A new iteration for computing the eigenvalues of semiseparable (plus diagonal) matrices

Raf Vandebril
Marc Van Barel
Nicola Mastronardi

Report TW 507, October 2007

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

A new iteration for computing the eigenvalues of semiseparable (plus diagonal) matrices

Raf Vandebril
Marc Van Barel
Nicola Mastronardi

Report TW 507, October 2007

Department of Computer Science, K.U.Leuven

Abstract

This paper proposes a new type of iteration based on a structured rank factorization for computing eigenvalues of semiseparable and semiseparable plus diagonal matrices. Also the case of higher order semiseparability ranks is included. More precisely, instead of the traditional QR-iteration, a QH-iteration will be used. The QH-factorization is characterized by a unitary matrix Q and a Hessenberg-like matrix (often called lower semiseparable matrix) H, having the lower triangular part of semiseparable form. The Q factor of this factorization determines the similarity transformation of the QH-method.

It will be shown that this iteration is extremely useful for computing the eigenvalues of structured rank matrices. Whereas the traditional QR-method applied onto semiseparable (plus diagonal) and Hessenberg-like matrices uses similarity transformations involving $2p(n-1)$ Givens transformations (where p denotes the semiseparability rank), this iteration only needs $p(n-1)$ iterations, which is comparable to the tridiagonal and Hessenberg situation in case of $p = 1$. It will also be shown that this method can in some sense be interpreted as an extension of the traditional QR-method for Hessenberg matrices, i.e., the traditional case also fits into this framework.

Based on results in another paper, it will be shown that this iteration also exhibits an extra type of convergence behavior, w.r.t. the traditional QR-method.

The algorithm will be implemented in an implicit way, based on the Givens-weight representation of the involved structured rank matrices.

Numerical experiments will show the viability of the approach. It will be shown that the new approach yields a better complexity and also gives rise to more accurate results.

Keywords : QH-algorithm, structured rank matrices, implicit computations, eigenvalue, QR-algorithm, rational QR-iteration
MSC : Primary : 65F15 Secondary : 15A18.

A new iteration for computing the eigenvalues of semiseparable (plus diagonal) matrices ⋆

Raf Vandebril[1], **Marc Van Barel**[1], **Nicola Mastronardi**[2]

[1] KULeuven, Dept. of Computer Science, 3001 Leuven(Heverlee)
e-mail: {raf.vandebril,marc.vanbarel}@cs.kuleuven.ac.be
[2] Istituto per le Applicazioni del Calcolo M. Picone, sez. Bari,
e-mail: irmanm21@area.ba.cnr.it

12th October 2007

Abstract This paper proposes a new type of iteration based on a structured rank factorization for computing eigenvalues of semiseparable and semiseparable plus diagonal matrices. Also the case of higher order semiseparability ranks is included. More precisely, instead of the traditional QR-iteration, a QH-iteration will be used. The QH-factorization is characterized by a unitary matrix Q and a Hessenberg-like matrix (often called lower semiseparable matrix) H, having the lower triangular part of semiseparable form. The Q factor of this factorization determines the similarity transformation of the QH-method.

It will be shown that this iteration is extremely useful for computing the eigenvalues of structured rank matrices. Whereas the traditional QR-method applied onto semiseparable (plus diagonal) and Hessenberg-like matrices uses similarity transformations involving $2p(n-1)$ Givens transformations (where p denotes the semiseparability rank), this iteration only needs $p(n-1)$ iterations, which is comparable to the tridiagonal and Hessenberg situation in case of $p=1$. It will also be shown that this method can in some sense be interpreted as an extension of the traditional QR-method for Hessenberg matrices, i.e., the traditional case also fits into this framework.

Based on results in another paper, it will be shown that this iteration also exhibits an extra type of convergence behavior, w.r.t. the traditional QR-method.

The algorithm will be implemented in an implicit way, based on the Givens-weight representation of the involved structured rank matrices. Numerical experiments will show the viability of the approach. It will be shown that the new approach yields a better complexity and also gives rise to more accurate results.

Key words QH-algorithm, structured rank matrices, implicit computations, eigenvalue, QR-algorithm, rational QR-iteration

1 Introduction and preliminary results

Nowadays many authors are investigating efficient algorithms for computing the eigenvalues of structured rank matrices. All discussed methods focus attention towards QR-algorithms for computing the eigenvalues of these matrices. Various QR-type algorithms exist for higher order struc-

⋆ The research of the first two authors, was partially supported by the Research Council K.U.Leuven, project OT/05/40 (Large rank structured matrix computations), CoE EF/05/006 Optimization in Engineering (OPTEC), by the Fund for Scientific Research–Flanders (Belgium), G.0455.0 (RHPH: Riemann-Hilbert problems, random matrices and Padé-Hermite approximation), G.0423.05 (RAM: Rational modelling: optimal conditioning and stable algorithms), and by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office, Belgian Network DYSCO (Dynamical Systems, Control, and Optimization). The first author has a grant as "Postdoctoraal Onderzoeker"from the Fund for Scientific Research–Flanders (Belgium). The work of the third author was partially supported by MIUR, grant number 2004015437. The scientific responsibility rests with the authors.

tured rank matrices, generalized eigenvalue problems, polynomial root finding algorithms and so forth [1,2,3,4,5].

The QR-factorization of a Hessenberg (tridiagonal[1]) matrix can be computed easily by performing a sequence of $n-1$ Givens transformations from top to bottom, annihilating in each of the $n-1$ steps one subdiagonal element [6,7]. The corresponding (single shift) implicit QR-algorithm uses the same number of $n-1$ Givens transformations. The implicit QR-algorithm consists of an initial Givens similarity transformation, applied onto the Hessenberg (tridiagonal) matrix. This introduces a disturbing element, the so-called bulge, in the structure. In the implicit version, one constructs the remaining $n-2$ Givens transformations such that the bulge is removed and we obtain again a Hessenberg (tridiagonal) matrix [8]. Implicitly one has performed now a step of the shifted QR-method.

The QR-factorization of a semiseparable (Hessenberg-like) matrix plus a diagonal[2] consists of $2n-2$ Givens transformations [9]. A first sequence of Givens transformations from bottom to top transforms the semiseparable (Hessenberg-like) plus diagonal matrix into a Hessenberg matrix, whereas the second sequence of transformations will bring the Hessenberg matrix to upper triangular form. The implicit QR-algorithm connected to this type of QR-factorization can also be decomposed into two steps. A first step corresponds to a similarity transformation involving $n-1$ Givens transformations. (See [5,1].) Secondly a disturbance is introduced and $n-2$ Givens transformations are needed for restoring the structure. Unfortunately this implicit QR-algorithm uses twice as much Givens transformations as the corresponding algorithm for the Hessenberg (tridiagonal) case.

This paper will introduce a new type of algorithm for structured rank matrices. This new algorithm will be based on a so-called QH-factorization. This is a factorization of a matrix $A = \check{Q}\check{Z}$, in which $\check{Q}$ is unitary and in which $\check{Z}$ is a Hessenberg-like matrix (having the lower triangular part of the matrix of semiseparable form). This unitary matrix $\check{Q}$, will then be used for defining the new iterate $A_{QH} = \check{Q}^H A \check{Q}$. It will be shown that this iteration can be performed in an efficient manner for structured rank matrices. More precisely, the QH-factorization of a Hessenberg-like minus shift matrix $Z - \sigma I$ will also consist of $n-1$ Givens transformations. The QH-algorithm can be implemented in an implicit way, such that also $n-1$ Givens transformations instead of the traditional $2n-2$ are needed. Besides the fact that the method is cheaper for structured rank matrices, we will also show that this new iteration inherits a new type of convergence behavior, which can be advantageous in many cases.

The paper is organized as follows. This section continues by briefly introducing the classes of semiseparable, Hessenberg-like (plus diagonal) matrices as well as the Givens-weight representation. In Section 2 various methods for computing the QR-factorization of structured rank matrices are introduced. Based on these different types of QR-factorizations one can deduce different types of QR-algorithms. The different ways of computing these QR-algorithms are discussed in Section 3. Section 4 discusses the QH-factorization, which will form the basis for the new QH-method. Section 5, will briefly discuss the preservation of structure of the structured rank matrices under a QH-iteration, and also intuitive proofs of convergence will be given. A more rigorous treatment of the convergence will be presented in Section 6. An implicit version of the method for Hessenberg-like plus diagonal matrices is presented in Section 7. Before providing numerical experiments in Section 9, we briefly show that the QR-method for Hessenberg matrices can be considered as a special case of the QH-method, this is done in Section 8.

1.1 Definition

The class of semiseparable and Hessenberg-like matrices, considered in this paper is defined as follows:

[1] When discussing tridiagonal and semiseparable matrices in the context of eigenvalue computations, we assume them to be symmetric.

[2] The diagonal is necessary for introducing the shift matrix $-\sigma I$ in the shifted QR-algorithm. In the Hessenberg (tridiagonal) case this does not influence the structure, whereas in the structured rank case it does.

Definition 1 *A square matrix S is called a $\{p,q\}$-semiseparable matrix if the following relations are satisfied:*

$$\operatorname{rank}(S(1:i+q-1,i:n)) \leq q \text{ and } \operatorname{rank}(S(i:n,1:i+p-1)) \leq p,$$

for all feasible i. A matrix is called $\{p\}$-semiseparable if it is $\{p,p\}$-semiseparable and semiseparable if it is $\{1,1\}$-semiseparable.

Definition 2 *A square matrix Z is called a $\{p\}$-Hessenberg-like (or lower semiseparable) matrix if the following relations are satisfied:*

$$\operatorname{rank}(Z(i:n,1:i+p-1)) \leq p,$$

for all feasible i.

For simplicity reasons we will focus towards Hessenberg-like (plus diagonal) matrices in this document. There is no loss in generality because only the structure of the lower triangular part of the involved matrices is important in this theoretical analysis. Hence for most derivations, we do not need to know the structure of the upper triangular part involved. For the actual implementations this is, however, very important, to obtain the lowest possible computational complexity.

More important to remark is that in this paper we will develop algorithms for two main classes, the class of Hessenberg-like matrices $\{Z\}$ and the class of Hessenberg-like plus diagonal matrices $\{Z+D\}$, in which D represents an arbitrary diagonal matrix. For several of the topics discussed in this paper we do not need to distinguish between both methods, e.g. the computation of the QR-factorization. For the convergence properties and the actual implementation of the method, we do need to make a distinction because there are several important differences between both approaches.

If both methods differ, we will indicate this by working with the matrix $Z-\sigma I$ for the shifted Hessenberg-like matrix, or by $Z+D-\sigma I$ for the shifted Hessenberg-like plus diagonal matrix. For some topics, we can assume the shift matrix to be incorporated in the diagonal D, e.g., when computing the QR-factorization. In these cases we omit the shift matrix $-\sigma I$.

1.2 Representation

The above defined matrices are dense in the sense that they contain mostly nonzero elements. But, these matrices can be represented by using only a limited number of parameters. They admit, for example, a sparse representation based on Givens transformations. This representation is the so-called Givens-weight representation for the general structured rank case (see [10]), or the Givens-vector representation for the $\{1\}$-semiseparable matrices.[3] More precisely the Givens-weight representation for representing the lower triangular part of a $\{p\}$-Hessenberg-like matrix Z, consists of p sequences of Givens transformations. In fact it is a sort of QR-factorization of the matrix:

$$Q_1^H Q_2^H \dots Q_p^H Z = R \text{ and } Z = Q_1 Q_2 \dots Q_p R = QR, \tag{1}$$

where every unitary matrix Q_i^H consists of $n-1$ Givens transformations, pealing off a rank 1 part from the Hessenberg-like matrix Z. Each of the matrices Q_i contains a descending sequence of Givens transformations. This means that for a particular Q_i, the first Givens transformation acts on rows 1 and 2, the second on rows 2 and 3, the third on rows 3 and 4 and so forth. They start changing the top rows of the matrix and go downwards, hence the name descending. Similarly we name the sequence corresponding to Q_i^H ascending.

