

A rational QR -iteration

Raf Vandebril
Marc Van Barel
Nicola Mastronardi

Report TW 504, September 2007



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

A rational QR -iteration

Raf Vandebril
Marc Van Barel
Nicola Mastronardi

Report TW 504, September 2007

Department of Computer Science, K.U.Leuven

Abstract

In this manuscript a new type of QR -iteration will be presented. Each step of this new iteration consists of two substeps. In the explicit version, first an RQ -factorization of the initial matrix $A - \kappa I = RQ$ will be computed, followed by a QR -factorization of the matrix $(A - \sigma I)Q^H$. Applying the unitary similarity transformation defined by the QR -factorization of the transformed matrix $(A - \sigma I)Q^H$, will yield interesting convergence properties. It will be shown that the convergence behavior is related to a subspace iteration based on a rational function in A namely $(A - \sigma I)(A - \kappa I)^{-1}$. Convergence properties of this new iteration will be investigated and examples will be presented, illustrating the effectiveness of this approach with respect to some specific classes of matrices

Keywords : QR -algorithm, eigenvalues, rational functions

AMS(MOS) Classification : Primary : 65F15 Secondary : 15A18.

A rational QR -iteration ^{*}

Raf Vandebril¹, Marc Van Barel¹, Nicola Mastronardi²

¹ KULeuven, Dept. of Computer Science, 3001 Leuven(Heverlee)
e-mail: {raf.vandebril, marc.vanbarel}@cs.kuleuven.be

² Istituto per le Applicazioni del Calcolo M. Picone, sez. Bari,
e-mail: mastronardi@ba.iac.cnr.it

25th September 2007

Abstract In this manuscript a new type of QR -iteration will be presented. Each step of this new iteration consists of two substeps. In the explicit version, first an RQ -factorization of the initial matrix $A - \kappa I = RQ$ will be computed, followed by a QR -factorization of the matrix $(A - \sigma I)Q^H$. Applying the unitary similarity transformation defined by the QR -factorization of the transformed matrix $(A - \sigma I)Q^H$, will yield interesting convergence properties. It will be shown that the convergence behavior is related to a subspace iteration based on a rational function in A namely $(A - \sigma I)(A - \kappa I)^{-1}$. Convergence properties of this new iteration will be investigated and examples will be presented, illustrating the effectiveness of this approach with respect to some specific classes of matrices.

Key words QR -algorithm, eigenvalues, rational functions

1 Introduction and preliminary results

If one wants to compute all eigenvalues (and corresponding eigenvectors) of a matrix A , quite often the standard QR -method is used. For a certain matrix A , and a specific shift σ we obtain the following equations, determining the QR -method:

$$\begin{aligned}A - \sigma I &= QR, \\ \hat{A} &= RQ + \sigma I = Q^H A Q,\end{aligned}$$

where \hat{A} yields the new iterate. When the shift σ is well-chosen, the method will converge to an upper triangular matrix containing the eigenvalues of the original matrix A (see, e.g., [1,2]). In the equation above \hat{A} can be computed in two different ways, the first manner is referred to as the explicit QR -method, whereas the formula $Q^H A Q$ can lead to an implicit approach (see, e.g., [3]). The new algorithm presented in this manuscript admits both types of methods. We will not design an implicit method, as it involves a lot of technical details, and heavily depends on the matrix class considered. In this manuscript we only derive the new algorithm and we do not restrict ourselves to any specific class of matrices. Details on how to implement an implicit version will be given.

* 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 Poles of Attraction Poles, 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.

The cost of the QR -iteration is mostly determined by the structure of the unitary matrix Q , coming from the QR -factorization of the matrix $A - \sigma I$. Thus its complexity is determined by the structure of the matrix A . For example, if the matrix A is of Hessenberg form, the matrix Q can be written as a product of $n - 1$ Givens transformations, chosen to remove the subdiagonal elements of the matrix A . These $n - 1$ Givens transformations result in a low complexity QR -step ($O(n^2)$). Other structures of matrices admitting a low complexity QR -step are tridiagonal ($O(n)$, see, e.g., [4, 2]), semiseparable ($O(n)$, see, e.g., [5, 6]), semiseparable plus diagonal matrices ($O(n)$, see, e.g., [7, 8]) and others. Essential for retaining a low complexity method is the preservation of the structure of the matrix A under a step of the QR -algorithm. This preservation of structure is guaranteed in the classes of matrices mentioned before.

The convergence of the traditional QR -algorithm, and all its variants (generically called GR -algorithms¹ [9]) can be interpreted as subspace iteration, determined by polynomials in the matrices A , followed by a coordinate transformation. Standard polynomial forms for one step of the standard QR -algorithm look like $p(\lambda) = \lambda - \sigma$, corresponding to the single shift strategy and $p(\lambda) = (\lambda - \sigma_1)(\lambda - \sigma_2)$ corresponding to the double shift strategy. Based on these polynomials one can derive bounds on the convergence of GR -algorithms.

In this manuscript we will provide a method for easily computing another type of QR -iteration, such that the resulting subspace iteration will be of rational type. This means that the corresponding subspace iteration will be performed with matrices $(A - \sigma I)(A - \kappa I)^{-1}$, corresponding to functions $(\lambda - \sigma)(\lambda - \kappa)^{-1}$. Theoretically, in order to perform a step of an iteration described above, one needs to compute the QR -factorization of the matrix $(A - \sigma I)(A - \kappa I)^{-1}$. It will be shown how one can compute the unitary matrix Q of this factorization without explicitly computing the inverse of the matrix $(A - \kappa I)$. Moreover, we will see that we want κ to be close to an eigenvalue for speeding up the convergence, this makes of course inversion totally impossible. Issues such as retaining of the structure, and the convergence properties will be discussed. Results on the convergence properties, with specific well-chosen shifts will be presented and it will be shown that this convergence ‘can’ supersede the traditional QR -algorithm. Moreover, examples will illustrate that for certain classes of matrices computing this rational QR -factorization can be achieved by less computational effort than computing the standard QR -factorization.

The manuscript is organized as follows: in the first section the new iteration will be described. This section will also describe the interpretation as a rational QR -step. In Section 3, the convergence speed of the new iteration will be investigated, using subspace iteration results. Section 4 will discuss some results on how to design implicit versions for the presented method. Indications on how to choose the shifts are given. In Section 5 some examples of matrices suitable for this iteration are presented, followed by numerical experiments in Section 6. Finally, conclusions and future work will be presented.

As this rational QR -method is related in some sense to rational Krylov methods, we will briefly mention some results related to rational Krylov methods.

1.1 Rational Krylov methods

Most of the iterative methods for computing eigenvalues of structured/sparse matrices are based on the so-called Krylov subspaces. Given a matrix A whose eigenvalues one would like to compute and an initial vector \mathbf{v} we have the following Krylov subspace

$$\mathcal{K}^j(A, \mathbf{v}) = \{\mathbf{v}, A\mathbf{v}, \dots, A^{j-1}\mathbf{v}\}.$$

The idea of using rational functions in A instead of the standard powers of A can be found in [10]. The idea is to work with the following Krylov sequence, in which all $\phi_i(\lambda)$ are rational functions in λ :

$$\mathcal{F}^j(A, \mathbf{v}) = \{\phi_1(A)\mathbf{v}, \phi_2(A)\mathbf{v}, \dots, \phi_3(A)\mathbf{v}\}.$$

¹ This covers a more general class of iterations such as QR -methods, LR -methods and so forth.

The idea in Rational Krylov methods is to choose the functions in an intelligent way, to speed up the convergence of the iterative methods.

Similarly one chooses a good shift to speed up convergence in case of the QR -method. In this manuscript we will develop a technique to perform rational QR -steps onto matrices, in order to speed up the convergence towards the eigenvalues.

