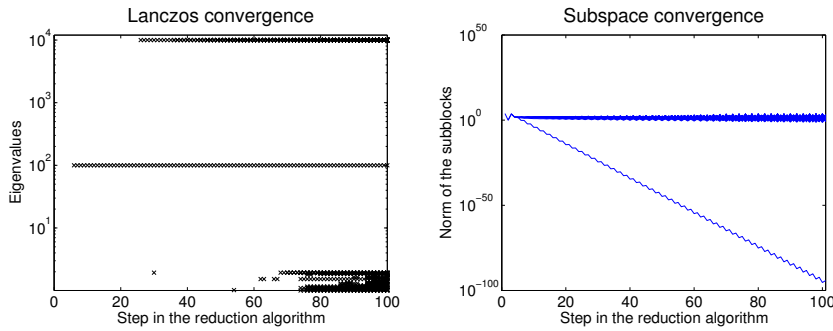


Fig. 5. Example 4

7.4 The multi-shift convergence speed

In this last subsection with numerical experiments we will investigate the upper bound for the convergence speed as presented in Section 6.⁶ In the following experiments the figures show the norm of the submatrix, which should decrease in norm, and the upper bound for this

⁶ The implementation used in the previous sections is based on the Givens-vector representation (see [13]) and is therefore more stable than the implementation used in this section, for generating the figures and calculating the constant C . That is why we get a horizontal line in the figures, once the norm of a subblock reaches the machine precision. This means that using this implementation, the norm of the subblocks cannot, relatively spoken, go below the machine precision, in contrast to the figures in the previous section.

subblock. The upperbound is dependent of the following factor:

$$\frac{\max_{1 \leq j \leq l-1} |\hat{p}_\eta(\lambda_j)|}{\min_{l \leq j \leq n} |\hat{p}_\eta(\lambda_j)|}.$$

To obtain the norm of a subblock, we need to reorder at every step of the method the eigenvalues such that $|\hat{p}_\eta(\lambda_1)| \leq |\hat{p}_\eta(\lambda_2)| \leq \dots$. In our computation of the upper bound, we assume however, that we know to which eigenvalues convergence will occur. Hence, we can divide the eigenvalues into two clusters, a cluster $\lambda_1, \dots, \lambda_{l-1}$ and a cluster $\lambda_l, \dots, \lambda_n$. We know that when there is convergence we have that

$$\max_{1 \leq j \leq l-1} |\hat{p}_\eta(\lambda_j)| \leq \min_{l \leq j \leq n} |\hat{p}_\eta(\lambda_j)|.$$

Hence our computed upper bound will be the correct one, in case there is convergence to the eigenvalue $\lambda_l, \dots, \lambda_n$.

For every example we also give the constant C , which is a measure for the influence of the Lanczos convergence behavior on the subspace iteration. (see Section 6.)

The first example is similar as in the previous two sections. A matrix is constructed with eigenvalues $A = [1; 2; 3; 10 + rand(22, 1)]$, the diagonal $d = 10 * ones(25, 1)$. We expect convergence to a 3×3 block containing the three eigenvalues $[1, 2, 3]$. The norm of the off-diagonal block is plotted and decreases linearly as can be seen in the left figure of Figure 6. The size of the constant C , for calculating the convergence rate equals $1.0772 * 10^4$. In the right figure, we plotted almost the same example, but the sizes of the larger eigenvalues, are chosen smaller now. The eigenvalues were $A = [1; 2; 3; 5 + rand(22, 1)]$, and $d = 5 * ones(25, 1)$, this clearly has an effect on the slope of line representing the convergence speed, the size of the constant C equals $1.8409 * 10^3$. In both figures we see that our upper bound predicts well the convergence behavior.

Also in the case of a diagonal with varying elements, our upper bound provides an accurate estimate of the decay of the corresponding subblock. We consider the example with eigenvalues $A = [1; 2; 10 + rand(21, 1); 100; 101]$. If we choose as diagonal $d = 10 * ones(25, 1)$, we get a similar behavior as above (see the left of Figure 7), and convergence to the cluster containing the eigenvalues 100, 101. The constant $C = 3.4803 * 10^3$. To obtain however the cluster 1, 2 we need to change our diagonal to e.g. $d = [100; 101; 100; 101; 100; 10 * ones(21, 1)]$. Also in this case our upper bound provides an accurate estimate of the decay (see the right of Figure 7). The constant $C = 2.0396 * 10^3$.