In an actual implementation one does not really store the matrix R, but a condensed form (called the weights). Also for the upper triangular part, in case it has rank structure, one can construct such a representation. The effective representation consists of p sequences of $n-1$ Givens transformations plus the weights. In case of a $\{p,q\}$-semiseparable matrix, one has p

[3] We remark that there are many more representations, such as the quasiseparable, generator representation and so forth.

sequences of $n-1$ Givens for storing the lower triangular part and q sequences for storing the upper triangular part plus all weights. The use of the weights is only necessary for implementation details. For theoretical purposes, we work with the QR-like formulation from Equation (1). More information can be found in [10,11].

The above representation is often referred to as the top bottom representation as it starts on the top row of the matrix R, and gradually fills up the matrix from the top to bottom. One can easily change this representation to another kind of factorization: $Z = RQ$, where the matrix Q consists again of p sequences of Givens transformations, now gradually filling up the low rank part of the matrix, from right to left, therefore one names this representation the representation from right to left. One can easily change in $O(pn)$ the representation from a top bottom form to a right left form.

2 The *QR*-factorization and its variants

The idea for the new iteration finds its origin in the different variants for computing the QR-factorization. These variants result of course in different QR-algorithms. Let us briefly discuss the different forms for computing the QR-factorization of structured rank matrices. For simplicity we assume to be working with a Hessenberg-like plus diagonal matrix; semiseparable plus diagonal matrices and higher order semiseparable plus diagonal matrices can be considered in the same way. Remember the previous remark: in this case we assume the shift matrix to be included into the diagonal D.

2.1 The traditional factorization: $\wedge$-pattern

For this type of QR-factorization firstly an ascending sequence of Givens transformations, followed by a descending sequence of Givens transformations is applied onto the Hessenberg-like plus diagonal matrix $Z+D$. More information on this type of QR-factorization can be found in [9,12,13]. The first sequence of Givens transformations acting on $Z+D$, denoted by Q_1^H consists of $n-1$ Givens transformations in which each Givens transformation acts on two successive rows of the matrix Z exploiting thereby the rank structure in the lower triangular part to annihilate all elements below the diagonal (these unitary transformations coincide with the ones from the top to bottom representation). We obtain

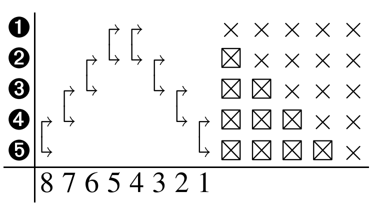
$$Q_1^H Z = R \quad \text{and} \quad Q_1^H (Z+D) = H,$$

in which H is a Hessenberg matrix. This is followed by a second sequence of $n-1$ Givens transformations from top to bottom for annihilating the subdiagonal elements of the matrix H. This gives

$$Q_2^H H = Q_2^H Q_1^H (Z+D) = Q^H (Z+D) = \hat{R},$$

in which $\hat{R}$ is the resulting upper triangular matrix. This is the standard QR-factorization, which is discussed in detail in the paper [9].

In the remainder of the paper, we will often work with a graphical interpretation related to Givens transformations and the matrix they are acting on. The matrix product $Q^H(Z+D)$ is graphically represented as follows:



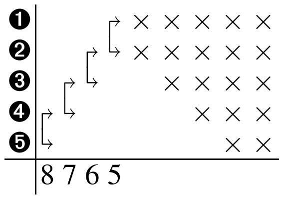
(2)

The right part consisting of $\times$ and $\boxtimes$ represents the matrix $Z+D$, the elements $\boxtimes$ denote the part of the matrix satisfying the rank structure. The other elements $\times$ denote arbitrary elements.

In this figure the elements on the diagonal are not includable in the rank structure because they are perturbed by the diagonal D. The left part, consisting of the brackets with arrows denote the Givens transformations.

The numbered circles on the vertical axis depict the rows of the matrix, to indicate on which rows the Givens transformations will act. The bottom numbers represent in some sense a time line to indicate in which order the Givens transformations are performed. The brackets in the table represent graphically a Givens transformation acting on the rows in which the arrows of the brackets are lying. The Givens transformations from columns 1 up to 4 represent the Givens transformations in the matrix Q_1^H. The ones in the columns 5 up to 8 denote these of the matrix Q_2^H.

Let us explain in more detail this scheme. First a Givens transformation is performed, the one in position 1 in Scheme 2, acting on row 5 and row 4. Secondly a Givens transformation is performed acting on row 3 and row 4 and this process continues. Applying the Givens transformations in position 1 up to 4 onto the matrix on the right results in the following graphical representation. This represents exactly the same matrix as in the previous scheme, but equals now $Q_2^H H$.



(3)

Applying the remaining four Givens transformations in Scheme 3 onto the Hessenberg matrix on the right, will remove the remaining subdiagonal elements. Hence we obtain the upper triangular matrix $\hat{R}$.

So, Scheme 2 gives a graphical way to represent the QR-factorization of a Hessenberg-like plus diagonal matrix.

Remark 1 Consider a $\{p\}$-Hessenberg-like plus diagonal matrix. First one removes the low rank part by applying p ascending sequences of Givens transformations. This gives us

$$Q_p^H \dots Q_1^H (Z+D) = R+H,$$

in which H is a generalized Hessenberg matrix, having p nonzero subdiagonals. To obtain the complete QR-factorization hence another p Givens transformations are needed, from top to bottom each sequence removing one subdiagonal from H.

Globally we have p ascending sequences of Givens transformations for removing the rank p structure, followed by p descending sequences of Givens transformations removing the p subdiagonals. This leads again to a so-called $\wedge$-pattern, having thicker legs.

Due to some specific properties of Givens transformations, we can obtain other patterns.

2.2 Some properties of Givens transformations

Briefly, two important properties of Givens transformations are mentioned here. We will also show their graphical interpretation.

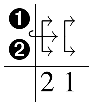
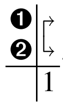
Lemma 1 *Suppose two Givens transformations*[4] G_1 *and* G_2 *are given:*

$$G_1 = \begin{bmatrix} c_1 & -\bar{s}_1 \\ s_1 & \bar{c}_1 \end{bmatrix} \text{ and } G_2 = \begin{bmatrix} c_2 & -\bar{s}_2 \\ s_2 & \bar{c}_2 \end{bmatrix}.$$

Then we have that $G_1 G_2 = G_3$ *is again a Givens transformation. We will call this the fusion of Givens transformations in the remainder of the text.*

[4] In fact these transformations are rotations. More information on Givens rotations can be found in [14].

The proof is trivial. In our graphical schemes, we will depict this as follows:

 resulting in .

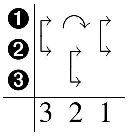
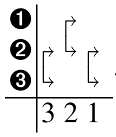
The following lemma is very powerful and will give the possibility to interchange the order of Givens transformations and to obtain different patterns. Quite often Givens transformations of higher dimensions, say n, are considered. This means that the corresponding 2×2 Givens transformation is embedded in the identity matrix of dimension n, still changing only two rows when applied to the left.

Lemma 2 (Shift through lemma) *Suppose three* 3×3 *Givens transformations* $\check{G}_1, \check{G}_2$ *and* $\check{G}_3$ *are given, such that the Givens transformations* $\check{G}_1$ *and* $\check{G}_3$ *act on the first two rows of a matrix, and* $\check{G}_2$ *acts on the second and third row (when applied on the left to a matrix).*
Then there esixt 3 *Givens transformations* $\check{G}_1, \check{G}_2$ *and* $\check{G}_3$ *such that*

$$\check{G}_1 \check{G}_2 \check{G}_3 = \hat{G}_1 \hat{G}_2 \hat{G}_3,$$

where $\hat{G}_1$ *and* $\hat{G}_3$ *work on the second and third row and* $\hat{G}_2$*, works on the first two rows.*

This result is well-known. The proof can be found in [13] and is simply based on the fact that one can factorize a 3×3 unitary matrix in different ways. Graphically we will depict this rearrangement as follows.

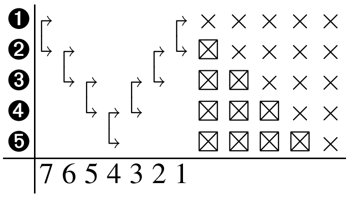
 resulting in .

Of course there is also a variant in the other direction (from the right to the left figure depicted by ↺).

2.3 The $\vee$-pattern

It will be illustrated how one can rewrite the order of the Givens transformations in Scheme 2. Finally we obtain a different graphical scheme, which represents exactly the same factorization, but the Givens transformations are performed now in a different order.

After applying Lemma 1 onto the Givens transformations in position 4 and 5 in Scheme 2 we can apply several times the shift through lemma (three times in this case) and change thereby the order, such that we obtain the following factorization.

 (4)

This gives us the $\vee$-pattern for computing the QR-factorization of a matrix. The order of the Givens transformations has changed, but we compute the same factorization (more information can be found in [15]):

$$\check{Q}_2^H \check{Q}_1^H (Z+D) = \hat{R}.$$

Remark 2 Some important remarks related to both the $\vee$ and $\wedge$-patterns have to be made.

- We have the following equality

$$\check{Q}_1 \check{Q}_2 = Q_1 Q_2,$$

  since $\hat{R}$ was not affected, we obtain an identical QR-factorization.

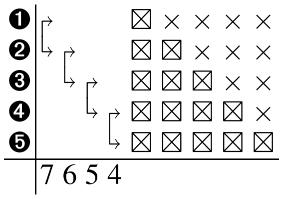
– But we also have generically:

$$\check{Q}_2 \neq Q_2$$
$$\check{Q}_1 \neq Q_1,$$

which means that the factorization of the unitary matrix in the QR-factorization is different in both patterns.

This pattern can also be decomposed into two parts. First a descending sequence of Givens transformations (position 1 up to 3) is applied followed by an ascending sequence of Givens transformations (position 4 up to 7). To distinguish between the $\vee$ and the $\wedge$-pattern we put a $\vee$ on top of the unitary transformations in case of the $\vee$-pattern.

The first three Givens transformations are in fact rank expanding Givens transformations. They lift up the rank structure. Hence after having applied these first Givens transformations we obtain the following scheme.



(5)

The figure clearly illustrates that the strictly lower triangular structure has lifted up and that the diagonal is includable into the lower triangular rank structure.

The remaining four Givens transformations from bottom to top will remove the rank 1 structure in the lower triangular part such that we obtain the upper triangular matrix $\hat{R}$.

Writing the above figure in mathematical formulas, we obtain ($\check{Z}$ denotes a Hessenberg-like matrix):

$$\check{Q}_2^H \check{Q}_1^H (Z+D) = \check{Q}_2^H \check{Z},$$
$$\check{Q}_1^H (Z+D) = \check{Z},$$
$$(Z+D) = \check{Q}_1 \check{Z}.$$

The final equation denotes a structured rank factorization of the matrix $Z+D$, since the matrix $\check{Z}$ is of Hessenberg-like form and $\check{Q}_1$ is a unitary transformation. This unitary-Hessenberg-like (QH) factorization will form the basis of the eigenvalue computations proposed in this paper.

Definition 1 *A factorization of the form*

$$A = \check{Q}\check{Z},$$

with $\check{Q}$ unitary and $\check{Z}$ a Hessenberg-like matrix is called a unitary-Hessenberg-like factorization, briefly a QH-factorization. In case the matrix $\check{Z}$ is a $\{p\}$-Hessenberg-like matrix, we still name this a QH-factorization, but we will specify the rank of the matrix $\check{Z}$.

Remark 3 This factorization is a straightforward extension of the QR-factorization as it can be considered as a QH-factorization in which the matrix $\check{Z}$ is of semiseparability rank 0, i.e. that this matrix has the strictly lower triangular part equal to zero.