How rational Krylov methods work can be found in, e.g., [10, 11, 12]

To conclude, we would like to mention the following. When considering a tridiagonal matrix T and the standard Krylov subspace, it is well-known that there is a relation between the QR -factorization of the matrix T and this Krylov sequence (see [4]). Similarly there is a relation between rational Krylov sequences and semiseparable plus diagonal matrices (see [13]).

The interested reader can find more information on rational Krylov methods in the references above.

2 The new method

In this section we will start by describing the new iteration. Then it will be shown that the orthogonal matrix Q computed in the new process can be seen as the Q -factor from the QR -factorization of a rational function in the matrix A . Finally, the preservation of the structure and nonunitary transformations similarity transformations will be discussed.

2.1 The iteration

Let us consider the matrix $A^{(1)} = A$, whose eigenvalues we would like to compute. It will be shown that the sequence $A^{(1)}, A^{(2)}, A^{(3)}, \dots$ converges to upper triangular form. For each $A^{(i)}$ (with $i = 1, 2, 3, \dots$) we compute two shifts $\kappa^{(i)}$ and $\sigma^{(i)}$.

We have the following relations:

$$A^{(i)} - \kappa^{(i)}I = R^{(i)}Q^{(i)}.$$

This will be used in the following equations:

$$\begin{aligned} A^{(i)} - \sigma^{(i)}I &= ((A^{(i)} - \kappa^{(i)})Q^{(i)H} - (\sigma^{(i)} - \kappa^{(i)})Q^{(i)H})Q^{(i)} \\ &= (R^{(i)} - (\sigma^{(i)} - \kappa^{(i)})Q^{(i)H})Q^{(i)}. \end{aligned}$$

Computing now the QR -factorization of the matrix $R^{(i)} - (\sigma^{(i)} - \kappa^{(i)})Q^{(i)H}$, and substituting this result in the equations above leads to the following result:

$$\begin{aligned} \hat{Q}^{(i)}\hat{R}^{(i)} &= R^{(i)} - (\sigma^{(i)} - \kappa^{(i)})Q^{(i)H}, \\ A^{(i)} - \sigma^{(i)}I &= \hat{Q}^{(i)}\hat{R}^{(i)}Q^{(i)}. \end{aligned}$$

Applying now the following unitary similarity transformation onto the matrix A gives us the next iterate:

$$A^{(i+1)} = \hat{Q}^{(i)H} A^{(i)} \hat{Q}^{(i)}.$$

Based on matrix $A^{(i+1)}$ we can continue with our iteration process. In the remainder of the manuscript we will show that this iteration can be interpreted as a rational QR -iteration, where Q is the unitary matrix coming from the Cayley transform of the original matrix A . The computational complexity of the method is not yet discussed, but we will show that this type of iteration can be computed very effectively for some classes of matrices.

Note that in the remainder of the manuscript, we will omit the superscripts when they are clear from the context.

2.2 A rational QR -factorization

The following analysis is extremely simple, but helps us throughout the remainder of the manuscript for understanding the behavior of the new method.

The previous iteration corresponds to performing a rational shift iteration step onto the matrix A . It is easy to prove that the computed matrix \hat{Q} is the one coming from the QR -factorization of the following matrix:

$$\begin{aligned} (A - \sigma I)(A - \kappa I)^{-1} &= \hat{Q}\hat{R}Q^HR^{-1} \\ &= \hat{Q}\hat{R}^{-1}. \end{aligned}$$

We can assume without loss of generality that the matrix $A - \kappa I$ is invertible. In fact, a singular $A - \kappa I$ is good news, as the RQ -factorization reveals an eigenvalue κ . It will be shown in the next section, that we want κ to be as close as possible to an eigenvalue of the matrix A . This means that the matrix $A - \kappa I$ will get very ill-conditioned, and therefore inverting the matrix makes no sense. Luckily, in the practical algorithm the inversion of $A - \kappa I$ is not needed, and has therefore no negative effect on the stability of the method. The inversion of $A - \kappa I$ is only needed for theoretical purposes.

2.3 Preservation of the structure

Essential in effective QR -algorithms is the preservation of the structure under a step of the QR -method.

The preservation of the matrix structure is a not so difficult result as one computes the QR -factorization of a rational function in A . This means that one can decompose this iteration into two different independent steps, both preserving the structure. Let us provide a more thorough explanation.

Suppose a matrix A is given, whose structure is preserved under an iteration of the QR -algorithm. Any kind of structure can be considered such as Hessenberg, tridiagonal, band, Hessenberg-like, semiseparable, unitary plus low rank (see, e.g., [14, 15]), etc.² Similarly as in the QR -case one can prove that a step of the RQ -algorithm preserves the structure of the matrix A .

We know that

$$(A - \sigma I)(A - \kappa I)^{-1} = \hat{Q}\hat{R},$$

and we want to prove that $\hat{A} = \hat{Q}^HA\hat{Q}$ has the same structure as the original matrix A .

The equation above can be decomposed into two steps. In a first step we compute the RQ -factorization of the matrix $(A - \kappa I)$, and we perform a step of the RQ -method.

$$\begin{aligned} (A - \kappa I) &= RQ, \\ \tilde{A} &= QAQ^H. \end{aligned}$$

It is known that the matrix \tilde{A} inherits the structure of the matrix A . In the second step a QR -step is performed on the matrix \tilde{A} . We obtain:

$$\begin{aligned} (\tilde{A} - \sigma I) &= \tilde{Q}\tilde{R}, \\ \hat{A} &= \tilde{Q}^H\tilde{A}\tilde{Q} = \tilde{Q}^HQAQ^H\tilde{Q} \end{aligned}$$

The traditional QR -method states now that the matrix \hat{A} satisfies the same structural constraints as \tilde{A} and hence of A .

The statement is proved, if \hat{A} is essentially equivalent to \tilde{A} .³

² Even though not mentioned, in case of a band matrix, a tridiagonal matrix, ... we implicitly assume the matrices to be symmetric.

³ With essentially equivalent is meant that the elements are the same up to a factor, where all factors are roots of unity.

The connection between the QR -factorization of $(A - \sigma I)(A - \kappa I)^{-1}$ and the above results is the following:

$$\hat{Q}\hat{R} = (A - \sigma I)(A - \kappa I)^{-1} = (A - \sigma I)Q^H R^{-1} \quad (1)$$

$$= Q^H Q(A - \sigma I)Q^H R^{-1} \quad (2)$$

$$= Q^H(\tilde{A} - \sigma I)R^{-1} \quad (3)$$

$$= Q^H \tilde{Q}\tilde{R}R^{-1}. \quad (4)$$

It is clear that the matrices \hat{Q} and $Q^H \tilde{Q}$ are essentially the same, hence they determine essentially the same similarity transformation. As a result the structure of the matrix A is preserved under the similarity transformation determined by the matrix \hat{Q} , from Equation (1).

In the following section we will investigate in more detail the convergence rate of the involved method. The analysis will be based on the subspace iteration induced by the rational function in A , namely $p(\lambda) = (\lambda - \sigma)(\lambda - \kappa)^{-1}$.