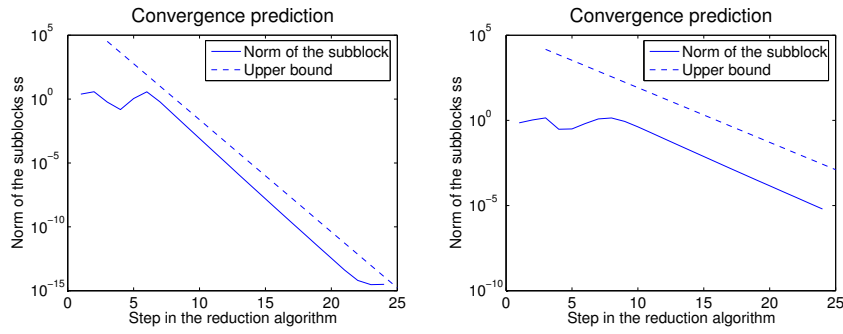


Fig. 6. Example 5

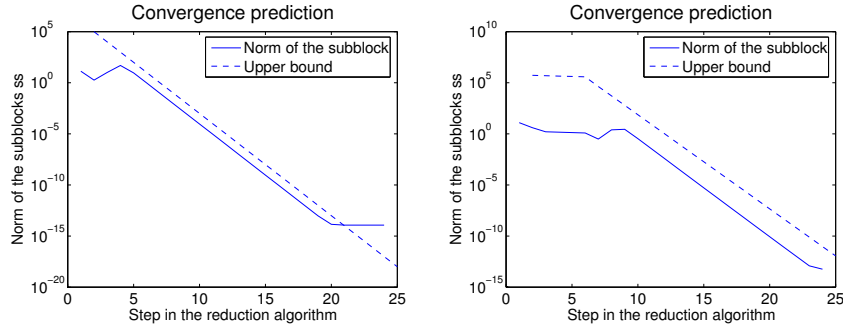


Fig. 7. Example 6

In the following experiment we illustrate, that also if convergence is slowed down, our upper convergence bound predicts a rather accurate estimate of the convergence rate. We consider here the same matrix three times, with varying values for the diagonal, used to reduce it to semiseparable plus diagonal form. The matrix has eigenvalues $\Lambda = [1 + \text{rand}(20, 1); 100 + \text{rand}(2, 1)]$. The diagonals considered for the reduction algorithm are the following ones:

$$\begin{aligned} d_1 &= [\text{zeros}(22, 1)], \\ d_2 &= [100; \text{zeros}(21, 1)], \\ d_3 &= [100; 100; 100; \text{zeros}(21, 1)]. \end{aligned}$$

In the first case we expect normal convergence, in the second case a delay, and in the third one an even larger delay. This behavior is shown in Figure 8, where the reduction with d_1, d_2 and d_3 from left to right is shown.

In the last experiments we illustrate false convergence, and how our upper bound can deal with, or predicts, it. Suppose we have a

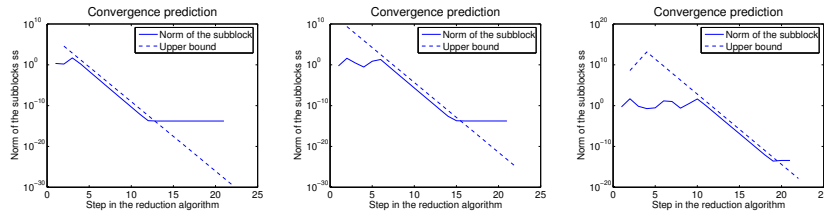


Fig. 8. Example 7

matrix with eigenvalues $\Lambda = [1 + rand(2, 1); 5 + rand(36, 1); 10 + rand(2, 1)]$, suppose the diagonal is chosen in the following way: $d = [10; 10; 10; 10; 10; 10; 5 * ones(38, 1)]$. We observe in the convergence behavior (left of Figure 9), that first there is convergence, but then suddenly the subblock starts to diverge again. This divergence was predicted by the convergence bound. This means that our designed