Remark 4 Remark that for a $\{p\}$-Hessenberg-like plus diagonal matrix $Z+D$ we will use a higher order QH-factorization in which $\check{Z}$, the Hessenberg-like matrix, will have the lower triangular part of $\{p\}$-semiseparable form. More precisely, in this case, one needs $O(p(n-1))$ Givens transformations for obtaining the factorization.

3 The QR-algorithm and its variants

As there are different manners of computing the QR-factorization, the connected QR-algorithms will also be different, yielding however the same final result. In this section we will briefly discuss the two QR-algorithms connected to both the $\wedge$ and the $\vee$ pattern of Givens transformations. We remark once more that the final outcome of both transformations will be equal, the order in which the Givens transformations are performed will, however, change as well as the Givens transformations themselves.

3.1 The QR-algorithm connected to the $\wedge$-pattern

We consider the following iteration step (we will comment on the Hessenberg-like plus diagonal case afterwards):

$$\begin{aligned} Z - \sigma I &= Q_1 Q_2 \hat{R}, \\ Z_{QR} &= \hat{R} Q_1 Q_2 + \sigma I = Q_2^H Q_1^H Z Q_1 Q_2, \end{aligned}$$

in which Z_{QR} denotes the new iterate.

The single shift QR-algorithm based on the $\wedge$-pattern was firstly discussed in an implicit form in [5]. Explicit QR-algorithms for structured rank matrices, based on the $\wedge$-pattern are legion: for general rank structured matrices and unitary matrices [16,2] and for quasiseparable matrices in [1]. More general forms, for computing the eigenvalues of polynomials can be found in [3,4,17, 18].

Let us discuss the global flow of the iteration related to the $\wedge$-pattern. The iteration can be decomposed into two steps, each step corresponding to performing a sequence of $n-1$ Givens transformations. The first sequence is an ascending one denoted by $\setminus$ in the $\wedge$ and annihilates the low rank part in the Hessenberg-like matrix. The second sequence corresponds to the descending Givens transformations, denoted by $/$ in the $\wedge$-pattern which removes the subdiagonal elements.

Since the new iterate is defined as $Q_2^H Q_1^H Z Q_1 Q_2 = Q_2^H (Q_1^H Z Q_1) Q_2$, two similarity transformations need to be applied onto the matrix Z. One is determined by Q_1 and the other by Q_2.

- The first similarity transformation (related to Q_1) computes the following (see Subsection 2.1):

  $$\begin{aligned} \tilde{Z} &= Q_1^H Z Q_1 = \\ &= R Q_1. \end{aligned}$$

  This corresponds to performing a step of the QR-method without shift onto the matrix Z. As a result we obtain another Hessenberg-like matrix $\tilde{Z}$.
- The second similarity transformation (related to Q_2) is quite often performed explicitly (see, e.g., [4,1]).
  In an implicit manner (see [5]) one can do this as follows. Determine the first Givens transformation $\tilde{G}$ of Q_2, for annihilating the element in position $(2,1)$ of the Hessenberg matrix $Q_1^H(Z - \sigma I) = H$. Applying this Givens transformation $\tilde{G}$ as a similarity transformation onto the Hessenberg-like matrix $\tilde{Z}$ disturbs the specific rank structure of this Hessenberg-like matrix.
  The implicit part of the method consists of finding the remaining $n-2$ Givens transformations and applying them onto $\tilde{G}^H \tilde{Z} \tilde{G}$, such that the resulting matrix is back of Hessenberg-like form. Based on the implicit Q-theorem for Hessenberg-like matrices (see [19]) one knows that the result of this approach is again a Hessenberg-like matrix which is essentially the same as the one resulting from an explicit step of the QR-method.

Remark 5 The first similarity transformation based on Q_1 is independent from the chosen shift σ. The second similarity transformation is dependent of the shift σ.

The multishift method as presented in [20] is also based on the $\wedge$-pattern. Also in this case one can decompose the algorithm into two main steps. The first step can be interpreted as performing several steps of the QR-method without shift. The second step consists again of a chasing technique, involving now more than one disturbing Givens transformation.

The QR-method for Hessenberg-like plus diagonal or $\{p,q\}$-semiseparable plus diagonal matrices is identical. First performing a number of Givens transformations, corresponding to a step of QR-without shift, followed by a similarity transformation determined by Q_2. For restoring the structure in the Hessenberg-like plus diagonal case, one needs to take into consideration the structure of the diagonal, as the diagonal is preserved under a step of the QR-method. (More information can be found in [21].)

3.2 The QR-algorithm connected to the $\vee$-pattern

We consider the following iteration step:

$$\begin{aligned} Z - \sigma I &= \check{Q}_1 \check{Q}_2 \hat{R}, \\ Z_{QR} &= \hat{R} \check{Q}_1 \check{Q}_2 + \sigma I = \check{Q}_2^H \check{Q}_1^H Z \check{Q}_1 \check{Q}_2, \end{aligned}$$

in which Z_{QR} denotes the new iterate.

The QR-algorithm based on the $\vee$-pattern has not yet been discussed in the literature. The idea is however a straightforward generalization of the QR-algorithm based on the $\wedge$-pattern. Due to the fact that we have switched in some sense the order of both sequences of $n-1$ Givens transformations, we can also switch the interpretation of this algorithm.

We have again two similarity transformations to be performed: $\check{Q}_2^H(\check{Q}_1^H Z \check{Q}_1)\check{Q}_2$. Now, $\check{Q}_1$ is a descending sequence of Givens transformations for expanding the rank structure and $\check{Q}_2$ is an ascending sequence of Givens transformations for removing the newly created rank structure of the intermediate Hessenberg-like matrix.

- The first step can be performed in an implicit manner, similar to the $\wedge$-case. An initial disturbing Givens transformation is applied, followed by $n-2$ structure restoring transformations[5]. As a result we obtain the Hessenberg-like matrix[6]

$$\tilde{Z} = \check{Q}_1^H Z \check{Q}_1$$

- One can prove that the second step (corresponding to the Givens transformations from bottom to top) can again be seen as performing a step of the QR-method without shift onto the newly created Hessenberg-like matrix $\tilde{Z}$. After performing the similarity transformation corresponding to $\check{Q}_2$, we obtain the result of performing one step of the QR-method without shift applied onto the Hessenberg-like matrix Z.

Remark 6 In the similarity transformation related to the $\vee$-pattern we have that the first step is dependent on the shift σ, whereas the second step is independent from σ. See also Remark 5 for the iteration related to the $\wedge$-pattern.

Remark 7 The remark above makes it clear that this algorithm (as well as the algorithm related to the $\wedge$-pattern) has a kind of contradicting convergence behavior in it. When we look at teh bottom-right of the matrix, we have that:

- The first step is determined by the shift, and creates hence convergence to the eigenvalue(s) closest to the shift.
- The second step corresponds to a QR-step without shift, and hence converges to the smallest eigenvalue(s) in modulus.

---

[5] Details on how to compute this Givens transformation and how to perform the chasing will be presented in the upcoming sections.

[6] In the upcoming sections, we will prove that the matrix $\tilde{Z}$ is indeed of Hessenberg-like form.

Clearly both convergence behaviors, do not necessarily cooperate. In some sense the second step can slightly destroy the good improvements made by the first step.

Therefore, we opt to simply remove the second similarity transformation, unfortunately we will not have a *QR*-factorization and the corresponding *QR*-method anymore. In the remainder of the paper we will investigate what kind of new iteration is performed, if it preserves the matrix structure, how its convergence properties are and how to implement it in an implicit way.

Based on the comments above we would like to use only the factor $\check{Q}_1$ for performing an orthogonal similarity transformation onto the matrix Z. As $\check{Q}_1$ is closely related to the QH-factorization, a naive approach would be the following:

$$Z - \sigma I = \check{Q}\check{Z},$$

which is a QH-factorization of the matrix $Z - \sigma I$. We can define the new iteration as

$$Z_{QH} = \check{Q}^H Z \check{Q}.$$

Unfortunately this will create some problems as we will see in the next section.

4 More on the QH-factorization and the new QH-algorithm

The QH-factorization will be the basic step in the new QH-method. Unfortunately this QH-factorization as proposed before, is not essentially unique. Hence we need more constraints to be posed on this factorization.

Example 1 Suppose we have the following matrix:

$$Z = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

This matrix is obviously already of Hessenberg-like form. Hence the factorization $Z = IZ$ is a QH-factorization. But in fact one can apply an arbitrary 2×2 Givens transformation acting on the last two rows, without disturbing the structure. This means that we have an infinite number of QH-factorizations for this matrix.

One can also clearly see in Schemes 4 and 5 that the first three Givens transformations already applied to the matrix create the desired structure. This means that in general one needs $n-2$ Givens transformations to obtain a matrix of the following form (e.g. a 4×4 matrix):

$$Z = \begin{bmatrix} \boxtimes & \times & \times & \times \\ \boxtimes & \boxtimes & \times & \times \\ \boxtimes & \boxtimes & \boxtimes & \times \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix}.$$

This matrix is clearly of Hessenberg-like form and an arbitrary Givens transformation, acting on the last two rows can never destroy this low rank structure.

Remark 8 For the higher order case, a similar remark concerning uniqueness can be made. Suppose one has a QH-factorization QZ, with Z of $\{p\}$-Hessenberg-like form. One can apply an arbitrary unitary transformation involving the last $p+1$ rows, without disturbing the factorization.

The non uniqueness of the factorization has its direct impact onto the QH-method, as we cannot guarantee the preservation of the structure anymore. Later on we will show that we can guarantee this, after having defined our QH-factorization in an essentially unique way.

Example 2 Suppose we have the following 3×3 matrix Z and a QH-factorization of this matrix. The given matrix Z is clearly a Hessenberg-like matrix, which has its structure preserved under the standard QR-algorithm. Let us construct a QH-factorization of this matrix:

$$Z = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & \\ 1 & 0 & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 0 & -1 \\ & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = (\check{G}_1 \check{G}_2)\check{Z} = \check{Q}\check{Z},$$

in which $\check{G}_1\check{G}_2 = \check{Q}$, with G_1 and G_2 two Givens transformations and $\check{Z}$ a Hessenberg-like matrix. Performing the similarity transformation with the unitary matrix Q, we obtain:

$$Z_{QH} = \check{Q}^H Z \check{Q} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

The new iterate Z_{QH} after a step of the QH-method with this factorization, is clearly not of Hessenberg-like form anymore.

Hence, it is clear, that we have to pose some extra constraints onto the QH-factorization for making it essentially unique. Some extra conditions will be put onto the QH-factorization guaranteeing an essentially unique factorization in case of $\{p\}$-Hessenberg-like plus diagonal matrices.

Let us consider the following constructive procedure. Suppose we want to compute the QH-factorization of the matrix $Z+D$, in which the matrix D has all diagonal elements different from zero and Z is nonsingular (this is necessary to guarantee uniqueness of the considered factorization). For the Hessenberg-like case $D = -\sigma I$, for the Hessenberg-like plus diagonal case D incorporates the shift matrix $-\sigma I$. We can write the matrix Z as follows:

$$Z = RQ,$$

in which Q consists of p sequences of Givens transformations and which is uniquely determined since Z is nonsingular. Since D has all elements different from zero, the matrix DQ^H is a $\{p\}$-generalized Hessenberg matrix, having all elements on the pth-subdiagonal different from zero.

We obtain now:

$$\begin{aligned} Z + D &= RQ + DQ^H Q = (R + DQ^H)Q \\ &= \check{Q}\check{R}Q, \end{aligned}$$

where $\check{Q}\check{R} = R + DQ^H$, which is the QR-factorization of the left factor in the product. This corresponds to a QH-factorization of the original matrix $Z+D$:

$$Z + D = \check{Q}\check{R}Q = \check{Q}\check{Z},$$

with $\check{Z}$ a $\{p\}$-Hessenberg-like matrix. Important to remark is that the matrix $\check{Z} = \check{R}Q$ has exactly the same Q-factor in its representation from right to left as the original matrix $Z = RQ$, only the upper triangular matrices $\check{R}$ and R differ. One can consider this as an extra constraint posed on the matrices.