2.4 Non-unitary factorizations

Until now, we have always assumed to be working with unitary transformations. In general, however, one can easily loosen this constraint. Without any loss of generality one can use non-unitary transformations. Suppose we have the following equations (with G and \hat{G} invertible, not necessarily unitary, matrices):

$$A - \kappa I = RG,$$

for which R is an upper triangular matrix. This will be used in the following equations:

$$A - \sigma I = ((A - \kappa)G^{-1} - (\sigma - \kappa)G^{-1})G$$

$$= (R - (\sigma - \kappa)G^{-1})G,$$

$$A - \sigma I = \hat{G}\hat{R}G,$$

the matrix \hat{R} is again upper triangular. The following similarity transformation is then determined by the matrix \hat{G} :

$$\hat{A} = \hat{G}^{-1}A\hat{G}.$$

This brings us back to the case of GR -algorithms (see [9]). Unfortunately loosening the unitarity constraint can result in the use of ill conditioned matrices. One can also use a combination of unitary and non-unitary transformations. For example using a non-unitary transformation $G^{(1)}$ and a unitary $\hat{G}^{(1)}$ still determines a unitary similarity transformation onto the matrix $A^{(1)}$, instead of simply a similarity transformation. This creates flexibility in the design of the method.

In the next section we will assume we are in the general GR -case, for proving results related to the convergence of the method.

3 The convergence

Based on the results of the previous section we will derive here the convergence speed of the presented algorithm, based on subspace iteration theory. Our analysis uses a similar approach as presented in [9, 16]. We will recall some basic definitions for a full understanding of the presented convergence analysis. In the numerical experiments section, we will illustrate that the computed bound below is rather sharp. We note that we adapted the presented results such that one can use rational functions.

3.1 General subspace iteration theory related to GR-algorithms

A more elaborate study on this subject can be found in [9, 4].

Subspace iteration can be seen as the following iteration:

$$S_i = p_i(A)S_{i-1},$$

starting from an initial subspace $S_0 = S$. The $p_i(\lambda)$ are suitably chosen polynomials (or rational functions in our case), in order to speed up the convergence. Under some mild constraints this will converge to an invariant subspace. Defining $\hat{p}_i(\lambda) = p_i(\lambda) \dots p_2(\lambda)p_1(\lambda)$, we also have

$$S_i = \hat{p}_i(A)S_0.$$

A GR-method, in which R defines an upper triangular matrix, and G is an invertible transformation matrix is based on transformations of the following form: $p_i(A^{(i)}) = G^{(i)}R^{(i)}$, where the new iterate is defined as follows:

$$A^{(i+1)} = G^{(i)-1}A^{(i)}G^{(i)}. \quad (5)$$

This can easily be interpreted as a subspace iteration step, followed by a coordinate transformation. Namely first a subspace iteration step is performed onto $\langle \mathbf{e}_1, \dots, \mathbf{e}_k \rangle$ (for all $1 \leq k \leq n$ at the same time):

$$p_i(A^{(i)})\langle \mathbf{e}_1, \dots, \mathbf{e}_k \rangle = \langle \mathbf{g}_1, \dots, \mathbf{g}_k \rangle,$$

where the \mathbf{g}_i are the columns of the matrix $G^{(i)}$, which means that $G^{(i)} = [\mathbf{g}_1, \dots, \mathbf{g}_n]$.

To conclude a coordinate transformation has to be performed, defined by Equation (5). This maps the subspace $\langle \mathbf{g}_1, \dots, \mathbf{g}_k \rangle$ back to the subspace $\langle \mathbf{e}_1, \dots, \mathbf{e}_k \rangle$.

Hence, in case of a GR-method, one does not work with a sequence of changing subspaces, but with a sequence of changing matrices. Gradually the lower triangular part of the matrices $A^{(i)}$ should converge to zero, thereby revealing the eigenvalues on the diagonal of the matrix $A^{(i)}$ ⁴.

We did not yet specify the $p_i(\lambda)$ in this case. Considering Equation (5), one can easily see that for the standard QR-method one considers in every step $p_i(\lambda) = \lambda - \sigma_i$. Where σ_i is a suitable chosen shift, e.g., the Rayleigh or the Wilkinson shift. In our case the considered rational functions are of the form $p_i(\lambda) = (\lambda - \sigma_i)(\lambda - \kappa_i)$.

3.2 Convergence of subspace iteration

Generally we can state the following results on the convergence of subspace iteration methods, related to the functions (that can also be rational) $p_i(\lambda)$.

Given two subspaces S and \mathcal{T} in \mathbb{C}^n and denote with P_S and $P_{\mathcal{T}}$ the orthonormal projector onto the subspace S and \mathcal{T} respectively. The standard metric between subspaces (see [4]) is defined as

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

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

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

Theorem 1 (Theorem 5.1 from [9]) *Given a simple⁵ matrix $A \in \mathbb{C}^{n \times n}$ with $\lambda_1, \lambda_2, \dots, \lambda_n$ as eigenvalues and associated linearly independent eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$. Let $V = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n]$ and κ_V be the condition number of V , w.r.t. to the spectral⁶ norm. Let k be an integer $1 \leq k \leq n - 1$, and define the invariant subspaces $\mathcal{U} = \langle \mathbf{v}_{k+1}, \dots, \mathbf{v}_n \rangle$ and $\mathcal{T} = \langle \mathbf{v}_1, \dots, \mathbf{v}_k \rangle$. Denote with $(p_i)_i$ a*

⁴ In some cases one forces convergence to blocks on the diagonal, which then contain the eigenvalues.

⁵ A matrix is called simple if it has n linearly independent eigenvectors.

⁶ The spectral norm is naturally induced by the $\|\cdot\|_2$ norm on vectors.

sequence of rational functions and let $\hat{p}_i = p_i \dots p_2 p_1$. Suppose that the functions have no zeros and no poles in the λ_j , more precisely

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

for all i , and let

$$\hat{r}_i = \frac{\max_{k+1 \leq j \leq n} |\hat{p}_i(\lambda_j)|}{\min_{1 \leq j \leq k} |\hat{p}_i(\lambda_j)|}. \quad (6)$$

Let S_0 be a k -dimensional subspace of \mathbb{C}^n , satisfying

$$S_0 \cap \mathcal{U} = \{0\}.$$

Let $S_i = \hat{p}_i(A)S_0, i = 1, 2, \dots$. Then there exists a constant C (depending on S_0) such that for all i ,

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

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

$$C = \frac{d(V^{-1}S_0, V^{-1}\mathcal{T})}{\sqrt{1 - d(V^{-1}S_0, V^{-1}\mathcal{T})}}.$$

We remark, that similar theorems exist for defective matrices. Also more information concerning the conditions put on the matrices in Theorem 1, can be found in [9]. We will not go into these details.

Note 1 In the previous theorem the accumulated contraction rate was denoted by \hat{r}_i . This contraction rate is related to several steps of the new method at once. One can define in a similar fashion the contraction rate to go from subspace S_{i-1} to subspace S_i (one needs to replace in this case S_0 by S_i). This the contraction rate of a single step of the iteration. We will denote this contraction rate by r_i . In the case of a single step, one only has to take into consideration the rational function $p_i(\lambda)$ instead of the accumulated rational function $\hat{p}_i(\lambda)$.

Note that not necessarily $\hat{r}_i = r_i r_{i-1} \dots r_1$. There is a connection, but as the final result on $d(S_i, \mathcal{T})$ also incorporates a constant C , which is dependent on S_i in this case, these values change.

From a certain viewpoint, only the global contraction ratio \hat{r}_i is significant, because the r_i 's can indicate false convergence for a few steps.

Nevertheless, the values r_i are most often good indicators for the observed convergence. As the single step case is more easy to analyze we will restrict ourselves in the upcoming analysis to a single step of the new method. It is clear from the above theorem that we would like to get the value of \hat{r}_i or r_i , in case of a single step, as small as possible. A small value will force the subspace S_i to get closer and closer to \mathcal{T} , such that convergence is obtained. In order to obtain a small for \hat{r}_i or r_i , one can choose the shifts such that the numerator gets as small as possible and the denominator as large as possible.

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

Lemma 1 (Lemma 6.1 from [9]) 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 S to be the subspace spanned by the first k columns of G . (The subspace 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:

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

where κ_G denotes the condition number of the matrix G .

