An algorithm for the complete solution of the quartic eigenvalue problem

Zlatko Drmač and Ivana Šain Glibić

University of Zagreb, Faculty of Science, Department of Mathematics

Abstract

Quartic eigenvalue problem \((\lambda^4 A + \lambda^3 B + \lambda^2 C + \lambda D + E) x = 0\) naturally arises e.g. when solving the Orr–Sommerfeld equation in the analysis of the stability of the Poiseuille flow, in theoretical analysis and experimental design of locally resonant phononic plates, modeling a robot with electric motors in the joints, calibration of catadioptric vision system, or e.g. computation of the guided and leaky modes of a planar waveguide. This paper proposes a new numerical method for the full solution (all eigenvalues and all left and right eigenvectors) that is based on quadratification, i.e. reduction of the quartic problem to a spectraly equivalent quadratic eigenvalue problem, and on a careful preprocessing to idetinfy and deflate zero and infinite eigenvalues before the linearized quadratification is forwarded to the QZ algorithm. Numerical examples and backward error analysis confirm that the proposed algorithm is superior to the available methods.

1 Introduction and preliminaries

We propose a new method for numerical solution of the quartic eigenvalue problem

\[
(\lambda^4 A + \lambda^3 B + \lambda^2 C + \lambda D + E) x = 0,
\]

where the coefficient matrices \(A, B, C, D, E \in \mathbb{C}^{n \times n}\) are assumed general, with no particular structure (such as symmetry, sparsity). We are interested in the full solution, i.e. computation of all eigenvalues with the corresponding (left and/or right) eigenvectors, and our ultimate goal is to provide a robust mathematical software that can be used in ever increasing number of applications in applied sciences and engineering.

The quartic eigenvalue problem naturally arises in solving the Orr–Sommerfeld equation which appears in the hydrodynamic analysis of the stability of the Poiseuille flow by elimi\-nating the pressure from the linearized Navier-Stokes equation. Other applications include e.g. theoretical analysis and experimental design of locally resonant phononic plates [19], finite element analysis of two dimensional phononic crystals [16], modeling a robot with electric motors in the joints [12], computing deformation modes of thin-walled structures [17], or e.g. computation of the guided and leaky modes of a planar waveguide [14]. In these examples the matrix eigenvalue problem is the result of discretization of differential operators and thus (depending on the discretization method) the coefficient matrices are sparse and usually only some eigenvalues are needed – those may be prescribed by specifying e.g. a region of interest in the complex plane. In such cases, methods for large sparse problems such as e.g. NLFEAST [9].
[13], [20], [4] will find a subspace that contains eigenvectors of interest, and then Rayleigh–Ritz extraction uses the projected problem in which the Rayleigh quotients are medium size dense matrices. Reliable solution of the projected problem is important both for the convergence of the iterations towards the wanted part of the spectrum (e.g. for robust implementation of locking and purging) and for the accuracy of the computed solution.

These examples illustrate the wide spectrum of important applications of the quartic eigenvalue problem, and justify, even demand, development of methods specialized for (1.1). Yet, to the best of our knowledge, there is no published custom-built solver with a supporting analysis that would provide certain level of confidence/guarantee that is comparable e.g. to the currently available solvers of the quadratic eigenvalue problem such as \texttt{polyeig}. Instead, (1.1) is usually numerically solved by a standard linearization and deployment of the solvers such as \texttt{polyeig} in Matlab. On the other hand, numerical difficulties in solving nonlinear eigenvalue problems become nontrivial even in the simplest case of the polynomial quadratic problem, which is at the core of the theory and applications of mechanical systems. More carefully designed custom-made algorithm often proves much better than a generic solver – an excellent example is the quadratic eigenvalue problem, where the algorithms \texttt{quadeig} [10] and \texttt{KVADeig} [7] outperform \texttt{polyeig}, in particular when the spectrum contains multiple infinite eigenvalues.

1.1 A quandary about the infinite eigenvalues

The presence of infinite eigenvalues, indicated by the rank deficiency of $A$, may cause difficulties in the QZ algorithm, which is usually deployed for solving the linearized problem; infinite eigenvalues may not be identified correctly, they may have negative impact on the accuracy of the computed finite eigenvalues. It is then advantageous to remove infinite eigenvalues by a deflation and proceed with a problem of smaller dimension, with only finite eigenvalues. This framework, introduced in [10], proved much better than direct solution of the linearized problem.

In some cases, certain number of infinite eigenvalues of (1.1) can be identified and removed already during the problem formulation. An illustrative example is given in the eigenvalue problem for the channel and Blasius boundary layer in semi-infinite domain [5]. The Orr–Sommerfel differential equation is discretized using the Chebyshev collocation matrix method, and the boundary conditions are imposed in $E$; $A$, $B$, $C$, $D$ have the corresponding last four rows equal to zero. In the case of linearly independent boundary conditions, by a clever column permutation, four infinite eigenvalues can be separated and deflated, see [5] for technical details.

The structure of the infinite eigenvalue (the number and the dimensions of blocks in the Kronecker Canonical Form (KCF)) cannot be inferred by only inspecting the rank of $A$. Rank deficiency in $A$ reveals only certain number of infinite eigenvalues and further steps are necessary to either confirm that there are no more infinite eigenvalues or to reveal more blocks in the KCF carrying $\lambda = \infty$. These steps involve decisions on the numerical ranks of some intermediate matrices that have been contaminated by the roundoff noise from the previous steps. If the data is not well scaled and if the computation cannot be interpreted as backward stable in terms of the original coefficients, then there may be quite a few spurious eigenvalues with large absolute values. The backward stability of the beginning steps that carry the critical responsibility of removing infinite eigenvalues must be as much as possible in terms of the initial coefficient matrices, and it has to be as much as possible structured, e.g. column-wise small (backward error in each column small relative to that column’s norm) instead of only small in matrix norm.
1.2 Quadratification

We build upon the algebraic tool of quadratification introduced in [6], which allows us to reduce the quartic problem (1.1) to an equivalent quadratic eigenvalue problem, and we turn it into a robust numerical algorithm by extending the framework introduced in [10] and recently revisited and upgraded in [7].

Let \( P(\lambda) = \sum_{\ell=0}^{k} \lambda^{\ell} A_{\ell} \in \mathbb{C}^{n \times n} \) be a matrix polynomial of degree \( k \geq 1 \). A pencil \( L(\lambda) = A - \lambda B \in \mathbb{C}^{kn \times kn} \) is a linearization of \( P(\lambda) \) if there exist unimodular matrix polynomials (i.e. with constant nonzero determinant) \( E(\lambda), F(\lambda) \) such that

\[
E(\lambda)L(\lambda)F(\lambda) = \begin{pmatrix} P(\lambda) & 0 \\ 0 & \mathbb{I}_{(k-1)n} \end{pmatrix}, \quad \text{i.e.} \quad L(\lambda) \sim \begin{pmatrix} P(\lambda) & 0 \\ 0 & \mathbb{I}_{(k-1)n} \end{pmatrix}. \tag{1.2}
\]

In other words, \( L(\lambda) \) is a linearization of \( P(\lambda) \) if \( L(\lambda) \) and \( P(\lambda) \oplus \mathbb{I}_{(k-1)n} \) are equivalent.

Quadratification is an extension of the idea of linearization – it means establishing an equivalence between \( P(\lambda) \) and a quadratic matrix polynomial. For more general concept of \( \ell \)-ification we refer to [6].

In order to introduce precise definition for the quadratification, and for the reader’s convenience, we recall the two basic definitions from [6].