Under these assumptions and the essential uniqueness of all considered factorizations above we can guarantee essential uniqueness of the QH-factorization. This factorization will be used for the QH-method.

Remark 9 Two remarks have to be made.

- One might consider the demands of nonsingularity of the matrices Z and D is strict. Luckily they are not so strict and they can be checked rather easily. Since $Z = RQ$ needs to be computed, one immediately knows that Z is singular or not. In a forthcoming section it is shown how to overcome this problem.

– Secondly, important to remark is the fact that the ranks of the lower left subblocks in the matrix Z are completely determined by the unitary factor Q (see [10]). Hence the new Hessenberg-like matrix $\check{Z}$, will satisfy the same upper bounds for the ranks of the subblocks of the original matrix Z:

$$\operatorname{rank}(\check{Z}(k:n,1:l)) \leq \operatorname{rank}(Z(k:n,1:l)), \text{ with } l \leq k.$$

This means that the ranks in the subblocks of the matrix $\check{Z}$ can only decrease, w.r.t. the ranks of the matrix Z.

Reconsidering both examples above, we can clearly see that the structured rank blocks of rank zero, directly below the diagonal are not maintained when considering the proposed QH-factorization. More precisely in both examples the block $Z(3,1:2)=0$, but in the matrix $\check{Z}(3,1:2)$ this block is of rank 1. Hence, we can conclude that the proposed QH-factorization in these examples is not suitable for the QH-method.

Definition 2 *The new iteration proposed in this paper is of the following form. Assume a Hessenberg-like plus diagonal matrix $Z+D$ is given and we have shift σ (with RQ an RQ-factorization of Z):*

$$\begin{aligned} Z+(D-\sigma I) &= RQ+(D-\sigma I)Q^H Q \\ &= (R+(D-\sigma I)Q^H)Q \\ &= \check{Q}\check{R}Q \\ &= \check{Q}\check{Z}, \end{aligned}$$

which gives us a specific QH-factorization of the matrix $Z+D$.

The new iterate is defined as follows

$$\begin{aligned} Z_{QH}+D_{QH} &= \check{Z}\check{Q}+\sigma I \\ &= \check{Q}^H(Z+D)\check{Q}, \end{aligned}$$

5 Convergence and preservation of the structure

The construction in the previous section defines an essentially unique QH-factorization, which we will use for the QH-method. As generically a QH-factorization is not unique and will not lead to convergence, we will base our convergence analysis on the formulas above. Before utilizing results from a previous paper concerning the convergence speed, we will provide an intuitive convergence analysis based on knowledge of the Hessenberg case. Important is also that this intuitive convergence analysis will reveal some constraints which need to be satisfied in order to obtain convergence. Secondly we will prove the preservation of the structure under this iteration.

5.1 Proof of convergence

Let us first discuss the simple Hessenberg-like case, without the diagonal. The idea is that we have a Hessenberg-like matrix Z, and we choose the shift matrix $Z-\sigma I$ in such a way that $Z-\sigma I$ becomes 'almost' singular.

Without loss of generality we can assume the matrix Z not to be of the form

$$Z=\begin{bmatrix} Z_1 & \times \\ 0 & Z_2 \end{bmatrix},$$

with Z_1 and Z_2 square matrices, otherwise we could deflate both blocks and continue the process independently. This assumption implies, that none of the Givens transformations, used in the right left representation of the matrix equals the identity matrix.

Computing the QH-factorization of $Z-\sigma I$ gives us

$$\begin{aligned} Z-\sigma I &= RQ-\sigma Q^H Q = (R-\sigma Q^H)Q \\ &= \check{Q}\check{R}Q. \end{aligned}$$

Due to the structure of the matrix Q^H, we obtain that the matrix σQ^H is an unreduced Hessenberg matrix (assuming $\sigma \neq 0$).

Therefore, $R - \sigma Q^H$ is also an unreduced Hessenberg matrix. This means that in the following factorization,

$$(R - \sigma Q^H) = \check{Q}\check{R},$$

the matrix $\check{R}$, can only have a zero on the diagonal in the lower right position. This zero only occurs in case $Z - \sigma I$ is singular, i.e., when σ is an eigenvalue of Z. In this case, completing the similarity transformation leads to:

$$Z_{QH} = \check{Q}^H Z \check{Q} = (\check{R}Q + \sigma \check{Q}^H)\check{Q} = \check{R}Q\check{Q} + \sigma I,$$

which is a matrix having the last row equal to zero except for the diagonal element which is equal σ.

The above analysis shows that in case the shift is an approximation of an eigenvalue of the original matrix Z, the QH-iteration will reveal this as an eigenvalue. Steps of this method without shift make no sense, as a similarity transformation will be performed with the identity matrix.

Let us now consider the case of a Hessenberg-like plus diagonal matrix $Z + D$, which also does not have zero blocks below the diagonal.

Suppose we perform a step of the shifted QH-method on this matrix (in this case we can also perform steps without shift!).

We obtain the following relations:

$$\begin{aligned} Z + D - \sigma I &= \left(R + (D - \sigma I)Q^H\right) Q \\ &= \check{Q}\check{R}Q. \end{aligned}$$

If $\sigma \neq d_i$, with d_i the diagonal elements of the diagonal matrix D, then we can apply a similar analysis as above and prove convergence towards σ if $Z + D - \sigma I$ is singular, which means that σ is an eigenvalue of the matrix $Z + D$. In case $\sigma = d_i$ convergence is not guaranteed in general, and the implicit method will break down. We will come back to this later on.

Remark 10 In the papers [5,21], the definition of unreducedness of the matrices to prevent break-downs, was quite complicated.

The constraints posed in this setting seem to be much more simple. For the Hessenberg-like case we only need to apply deflation, just like in the standard Hessenberg case, for the Hessenberg-like plus diagonal an extra constraint is posed, namely the diagonal elements need to be different from the shift.

We indicated that in case the above constraints are satisfied, that convergence will occur. Unfortunately we do not yet know that the structure of the Hessenberg-like (plus diagonal) matrix is maintained under the iteration. This is very important, because this guarantees that we can develop a low complexity algorithm for computing the eigenvalues.

5.2 Preservation of the structure

If one desires to use this factorization for computing eigenvalues, it is important to use a matrix structure, invariant under this iteration. We will prove here, that the Hessenberg-like structure is maintained under the QH-iteration as described above. The proof uses theoretical results from the QR-iteration for Hessenberg-like matrices. First we will discuss the Hessenberg-like case, followed by the Hessenberg-like plus diagonal case.

5.2.1 The Hessenberg-like case

Theorem 1 *Suppose Z to be a Hessenberg-like matrix. Suppose we have the following equalities:*

$$\begin{aligned} Z - \sigma I &= \check{Q}\check{Z} \\ Z_{QH} &= \check{Z}\check{Q} + \sigma I, \end{aligned}$$

where σ is a suitably chosen shift and $\check{Q}\check{Z}$ denotes the QH-factorization of the matrix $Z - \sigma I$, constructed as above. The resulting matrix Z_{QH} will again be of Hessenberg-like form.

Proof Let us denote the matrix $\check{Q} = \check{Q}_1$. We know that the matrix $\check{Q}_1$ consists of a sequence of Givens transformations from top to bottom. Let us denote the QR-factorization of the matrix $\check{Z}$ as $\check{Q}_2\check{R}$. We have the following relations:

$$\begin{aligned} \check{Q}_1^H (Z - \sigma I) &= \check{Q}_2\check{R} \\ \check{Q}_2^H \check{Q}_1^H (Z - \sigma I) &= \check{R}. \end{aligned}$$

We divide the proof in several parts.

- Due to the structure of the matrix $\check{Q}_1$, we know that the matrix $\check{Z}\check{Q}_1$ always has the strictly lower triangular part of Hessenberg-like form[7]. Therefore the resulting matrix Z_{QH} has already the strictly lower triangular part of semiseparable form. Moreover as the unitary matrix $\check{Q}_1$ only acts on the columns, the dependencies between the rows do not change, these dependencies in the strictly lower triangular part are exactly the sames as the one in the matrix $\check{Z}$.
- Moreover we know that the matrix

  $$Z_{QR} = \check{Q}_2^H \check{Q}_1^H Z \check{Q}_1 \check{Q}_2 = \check{Q}_2^H Z_{QH} \check{Q}_2,$$

  is of Hessenberg-like form, this is simply a standard QR-step applied onto the matrix Z [5]. Let us assume now that the matrix Z_{QH} has the strictly lower triangular part of semiseparable form, and the diagonal is not includable in the strictly lower triangular structure.
  Based on the previous item, and assuming that the diagonal is not includable in the semiseparable structure we obtain that $\check{Q}_2^H Z_{QH} = H$, which is a Hessenberg matrix with nonzero subdiagonal elements. Completing now the procedure and applying $\check{Q}_2$ to the right of $\check{Q}_2^H Z_{QH}$, will fill up the zeros in the strictly lower triangular part of the Hessenberg matrix H. Based on the structure of the matrix $\check{Q}_2$ (which is a sequence of Givens transformations from right to left) one can easily verify that the matrix $\check{Q}_2^H Z_{QH} \check{Q}_2$ has the strictly lower triangular part of semiseparable form. Moreover the diagonal is not includable in the structure.
  But, we know that Z_{QR} needs to be a Hessenberg-like matrix, having therefore the lower triangular part of semiseparable form. This means that our initial assumption is wrong, hence the matrix Z_{QH} needs to have the lower triangular part of semiseparable form and will therefore be of Hessenberg-like form.

Based on the preservation of structure, we will present in a forthcoming section an implicit method for restoring the structure, when the initial disturbing Givens transformation is performed.

5.2.2 The Hessenberg-like plus diagonal case We remark also that after a step of this QH-method applied onto the Hessenberg-like plus diagonal matrix we obtain again a Hessenberg-like plus diagonal matrix, but the diagonal changes, whereas the diagonal remains the same in the QR-method.

Let us prove the above statement in a more formal way. We have the following theorem, which can e.g. be found in [23].

[7] When the diagonal is not includable into the structure one often names these matrices quasiseparable [22].

Theorem 2 *Suppose a Hessenberg-like plus diagonal matrix $Z+D$, $D = \mathrm{diag}([d_1,\ldots,d_n])$ is given, with*

$$Z = QR,$$

the QR-factorization of the matrix Z. We obtain that the matrix $Q^H(Z+D)Q$ will again be of Hessenberg-like plus diagonal form $\tilde{Z}+\tilde{D}$, where the diagonal elements of $\tilde{D}$ are shifted down one position w.r.t. the diagonal elements of the matrix D, i.e.

$$\tilde{D} = \mathrm{diag}([\alpha, d_1, d_2, \ldots, d_{n-1}]),$$

where α is a freely chosen element.

Based on the above theorem we can formulate the following theorem, predicting the structure of the Hessenberg-like plus diagonal matrix after one iteration of the QH-method.

Theorem 3 *Suppose a Hessenberg-like plus diagonal matrix $Z+D$, $D = \mathrm{diag}([d_1,\ldots,d_n])$ is given, with*

$$Z+D-\sigma I = \check{Q}\check{Z},$$

constructed as presented above. Then the matrix $\check{Q}^H(Z+D)\check{Q}$ will again be a Hessenberg-like plus diagonal matrix $Z_{QH}+D_{QH}$, where the diagonal elements of D_{QH} are shifted up one position w.r.t. the diagonal elements of the matrix D, i.e.

$$D_{QH} = \mathrm{diag}([d_2, \ldots, d_{n-1}, d_n, \beta]),$$

where β is a freely chosen element.

Proof The proof is a combination of different theoretical results. We know that the structure of a semiseparable plus diagonal matrix is preserved under the QR-algorithm (see [24]), having even the same diagonal D. Based on the previous section we can write the new iterate from the QR-method as follows:

$$Z_{QR}+D_{QR} = Z_{QR}+D = \check{Q}_2^H\check{Q}_1^H(Z+D)\check{Q}_1\check{Q}_2,$$

where $D_{QR} = D$. Moreover, without loss of generality we can assume $\check{Q}_1 = \check{Q}$.