We are now ready to use these theorems, to derive an upper bound on the norm of the sub-blocks, while performing the rational QR -iteration. To get a clear indication on how this method behaves w.r.t. the traditional QR -method. We will compare the convergence speed of this approach, with the traditional convergence results.

3.3 Choice of the shifts

In this subsection we will consider the choice of the shifts to obtain an ‘optimal’ convergence behavior. Suppose we have a matrix A , with given eigenvalues $\lambda_1, \dots, \lambda_n$, and eigenvectors $\mathbf{v}_1, \dots, \mathbf{v}_n$. Before deriving some results, we will rewrite the convergence rate from Theorem 1 to get another view towards the convergence. Reconsidering the convergence rate in Equation (6) we can see that in order to obtain a good rate one should choose the shifts such to make the maximum in the numerator in the equation as small as possible and to make the minimum in the denominator as large as possible.

Due to the continuity and monotonicity of the function $1/x$ in the interval $]0, \infty[$, we can rewrite our convergence rate r_i as follows, for a single step, i.e. to go from S_{i-1} to S_i (for simplicity we will consider single steps, an analysis of all steps combined can be given similarly):

$$r_i = \frac{\max_{k+1 \leq j \leq n} |p_i(\lambda_j)|}{\min_{1 \leq j \leq k} |p_i(\lambda_j)|} = \max_{k+1 \leq j \leq n} |p_i(\lambda_j)| \max_{1 \leq j \leq k} \left| \frac{1}{p_i(\lambda_j)} \right|.$$

To simplify the notation above we define the new functions $q_i(\lambda)$ as follows:

$$q_i(\lambda) = \frac{1}{p_i(\lambda)} = \frac{\lambda - \kappa_i}{\lambda - \sigma_i}.$$

Hence we can rewrite the equation above as follows:

$$r_i = \max_{k+1 \leq j \leq n} |p_i(\lambda_j)| \max_{1 \leq j \leq k} |q_i(\lambda_j)| = \max_{k+1 \leq j \leq n} \left| \frac{\lambda_j - \sigma_i}{\lambda_j - \kappa_i} \right| \max_{1 \leq j \leq k} \left| \frac{\lambda_j - \kappa_i}{\lambda_j - \sigma_i} \right|. \quad (7)$$

In order to obtain a fast convergence (which means a small r_i) we have to choose the shifts such as to minimize both maxima. In some sense both equations can be considered independently from each other. From the formula above it is quite clear that both shifts should be chosen different from each other, as there is no convergence otherwise. Let us distinguish between two cases now, afterwards we will compare this convergence rate with the standard QR convergence rates. To clearly see the difference we already depict here the convergence rates related to the traditional QR -method with shift σ_i :

$$r_i^{(QR)} = \frac{\max_{k+1 \leq j \leq n} |\lambda_j - \sigma_i|}{\min_{1 \leq j \leq k} |\lambda_j - \sigma_i|}. \quad (8)$$

Suppose we have the following set of eigenvalues $\{\lambda_1, \lambda_2, \dots, \lambda_{n-1}, \lambda_n\}$. The convergence rate above is true for every value of k . Specifying k , we can see how to choose the shifts such to obtain convergence.

- Assume we would like to force convergence towards λ_n , this means $k = n - 1$. In the traditional QR -algorithm one chooses the shift σ_i close to λ_n . The new method behaves similarly. To force convergence to this last eigenvalue λ_n , one tries to minimize the first factor in Equation (7). Hence choosing a sequence of $\sigma_i \rightarrow \lambda_n$ will force convergence to the last eigenvalue, independently from the choice of κ_i . Corresponding Lemma 1, we will refer to this convergence as convergence in the lower right corner of the matrix.
- Similarly, one can force convergence to the upper left position, i.e. $k = 1$ (with regard to Lemma 1. Suppose we have the following two sets of eigenvalues $\{\lambda_1\}$ and $\{\lambda_2, \dots, \lambda_n\}$. To obtain convergence to the eigenvalue λ_1 , one needs to minimize the second factor in Equation (8). Hence a sequence of shifts $\kappa_i \rightarrow \lambda_1$ is needed to obtain convergence.

It is obvious that both convergence rates interfere with each other. In the following description we will derive some upper bounds on this interference and compare this with the convergence rate of the traditional QR -method.

3.4 Relation to the QR -method

The convergence rate corresponding to the standard QR -method with shift σ_i is the following:

$$r_i^{(QR)} = \frac{\max_{k+1 \leq j \leq n} |\lambda_j - \sigma_i|}{\min_{1 \leq j \leq k} |\lambda_j - \sigma_i|}. \quad (9)$$

In order to compare this rate with the convergence rate of the new method, we need to introduce some constants:

$$\begin{aligned} \omega_i &= \min_{k+1 \leq j \leq n} \{|\lambda_j - \kappa_i|\}, \\ \Omega_i &= \max_{1 \leq j \leq k} \{|\lambda_j - \kappa_i|\}. \end{aligned}$$

Calculating an upper bound for the convergence, in case of $k = n - 1$, independent from κ_i , we obtain:

$$r_i = \max_{k+1 \leq j \leq n} \left| \frac{\lambda_j - \sigma_i}{\lambda_j - \kappa_i} \right| \max_{1 \leq j \leq k} \left| \frac{\lambda_j - \kappa_i}{\lambda_j - \sigma_i} \right| \leq \frac{\Omega_i \max_{k+1 \leq j \leq n} |\lambda_j - \sigma_i|}{\omega_i \min_{1 \leq j \leq k} |\lambda_j - \sigma_i|} = \frac{\Omega_i}{\omega_i} r_i^{(QR)}.$$

The above formula shows that up to a constant both convergence rates are almost equal. In case $\omega_i > \Omega_i$, we get an even better convergence rate (this will be shown in the numerical experiments). In case $\Omega_i > \omega_i$, we get a delay. This delay can be arbitrarily large, when choosing outrageous values for σ_i . But normally σ_i is located within the bounds of the spectrum and does therefore not tend to create a large delay. In the following example we show that the effect of the σ_i is quite limited.

Example 1 Assume we have the following spectrum $0 \leq \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$. The shift σ_i is chosen to be close to λ_n . Normally we choose κ_i to be close to λ_1 . Assume we would like to investigate convergence towards the largest eigenvalue λ_n , i.e., $k = n - 1$. We obtain the following convergence rate:

$$r_i \leq \frac{\Omega_i}{\omega_i} \frac{|\lambda_n - \sigma_i|}{\min_{1 \leq j \leq k} |\lambda_j - \sigma_i|},$$

with

$$\begin{aligned} \omega_i &= |\lambda_n - \kappa_i|, \\ \Omega_i &= \max_{1 \leq j \leq n-1} \{|\lambda_j - \kappa_i|\}. \end{aligned}$$

If $\lambda_{n-1} \gg \kappa_i$, the factor Ω_i/ω_i will be close to 1. If λ_{n-1} is close to λ_1 and hence close to κ_i , $\Omega_i/\omega_i \ll 1$ and hence convergence is even speed up. In both cases, convergence is guaranteed.

Note 2 We remark that gradedness of the matrix is quite advantageous for this approach, with the double shift. Problems can occur when bad choices for κ_i and σ_i are made, such as $\sigma_i < \kappa_i$.

The constraints posed on κ_i are not so strong and quite often they are satisfied in a natural manner.