**Definition 1.1.** Let \( P \) and \( Q \) be two matrix polynomials of degrees \( k \) and \( h \), respectively, not necessarily of the same size, and let \( \text{rev} \; P, \text{rev} \; Q \) be the corresponding reversed polynomials. Then:

(i) \( P \) and \( Q \) are extended unimodularly equivalent, denoted by \( P \sim Q \), if for some \( r, s \geq 0 \) we have \( \text{diag}(P, \mathbb{I}_r) \sim \text{diag}(Q, \mathbb{I}_s) \).

(ii) \( P \) and \( Q \) are spectrally equivalent, denoted by \( P \simeq Q \), if \( P \sim Q \) and \( \text{rev} \; P \sim \text{rev} \; Q \).

**Definition 1.2.** Let \( P(\lambda) \) be a matrix polynomial of degree \( k \).

(i) A matrix pencil \( L(\lambda) \) is a linearization of \( P(\lambda) \) if \( L(\lambda) \sim P(\lambda) \). It is strong linearization if \( L(\lambda) \asymp P(\lambda) \).

(ii) A quadratic matrix polynomial \( Q(\lambda) \) is said to be a quadratification of \( P(\lambda) \) if \( Q(\lambda) \sim P(\lambda) \). A quadratification is said to be strong if, in addition, \( \text{rev} \; Q(\lambda) \sim \text{rev} \; P(\lambda) \). Equivalently, a pencil \( Q(\lambda) \) is a strong quadratification for \( P(\lambda) \) if \( Q(\lambda) \asymp P(\lambda) \).

We are interested only in strong quadratifications, because spectral equivalence is necessary for our computational task. The companion forms of grade 2, described next, meet that condition.

1.2.1 Companion form of grade 2

Analogously to the linearization by companion form, the first and the second companion form of grade 2 are introduced in [6] as follows. First, define matrix polynomials

\[
B_1(\lambda) = \lambda^2 C + \lambda D + E, \quad B_2(\lambda) = \lambda^2 A + \lambda B. \tag{1.3}
\]

The first companion form of grade 2 is then defined as

\[
C_1^2(\lambda) = \begin{pmatrix} B_2(\lambda) & B_1(\lambda) \\ -\mathbb{I}_n & \lambda^2 \mathbb{I}_n \end{pmatrix} = \begin{pmatrix} \lambda^2 A + \lambda B & \lambda^2 C + \lambda D + E \\ -\mathbb{I}_n & \lambda^2 \mathbb{I}_n \end{pmatrix} = \lambda^2 \begin{pmatrix} A & C \\ 0 & \mathbb{I}_n \end{pmatrix} + \lambda \begin{pmatrix} B & D \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & E \\ -\mathbb{I}_n & 0 \end{pmatrix}.
\]
We will use the second companion form of grade 2 because its structure is compatible with the deflation framework of \texttt{quadeig} and \texttt{KVADeig}.

\[
C^2_2(\lambda) = \begin{pmatrix} B_2(\lambda) & -\mathbb{I}_n \\ B_1(\lambda) & \lambda^2 \mathbb{I}_n \end{pmatrix} = \begin{pmatrix} \lambda^2 A + \lambda B & -\mathbb{I}_n \\ \lambda^2 C + \lambda D + E & \lambda^2 \mathbb{I}_n \end{pmatrix} = \lambda^2 \begin{pmatrix} A & 0 \\ C & D \end{pmatrix} + \lambda \begin{pmatrix} B & 0 \\ E & 0 \end{pmatrix} + \begin{pmatrix} 0 & -\mathbb{I}_n \\ 0 & 0 \end{pmatrix} = \lambda^2 \mathbb{M} + \lambda \mathbb{C} + \mathbb{K}. \tag{1.4}
\]

Both $C^2_1(\lambda)$ and $C^2_2(\lambda)$ are strong in the sense of Definition 1.2, see [Theorem 5.3, Theorem 5.4] [6].

### 1.3 Our contribution and outline of the paper

The starting point of the development of the proposed algorithm is the quadratic pencil (1.4). It can be further linearized using e.g. the second companion form. In that case, the final matrix pencil of size $4n \times 4n$, that represents a linearization of the quartic problem 1.1 reads

\[
A - \lambda B = \begin{pmatrix} C & -\mathbb{I}_{2n} \\ \mathbb{K} & 0_{2n} \end{pmatrix} - \lambda \begin{pmatrix} -\mathbb{M} & 0_{2n} \\ 0_{2n} & -\mathbb{I}_{2n} \end{pmatrix} = \begin{pmatrix} B & 0_n \\ D & 0_n \\ 0_n & -\mathbb{I}_n \end{pmatrix} \begin{pmatrix} -\mathbb{I}_n & 0_n \\ 0_n & -\mathbb{I}_n \end{pmatrix} - \lambda \begin{pmatrix} 0_n & 0_n \\ 0_n & -\mathbb{I}_n \end{pmatrix}. \tag{1.5}
\]

Now, we can follow the structure of \texttt{quadeig}/\texttt{KVADeig}, attempting to deflate the infinite eigenvalues of $\lambda^2 \mathbb{M} + \lambda \mathbb{C} + \mathbb{K}$. Even if that is expected to perform better than a straightforward companion type linearization followed by \texttt{polyeig}, it is not the best one can do – the goal is to implement the beginning critical steps with small (hopefully to some extent structured) backward error in the original coefficient matrices $A$, $B$, $C$, $D$, $E$. Therefore, on the global level, we follow the strategy of \texttt{KVADeig} [7, Algorithm 3.1] to bring (1.5) to an upper triangular Kronecker Canonical Form (KCF), but the elementary steps are rewritten in terms of the original matrices whenever feasible.

The new algorithm, designated as \texttt{KVARTeig}, is described in detail in §2. For the sake of completeness, the formulas for recovering the eigenvectors of (1.1) from those of (1.5) are provided in §4. In §1 we provide details backward error analysis of the first two steps that are critical for removing zero and infinite eigenvalues. We clearly identify moments in the algorithm where scaling of the data plays the key role in keeping the backward error in the initial data small. The numerical experiments, presented in §5, show the power of our method in comparison to the MATLAB’s function for the computation of the polynomial eigenvalue problem, \texttt{polyeig}, and to the \texttt{quadeig} and \texttt{KVADeig} (applied to the quadratification) as well. Although \texttt{polyeig} in many cases performs well (mostly thanks to the QZ method), it completely fails to find the solution of the quartic eigenvalue problem obtained from Orr–Sommerfeld equation of the dimension $n = 1000$, whereas \texttt{KVARTeig} computes the solution with acceptable backward error.

The material of this paper should be considered as the second part of [7], and numerical results in §5 once more illustrate the power of the improvements introduced in \texttt{KVADeig}.

### 2 The KVARTeig algorithm

We now describe the main ideas of the proposed procedure for deflation of infinite and/or zero eigenvalues. Our plan is to adapt the deflation scheme from \texttt{quadeig}/\texttt{KVADeig} to the quadratic
problem (1.4), but keeping in mind the structure (1.5). We stress that applying a robust quadratic solver to the quadratification blindly (i.e. by ignoring the origin of the quadratic problem) is not satisfactory – this is illustrated in [5].

2.1 Scaling

The procedure for computing the eigenvalues and the corresponding right and left eigenvectors for polynomial eigenvalue problem consists of three main steps: linearization, i.e. definition of the equivalent generalized eigenvalue problem, using the QZ algorithm to find the eigenpairs for the linearization, and the reconstruction of eigenpairs for the original problem. However, it can happen that the eigenpair obtained by this procedure has high norm-wise backward error, although the norm-wise backward error for the eigenpair of the corresponding linearization is low. This phenomenon is further analyzed in [11], and it is proven that this kind of variation in the norm-wise backward errors is due to the fact that the norms of the coefficient matrices of the original problem are not equilibrated.