We know that, using the construction from above:

$$\begin{aligned}(Z+D-\sigma I) &= \check{Q}_1\check{Z},\\ \check{Q}_1^H(Z+D)\check{Q}_1 &= Z_{QH}+D_{QH},\end{aligned}$$

needs to be a Hessenberg-like plus diagonal matrix (in a similar manner as in the proof of Theorem 1), with $\check{Z} = \check{Q}_2\check{R}$. Combining this with Theorem 2 leads to the desired result.

We have proved now that the Hessenberg-like (plus diagonal) structure is maintained under the QH-method, making thereby these matrix structures suitable for this iteration. In the following section we will discuss more in detail the theoretical convergence rates of this new method.

For simplicity reasons we did not discuss the general $\{p\}$-Hessenberg-like case anymore. The results can however be generalized to this type of matrices.

6 Convergence of the QH-method

We already derived an intuitive proof of convergence and theorems on the preservation of the structure. We can consider however this method as a specific case of a more general framework presented in [25]. This framework discusses rational QR-iteration steps. In this paper general convergence theoretical results as well as results on the preservation of structure and so forth are presented. We will only use the results applicable to our case.

6.1 A rational QR-iteration

Let us interpret the QH-iteration in terms of a rational QR-iteration. The analysis presented here is similar to the one in [26,27,28] and is a special case of the rational QR-iteration, which was presented in [25].

As discussed in the previous section, the global iteration is of the following form:

$$\begin{aligned} Z &= RQ, \\ Z+(D-\sigma I) &= (R+(D-\sigma I)Q^H)Q = \check{Q}\check{R}Q, \\ Z_{QH}+D_{QH} &= \check{Q}^H(Z+D)\check{Q}, \end{aligned}$$

where $Z_{QH}+D_{QH}$ defines the new iterate in the method.

One can rewrite the above formulas and obtain that the matrix $\check{Q}$ is the Q-factor in the QR-factorization of the matrix product $(Z+(D-\sigma I))Z^{-1}$:

$$\begin{aligned} (Z+(D-\sigma I))Z^{-1} &= \left(\check{Q}\check{R}Q\right)\left(Q^H R^{-1}\right) \\ &= \check{Q}\check{R}R^{-1}. \end{aligned}$$

This formula illustrates that we have computed the unitary factor of a special function in Z. Depending on the diagonal matrix D, we have to distinguish between two cases. The case in which D is zero, which is the Hessenberg-like case, or the case in which D is an arbitrary chosen diagonal.

6.2 The Hessenberg-like case

In this case the diagonal matrix D equals zero, and σ is a suitably chosen shift. The equation above simplifies and we obtain:

$$(Z-\sigma I)Z^{-1} = \check{Q}\check{R}R^{-1}.$$

This means that the convergence properties of the iteration performed on the matrix Z are defined by the subspace convergence properties, defined by the rational function $p(\lambda) = (\lambda - \sigma)\lambda^{-1}$.

These convergence properties, and more advanced results for a general rational iteration of the form $p(\lambda) = (\lambda-\sigma)(\lambda-\kappa)^{-1}$, were extensively discussed in [25].

Some initial theoretical results on subspace iteration theory are necessary. Given two subspaces $\mathcal{S}$ and $\mathcal{T}$ in $\mathbb{C}^n$ 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 [6]) is defined as

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

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 rational functions.

Theorem 4 (Theorem 5.1 from [27]) *Given a simple*[8] *matrix $A \in \mathbb{C}^{n\times n}$ with $\lambda_1,\lambda_2,\ldots,\lambda_n$ the eigenvalues and associated linearly independent eigenvectors $\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_n$. Let $V = [\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_n]$ and let κ_V is the condition number of V, w.r.t. to the spectral*[9] *norm. Let k be an integer $1 \le k \le n-1$, and define the invariant subspaces $\mathcal{U} = \langle \mathbf{v}_{k+1},\ldots,\mathbf{v}_n\rangle$ and $\mathcal{T} = \langle \mathbf{v}_1,\ldots,\mathbf{v}_k\rangle$. Denote with $(p_i)_i$ a sequence of rational functions and let $\hat{p}_i = p_i \ldots p_2 p_1$. Suppose that the*

$$\begin{aligned} p_i(\lambda_j) &\neq 0 \qquad j = 1,\ldots,k \\ p_i(\lambda_j) &\neq \pm\infty \qquad j = k+1,\ldots,n \end{aligned}$$

[8] A matrix is called simple if it has n linearly independent eigenvectors.
[9] The spectral norm is naturally induced by the $\|.\|_2$ norm on vectors.

for all i, and let

$$\hat{r}_i = \frac{\max_{k+1\leq j\leq n} |\hat{p}_i(\lambda_j)|}{\min_{1\leq j\leq k} |\hat{p}_i(\lambda_j)|}.$$

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

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

Let $\mathcal{S}_i = \hat{p}_i(A)\mathcal{S}_0, i = 1, 2, \ldots$, with $\mathcal{S}_0 = \mathcal{S}$. Then there exists a constant C (depending on $\mathcal{S}$) such that for all i,

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

In particular $\mathcal{S}_i \to \mathcal{T}$ if $\hat{r}_i \to 0$. More precisely we have that

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

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

Lemma 3 (Lemma 6.1 from [27]) *Suppose $A \in \mathbb{C}^{n\times n}$ 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 first 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 $B_{21} \in \mathbb{C}^{(n-k)\times k}$. Then we have:

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

where σ_G denotes the condition number of the matrix G.

For the Hessenberg-like case, the functions are of the following form

$$p_i(\lambda) = (\lambda - \sigma_i)\lambda^{-1}.$$

Let us compare the convergence behavior of this new iteration w.r.t. the standard QR-iteration with shift σ_i. We consider only one iterate, i.e. r_i denotes the contraction rate from step i in the iteration process. For the standard QR-algorithm we obtain the following contraction ratio:

$$r_i^{(QR)} = \frac{\max_{k+1\leq j\leq n} |\lambda_j - \sigma_i|}{\min_{1\leq j\leq k} |\lambda_j - \sigma_i|}. \tag{6}$$

We introduce the following constants:

$$\omega = \min_{k+1\leq j\leq n} \{|\lambda_j|\},$$
$$\Omega = \max_{1\leq j\leq k} \{|\lambda_j|\}.$$

Calculating now an upper bound for the convergence towards the eigenvalue closest to the shift σ_i, for the QH-method gives us:

$$r_i^{(QH)} = \max_{k+1\leq j\leq n} \left|\frac{\lambda_j - \sigma_i}{\lambda_j}\right| \max_{1\leq j\leq k} \left|\frac{\lambda_j}{\lambda_j - \sigma_i}\right| \leq \frac{\Omega}{\omega} \frac{\max_{k+1\leq j\leq n} |\lambda_j - \sigma_i|}{\min_{1\leq j\leq k} |\lambda_j - \sigma_i|} = \frac{\Omega}{\omega} r_i^{(QR)}$$

This indicates that convergence of the new iteration is comparable (up to a constant) with the convergence of the standard QR-method. This constant will only create a small, neglectable delay in the convergence. This means that if the traditional QR-method converges to an eigenvalue in the lower right corner, the QH-method will also converge. Hence, to obtain convergence to a specific eigenvalue λ_j, we choose σ_i close to this eigenvalue. The convergence results prove that this eigenvalue will then be revealed in both the *QR*- and the *QH*-method in the lower right corner.

Moreover we also have extra convergence, which is not present in the standard *QR*-case, and which is created by the factor λ^{-1} in the rational functions.

Define the following constants:

$$\Delta_i = \max_{k+1 \leq j \leq n} \{|\lambda_j - \sigma_i|\},$$
$$\delta_i = \min_{1 \leq j \leq k} \{|\lambda_j - \sigma_i|\}.$$

Similarly as above, we can define the following contraction ratio, for all k.

$$r_i = \frac{\Delta_i}{\delta_i} \frac{\max_{1 \leq j \leq k} |\lambda_j|}{\min_{k+1 \leq j \leq n} |\lambda_j|}.$$

Assume now (without loss of generality), all eigenvalues to be ordered, i.e. $|\lambda_1| \leq |\lambda_2| \leq \ldots \leq |\lambda_n|$. This means that our convergence rate can be simplified as follows:

$$r_i = \frac{\Delta_i}{\delta_i} \frac{|\lambda_k|}{|\lambda_{k+1}|}.$$

This means that we get a contraction for all k determined by the ratio λ_k/λ_{k+1}. This is a basic non shifted subspace iteration taking place for all k at the same time. Remark that this convergence takes place in addition to the convergence imposed by the shift σ_i, which can force for example extra convergence towards the bottom right element.

More information on this specific type of subspace iteration can be found in the paper [25].

6.3 The Hessenberg-like plus diagonal case

The convergence theory related to the Hessenberg-like plus diagonal case is more complicated. In each step of the above method, one will perform now a step of the shifted *QR*-iteration, combined with a nested multishift iteration. The convergence analysis of this method is not so easy, w.r.t. the standard *QH*-method for Hessenberg-like matrices. We will not present the global convergence theory, but a brief explanation on the behavior. Based on the results in [25, 29] one can derive global convergence results and predictions on the convergence ratios.

We will distinguish between two cases. First we will discuss the case in which $\sigma = 0$. As we want to compute the specific *QH*-factorization of the matrix $A = Z + D$, in which Z is a Hessenberg-like matrix and D an arbitrary diagonal, we apply the following algorithm:

$$\begin{aligned} Z &= RQ, \\ Z + D &= (R + DQ^H)Q \\ &= \check{Q}\check{R}Q. \end{aligned}$$

Applying the traditional analysis from above, we obtain the following equalities:

$$(Z+D)Z^{-1} = A(A-D)^{-1} = \check{Q}\check{R}R^{-1}.$$

Hence we have computed the *QR*-factorization of the original matrix A multiplied with the inverse of A minus a diagonal shift matrix. This diagonal shift will create the nested multishift iteration, with shifts equal to the diagonal elements, similarly as in the reduction to semiseparable plus diagonal form.

Assuming our shift $\sigma \neq 0$ we obtain the following relations:

$$\begin{aligned} Z &= RQ, \\ Z + D - \sigma I &= \left(R + (D - \sigma I)Q^H\right)Q \\ &= \check{Q}\check{R}Q. \end{aligned}$$

and we also get:

$$(Z+D-\sigma I)Z^{-1} = (A-\sigma I)(A-D)^{-1} = \hat{Q}\hat{R}R^{-1}.$$

This implies that we perform a step of the traditional *QR*-method combined again with the multishift iteration.

Hence in the Hessenberg-like plus diagonal case we obtain also the classical convergence of the *QR*-method plus an extra nested multishift iteration. An interpretation of this kind of subspace iteration and its convergence properties can be found in [29].

In a previous section it was mentioned then shift σ needed to be different from the diagonal elements. This analysis makes clear that if the shift equals a diagonal element that the there will be no convergence on a specific subspace!

6.4 Summary

The summary of both convergence behaviors is very closely related to the reduction algorithms to respectively Hessenberg-like and Hessenberg-like plus diagonal form:

- The unitary similarity reduction of an arbitrary matrix to Hessenberg-like form had an extra convergence property w.r.t. the traditional reduction to tridiagonal form. In every step of the reduction process also a kind of nested non shifted subspace iteration took place. This nested non shifted subspace iteration can also be found here in the new QH-iteration. The standard convergence results of the *QR*-iteration are present, plus an extra subspace iteration convergence.
- The unitary similarity transformation to Hessenberg-like plus diagonal form had an even more advanced convergence behavior than the reduction to Hessenberg-like form. Namely a nested multishift subspace iteration. Also here, in the QH-iteration a similar thing happens, in every step of the iteration we have the traditional convergence properties plus an extra shifted iteration which we can see, when combining multiple steps as a multishift iteration.