Similarly as above one can derive an upper bound for the convergence based on the shift κ_i . Define the following constants:

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

Based on these constants we obtain the following convergence rate for convergence based on the shift κ_i .

$$r_i = \max_{k+1 \leq j \leq n} \left| \frac{\lambda_j - \sigma_i}{\lambda_j - \kappa_i} \right| \max_{1 \leq j \leq k} \left| \frac{\lambda_j - \kappa_i}{\lambda_j - \sigma_i} \right| \leq \frac{\Delta_i}{\delta_i} \frac{\max_{1 \leq j \leq k} |\lambda_j - \kappa_i|}{\min_{k+1 \leq j \leq n} |\lambda_j - \kappa_i|}. \quad (10)$$

Reconsidering now the following 3 sets $\{\lambda_1\}$, $\{\lambda_2, \dots, \lambda_{n-1}\}$ and $\{\lambda_n\}$ and let $\sigma_i \rightarrow \lambda_n$ and let $\kappa_i \rightarrow \lambda_1$. From the equations above one can see that, under some mild conditions, convergence will occur both at the top and at the bottom level at the same convergence rates as the QR -method would converge.

3.5 The case $\kappa_i = 0$

Let us briefly consider the special case $\sigma_i \neq 0$ and $\kappa_i = 0$. This analysis coincides with the case $\sigma_i = 0, \kappa_i \neq 0$. Both shifts equal to each other makes no sense, as nothing happens.

We will prove that besides the convergence posed by the shift σ_i , basic non shifted subspace iteration takes place for all k at the same time.

Consider therefore the convergence rate derived in Equation (10), simplified with $\kappa_i = 0$:

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

Assume now (without loss of generality), all eigenvalues to be ordered, i.e. $|\lambda_1| \leq |\lambda_2| \leq \dots \leq |\lambda_n|$. 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 next to the convergence imposed by the shift σ_i , which can force for example extra convergence towards the bottom right element.

The case $\kappa_i \neq 0$ and $\sigma_i = 0$ is almost identical. We will have standard convergence for $k = 1$ and a subspace iteration taking place for all other values of k .

4 Implicit and explicit version

It might seem at first that the presented method converges twice as fast, but is also twice as expensive. Twice as expensive, because one has to compute two factorizations: one QR - and one RQ -factorization to fully determine the complete QR -step.

In an explicit version of the method it is true that one has to compute two full factorizations. One has to compute the complete matrix Q to determine the next iterate. Implicit methods, in general, do not need the full matrix Q . Only the first column $Q\mathbf{e}_1$ is sufficient for determining the iteration in a unique way.

An implicit QR -method performed onto the structured matrix A , has only few building blocks (see also [3,4,17]):

- Determine the first column of the following matrix product $(A - \sigma_i)(A - \kappa_i)^{-1}$, i.e., determine $\mathbf{v} = (A - \sigma_i)(A - \kappa_i)\mathbf{e}_1$.
- Determine now an orthogonal transformation W_1 , either a single Householder transformation, or few Givens transformations, such that $W_1^H \mathbf{v} = \|\mathbf{v}\|\mathbf{e}_1$.
- Apply this transformation to the matrix A :

$$\tilde{A} = W_1^H A W_1.$$

The result of this transformation is a new matrix \tilde{A} , which does in general not meet the structural constraints posed on the matrix A , anymore.

- Restore the structure of the matrix \tilde{A} with on orthogonal transformation W_2 , such that $W_2\mathbf{e}_1 = \mathbf{e}_1$. Use this transformation to perform another similarity transformation onto the matrix \tilde{A} :

$$\hat{A} = W_2^H \tilde{A} W_2.$$

- Using a kind of implicit Q -theorem, one can prove that the result of this high-level algorithm gives us the matrix \hat{A} , which is essentially the same as the matrix resulting from a step of the QR -method onto the matrix A . In certain sense only the first column of the matrix Q , determining the orthogonal similarity transformation is essential for an essentially unique result.

Applying the technique mentioned above can lead to a fast implementation for the rational QR -method, since only the first column of the matrix Q is needed. It will be shown in a forthcoming manuscript that the vector \mathbf{v} determining the rational QR -step for semiseparable matrices is even more simple than the vector \mathbf{v} from the QR -method applied onto the semiseparable matrix.

Since an implicit technique also involves determining the similarity transformation W_2 , and this is dependent on the structure of the matrix, we will not design implicit techniques here. In the forthcoming section we will, however, show some examples of matrices for which the algorithm leads to a low complexity algorithm, with an advantageous convergence behavior.

5 Examples

In this section we will briefly present some classes of matrices suitable for applying the new method. With suitable we mean that the new iteration is as expensive or cheaper than the standard QR -iteration. A complete implementation of the new method, with all details would be too extended to be included in this manuscript, and it would lead us too far from the initial goal of the manuscript.

We will only show that based on the specific structure of the following matrices the new iteration does not necessarily lead to a more expensive step of the new method w.r.t. the traditional QR -method.

5.1 Semiseparable matrices

Semiseparable matrices are suitable for the new QR -iterate with $\kappa_i = 0$. With suitable we mean that implementing the new QR -iteration will be cheaper w.r.t. the standard QR -iteration. With cheaper, we did not take into consideration the extra convergence behavior, only the computational complexity of 1 step of the new method.

Let us define the class of higher order semiseparable matrices.

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

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

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

More precisely this means that a matrix has all subblocks taken out of the matrix below a specific superdiagonal satisfying a certain rank constraint (a similar remark holds for the upper triangular part). For example the elements \boxtimes in the following matrix, make up the lower triangular semiseparable part of a $\{1\}$ -semiseparable matrix. This means that all submatrices taken out of the part of the matrix marked with \boxtimes are of rank at most one:

$$S = (s_{ij})_{ij} = \begin{bmatrix} \boxtimes & \times & \times & \times & \times \\ \boxtimes & \boxtimes & \times & \times & \times \\ \boxtimes & \boxtimes & \boxtimes & \times & \times \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \times \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix}.$$

In some sense (neglecting thereby singular matrices) one can consider $\{p, q\}$ -semiseparable matrices as being the inverses of $\{p, q\}$ -band matrices.

In a straightforward way one can compute the RQ -factorization of such matrices (similarly as computing the QR -factorization, but now based on Givens transformations acting on the right, thereby changing the columns of the matrix). Let us first illustrate this for a $\{1\}$ -semiseparable matrix $S = (s_{ij})_{ij}$. First a Givens transformation acting on column 1 and 2 annihilating all elements in the first column below s_{11} is applied. This transformation is followed by a Givens transformation acting on column 2 and 3, annihilating all elements in the second column below s_{22} . This procedure can easily be continued. Thus applying $n - 1$ Givens transformations on the right creates an upper triangular matrix R in case of a $\{1\}$ -semiseparable matrix. In general, one needs $p(n - 1)$ Givens transformations for making a $\{p\}$ -semiseparable matrix upper triangular.

In general we obtain the following relation for a $\{p\}$ -semiseparable matrix S (see, e.g., [18, 19, 20]):

$$S = RQ,$$

with Q a product of $p(n - 1)$ Givens transformations⁷. This leads to

$$\begin{aligned} S - \sigma I &= (SQ^H - \sigma Q^H)Q \\ &= (R - \sigma Q^H)Q. \end{aligned}$$

Due to the specific structure of the matrix Q^H , we can see that the matrix σQ^H will be a unitary generalized Hessenberg matrix, having p subdiagonals different from zero.

Hence we obtain

$$\begin{aligned} S - \sigma I &= HQ \\ &= \hat{Q}\hat{R}Q. \end{aligned} \tag{11}$$

More precisely the matrix \hat{Q} can be seen as the product of p sequences of Givens transformations each annihilating one subdiagonal of the matrix H . To conclude, we need p sequences of Givens transformations to be performed onto the matrix S in order to perform a step of the new iteration yielding the same convergence as the standard QR -method due to the shift σ_i , plus an extra non shifted subspace iteration taking place for all k at the same time.