The problem can be alleviated by parameter scaling as follows. Write \( \lambda = \gamma \nu \), where \( \gamma > 0 \) is parameter to be determined, and define the scaled quartic polynomial as

\[
\theta E + \nu(\gamma \theta D) + \nu^2(\gamma^2 \theta C) + \nu^3(\gamma^3 \theta B) + \nu^4(\gamma^4 \theta A) \equiv \hat{E} + \nu \hat{D} + \nu^2 \hat{C} + \nu^3 \hat{B} + \nu^4 \hat{A},
\]

with another free parameter \( \theta > 0 \). The parameters \( \gamma \) and \( \theta \) are defined so that the new coefficient matrices do not vary much in norm. This can be done by adapting the Fan, Lin and Van Dooren’s scaling [8]. For \( \gamma \), we choose \( \gamma = \sqrt[4]{\frac{\| \hat{E} \|_2}{\| \hat{A} \|_2}} \), which is the optimal \( \gamma \) for minimizing the factor \( \max(1, \| \hat{A} \|_2, \| \hat{B} \|_2, \| \hat{C} \|_2, \| \hat{D} \|_2, \| \hat{E} \|_2)^2/\min(\| \hat{E} \|_2, \| \hat{A} \|_2) \) in the backward error ratio bounds [1], and for \( \delta \), we choose \( \delta = \| E \|_2 + \gamma \| D \|_2 + \gamma^2 \| C \|_2 + \gamma^3 \| B \|_2 \), as in [3].

**Remark 2.1.** In addition to parameter scaling, we can use diagonal scaling matrices \( \Delta_\ell \) and \( \Delta_r \), for scaling all coefficients from the left with \( \Delta_\ell \) and from the right with \( \Delta_r \). These scaling matrices can be computed by a simple extension of the scheme described in [3, §4.2].

2.2 Numerical rank and block-structure

Consider the matrix \( \mathbb{M} \) in the quadratification (1.4). The obvious algebraic fact \( \text{rank}(\mathbb{M}) = n + \text{rank}(A) \) becomes more important in numerical setting where we have to determine the numerical rank. Namely, applying a rank revealing decomposition (such as the SVD or the pivoted QR factorization) to \( \mathbb{M} \) would mean looking for a small perturbation \( \delta \mathbb{M} \) such that \( \mathbb{M} + \delta \mathbb{M} \) has lower rank that cannot be further reduced by a small perturbation. Such a construction does not respect the block structure of \( \mathbb{M} \), and better way is to think at this step in terms of the numerical rank with constrained perturbation. If \( J \) denotes the first \( n \) columns of \( \mathbb{I}_n \) then the allowed perturbation might be \( \delta \mathbb{M} = J \delta A J^T \) with an \( n \times n \) \( \delta A \). Similarly, the numerical rank of \( \mathbb{K} \) will be determined under the constraint that only \( \mathbb{K}(n + 1 : 2n, 1 : n) = E \) is allowed to change. For a systematic treatment of the general case using the generalized SVD, see [21].

Let \( r_A = \text{rank}(A) \), \( r_E = \text{rank}(E) \) and let

\[
A \Pi_A = Q_A R_A, \quad R_A = \begin{pmatrix} \hat{R}_A \\ 0_{n-r_A,n} \end{pmatrix}, \quad E \Pi_E = Q_E R_E, \quad R_E = \begin{pmatrix} \hat{R}_E \\ 0_{n-r_E,n} \end{pmatrix}, \quad \text{be the rank revealing QR factorizations for } A \text{ and } E.
\]
Now, we can use (2.1) to get a structure preserving rank revealing decomposition of the matrix $M$. Finally, the rank revealing factorization of $M$ is given by

$$M \Pi_M = Q_M R_M, \quad Q_M = \begin{pmatrix} 0 & Q_A \\ \frac{I_n}{n} & 0 \end{pmatrix}, \quad \Pi_M = \begin{pmatrix} 0 & P_A \\ I_n & 0 \end{pmatrix}, \quad R_M = \begin{pmatrix} -I_n & -CP_A \\ 0 & -R_A \end{pmatrix}. \quad (2.2)$$

Similarly, the rank revealing factorization of the matrix $K$ is

$$K \Pi_K = Q_K R_K, \quad Q_K = \begin{pmatrix} I_n & 0 \\ 0 & Q_E \end{pmatrix}, \quad \Pi_K = \begin{pmatrix} 0 & I_n \\ 0 & 0 \end{pmatrix}, \quad R_K = \begin{pmatrix} I_n & 0 \\ 0 & R_E \end{pmatrix}. \quad (2.3)$$

However, notice that the permutation of the column blocks only ensures that the matrix $R_K$ is upper triangular. If this structure is not important for the process, we can skip the permutation step and just make the following transformation

$$\begin{pmatrix} I_n & 0 \\ 0 & Q_E \end{pmatrix} K \begin{pmatrix} I_n & 0 \\ 0 & I_n \end{pmatrix} = \begin{pmatrix} 0 & I_n \\ R_E & 0 \end{pmatrix}. \quad (2.4)$$

**Remark 2.2.** To determine the numerical rank, we use the thresholding strategies as in [7, §2.3.1]. For a softer thresholding we look for a drop-off of two consecutive diagonal entries in the upper triangular form.

### 2.3 The decision tree of KVARTeig

Again, as in the KVADeig, there are three standard cases: both $A$ and $E$ regular; only one matrix is singular; and both $A$ and $E$ are singular. The algorithm is designed to remove zero eigenvalues; the infinities are removed by switching to the reversed pencil.

#### 2.3.1 Both matrices $A$ and $E$ regular

If both matrices $A$ and $E$ are regular, we can use the factorization (2.2) to reduce the matrix $B$ from (1.5) to upper triangular form, since this is already the first step of the QZ algorithm.

$$\begin{pmatrix} Q_M^* & 0 \\ 0 & I_{2n} \end{pmatrix} \begin{pmatrix} C & -I_{2n} \\ K & 0 \end{pmatrix} \begin{pmatrix} -M & 0 \\ 0 & -I_{2n} \end{pmatrix} \left( \begin{pmatrix} \Pi_M & 0 \\ 0 & \Pi_{2n} \end{pmatrix} - \lambda \left( \begin{pmatrix} -I_n & -CP_A \\ 0 & -R_A \end{pmatrix} \right) \right) \left( \begin{pmatrix} -I_n & -CP_A \\ 0 & -R_A \end{pmatrix} \right) = \lambda \left( \begin{pmatrix} -I_n & -CP_A \\ 0 & -R_A \end{pmatrix} \right). \quad (2.5)$$

The rest of the computation depends on the QZ algorithm. Note that the special structure of the pencil (2.5) could be exploited for designing a more efficient Hessenberg-triangular decomposition. However, we will not tackle that issue in this work.