These simple arguments state that the iteration presented here is a straightforward extension of the basic reduction algorithms to both semiseparable and semiseparable plus diagonal form.

7 The implicit QH-iteration for Hessenberg-like (plus diagonal) matrices

Even though the presented theoretical results might be complicated to prove, the actual implementation is quite simple, more simple than the implementation of the *QR*-method.

In this section, we will derive the implicit chasing technique developed for Hessenberg-like plus diagonal matrices. Moreover, remark that this approach is also valid for Hessenberg-like matrices, by setting the diagonal equal to zero.

7.1 An implicit algorithm

In this section we will design an implicit way for performing an iteration of the QH-method onto a Hessenberg-like plus diagonal matrix.

Based on the results above we can compute the following factorization:

$$Z+(D-\sigma I) = \check{Q}\check{Z},$$

the matrix $\check{Q}$ is then used for performing a unitary similarity transformation onto the matrix $Z+D$:

$$Z_{QH}+D_{QH} = \check{Q}^H(Z+D)\check{Q}.$$

The idea of the implicit method is to compute the unitary similarity transformation $\check{Q}^H(Z+D)\check{Q}$, based on only the first column of $\check{Q}$ and on the fact that the matrix Z_{QH} satisfies some structural constraints. This idea is completely similar to the implicit QR-step for tridiagonal matrices [7,6] (and also semiseparable matrices [5]).

Because the matrix $\check{Q} = \check{G}_1\check{G}_2 \ldots \check{G}_{n-1}$ consists of a descending sequence of $n-1$ Givens transformations only the first Givens transformation $\check{G}_1$ is necessary for determining the first column of the matrix $\check{Q}$. Having determined this Givens transformation, it will be applied onto the matrix $(Z+D)$ disturbing thereby the Hessenberg-like plus diagonal structure. The remaining $n-2$ Givens transformations are constructed in such a manner to restore the structure of the Hessenberg-like matrix.

After having performed these transformations, we know, based on the implicit Q-theorems for Hessenberg-like (plus diagonal) matrices (see [19,30,31]), that we have performed a step of the QH-method in an implicit manner.

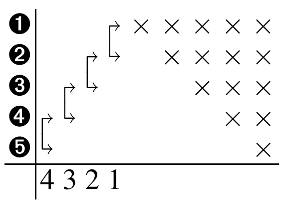
7.2 Assumptions

Before starting the construction of the implicit algorithm we need to assume some things about the Hessenberg-like (plus diagonal) matrix. In the Hessenberg case one only assumes irreducibility, i.e. the matrix cannot be split up into several subblocks. Assume we are working with the matrix $Z+D$, where the shift matrix $-\sigma I$ is included in the diagonal matrix D.

- Similarly as above we assume our matrix cannot be divided in subblocks.
- We also assume the diagonal D to have no zeros on the diagonal. More precisely, this means that in the Hessenberg-like case the σ needs to be different from zero, and in the Hessenberg-like plus diagonal case, σ cannot be equal to one of the diagonal elements from the Hessenberg-like plus diagonal matrix.

7.3 Computing the initial disturbing Givens transformations

For the actual implementation, we assume the Hessenberg-like matrix Z to be represented with the Givens-vector representation. This can be seen as the QR-factorization of this matrix $Z = QR$. We remind that the matrix $Q = G_{n-1}G_{n-2} \ldots G_1$ can be factored as a sequence of Givens transformations, where each Givens transformation G_i acts on two successive rows i and $i+1$. Graphically this representation $Z = QR$ is depicted as follows:



(7)

The Givens transformations in positions 1 to 4 make up the matrix Q, and the upper triangular matrix R is shown on the right.

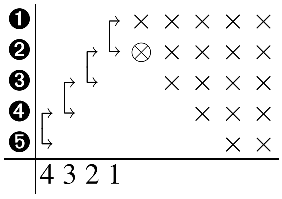
We have the following equations

$$Z + (D - \sigma I) = QR + (D - \sigma I) = Q(R + H),$$

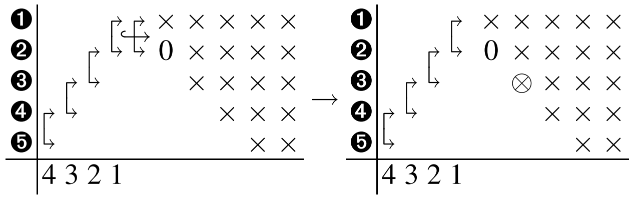
in which H is a Hessenberg matrix. We want to apply now a sequence of descending Givens transformations onto this matrix $Z + (D - \sigma I)$ such that we obtain a Hessenberg-like matrix $\check{Z}$. We remark that the construction of the QH-factorization here corresponds to the construction used before, preserving the dependencies in the low rank part. Here however the procedure is optimized, for working with our Givens-vector representation.

Using the graphical representation we can represent $Q(R+H)$ as follows (the Givens transformations making up the matrix Q are shown on the left, whereas the Hessenberg matrix $R+H$

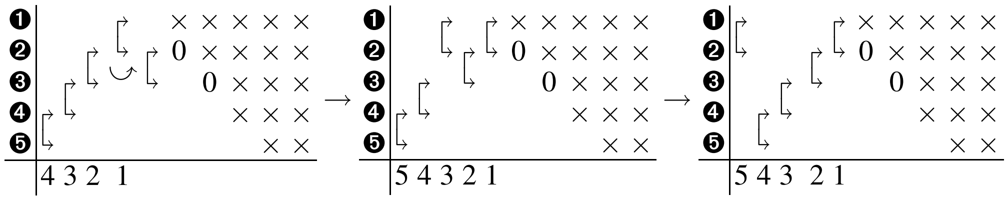
is shown on the right).



The element marked with $\otimes$ should be annihilated, because we want to obtain a Givens-vector representation of a new Hessenberg-like matrix namely $\check{Z}$, as in Scheme 7. Removing this element by placing a new Givens transformation in position one, and applying the indicated fusion, gives us the following result.



Annihilating the element marked in position $(3,2)$, by a Givens transformation and performing the shift-through operation at the indicated position we obtain the following figure.



We remark that the rightmost figure still represents the original matrix $Z+D-\sigma I$. Due to the rewriting of the matrix we can however clearly see that performing the hermitian conjugate of the Givens transformation in position 5 to the left of the matrix $Z+D$, will give already a Hessenberg-like structure in the upper left corner of this matrix. This is due to the fact that this upper left part is already represented in the Givens-vector representation.

When continuing this process, another Givens transformation annihilating the element in position $(4,3)$ can be dragged through the representation, and so forth. As a result we obtain the QH-factorization of the matrix $Z+D$, constructed simply by applying few times the shift through operation. The reader can verify that the constructed QH-factorization coincides with the desired one from the previous subsections. Moreover, as a result of these computations we have immediately the matrix $\check{Z}$ in its Givens-vector representation.

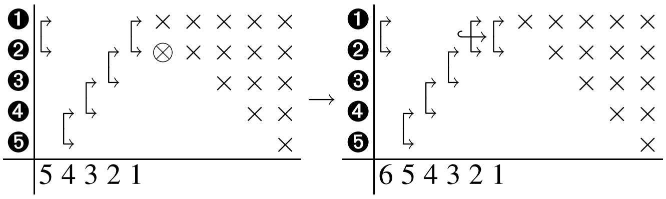
For our purposes however, only the transpose of the initial Givens transformation, working on rows 1 and 2, and presented in position 5 on the rightmost figure above, is needed. Having calculated this Givens transformation, we can apply it as a similarity transformation onto the matrix Z and then to complete the implicit chasing procedure, restore the structure of this matrix, never interfering anymore with the first column and row. In the following subsection we illustrate how to restore the structure of this matrix based on an initial disturbing Givens transformation.

7.4 Restoring the structure

We have a Hessenberg-like plus diagonal matrix $Z+D$, in which $D=\operatorname{diag}([d_1,d_2,\ldots,d_n])$. We know that after a step of the QH-method our resulting matrix will be a Hessenberg-like plus diagonal matrix $\hat{Z}+\hat{D}$ in which $\hat{D}=\operatorname{diag}([d_2,d_3,\ldots,d_n,\beta])$. Assume in the following graphical schemes that all transformations are well-defined. After the global flow of the algorithm, we will come back, in the next section to some specific cases.

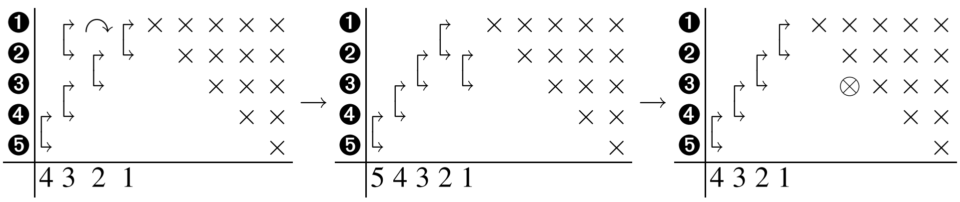
After having computed the initial disturbing Givens transformation, we will apply this transformation onto the matrix $Z+D$. Before being able to perform the first transformation we need to rewrite our matrix $Z+D = Z_1 + D_1$, in which Z_1 is again a Hessenberg-like matrix, differing from Z only in the upper left element, and the diagonal equals $D_1 = \mathrm{diag}([d_2, d_2, d_3, \dots, d_n])$. Applying the similarity transformation gives us $\check{G}_1^H (Z_1 + D_1)\check{G}_1 = \check{G}_1^H Z_1 \check{G}_1 + D_1$. The diagonal D_1 does not change, because the Givens transformation acts on the first two rows and columns, and the diagonal elements in these positions are both equal to d_2. Our matrix Z_1 can be represented as in Scheme 7. After applying the disturbing transformation, this scheme will also be disturbed. Then we will try to obtain again Scheme 7 by applying similarity transformations, not involving the first column and row of the matrix anymore.

In the following figures, we will not show the diagonal, but only the effect of the similarity transformation $\check{G}_1$ acting on the matrix Z_1. For simplicity reasons we assume our matrix to be of size 5×5. Let us name $\tilde{Z}_2 = \check{G}_1^H Z_1 \check{G}_1$.



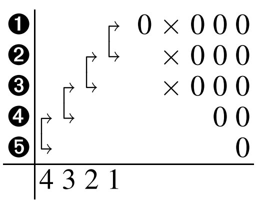
The transformation $\check{G}_1$ applied to the right creates the bulge, marked with $\otimes$, whereas the Givens transformation $\check{G}_1^H$ applied to the left can be found in position 5. The bulge marked with $\otimes$ can be annihilated by a Givens transformation as depicted above.

In the following figure, we have combined the Givens transformations in position 1 and 2, by a fusion. We have moved the transformation from position 6 to position 3 and we depicted where to apply the shift-through lemma. The right figure shows the result after having applied the shift-through lemma and after having created the bulge, marked with $\otimes$.



We remark once more that the above rearrangements of the Givens transformations did not affect the diagonal matrix D_1. To continue now we do need to incorporate again the diagonal matrix D.

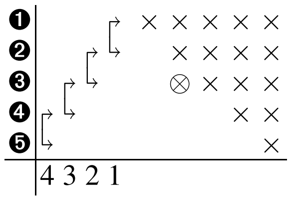
The next similarity Givens transformation we will perform will act on columns and rows 2 and 3. To perform the procedure correctly, we will first transform our diagonal matrix from $D_1 = \mathrm{diag}([d_2, d_2, d_3, \dots, d_n])$ to $D_2 = \mathrm{diag}([d_2, d_3, d_3, \dots, d_n])$. This change in the diagonal $\tilde{D}_2$ with $D_1 = D_2 + \tilde{D}_2$, and $\tilde{D}_2 = \mathrm{diag}([0, d_2 - d_3, 0, \dots, 0])$ needs to be incorporated in the scheme above, in the rightmost figure, namely matrix $\tilde{Z}_2$. To incorporate the matrix $\tilde{D}_2$ into $\tilde{Z}_2$, we use the factorization of the matrix $\tilde{Z}_2 = U_2 S_2$ depicted in the rightmost scheme above, where U_2 depicts the combination of the Givens transformations in positions 1 to 4 and S_2 is the upper triangular matrix with the bulge on the right. We obtain that the matrix $\tilde{D}_2 = U_2 U_2^H \tilde{D}_2 = U_2(U_2^H \tilde{D}_2)$ equals the following scheme. The Givens transformations in the positions 1 to 4 coincide with U_2 and the sparse matrix on the right equals $(U_2^H \tilde{D}_2)$.