Let us compare the above results with the standard QR -method for these matrices. For computing the QR -factorization of the matrix $S - \sigma I$, $2p(n - 1)$ Givens transformations are needed. Moreover, this QR -factorization does not have the extra non shifted subspace iteration, only convergence based on the single shift strategy is present.

As a result we obtain that applying the new unitary similarity transformation onto the semiseparable matrix S yields a much better computational complexity as only half the number of Givens transformations, w.r.t. the standard QR -iteration are involved. Moreover, also an enhanced convergence behavior can be observed. This means that cheaper QR -iterates are obtained, with a faster convergence behavior.

As mentioned before, the details on how to perform these operations effectively and in an implicit way will not be described here, they would lead us to far from the general purpose of the manuscript. In a forthcoming manuscript (see [21]), we will discuss the issues such as:

- An effective computation of the RQ -factorization. Based on the Givens-weight representation of semiseparable matrices we can even omit this computation.
- An implicit implementation of the involved method.
- Detailed complexity counts and comparisons with the standard QR -method for semiseparable matrices.

We can already say that the resulting method is more accurate than the existing QR -method and moreover the QR -method is 50% more expensive than this new iteration.

⁷ We remark that other types of factorizations based on Gauss transforms can also be computed in a similar fashion.

5.2 Unitary Hessenberg matrices

In this section we will illustrate the applicability of our approach to unitary Hessenberg matrices. We will illustrate that the new iteration (with $\kappa = 0$) can be computed at the same cost as the standard QR -factorization. Unfortunately, we will see that the convergence is equally fast in both methods.

The Schur representation for unitary Hessenberg matrices is widespread and presents an easy manner of dealing with these matrices (see [22]).

Suppose a unitary Hessenberg matrix $H \in \mathbb{C}^{n \times n}$ is given. An easy calculation reveals that this matrix can be written as the product of $n - 1$ Givens transformations (this is the so-called Schur representation):

$$H = G_1 G_2 \dots G_{n-1},$$

where G_i is a Givens transformation acting on columns (rows) i and $i + 1$.

For a lower unitary Hessenberg matrix Z we have a similar factorization as

$$H = G_{n-1} G_{n-2} \dots G_1,$$

where again G_i is the Givens transformation acting on columns (rows) i and $i + 1$.

The Givens transformation G_i embeds a 2×2 Givens transformation of the following form:

$$\begin{bmatrix} c_i & -\bar{s}_i \\ s_i & \bar{c}_i \end{bmatrix}.$$

Based on this factorization, one can see that in case of a unitary Hessenberg matrix the upper triangular part is of semiseparable form and in case of a lower triangular unitary Hessenberg matrix, the lower triangular part is of semiseparable form. Let us briefly recapitulate the traditional QR -algorithm for unitary Hessenberg matrices in order to compare it with the new QR -method for unitary Hessenberg matrices.

5.2.1 The traditional QR -algorithm For applying the traditional QR -algorithm, we assume we are working with an upper Hessenberg matrix H , represented by its Schur factorization. Let us recapitulate the traditional ‘implicit’ QR -algorithm in a nutshell.

Suppose we want to apply a single shift QR -step onto the matrix H , where the shift is denoted by κ . Hence we need to compute the QR -factorization of the matrix $H - \kappa I = QR$. Due to the implicit Q -theorem, we know that it is sufficient to know the first column of the unitary matrix Q , in case H is irreducible, to uniquely determining the unitary similarity transformation $Q^H H Q$ restoring the structure. This corresponds to determining the initial unitary transformation \tilde{Q} , such that $\tilde{Q}^H (H - \kappa I) \mathbf{e}_1 = \beta \mathbf{e}_1$.

Hence, in order to apply an implicit step of the QR -method onto the matrix $H = G_1 G_2 \dots G_{n-1}$, we need to determine \tilde{Q} such that

$$\tilde{Q}^H [c_1 - \kappa, s_1, 0, \dots, 0]^T = \beta \mathbf{e}_1. \quad (12)$$

It is clear that this can be accomplished by a simple Givens transformation. Next, one executes the following unitary similarity transformation:

$$\tilde{Q}^H H \tilde{Q}, \quad (13)$$

followed by structure restoring unitary similarity transformations performed on the matrix in Equation (13), not interfering with the first column of the matrix \tilde{Q} . These structure restoring unitary transformations also consist of a sequence of $n - 2$ Givens transformations. Based on the so-called shift through lemma for reordering Givens transformations (see, e.g., [23]), one can perform these remaining $n - 2$ Givens transformations, restoring the Hessenberg structure and returning the new Schur parameterization.

One can conclude that in order to start with the implicit chasing procedure, one needs to compute a first Givens transformation, satisfying Equation (12) followed by a sequence of $n - 2$ Givens transformations.

5.2.2 The new QR-algorithm To illustrate the applicability of the new QR-algorithm onto unitary Hessenberg matrices, we assume we are working with a lower unitary Hessenberg matrix $Z = G_{n-1}G_{n-2}\dots G_1$. There is no loss of generality in this assumption, as one can operate also on H^H .

To start with the new approach (we will only consider the case $\kappa = 0$), one needs to compute the QR-factorization of the matrix Z . Fortunately the matrix is already factored in this form:

$$Z = RQ = R(G_{n-1}G_{n-2}\dots G_1) = I(G_{n-1}G_{n-2}\dots G_1)$$

in which R is simply the identity matrix.

To proceed with the new method we obtain, for a suitable chosen shift σ :

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

The unitary transformation \hat{Q} determines the new unitary similarity transformation to be performed onto the matrix Z . Similarly as in the traditional case the unitary similarity transformation is uniquely determined by the first column of the matrix \hat{Q} .

The first column is in turn determined by the unitary transformation \tilde{Q} satisfying $\tilde{Q}^H(I - \sigma Q^H)\mathbf{e}_1 = \beta\mathbf{e}_1$. Again this transformation \tilde{Q} is a Givens transformation, chosen such that

$$\tilde{Q}^H[1 - \sigma\bar{c}_1, -s, 0, \dots, 0] = \beta\mathbf{e}_1. \quad (14)$$

This transformation needs to be applied onto the lower unitary Hessenberg matrix Z , followed by a structure restoring chasing which will also consist of $n - 2$ Givens transformations. Again also the shift through lemma can be used to obtain the new Schur parameterization.

5.2.3 Convergence rates As both methods are equally expensive, one would prefer method 2, to obtain the enhanced convergence behavior. Due to the fact that $\kappa = 0$, this convergence behavior induces an extra non shifted subspace iteration.

Unfortunately it seems that both methods converge equally fast, hence there is no trace of the extra convergence behavior. This is easily explained by the properties of the eigenvalues of unitary matrix. All eigenvalues have modulo 1, as they lie on the unit circle. Due to the fact that the extra convergence behavior is determined by ratio of the moduli between two successive eigenvalues, there will be no extra convergence behavior. Both methods are each others duals.

6 Numerical experiments

In the following numerical experiments we will illustrate the convergence rate of our method, w.r.t. the earlier derived bounds. It will be shown that the new iteration can create convergence behavior both at the top as at the bottom. The numerical experiments are based on an explicit version of the method, in which the full orthogonal matrix Q is constructed. Only results concerning convergence speed and the number of iterations are included. No complexity comparisons are made, as the implementations need to be tuned to the matrix working with.

6.1 Convergence rates

In the following figures we will plot the effective norm of the subblock, and the predicted upper bound for the convergence of this subblock, based on Theorem 1. The examples are based on arbitrary matrices. The computational complexity of the presented method on these matrices is not taken into consideration, only the convergence w.r.t. the choice of the shifts is given.

Firstly we will illustrate the convergence of the method for $\sigma \neq 0$ and $\kappa \neq 0$. We take σ to be the shift equal to lower right element of the matrix A , κ is taken equal to the upper left element.⁸

⁸ Similarly one can choose the Wilkinson shift. Here we used the Rayleigh shift.