#### 2.3.2 Only one matrix is singular

Assume first that $E$ is singular, $r_E < n$, and thus there are at least $n - r_E$ zero eigenvalues which can be deflated. If our setup is to remove only the block of zero eigenvalues that is revealed by the null space of $E$, then we can achieve that and, at the same time, transform the
matrix $\mathbb{B}$ to upper triangular form by the equivalence transformation

$$
\begin{pmatrix}
Q^*_M & 0 \\
0 & Q^*_K
\end{pmatrix}
\begin{pmatrix}
\mathbb{C} & -\mathbb{I}_{2n} \\
\mathbb{K} & 0
\end{pmatrix}
- \lambda
\begin{pmatrix}
-M & 0 \\
0 & -\mathbb{I}_n
\end{pmatrix}
\begin{pmatrix}
\Pi_M & 0 \\
0 & Q_K
\end{pmatrix}
= 
\begin{pmatrix}
DP_A & 0 & -Q_E \\
0 & -Q^*_A & 0 \\
-\mathbb{I}_n & 0 & 0
\end{pmatrix}
- \lambda
\begin{pmatrix}
-\mathbb{I}_n & -CP_A & 0 \\
0 & -R_A & 0 \\
0 & 0 & -\mathbb{I}_{2n}
\end{pmatrix},
$$

(2.6)

The $n - r_E$ zero eigenvalues are now deflated implicitly by working with the leading $(3n + r_E) \times (3n + r_E)$ sub-pencil of (2.6). If we want to check for the existence of further blocks corresponding to $\lambda = 0$, then it is convenient to use the following transformation:

$$
\begin{pmatrix}
Q^*_K & 0 \\
0 & Q^*_K
\end{pmatrix}
\begin{pmatrix}
\mathbb{C} & -\mathbb{I}_{2n} \\
\mathbb{K} & 0
\end{pmatrix}
- \lambda
\begin{pmatrix}
-M & 0 \\
0 & -\mathbb{I}_{2n}
\end{pmatrix}
\begin{pmatrix}
\Pi_{2n} & 0 \\
0 & Q_K
\end{pmatrix}
= 
\begin{pmatrix}
B & 0 \\
Q^*_E D & 0 \\
0 & -\mathbb{I}_n
\end{pmatrix}
- \lambda
\begin{pmatrix}
-A & 0 \\
-Q^*_E C & -Q^*_E \\
0 & 0
\end{pmatrix}
\begin{pmatrix}
\Pi_{2n} & 0 \\
0 & -\mathbb{I}_{2n}
\end{pmatrix},
$$

(2.7)

The deflated pencil of order $3n + r_E$ reads

$$
A_{22} - \lambda \mathbb{B}_{22} = 
\begin{pmatrix}
B & 0 \\
Q^*_E D & 0 \\
0 & -\mathbb{I}_n
\end{pmatrix}
- \lambda
\begin{pmatrix}
-A & 0 \\
-Q^*_E C & -Q^*_E \\
0 & 0
\end{pmatrix}
\begin{pmatrix}
\Pi_{n+r_E} & 0 \\
0 & -\mathbb{I}_{n+r_E}
\end{pmatrix},
$$

(2.8)

where $Q^*_{E,1} = Q^*_E(1 : r_E,:)$ and $Q^*_E = Q^*_E(r_E + 1 : n,:).$ Note that $A_{22} - \lambda \mathbb{B}_{22}$ is the block at the position $(1,1)$ of a block-upper triangular pencil (2.7); the block position $(2,2)$ corresponds to the deflated $n - r_E$ zeros. Denote the left and the right transformation matrices from (2.7) with $P_1$ and $Q_1$ respectively, and the linearization pencil with $A - \lambda \mathbb{B} = \tilde{A}_{11} - \lambda \tilde{\mathbb{B}}_{11}$. After the first deflation step we have

$$
P_1(\tilde{A}_{11} - \lambda \tilde{\mathbb{B}}_{11})Q_1 = 
\begin{pmatrix}
\tilde{A}_{22} - \lambda \tilde{\mathbb{B}}_{22} & \blacklozenge \\
0 & -\lambda \tilde{\mathbb{B}}_{11}
\end{pmatrix},
$$

(2.9)

The next step in the deflation process is to determine the rank of the matrix $\tilde{A}_{22}$. From the structure of the matrix, we conclude that the rank of $\tilde{A}_{22}$ is equal to $2n + r_E$ + "the rank of the $n \times n$ matrix \((Q^*_{E,2} D / R^*_E P_E^*)\)" , which is defined in terms of the coefficient matrices $D$ and $E$ of the original problem. So, we compute the rank revealing factorization

$$
\begin{pmatrix}
Q^*_{E,2} D \\
R^*_E P_E^*
\end{pmatrix}
\Pi_{A_{22}} = Q_{A_{22}} R_{A_{22}}.
$$

(2.10)

If (2.10) is of full rank $n$, then $A_{22}$ is regular, there are no more zeros in the spectrum, and the single deflation step is done by removing the trailing $n - r_E$ rows and columns in (2.6). If, on

\footnote{See [7] §5.2 for more details.}
the other hand, (2.10) is rank deficient with rank\((R_{A_{22}}) = r_2 < n\), the corresponding number of \(n - r_2\) zero eigenvalues can be deflated. To that end, note that \(R_{A_{22}} = \left(\begin{array}{c} R_{A_{22}} \\ 0_{n-r_2,n} \end{array}\right)\) and transform the pencil (2.8) to get zero rows at the bottom of \(A_{22}\). This is done by the permutation \(\pi = (1 : n + r_E, \ 2n + 1 : 3n, \ n + r_E + 1 : 2n, \ 3n + 1 : 3n + r_E)\). If \(\Pi\) is the corresponding row permutation matrix, and if we set
\[
\hat{P}_2 = \left(\begin{array}{c} I_{2n+r_E} \\ Q_{A_{22}}^* \end{array}\right) \Pi,
\]
then the transformed pencil is
\[
\hat{P}_2 A_{22} = \left(\begin{array}{c} B \\ Q_{E,1}^* D \\ 0 \\ -I_n \\ 0 \\ 0 \\ 0_{n \times (n+r_E)} \end{array}\right), \quad \hat{P}_2 B_{22} = \left(\begin{array}{c} -A \\ -Q_{E,1}^* C \\ 0 \\ 0 \\ -N_{[1]} \\ -N_{[2]} \\ N_{[3]} \\ N_{[4]} \end{array}\right).
\]
(2.12)

To deflate the additional \(n - r_2\) zeros, we reduce the trailing \(n - r_2\) rows of the blocks \(-N_{[1]}, -N_{[2]}\) and \(N_{[3]}\) to zero. This is done by the complete orthogonal decomposition
\[
(\hat{P}_2 B_{22})(2n + r_E + r_2 + 1 : 3n + r_E, :) = U_{BB} R_{BB} V_{BB}^T,
\]
so that \((\hat{P}_2 B_{22})(2n + r_E + r_2 + 1 : 3n + r_E, :) V_{BB} = (0 \ \bar{B}_{22})\). Finally, the deflated pencil is
\[
\hat{P}_2 A_{22} V_{BB} - \lambda \hat{P}_2 B_{22} V_{BB} = \left(\begin{array}{c} A_{33} - \lambda \bar{B}_{33} \\ 0 \end{array}\right).
\]
(2.14)

This reduction process continues by forwarding \(A_{33} - \lambda \bar{B}_{33}\) to the next step of reduction toward an upper triangular KCF, as described in [7].

**Remark 2.3.** For a more structured backward error in case of graded matrices, the complete orthogonal (URV) decomposition (2.13) should be computed as in [7, §2.2].

**Remark 2.4.** If the matrix \(A\) is rank deficient, and \(E\) is full rank, we process the reversed problem \((\mu^2 E + \mu^3 D + \mu^2 C + \mu B + A)x = 0, \ \mu = 1/\lambda,\) and the corresponding truncated linearization pencil of order \(3n + r_A\) reads
\[
A_{22} - \lambda \bar{B}_{22} = \left(\begin{array}{c} D \\ Q_{A,1}^* B \\ 0 \\ -I_n \\ 0 \\ 0 \\ 0_{n+r_A} \end{array}\right) - \lambda \left(\begin{array}{c} -E \\ -Q_{A}^* C \\ 0 \\ -I_n \\ 0_{(n+r_A) \times (2n)} \end{array}\right),
\]
and the rank of matrix \(A_{22}\) is now \(2n + r_A\) + the rank of the \(n \times n\) matrix \(\left(\begin{array}{c} Q_{A}^* B \\ R_{A} P_{A}^* \end{array}\right)\).