Rewriting all of this into formulas we obtain:

$$
\begin{aligned}
\check{G}_1^H(Z_1+D_1)\check{G}_1 &= \check{G}_1^H Z_1 \check{G}_1 + D_1 \\
&= \tilde{Z}_2 + D_1 \\
&= \tilde{Z}_2 + \tilde{D}_2 + D_2 \\
&= U_2 S_2 + U_2(U_2^H \tilde{D}_2) + D_2 \\
&= U_2(S_2 + U_2^H \tilde{D}_2) + D_2 \\
&= Z_2 + D_2
\end{aligned}
$$

Important now is the fact the matrices $\tilde{Z}_2$ and Z_2 are factored by the same matrix U_2 and moreover that they have the bulge in exactly the same position. Hence we can proceed with a similar scheme as above, where we assume now to be working with the matrix Z_2 instead of $\tilde{Z}_2$, namely



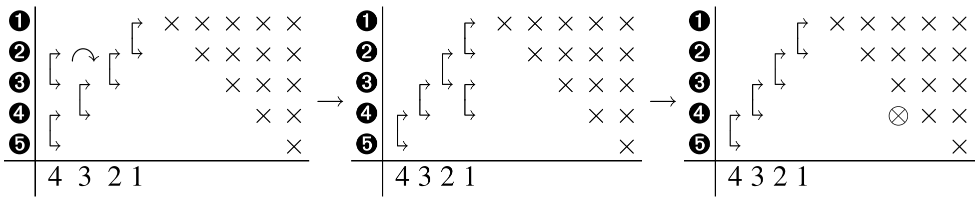
Remark that the scheme looks similar, but few elements, including the bulge have changed now.

To continue the implicit procedure, we want to remove the bulge in position $(3,2)$. In order to do so, we choose a Givens transformation $\check{G}_2$ acting on column 2 and 3, which will remove the bulge. Performing this Givens transformation as a similarity transformation onto the matrix Z_2+D_2 we obtain:

$$
\begin{aligned}
\check{G}_2^H(Z_2+D_2)\check{G}_2 &= \check{G}_2^H Z_2 \check{G}_2 + D_2 \\
&= \tilde{Z}_3 + D_2
\end{aligned}
$$

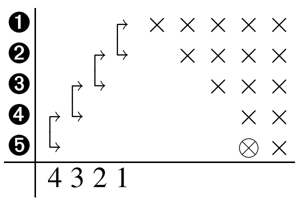
The diagonal remains unchanged as the diagonal elements on the second and third position are equal to each other.

The similarity transformation onto Z_2 is schematically depicted as follows:

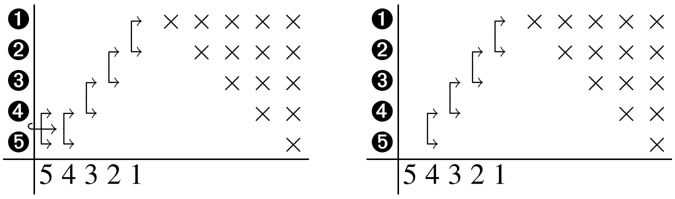


We see that we have created now a new bulge in position $(4,3)$. A similar technique can now be applied to change the diagonal D_2 to D_3 and to transform $\tilde{Z}_3$ into Z_3. Since the upper triangular part of the involved matrices is dense, such a chasing step involves $\mathcal{O}(n)$ operations, leading to a global complexity of $\mathcal{O}(n^2)$.

We will show only the final step. Assume we have our matrix Z_4, being of the following form.



Again we choose the similarity Givens transformation $\check{G}_4$ to annihilate the element in position $(5,4)$. Applying this transformation results in the lower left figure. Instead of applying now the shift-through lemma we only need to combine the Givens transformations in position 4 and 5, resulting in a Hessenberg-like matrix as we wanted. Moreover, we immediately have the new representation of this Hessenberg-like matrix and therefore we can immediately perform a new step of the iteration.



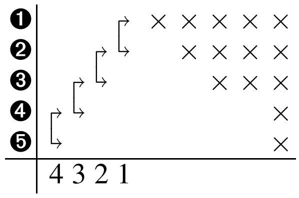
The resulting diagonal $D_5 = \mathrm{diag}([d_2, d_3, \ldots, d_n, \beta])$, in which β is a freely chosen element.

7.5 Special cases

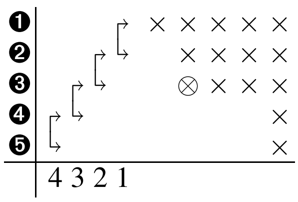
In the previous section the global flow of the algorithm was designed. Here we will investigate some cases in which breakdowns might occur.

- In the papers [5,21], it was assumed that the involved Hessenberg-like matrix was nonsingular. Singularity of the involved matrix implies that structured rank blocks crossing the diagonal are present (see [5,21]). More precisely this means that there are superdiagonal elements includable in the low rank structure. Here, this constraint is not necessary anymore. If there are such elements present in the matrix, these specific low rank blocks will gently move upwards until they appear in the upper left position and reveal the singularity. Let us show this in more detail. Assume we have a Hessenberg-like matrix Z having the element in position $(3,4)$ includable in the structure (see the left figure), with its representation on the right.

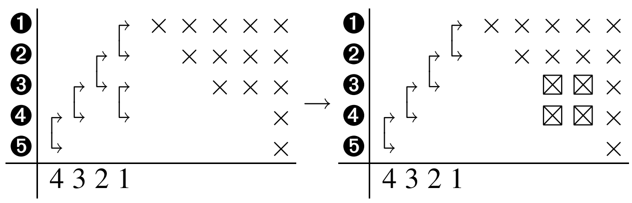
$$Z = \begin{bmatrix} \boxtimes & \times & \times & \times & \times \\ \boxtimes & \boxtimes & \times & \times & \times \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \times \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \times \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} \quad \rightarrow$$



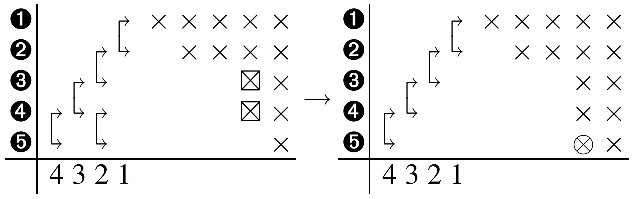
It is clear that in the representation there appears an extra zero on the diagonal.
Suppose that we are doing our chasing procedure and at a certain point we arrive at the following situation:



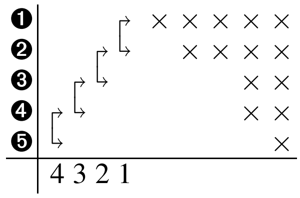
Similarly as in the previous section we will remove this bulge by a similarity transformation, such that the transformation on the right annihilates the element $\otimes$.
Performing the similarity Givens transformation results in the following figures:



As a result we obtain a bulge, which is part of a 2×2 rank 1 part (depicted by the elements $\boxtimes$). Removing now the bulge by a similarity transformation, will create more zeros due to the rank 1 structure. As a result we obtain the following.



Removing the final bulge results in the following representation of the matrix Z.



Comparing the final result with the original matrix, we see that the zero on the main diagonal of the upper triangular factor in the representation has shifted up one position. This means that eventually the singularity will pop up at the top left position of the matrix Z.

Remark 11 A similar result holds for Hessenberg-like plus diagonal matrices.

– A more delicate problem arises when a diagonal entry of the matrix D equals 0. We know from the construction of the QH-factorization, that the nonzero condition is compulsory for obtaining an essentially unique factorization. Hence we might expect serious difficulties in the implicit algorithm, which can even result in a breakdown. Due to the fact that the matrices resulting from the explicit and the implicit algorithm are not essentially identical anymore. This condition is however checked in a straightforwad way. Hence we assume our diagonal matrix to have all diagonal entries different from the shift.

8 The QR-iteration on Hessenberg matrices is a disguised QH-iteration

In the previous part of the paper we constructed an essentially unique QH-factorization to make the QH-method suitable for working with Hessenberg-like and Hessenberg-like plus diagonal matrices.

Let us see now how we can easily construct an essentially unique QH-factorization of a Hessenberg matrix.

Let us compute now the QH-factorization of a Hessenberg matrix, based on a sequence of descending Givens transformations. Remark that a Hessenberg matrix has already the strictly lower triangular part of semiseparability rank 1. Hence the descending sequence of Givens transformations will be constructed in such a way to expand the strictly lower triangular rank structure, to include the diagonal into it. Let us first consider the structure of the involved Givens transformations.

Corollary 1 *Suppose the row $[e, f]$ and the following 2×2 matrix are given*

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Then there exists a Givens transformation

$$G = \frac{1}{\sqrt{1+t^2}} \begin{bmatrix} \bar{t} & -1 \\ 1 & t \end{bmatrix} \tag{8}$$

such that the second row of the matrix $G^H A$, and the row $[e, f]$ are linearly dependent. The value t in the Givens transformation G as in (8)*, is defined as*

$$t = \frac{af - be}{cf - de},$$

under the assumption that $cf - de \neq 0$, otherwise one could have taken $G = I_2$.

Proof The proof involves straightforward computations.

Hence, we want to apply a sequence of successive Givens transformations onto the Hessenberg matrix H, to obtain the QH-factorization. Denote the diagonal elements of the Hessenberg matrix as $[a_1,\dots,a_n]$ and the subdiagonal elements as $[b_1,\dots,b_{n-1}]$. The first Givens transformation acts on rows 1 and 2 and only the first two columns are important, we have the following matrix A (as in the corollary):

$$A = \begin{bmatrix} a_1 & h_{1,2} \\ b_1 & a_2 \end{bmatrix}, \tag{9}$$

and we want to make the last row dependent of $[0,b_2]$. A Givens transformation with t defined as $t = \frac{a_1 b_2}{b_1 b_2} = \frac{a_1}{b_1}$, is found (assuming b_1 and b_2 to be different from zero).

Computing the product $G^H A$ gives us the following equalities:

$$G^H A = \frac{1}{\sqrt{1+t^2}} \begin{bmatrix} \bar{t} & 1 \\ -1 & t \end{bmatrix}, \begin{bmatrix} a_1 & b_1 \\ b_1 & a_2 \end{bmatrix}, = \begin{bmatrix} \times & \times \\ 0 & \times \end{bmatrix}.$$

One can continue this process, and as a result we obtain the following equations:

$$H = \check{Q}\check{Z} = QR.$$

The Hessenberg-like matrix $\check{Z}$ will become an upper triangular matrix. Hence in this case the new QH-factorization coincides with the traditional QR-factorization and therefore also the QR-algorithm for Hessenberg (as well as tridiagonal) matrices fits perfectly into the framework.

9 Numerical experiments

In this section we will illustrate by various numerical experiments the speed and accuracy of the proposed method.