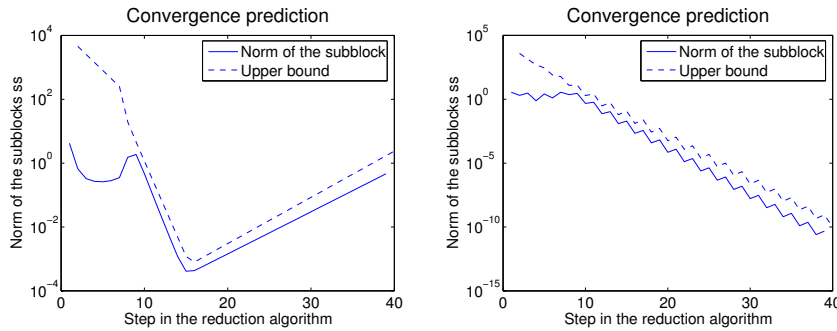


Fig. 9. Example 8

polynomial was not yet strong enough, to force convergence to the two small eigenvalues. If we however would have chosen our diagonal as $d = [10, 5, 10, 5, \dots]$, we would have been able to force convergence towards the two small eigenvalues (see right of Figure 9).

Let us conclude with almost the same experiment, but let us increase the number of eigenvalues in the middle of the spectrum. The eigenvalues are now $\Lambda = [1 + rand(2, 1); 5 + rand(66, 1); 10 + rand(2, 1)]$, and our diagonal values are $d = [10; 10; 10; 10; 10; 10; 10; 10; 5 * ones(10, 1); zeros(58, 1)]$. Similar like in the previous example, we observe first (see left of Figure 10) false convergence towards the small eigenvalues. We see that our upper bound goes up very fast. In reality however we see that norm of that subblock, starts decreasing again in size, so there is convergence towards a block. This block does however

not contain the small eigenvalues anymore, but the largest two. This is depicted in the second figure (right of Figure 10), where we plotted the upper bound related to convergence of the largest eigenvalues. It is clear that the upper limit predicts that the separated block will contain the largest eigenvalues, instead of the smallest ones

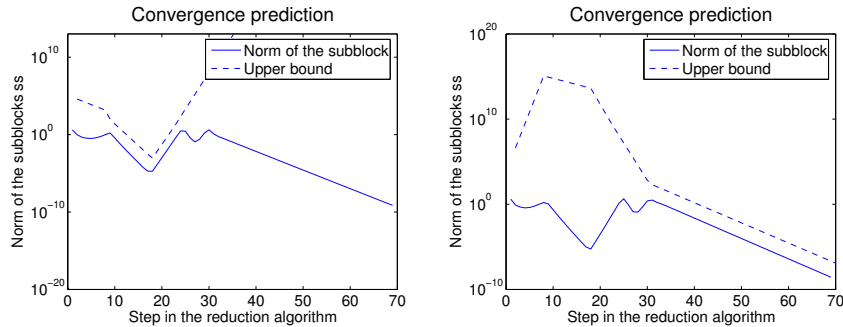


Fig. 10. Example 9

These last examples illustrate that the theoretical upper bound computed for this multishift subspace iteration is correct. Moreover this upper bound can also be used as a theoretical device to predict the eigenvalue convergence.

8 Conclusions

In this paper we presented theoretical results explaining the convergence behavior of the reduction to semiseparable plus diagonal form. As we proved that also the reduction to semiseparable and tridiagonal form can be seen as special cases of the reduction to semiseparable plus diagonal form, we know that also the presented theorems are valid for these reduction methods. Also a theoretical bound for the convergence speed was given.

In a final section we illustrated our theoretical findings by experiments, related to the tuning of the convergence behavior, the interaction between the Lanczos convergence and the subspace convergence and the bound on the convergence rate.

References

1. M. Van Barel, R. Vandebril, and N. Mastronardi. An orthogonal similarity reduction of a matrix into semiseparable form. *SIAM Journal on Matrix Analysis and its Applications*, 27(1):176–197, 2005.