### 2.3.3 Both matrices \(A\) and \(E\) are singular

When both matrices \(A\) and \(E\) are rank deficient, then, following the discussion from §2.3.2 the key information is in the numerical ranks of the matrices
\[
\Phi = \left(\begin{array}{c} Q_{A}^* B \\ R_{A} P_{A}^* \end{array}\right), \quad \Psi = \left(\begin{array}{c} Q_{E}^* D \\ R_{E} P_{E}^* \end{array}\right).
\]
(2.16)
Both $\Phi$ and $\Psi$ are full rank. In this case, in the KCF the zero and the infinite eigenvalue occupy single block each, induced by the rank deficiency of $E$ and $A$. The deflation process starts by creating $n-r_E$ and $n-r_A$ zero rows in the coefficients of the corresponding linearization as follows:

$$
\begin{pmatrix}
Q_M^* & 0 \\
0 & Q_K^*
\end{pmatrix}
\begin{pmatrix}
C & -I_{2n} \\
K & 0
\end{pmatrix}
- \lambda
\begin{pmatrix}
-M & 0 \\
0 & -I_{2n}
\end{pmatrix}
\begin{pmatrix}
I_{2n} & 0 \\
0 & Q_K
\end{pmatrix}
= 
\begin{pmatrix}
0_n & D \\
0_{n \times (n-r_A)} & Q_A^*(r_A + 1 : n,:)B \\
0_{n \times (n-r_A)} & Q_A^*(r_A + 1 : n,:)B
\end{pmatrix}
\begin{pmatrix}
0_n & -Q_E(:, 1 : r_E) \\
0_{n \times r_E} & 0_{r_A \times r_E}
\end{pmatrix}
\begin{pmatrix}
0_{n \times (n-r_E)} & -Q_E(:, r_E + 1 : n) \\
0_{(n-r_A) \times r_E} & 0_{(n-r_A) \times (n-r_E)}
\end{pmatrix}
= 
\begin{pmatrix}
0_n & 0_{n \times r_E} \\
0_{n \times (n-r_A)} & 0_{n \times (n-r_A)}
\end{pmatrix}
\begin{pmatrix}
0_n & 0_{n \times (n-r_E)} \\
0_{(n-r_A) \times r_E} & 0_{(n-r_A) \times (n-r_E)}
\end{pmatrix}
\begin{pmatrix}
0_{n \times (n-r_E)} & -Q_E(:, r_E + 1 : n) \\
0_{(n-r_A) \times r_E} & 0_{(n-r_A) \times (n-r_E)}
\end{pmatrix}.
\tag{2.17}
$$

The next step is to compute the complete orthogonal decomposition

$$
Q_A^*(r_A + 1 : n,:)B \\
Q_A^*(r_A + 1 : n,:)
0_{(n-r_A) \times r_E}
= Q_X \begin{pmatrix}
R_X & 0_{(n-r_A) \times (n+r_E + r_A)}
\end{pmatrix} Z_X,
\tag{2.18}
$$

and permute the first $(n-r_A)$ and the last $(n+r_E+r_A)$ columns to get

$$
Q_X \begin{pmatrix}
Q_A^*(r_A + 1 : n,:)B \\
Q_A^*(r_A + 1 : n,:)
0_{(n-r_A) \times r_E}
\end{pmatrix} Z_X^* \begin{pmatrix}
0 & I_{n-r_E} \\
I_{n+r_A+r_E} & 0
\end{pmatrix}
= \begin{pmatrix}
0_{(n-r_A) \times (n+r_E + r_A)} & R_X
\end{pmatrix}.
$$

Finally, to complete the deflation process the following left and right transformation matrices must be applied on the pencil $\begin{pmatrix} Q_A^*(r_A + 1 : n,:)B \\ Q_A^*(r_A + 1 : n,:) \end{pmatrix}$ $Z_X$.

$$
\begin{pmatrix}
I_{n+r_A} & 0 & 0 & 0 \\
0 & 0 & I_{r_E} & 0 \\
0 & 0 & I_n & 0 \\
0 & 0 & 0 & Q_X^*
\end{pmatrix},
\begin{pmatrix}
Z_X^* \begin{pmatrix}
0 & I_{n-r_E} \\
I_{n+r_A+r_E} & 0
\end{pmatrix} & 0 \\
0 & 0
\end{pmatrix}.
$$

After the transformation step, the deflation is finished by removing the last $2n-r_E-r_A$ rows and columns from the obtained pencil. The resulting pencil of dimension $2n+r_A+r_E$ is forwarded to the QZ algorithm.

Only one matrix in $\begin{pmatrix} Q_A^*(r_A + 1 : n,:)B \\ Q_A^*(r_A + 1 : n,:) \end{pmatrix}$ is singular. This means that there are at least two KCF blocks for zero (if $\Psi$ is singular) or infinite (if $\Phi$ is singular) eigenvalue. In either case, we deflate two blocks for the zero eigenvalue using the structure described in $\begin{pmatrix} Q_A^*(r_A + 1 : n,:)B \\ Q_A^*(r_A + 1 : n,:) \end{pmatrix}$ (see also $\begin{pmatrix} Q_A^*(r_A + 1 : n,:)B \\ Q_A^*(r_A + 1 : n,:) \end{pmatrix}$ §5.2, §6.1), meaning that the reversed problem is considered if there are more blocks for the infinite eigenvalues.
After deflating two blocks of zero eigenvalues, we obtain the pencil (2.14). Now, the existence of additional zero eigenvalues depends on the rank of the matrix $A_{22}$. To deflate possible additional zeros, the pencil $A_{22} - \lambda B_{22}$ is forwarded to the algorithm for computing the KCF [7, §3.2]. As the output we get the pencil $A_{\ell+1,\ell+1} - \lambda B_{\ell+1,\ell+1}$ and transformation matrices $Q_p$ and $P_p$, with $A_{\ell+1,\ell+1}$ regular. Denote with $n_{\ell+1}$ the dimension of the resulting pencil.

Finally, we have to deflate one block of infinite eigenvalues, which have been detected at the beginning. This is done by forwarding the reversed pencil $B_{\ell+1,\ell+1} - \lambda A_{\ell+1,\ell+1}$ to algorithm [7]. As the input to the algorithm we supply the information that there is only one block to be deflated, so that only one step of the algorithm is needed. In addition, we also send the number of infinite eigenvalues so that the rank determination of the matrix $B_{\ell+1,\ell+1}$ is omitted. As an output, we get the pencil $A_{\ell+1,\ell+1} - \lambda B_{\ell+1,\ell+1}$ with both $A_{\ell+1,\ell+1}$ and $B_{\ell+1,\ell+1}$ regular, and the corresponding transformation matrices $P_{p1}$ and $Q_{p1}$. The final transformation matrices $Q$ and $P$ are

$$Q = \begin{pmatrix} I_{2n} & 0 & 0 & Q_K \end{pmatrix} \begin{pmatrix} V_p^T & 0 & 0 & 0 \\ 0 & V_p & 0 & 0 \\ 0 & 0 & I_{n-r_E} & 0 \\ 0 & 0 & 0 & I_{2n-r_E} \end{pmatrix} \begin{pmatrix} Q_{p1} & 0 \\ 0 & I_{4n-n_{\ell+1}} \end{pmatrix}$$

$$P = \begin{pmatrix} P_{p1} & 0 & 0 & 0 \\ 0 & I_{4n-n_{\ell+1}} \end{pmatrix} \begin{pmatrix} V_p & 0 & 0 & 0 \\ 0 & V_p & 0 & 0 \\ 0 & 0 & Q_{A_{22}} & 0 \\ 0 & 0 & 0 & I_{2n-r_E} \end{pmatrix} \begin{pmatrix} Q_{z} & 0 \\ 0 & Q_K \end{pmatrix}.$$  

Both matrices in (2.16) are singular. This case is analogous to the previous one. The only difference is that, when we call the algorithm on the reversed pencil $B_{\ell+1,\ell+1} - \lambda A_{\ell+1,\ell+1}$, we provide additional information that there are least two steps of deflation ahead, as well as the dimensions of the first two blocks which were previously determined by the rank revealing decompositions of $A$ and $\Phi$.