9.1 Comparison with the traditional QR-method for symmetric semiseparable matrices

In the following experiment we constructed arbitrary symmetric semiseparable matrices, and computed their eigenvalues via the traditional QR-method for semiseparable matrices (the implementation from [20], was used). These eigenvalues were compared with the algorithm described in this paper. The eigenvalues computed by the MATLAB routine EIG were used to compare both solutions with. The following relative error norm was used: denote the vectors containing the eigenvalues as Λ, Λ_{QH} and Λ_{QR} for respectively EIG, the QH and the QR-method. The plotted error value equals

$$\frac{||\Lambda - \Lambda_{QH}||}{||\Lambda||} \text{ and } \frac{||\Lambda - \Lambda_{QR}||}{||\Lambda||},$$

for both methods. Five experiments were performed, and the line denotes the average accuracy of all five experiments combined. The x-axis denotes the problem sizes, ranging from 100 to 700, via steps of size 50. The cut-off criterion was chosen equal to 10^{-8}. In the figures, the $\circ$'s denote the results of individual experiments of the QR-iteration, whereas the $\star$'s denote these of the QH-iteration.

The following figures show the average number of iterations and the cputimings for both methods, we see that the new method needs in average much less iterations.

9.2 Comparison with nonsymmetric complex matrices

In this section we describe the results for similar experiment as above, but for complex, not necessarily symmetric matrices. The examples range from 100 to 700 via steps of size 50 and the cut-off criterion is set to 10^{-14} now.

The following figures show the average number of iterations and the cputimings, for both methods, we see that the new method needs in average much less iterations.

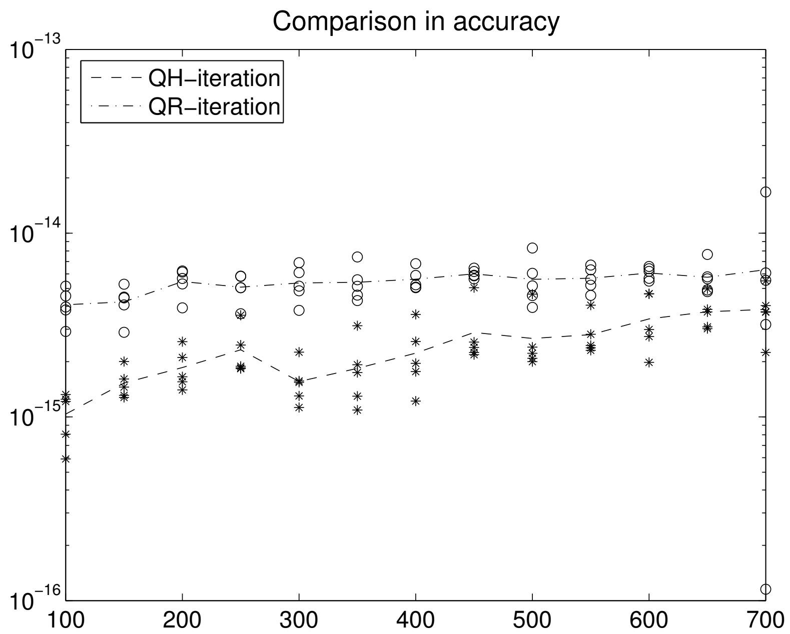


Figure 1. Accuracy comparison

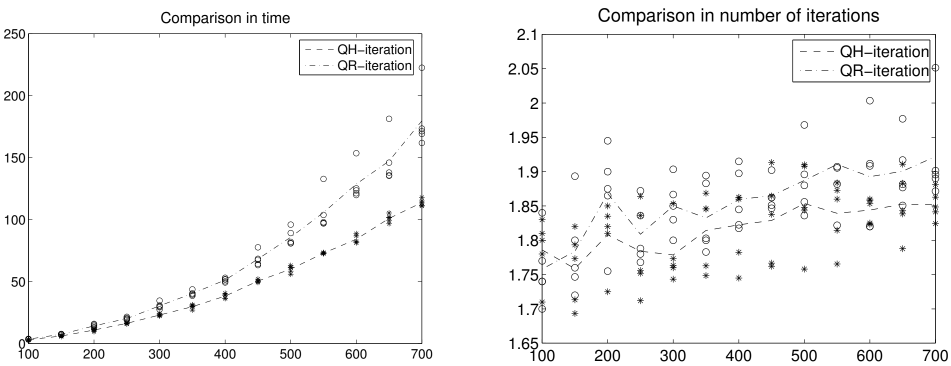


Figure 2. Timings and iterations

10 Conclusions

In this paper we proposed a new method which superseeds the traditional eigenvalue methods for both Hessenberg-like (higher order) and Hessenberg-like (higher order) plus diagonal. The complexity of the involved methods is halved w.r.t. the traditional *QR*-methods and moreover the convergence is enhanced.

References

1. 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.
2. S. Delvaux and M. Van Barel. The explicit QR-algorithm for rank structured matrices. Technical Report TW459, Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3000 Leuven (Heverlee), Belgium, May 2006.
3. D. A. Bini, F. Daddi, and L. Gemignani. On the shifted *QR* iteration applied to companion matrices. *Electronic Transactions on Numerical Analysis*, 18:137–152, 2004.
4. D. A. Bini, Y. Eidelman, L. Gemignani, and I. C. Gohberg. Fast QR eigenvalue algorithms for Hessenberg matrices which are rank-one perturbations of unitary matrices. *SIAM Journal on Matrix Analysis and its Applications*, 29(2):566–585, 2007.

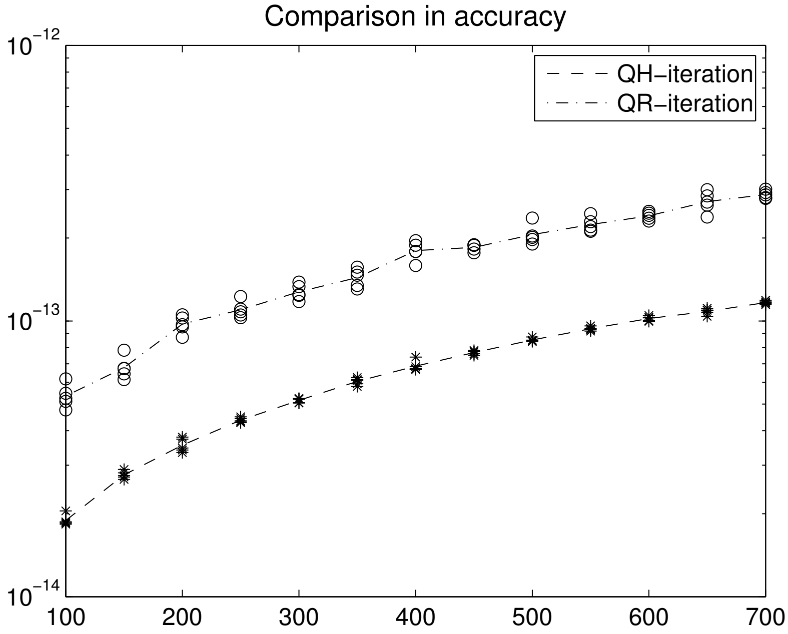


Figure 3. Accuracy comparison

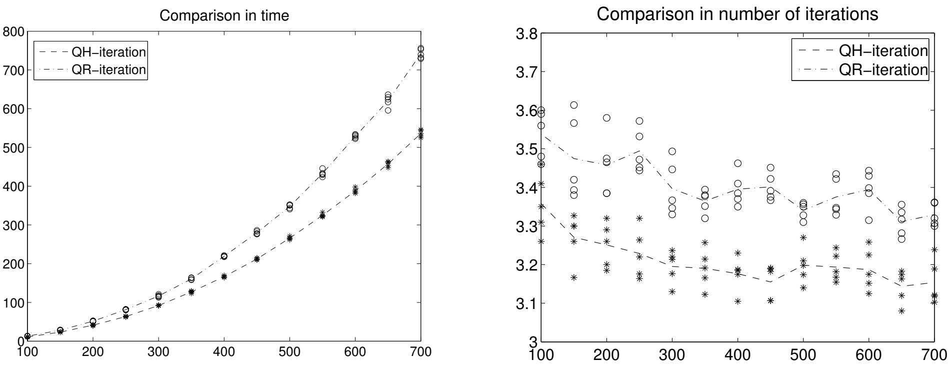


Figure 4. Timings and iterations

5. 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.
6. G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, Maryland, third edition, 1996.
7. B. N. Parlett. *The Symmetric Eigenvalue Problem*, volume 20 of *Classics in Applied Mathematics*. SIAM, Philadelphia, 1998.
8. D. S. Watkins and L. Elsner. Theory of decomposition and bulge-chasing algorithms for the generalized eigenvalue problem. *SIAM Journal on Matrix Analysis and its Applications*, 15(3):943–967, July 1994.
9. E. Van Camp, N. Mastronardi, and M. Van Barel. Two fast algorithms for solving diagonal-plus-semiseparable linear systems. *Journal of Computational and Applied Mathematics*, 164-165:731–747, 2004.
10. S. Delvaux and M. Van Barel. A Givens-weight representation for rank structured matrices. Technical Report TW453, Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3000 Leuven (Heverlee), Belgium, March 2006. (To appear in SIMAX).
11. 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.
12. S. Delvaux and M. Van Barel. A QR-based solver for rank structured matrices. Technical Report TW454, Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3000 Leuven (Heverlee), Belgium, March 2006.

13. R. Vandebril, M. Van Barel, and N. Mastronardi. A parallel QR-factorization/solver of structured rank matrices. Technical Report TW474, Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3000 Leuven (Heverlee), Belgium, October 2006.
14. D. Bindel, J. W. Demmel, W. Kahan, and O. Marques. On computing Givens rotations reliably and efficiently. *ACM Transactions on Mathematical Software*, 28(2):206–238, June 2002.
15. R. Vandebril, M. Van Barel, and N. Mastronardi. *Matrix Computations and Semiseparable Matrices, Volume I: System Solving*. Johns Hopkins University Press, 2007.
16. S. Delvaux and M. Van Barel. Eigenvalue computation for unitary rank structured matrices. Technical Report TW465, Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3000 Leuven (Heverlee), Belgium, July 2006.
17. 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, Largo Bruno Pontecorvo 5, 56127 Pisa, Italy, 2004.
18. D. A. Bini, L. Gemignani, and V. Y. Pan. Fast and stable *QR* eigenvalue algorithms for generalized companion matrices and secular equations. *Numerische Mathematik*, 100(3):373–408, 2005.
19. R. Vandebril, M. Van Barel, and N. Mastronardi. An implicit Q theorem for Hessenberg-like matrices. *Mediterranean Journal of Mathematics*, 2:59–275, 2005.
20. R. Vandebril, M. Van Barel, and N. Mastronardi. A multiple shift *QR*-step for structured rank matrices. Technical Report TW489, Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3000 Leuven (Heverlee), Belgium, April 2007.
21. 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.
22. Y. Eidelman and I. C. Gohberg. On a new class of structured matrices. *Integral Equations and Operator Theory*, 34:293–324, 1999.
23. E. Van Camp. *Diagonal-plus-semiseparable matrices and their use in numerical linear algebra*. PhD thesis, Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3000 Leuven (Heverlee), Belgium, May 2005.
24. S. Delvaux and M. Van Barel. Structures preserved by the QR-algorithm. *Journal of Computational and Applied Mathematics*, 187(1):29–40, March 2006.
25. R. Vandebril, M. Van Barel, and N. Mastronardi. A rational *QR*-iteration. Technical Report TW405, Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3000 Leuven (Heverlee), Belgium, September 2007.
26. D. S. Watkins. *QR*-like algorithms — An overview of convergence theory and practice. *Lectures in Applied Mathematics*, 32:879–893, 1996.
27. 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.
28. D. S. Watkins. Understanding the *QR* algorithm. *SIAM Review*, 24(4):427–440, 1982.
29. R. Vandebril, E. Van Camp, M. Van Barel, and N. Mastronardi. On the convergence properties of the orthogonal similarity transformations to tridiagonal and semiseparable (plus diagonal) form. *Numerische Mathematik*, 104:205–239, 2006.
30. D. Fasino. Rational Krylov matrices and QR-steps on Hermitian diagonal-plus-semiseparable matrices. *Numerical Linear Algebra with Applications*, 12(8):743–754, October 2005.
31. R. Bevilacqua and G. M. Del Corso. Structural properties of matrix unitary reduction to semiseparable form. *Calcolo*, 41(4):177–202, 2004.