2. R. Vandebril, E. Van Camp, M. Van Barel, and N. Mastronardi. Orthogonal similarity transformation of a symmetric matrix into a diagonal-plus-semiseparable one with free choice of the diagonal. Technical Report TW398, Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3000 Leuven (Heverlee), Belgium, August 2004.
3. G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, third edition, 1996.
4. J. W. Demmel. *Applied Numerical Linear Algebra*. SIAM, 1997.
5. N. Mastronardi, M. Schuermans, M. Van Barel, S. Van Huffel, and R. Vandebril. The Lanczos reduction to semiseparable matrices. In *Proceedings of the 17th IMACS World Congress on Scientific Computation, Applied Mathematics and Simulation*, pages 1–6, July 2005.
6. N. Mastronardi, M. Van Barel, and R. Vandebril. Computing the rank revealing factorization by the semiseparable reduction. Technical Report TW418, Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3000 Leuven (Heverlee), Belgium, May 2005.
7. N. Mastronardi, M. Van Barel, and E. Van Camp. Divide and conquer algorithms for computing the eigendecomposition of symmetric diagonal-plus-semiseparable matrices. *Numerical Algorithms*, 39(4):379–398, 2005.
8. S. Chandrasekaran and M. Gu. A divide and conquer algorithm for the eigendecomposition of symmetric block-diagonal plus semi-separable matrices. *Numerische Mathematik*, 96(4):723–731, February 2004.
9. R. Vandebril, M. Van Barel, and N. Mastronardi. An implicit QR -algorithm for semiseparable matrices to compute the eigendecomposition of symmetric matrices. Technical Report TW367, Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3000 Leuven (Heverlee), Belgium, August 2003.
10. D. A. Bini, L. Gemignani, and V. Y. Pan. QR -like algorithms for generalized semiseparable matrices. Technical Report 1470, Department of Mathematics, University of Pisa, 2004.
11. Y. Eidelman, I. C. Gohberg, and V. Olshevsky. The QR iteration method for Hermitian quasiseparable matrices of an arbitrary order. *Linear Algebra and its Applications*, 404:305–324, July 2005.
12. E. Van Camp. *Diagonal-plus-semiseparable matrices and their use in numerical linear algebra*. PhD thesis, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3001 Heverlee (Leuven), Belgium, May 2005.
13. R. Vandebril, M. Van Barel, and N. Mastronardi. A note on the representation and definition of semiseparable matrices. *Numerical Linear Algebra with Applications*, 12(8):839–858, October 2005.
14. L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM, 1997.
15. R. Vandebril and M. Van Barel. Necessary and sufficient conditions for orthogonal similarity transformations to obtain the Ritz values. Technical Report TW396, Katholieke Universiteit Leuven, Dept. Computerwetenschappen, Celestijnenlaan 200A, 3001 Heverlee (Leuven), Belgium, July 2004.
16. D. S. Watkins and L. Elsner. Convergence of algorithms of decomposition type for the eigenvalue problem. *Linear Algebra and its Applications*, 143:19–47, 1991.
17. D. S. Watkins. QR -like algorithms an overview of convergence theory and practice. *Lectures in Applied Mathematics*, 32:879–893, 1996.

18. R. Vandebril, M. Van Barel, and N. Mastronardi. An implicit QR -algorithm for symmetric semiseparable matrices. *Numerical Linear Algebra with Applications*, 12(7):625–658, 2005.
19. E. Van Camp, M. Van Barel, R. Vandebril, and N. Mastronardi. An implicit QR -algorithm for symmetric diagonal-plus-semiseparable matrices. Technical Report TW419, Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3000 Leuven (Heverlee), Belgium, March 2005.
20. J. J. M. Cuppen. A divide and conquer method for the symmetric tridiagonal eigenproblem. *Numerische Mathematik*, 36:177–195, 1981.
21. C. F. Borges and W. B. Gragg. Divide and conquer for generalized real symmetric definite tridiagonal eigenproblems. In *Proc. Shanghai International Conference on Numerical Linear Algebra and Applications*, pages 70–76, Shanghai, October 1992.