### 2.3.4 An illustrative example

Let us illustrate the action of the additional reduction steps toward the KCF. We use the mirror example from the NLEVP library; it originates from the calibration of catadioptric vision system [22]. The problem is of order $n = 9$.

Both $A$ and $E$ are rank deficient, with the rank $r_E = r_A = 2$, which means that there are at least 7 zero and 7 infinite eigenvalues. They are correctly identified and deflated in the preprocessing in quadeig; in the next step, the QZ algorithm found an additional zero eigenvalue, and two more infinite eigenvalues. On the other hand, polyeig identified in total only 2 zero and 9 infinite eigenvalues. This shows the advantage of the preprocessing introduced in quadeig.

However, the preprocessing in both KVDeig and KVARteig found additional two zero and two infinite eigenvalues, making the total of 9 zero and 9 infinite eigenvalues deflated before calling the QZ.
Figure 1 presents (for all four algorithms) the backward errors for all eigenvalues, sorted by the magnitude. It is clear from this figure that by just looking at the norm-wise backward error we cannot conclude that \texttt{polyeig} and \texttt{quadeig} did not find all zero eigenvalues because the backward errors are satisfactory low. This example shows the importance of checking whether there are more blocks in the KCF carrying zero and infinite eigenvalue and then deflating them.

If we look at the structure of the matrices $A$ and $E$ for this particular problem, we see that their ranks can be determined exactly because both have 7 zero columns. On the other hand, the block matrices (2.16), which are used to determine the existence of more than one block for zero and infinite eigenvalues, also have two zero columns each, and the remaining $9 \times 7$ submatrices are well conditioned. Thus we can argue that our algorithm has determined the correct numbers of zero and infinite eigenvalues.

3 Computing the eigenvectors

In the computation of the eigenvectors, we have two main computational tasks: (i) restore the eigenvectors of the quartic problem from the eigenvectors of its linearization via quadratification; (ii) assemble the eigenvectors of the linearization from the eigenvectors of the deflated (linearization) pencil, using the transformation matrices.

3.1 Quartic eigenvectors from the eigenvectors of the linearization

For an eigenvalue $\lambda$, the eigenvectors of the original problem (1.1) and the final linearization pencil (1.5) can be related using explicit formulas. For the reader’s convenience, we briefly outline the crux of this connection.

We use $z \in \mathbb{C}^{4n}$ and $w \in \mathbb{C}^{4n}$ to denote the right and the left eigenvector for the linearization, and $x \in \mathbb{C}^{n}$, $y \in \mathbb{C}^{n}$ to denote the right and the left eigenvector for the original problem. The eigenvalue $\lambda \in \mathbb{C}$ is now fixed as assumed nonzero and finite.

Let $z = \left( z_1^T, z_2^T, z_3^T, z_4^T \right)^T \in \mathbb{C}^{4n}$, $z_i \in \mathbb{C}^{n}$, $i = 1, 2, 3, 4$ be a right eigenvector for the

\footnote{It will be interesting to revisit this example after reading Example 5.6}
eigenvalue $\lambda$ ($0 < |\lambda| < \infty$) of the linearized problem, i.e. $(A - \lambda B)z = 0$:

$$
(A - \lambda B)z = \begin{pmatrix}
B & 0 & -I & 0 \\
D & 0 & 0 & -I \\
0 & -I & 0 & 0 \\
E & 0 & 0 & 0
\end{pmatrix} - \lambda \begin{pmatrix}
-A & 0 & 0 & 0 \\
-C & -I & 0 & 0 \\
0 & 0 & -I & 0 \\
0 & 0 & 0 & -I
\end{pmatrix} \begin{pmatrix}
z_1 \\
z_2 \\
z_3 \\
z_4
\end{pmatrix} = \begin{pmatrix}0 \\ 0 \\ 0 \\ 0\end{pmatrix}.

$$

(3.1)

By equating the corresponding block components on the left and on the right we get

$$
Bz_1 - z_3 + \lambda Az_1 = 0 \iff z_3 = (\lambda A + B)z_1, \tag{3.2}
$$

$$
Dz_1 - z_4 + \lambda Cz_1 + \lambda z_2 = 0 \iff Dz_1 + (1/\lambda)Ez_1 + \lambda Cz_1 + \lambda^2(\lambda A + B)z_1 = 0, \tag{3.3}
$$

$$
-z_2 + \lambda z_3 = 0 \iff z_2 = \lambda z_3, \tag{3.4}
$$

$$
Ez_1 + \lambda z_4 = 0 \iff \lambda z_4 = -Ez_1. \tag{3.5}
$$

It follows immediately that $z_1 \neq 0$: if $\det(\lambda A + B) \neq 0$, then, in addition, $z_3 \neq 0$ and $z_2 \neq 0$; if $\det(E) \neq 0$, then also $z_4 \neq 0$. Using (3.3) we easily check that $x = z_1/\lambda$ is an eigenvector of the original quartic problem. Further, (3.2) implies that $x$ satisfies $z_3 = \lambda(\lambda A + B)x$, and (3.4) yields $z_2 = \lambda^2(\lambda A + B)x$, and finally from (3.5) it follows that $z_4 = -Ex$. Similarly, if we initially assume that $x$ is an eigenvector of the quartic problem, these formulas for the $z_i$’s give an eigenvector of [3.1].

An analogous computation reveals a left eigenvector $y$, using the partitioned left eigenvector of the linearization, as $w = (w_1^T \quad w_2^T \quad w_3^T \quad w_4^T)^T$, $w_i \in \mathbb{C}^n$, $i = 1, 2, 3, 4$. Altogether, we obtain the following relations between the two sets of eigenvectors:

$$
z = \begin{pmatrix}z_1 \\ z_2 \\ z_3 \\ z_4\end{pmatrix} = \begin{pmatrix}\lambda x \\ \lambda^2(\lambda A + B)x \\ \lambda(\lambda A + B)x \\ -Ex\end{pmatrix}, \quad w = \begin{pmatrix}w_1 \\ w_2 \\ w_3 \\ w_4\end{pmatrix} = \begin{pmatrix}\lambda^3 y \\ \lambda^2 y \\ \lambda y \\ y\end{pmatrix}. \tag{3.6}
$$

For both the right and the left eigenvector there are four choices to recover $x$ and $y$. Reconstruction of the left eigenvector seems easier. We just choose one of the block components $w_1, w_2, w_3$ or $w_4$ and rescale appropriately.

For the right eigenvector we can choose $z_1, (\lambda A + B)^{-1}z_2, (\lambda A + B)^{-1}z_3$ or $E^{-1}z_4$. Notice that, for the last three choices we have to solve system of linear equations in order to compute the wanted vector. Given all the difficulties in numerical solution of nonlinear eigenvalue problem, we ought to use all alternatives in order to obtain better output – in this case, for instance, we can solve all systems and select the vector with smallest residual.

**Remark 3.1.** If $\lambda = 0$, for the corresponding right eigenvector we have $Ex = 0$ and $Ax = 0$. By the same reasoning as above, we conclude the following connection $z = (x^T \quad 0 \quad (Bx)^T \quad (Dx)^T)^T$.