Theorem 1, indicates that convergence will occur both to the lower right element, and the upper left element. This is depicted in the following figures, in which an arbitrary symmetric matrix A was generated. The left figure shows the convergence towards the upper left element, the right figure shows the convergence towards the lower right element. The x -axis depicts the number of QR -iterations performed, and the y -axis depicts the size of the off-diagonal blocks and the estimates. The cross-over between the two lines is due to the finite precision arithmetic. It is not shown in the figures, but in the middle of the matrix, no significant convergence occurs.

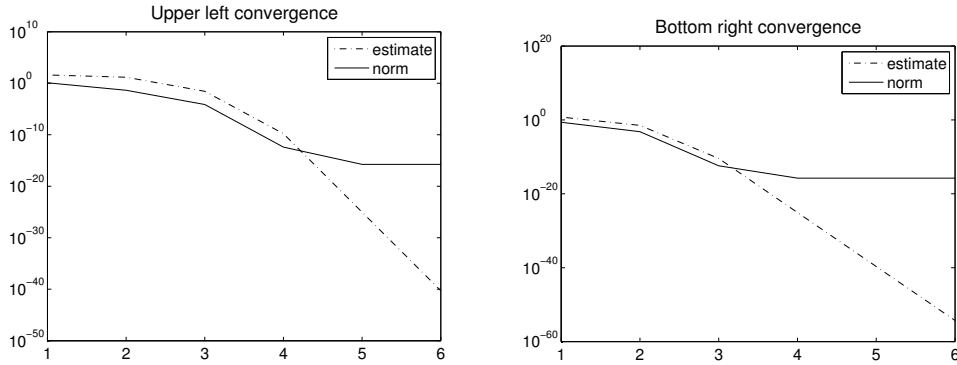


Figure 1. Convergence with two shifts

In the following three figures we illustrate convergence for an arbitrary symmetric matrix, in which κ was chosen equal to zero. As mentioned before, this creates a standard (non shifted) subspace iteration inside the complete matrix. Therefore, all off-diagonal blocks will gradually tend to become zero. The convergence rate is determined by the ratios between the moduli of the different eigenvalues. We choose a matrix having the following eigenvalues $[1 : 10, 31 : 40]$. The most left figure shows the convergence at the top left element, we see that it decreases linearly, just like the block corresponding with the gap shown in the middle. Due to the higher ratio in the middle block we see here a faster convergence rate. The right figure illustrates the standard fast convergence rate related to the lower right element. This convergence rate is comparable with the convergence rate of the standard QR -method.

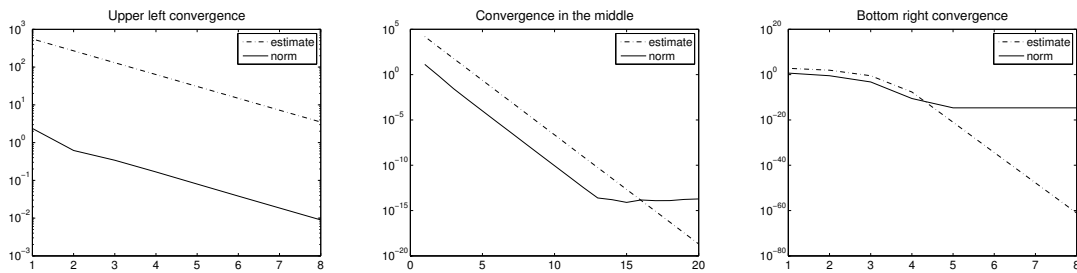


Figure 2. Convergence with two shifts

6.2 Number of iteration steps

In the following figures, we will show for few classes of matrices the number of iteration steps performed by the new method, w.r.t. the number of iteration steps performed by the traditional QR -method. Remark, however, that the number of steps does not necessarily indicate a more effective algorithm, as the complexity of each step independently needs to be taken into consideration. For

some classes of matrices we know however, that the complexity of one new step can be less than the complexity a traditional step.

In the following two figures the total and average number of iterations is depicted for computing the whole spectrum based on the two methods. Arbitrary symmetric matrices are created and deflation is applied, when the subblock norm is less than 10^{-9} . The bottom axis depicts the sizes of the matrices, whereas the y-axis depicts the number of steps. The left figure shows the total number of iterations, whereas the right figure counts only the average number of iterations. Three different experiments are shown in each graph counting the number of steps in case of the traditional QR -algorithm, the new QR -algorithm with both shifts κ and σ different from zero and the new QR -algorithm with $\kappa = 0$. Results for the case $\sigma = 0$ and $\kappa \neq 0$ are not included as the results are similar to the last experiment mentioned. The second experiment is exactly the same

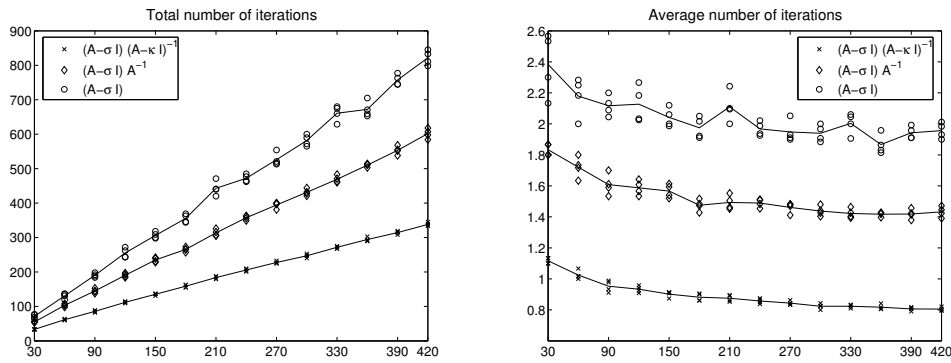


Figure 3. Global number of iterations

but for nonsymmetric matrices. We see that the differences are less extreme, in case $\kappa = 0$ but still significant.

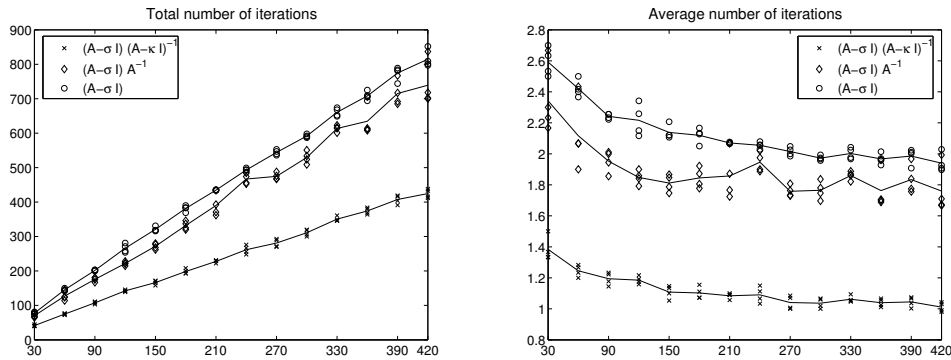


Figure 4. Global number of iterations

6.3 Some special examples

In this subsection we will apply the new method onto some specific generated examples, to illustrate the convergence behavior in case both shifts do not work separately, but interact with each other. See also Example 1. This means that for a matrix A convergence is forced by the shift σ_i to the block $A(k+1:n, k+1:n)$, and the shift κ_i forces convergence to the block $A(1:k, 1:k)$. Hence the convergence behavior of both shifts coincides. This behavior can create very fast con-

vergence, but it can also help convergence in difficult cases, where the traditional QR -algorithm might fail.