### 3.1.1 Computing $(\lambda A + B)^{-1}z_2, (\lambda A + B)^{-1}z_3$ or $E^{-1}z_4$ multiple times

Inverting $E$ (assuming $\det(E) \neq 0$) multiple times can be done by reusing initially computed LU decomposition. On the other hand computing $(\lambda A + B)^{-1}z_2, (\lambda A + B)^{-1}z_3$ for $4n$ values of $\lambda$ is not that simple because the coefficients of the linear system change with $\lambda$; $O(n^3)$ flops per eigenvalue to compute the corresponding eigenvector is prohibitive complexity. Fortunately, this can be reduced using a bag of tricks for solving shifted linear systems. In particular, this problem is similar to evaluating the transfer function of a descriptor LTI dynamical system at multiple frequencies.
We can compute the triangular-Hessenberg form of \((A, B)\), i.e. a unitary \(Q\), an upper triangular \(T\) and an upper Hessenberg matrix \(H\) can be constructed in \(O(n^3)\) time so that \(A = QTQ^*\), \(B = QHQ^*\). Hence, for any vector \(v\)

\[
(\lambda A + B)^{-1}v = Q[(\lambda T + H)^{-1}(Q^*v)],
\]

which has \(O(n^2)\) complexity because \(\lambda T + H\) is upper Hessenberg. This means that the total work (for all \(4n\) eigenvalues) of choosing the eigenvectors with smallest residuals remains \(O(n^3)\).

(Here, the tacit assumption is that \(A + \lambda B\) is nonsingular and well conditioned with respect to inversion.)

**Remark 3.2.** In some applications, such as e.g. computing deformation modes of thin-walled structures, the cubic term is zero, \(B = 0\), so that the shifted systems can be replaced with linear system matrix \(A\) for all \(\lambda\)'s. Other details include e.g. the case of real data and using the complex conjugate eigenpairs to save unnecessary computation. Here we omit those details and leave them for the detailed description of a software implementation, which is a subject of ongoing work.

### 3.1.2 Least squares reconstruction of the eigenvectors

Since in a finite precision computation the computed eigenvector \(z\) is only an approximation (thus noisy), and since \(A + \lambda B\) is not guaranteed to be well conditioned, it makes sense to turn the conditions (3.6) into a least squares problem, but keeping in mind than we may have to solve it \(4n\) times (i.e. we may take e.g. only two conditions to form the least squares problem).

So, for instance, we can compute \(x\) by solving the least squares problem

\[
\begin{align*}
\left\| \left( \begin{array}{c} \lambda I_n \\ E^* \\
\end{array} \right) x - \left( \begin{array}{c} z_1 \\ -z_4 \\
\end{array} \right) \right\|_2 & \rightarrow \min \quad (\text{or e.g. } \left\| \left( \begin{array}{c} I_n \\ E \\
\end{array} \right) x - \left( \begin{array}{c} z_1/\lambda \\ -z_4 \\
\end{array} \right) \right\|_2 \rightarrow \min) \quad (3.7)
\end{align*}
\]

If the data is well scaled (\(\|E\|_2 \approx \|I\|_2 = 1\)), this can be solved efficiently by (semi-)normal equations. Note that \(|\lambda|^2 I_n + E^*E\) is well conditioned if the matrices are well scaled; this is because \(E^*E\) as an additive perturbation of \(|\lambda|^2 I_n\) moves the eigenvalue \(|\lambda|^2\) (the spectrum of \(|\lambda|^2 I_n\)) to the right by at most \(\|E\|_2\), which is moderate in the case of well scaled data.

In general, the least squares problem (3.7) can be solved efficiently for any eigenvalue \(\lambda \neq 0\) by pre-computing the SVD \(E = U_E \Sigma_E V_E^*\) (which actually may be available if we used it for a strong rank revealing of \(E\)) and then, for each triple \(\lambda\), \(z_1\), \(z_4\), solving in \(O(n^2)\) flops the equivalent problem

\[
\begin{align*}
\left\| \left( \begin{array}{c} \lambda I_n \\ \Sigma_E \\
\end{array} \right) V_E^*x - \left( \begin{array}{c} V_E^*z_1 \\ -U_E z_4 \\
\end{array} \right) \right\|_2 & \rightarrow \min \quad (\text{or } \left\| \left( \begin{array}{c} I_n \\ \Sigma_E \\
\end{array} \right) V_E^*x - \left( \begin{array}{c} V_E^*z_1/\lambda \\ -U_E^* z_4 \\
\end{array} \right) \right\|_2 \rightarrow \min) \quad (3.8)
\end{align*}
\]

If \(\lambda = 0\), then, based on Remark 3.1, the corresponding eigenvector can be found from either of the following least squares problems

\[
\begin{align*}
\left\| \left( \begin{array}{c} I_n \\ B \\
\end{array} \right) x - \left( \begin{array}{c} z_1 \\ z_3 \\
\end{array} \right) \right\|_2 & \rightarrow \min, \\
\left\| \left( \begin{array}{c} I_n \\ D \\
\end{array} \right) x - \left( \begin{array}{c} z_1 \\ z_4 \\
\end{array} \right) \right\|_2 & \rightarrow \min, \quad (3.9)
\end{align*}
\]

which can be efficiently solved for all eigenvectors \(z\) of \(\lambda = 0\), using one of the approached discussed above. Other possibilities include e.g. using the bidiagonalization instead of the SVD (of \(B\) or \(D\)).
3.2 Assembling the eigenvectors of the linearization

Let \( \tilde{z} \) and \( \tilde{w} \) be the computed right and left eigenvector for the linearization pencil (1.3). Both right and left eigenvectors will have \( 4n \) elements if no deflation occurred, otherwise the number of elements will be \( 4n - d \), where \( d \) is the total number of zero and infinite eigenvalues deflated. \( 4n - d \) is also the dimension of the truncated pencil \( \tilde{A} - \lambda \tilde{B} = P(\tilde{A} - \lambda \tilde{B})Q \) which is passed to the QZ algorithm for computation of finite nonzero eigenvalues.

3.2.1 Case 1: No deflation has occurred

Let \( \tilde{z} \) and \( \tilde{w} \) be the right and the left eigenvector of the transformed pencil \( P(\tilde{A} - \lambda \tilde{B})Q \). The corresponding right and the left eigenvectors for the original linearization pencil are \( z = Q\tilde{z} \) and \( w = P^T\tilde{w} \). The right and the left eigenvector for the quartic problem are computed as described in §3.1.

3.2.2 Case 2: Deflation has occurred

Let \( n_{\ell+1} \) be the dimension of the deflated linearization \( A_{\ell+1,\ell+1} - \lambda B_{\ell+1,\ell+1} \), i.e. both \( A_{\ell+1,\ell+1} \) and \( B_{\ell+1,\ell+1} \) are regular. Let \( \tilde{z} \in \mathbb{C}^{n_{\ell+1}} \) and \( \tilde{w} \in \mathbb{C}^{n_{\ell+1}} \) be the right and the left eigenvector for a finite nonzero eigenvalue \( \lambda \).

To recover eigenvectors of the initial linearization, we must lift \( \tilde{z} \) and \( \tilde{w} \) to the \( 4n \)-dimensional space. For the right eigenvector this is easy; we just append \( 4n - n_{\ell+1} \) zeros to \( \tilde{z} \) to get \( z = Q \begin{pmatrix} \tilde{z}^T & 0_{1 \times (4n-n_{\ell+1})} \end{pmatrix}^T \).