For the first case we generated a random nonsymmetric matrix of size 100 having one eigenvalue of size 10 and 99 eigenvalues close to 1. These eigenvalues were constructed by adding a random perturbation of the size $\approx 10^{-1}$ to 1. This matrix is then processed by the new iteration and the traditional QR -iteration.

For the first plotted figures (Figure 5) we assumed no initial knowledge on the eigenvalues and just ran both methods. The traditional QR -method (shown in the right figure) uses the Rayleigh shift, and the new method (shown left) uses both bottom right and upper left element as a shift. In some cases both methods converge equally accurate but also in some cases we obtain the following results. The figures clearly illustrate that an initial badly chosen shift (e.g., close to 1), creates poor convergence for the QR -method, whereas the new method is less sensitive due both cooperating iterations at the top and bottom.

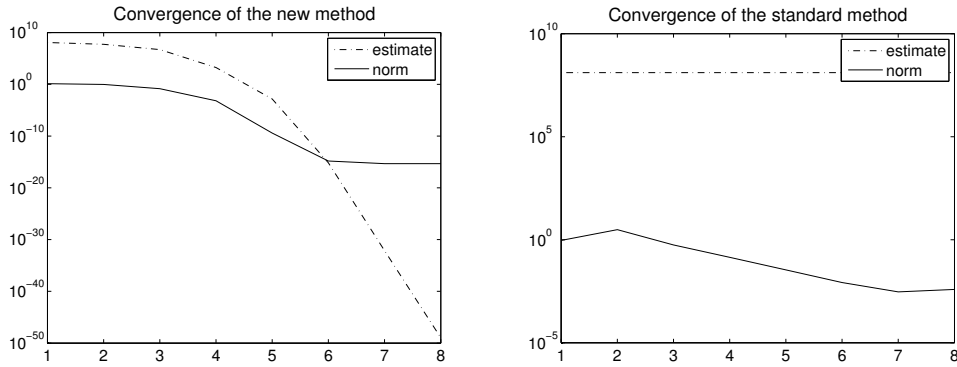


Figure 5. Convergence comparison

In the second figure (Figure 6), we repeat the same experiment, but we plot now an example in which both methods converge. It can be seen in the figures that the new approach converges almost twice as fast.

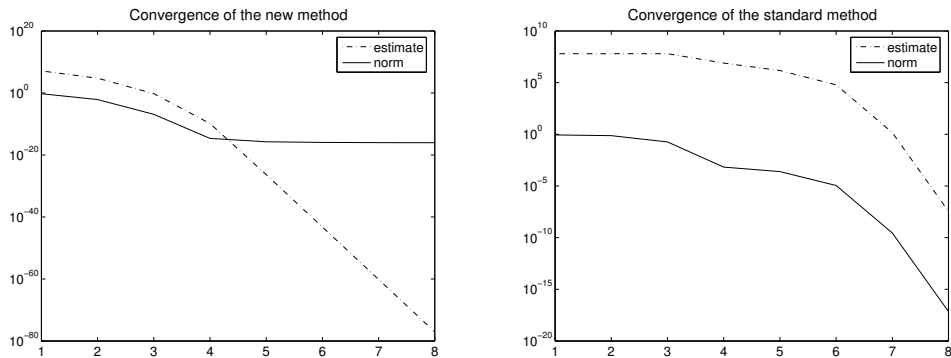


Figure 6. Convergence comparison

In the next example the difference is more clearly visible. We considered a matrix having two large clusters of 50 eigenvalues, 50 eigenvalues of size 1 randomly disturbed by elements of size 10^{-1} and 50 eigenvalues of size 10 plus a random element of size 1, in both cases we wanted to see if the methods revealed both clusters. The figures show that the new method has a higher convergence rate.

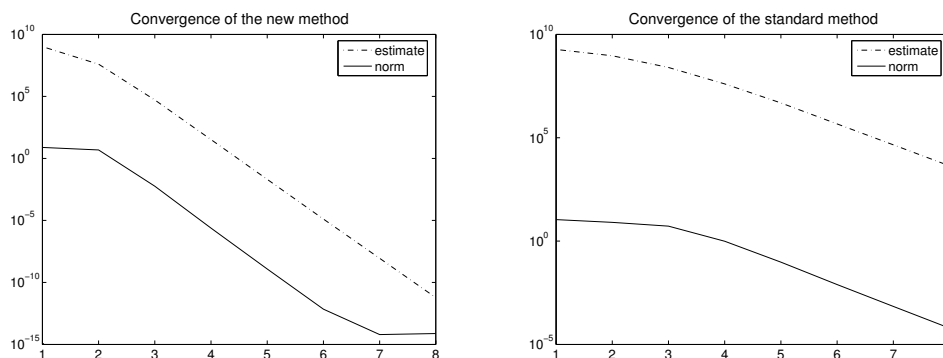


Figure 7. Convergence comparison

Of course there are many more similar experiments, but these illustrate the properties inherited by the new convergence behavior.

7 Conclusions & Future work

In this manuscript a new type of QR -iteration was presented. The approach corresponded to a kind of rational QR -iteration. The convergence properties of the proposed method were discussed. Some examples were presented, showing that this approach does not necessarily lead to an increase in complexity.

This manuscript opens several possibilities for further investigation. First there is the extension of this ‘basic’ rational step towards a more general ‘easy computable’ multishift setting allowing to compute efficiently the QR -factorization of a general rational function in the matrix A .

There is also the search for other classes of matrices allowing an easy implementation of this rational QR -algorithm.

References

1. J. H. Wilkinson. *The Algebraic Eigenvalue Problem*. Numerical Mathematics and Scientific Computation. Oxford University Press, Great Clarendon Street, Oxford OX2 6DP, 1999.
2. B. N. Parlett. *The Symmetric Eigenvalue Problem*, volume 20 of *Classics in Applied Mathematics*. SIAM, Philadelphia, 1998.
3. A. A. Dubrulle and G. H. Golub. A multishift QR iteration without computation of the shifts. *Numerical Algorithms*, 7(2-4):173–181, 1994.
4. G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, Maryland, third edition, 1996.
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. 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.
7. 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.
8. 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.
9. 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.
10. A. Ruhe. Rational Krylov sequence methods for eigenvalue computation. *Linear Algebra and its Applications*, 58:391–405, 1984.
11. A. Ruhe. Rational krylov algorithms for nonsymmetric eigenvalue problems, II: Matrix pairs. *Linear Algebra and its Applications*, 197/198:283–296, 1994.
12. A. Ruhe. The rational Krylov algorithm for nonsymmetric eigenvalue problems. III: Complex shifts for real matrices. *BIT*, 34:165–176, 1994.

13. 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.
14. 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.
15. 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.
16. D. S. Watkins. QR -like algorithms — An overview of convergence theory and practice. *Lectures in Applied Mathematics*, 32:879–893, 1996.
17. D. S. Watkins. Understanding the QR algorithm. *SIAM Review*, 24(4):427–440, 1982.
18. R. Vandebril, M. Van Barel, and N. Mastronardi. *Matrix Computations and Semiseparable Matrices, Volume I: System Solving*. Johns Hopkins University Press, 2007.
19. 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.
20. P. Dewilde and A.-J. van der Veen. *Time-varying systems and computations*. Kluwer Academic Publishers, Boston, June 1998.
21. R. Vandebril, M. Van Barel, and N. Mastronardi. A new iteration for computing the eigenvalues of semiseparable (plus diagonal) matrices. Technical report, Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3000 Leuven (Heverlee), Belgium, October 2007.
22. W. B. Gragg. The QR algorithm for unitary Hessenberg matrices. *Journal of Computational and Applied Mathematics*, 16:1–8, 1986.
23. S. Delvaux and M. Van Barel. Unitary rank structured matrices. *Journal of Computational and Applied Mathematics*, 2007. Accepted.