For the left eigenvector, let \( \tilde{w} \in \mathbb{C}^{n-n_{\ell+1}} \) be the vector satisfying \( \begin{pmatrix} \tilde{w}^T & \tilde{w}_2^T \end{pmatrix} P(\tilde{A} - \lambda \tilde{B})Q = 0 \). From
\[
\begin{pmatrix} \tilde{w}^T & \tilde{w}_2^T \end{pmatrix} P(\tilde{A} - \lambda \tilde{B})Q = \begin{pmatrix} \tilde{w}^T & \tilde{w}_2^T \end{pmatrix} \begin{pmatrix} \hat{A}_{\ell+1,\ell+1} - \lambda \hat{B}_{\ell+1,\ell+1} & X \\ 0 & Y \end{pmatrix} = 0
\]
we conclude that \( \tilde{w}_2 = -\tilde{w}^*XY^{-1} \). (It follows from §2.3.2 that \( Y \) is nonsingular.) Now, the left eigenvector for the original linearization is \( w = P^T \begin{pmatrix} \tilde{w}^T & \tilde{w}_2^T \end{pmatrix} \).

The right eigenvectors for the zero eigenvalue span the nullspace of the matrix \( E \). The corresponding basis is computed for the orthogonal complement of the range of \( E^* \). To compute this basis we can use the already computed QR factorization of \( E^* \) as follows. First compute the QR factorization of \( \Pi_E \widehat{R}_E^* = Q_{\widehat{R}_E^*} R_{\widehat{R}_E^*} \). Now, the last \( n - r_E \) columns of \( Q_{\widehat{R}_E^*} \) represent the basis for the nullspace of the matrix \( \widehat{E} \). Similarly, the right eigenvectors of the infinite eigenvalue span the nullspace of the matrix \( A \). The basis is computed using the already computed QR factorization \( 2.1 \). Again, compute the QR factorization of \( \Pi_A R_A^* = Q_{\widehat{R}_A} R_{\widehat{R}_A} \), and the last \( n - r_A \) columns of \( Q_{\widehat{R}_A} \) represent the basis for the nullspace of \( A \).

The left eigenvectors for the zero eigenvalue are determined as the last \( n - r_E \) columns of the unitary matrix \( Q_E \) from the corresponding QR factorization, and the left eigenvectors for the infinite eigenvalue are selected as the last \( n - r_A \) columns of the unitary matrix \( Q_A \) from the QR factorization of \( A \).

4 Backward error analysis

In this section, we develop a backward error analysis for the first two steps of the deflation procedure described in §2.3.2. The following proposition deals with the first step, that is, the deflation of the first batch of \( n - r_E \) zero eigenvalues.
Proposition 4.1. Let $E\tilde{\Pi}_E \approx \tilde{Q}_E \begin{pmatrix} \tilde{R}_E \\ 0 \end{pmatrix}$ be the computed rank revealing QR factorization of $E$, and let $\tilde{r}_E$ be the computed numerical rank of $E$. Further, let $\tilde{X} = \text{computed}(\tilde{Q}_E D)$, $\tilde{Y} = \text{computed}(\tilde{Q}_E C)$. Let

$$
\tilde{A} - \lambda \tilde{\mathbb{B}} = \begin{pmatrix} B & 0 & -I_n & 0 \\ X & 0 & 0 & 0 \\ 0 & -I_n & 0 & 0 \\ R_E \Pi_E^T & 0 & 0 & 0 \end{pmatrix} - \lambda \begin{pmatrix} -A & 0 & 0 & 0 \\ -Y & -Q_E^* & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0_{n+\tilde{r}_E} & 0_{n+\tilde{r}_E} & 0_{n+\tilde{r}_E} & 0_{n+\tilde{r}_E} \end{pmatrix}
$$

be the computed reduced pencil (2.7), extracted from the transformed linearization (2.7). There exists small structured perturbation

$$
\delta \mathbb{B} = \begin{pmatrix} 0 & 0 & 0_{2n} \\ 0 & -\delta Q_E^* & 0_{2n} \\ 0_{2n} & 0_{2n} & -0_{2n} \end{pmatrix}
$$

such that $\tilde{A} - \lambda (\tilde{\mathbb{B}} + \delta \mathbb{B})$ corresponds to an exact reduced quadratization of a quartic pencil

$$
\lambda^4 A + \lambda^3 B + \lambda^2 (C + \delta C) + \lambda (D + \delta D) + (E + \delta E + \Delta E)
$$

with at least $n - \tilde{r}_E$ zero eigenvalues, where, for all $i = 1, \ldots, n$,

$$
\|\delta C(:,j)\|_2 \leq \epsilon_C \|C(:,j)\|_2, \quad \|\delta D(:,i)\|_2 \leq \epsilon_D \|D(:,i)\|_2, \quad \|\delta E(:,i)\|_2 \leq \epsilon_{qr} \|E(:,i)\|_2,
$$

and the truncation error from the determination of the numerical rank of $E$ is\(^3\)

$$
\max_{j=1:n-k} \| (\Delta E) \tilde{\Pi}_E(:,j) \|_2 \leq \tau \min_{i=1:k} \| (E + \delta E) \tilde{\Pi}_E(:,i) \|_2, \quad (\Delta E) \tilde{\Pi}_E(:,1:k) = 0_{n,k}.
$$

Here $\epsilon_C, \epsilon_D, \epsilon_{qr}$ are bounded by a moderate function of $n$ times the machine precision $\epsilon$, and $\tau$ is prescribed threshold parameter.

**Proof.** The computed QR factorization of $E$, $E\tilde{\Pi}_E \approx \tilde{Q}_E \begin{pmatrix} \tilde{R}_E \\ 0 \end{pmatrix}$ can be represented as $(E + \delta E + \Delta E) \tilde{\Pi}_E = \tilde{Q}_E \begin{pmatrix} \tilde{R}_E \\ 0 \end{pmatrix}$, where $\tilde{Q}_E$ is exactly unitary and $\|\tilde{Q}_E - \tilde{Q}_E\|_F \leq \epsilon_{qr}$; the backward error $\delta E$ is induced by rounding errors during the factorization, and $\Delta E$ is the truncation error from the numerical rank. If we set $\delta Q_E = \tilde{Q}_E - Q_E$, then $Q_E = \tilde{Q}_E(1_n + \tilde{Q}_E^* \delta Q_E) = (1_n + \delta Q_E \tilde{Q}_E^*) \tilde{Q}_E$. We can also write $(E + \delta_1 E + \Delta E) \tilde{\Pi}_E = \tilde{Q}_E \begin{pmatrix} \tilde{R}_E \\ 0 \end{pmatrix}$, where $\delta_1 E = \delta E + \delta Q_E \begin{pmatrix} \tilde{R}_E \\ 0 \end{pmatrix}$, and thus

$$
\begin{pmatrix} \tilde{R}_E \\ 0 \end{pmatrix} \tilde{\Pi}_E^T = \tilde{Q}_E^T (E + \delta E + \Delta E) = \tilde{Q}_E^{-1}(E + \delta_1 E + \Delta E).
$$

There is an important subtlety here, and it is instructive to discuss it in more detail. In the actually computed matrix $\tilde{\mathbb{B}}$, stored in the computer memory, one of its blocks is the numerically computed numerically orthogonal $\tilde{Q}_E$. The backward stability of the QR factorization is usually stated in terms of an exactly unitary matrix $\tilde{Q}_E$, which is an unaccessible object as it is artificially constructed in the proof of backward stability. This is motivated by the desire to be able to say that we have computed the exact QR factorization of a nearby matrix. The matrices $\tilde{X}$ and $\tilde{Y}$ are computed by using the floating point matrix $\tilde{Q}_E$, possibly implicitly as

---

\(^3\)See Remark 2.2.
in the LAPACK subroutine xORMQR, or by explicit matrix multiply (xGEMM from BLAS) using explicitly formed $\tilde{Q}_E$, using xORGQR (LAPACK). The computed $X$, $Y$ can be represented as

$$\text{computed}(\tilde{Q}_E^* D) = \tilde{Q}_E^* D + \delta_0 D = \tilde{Q}_E^* (D + \delta D), \quad \delta D = \tilde{Q}_E^* \delta D, \quad |\delta D| \leq \epsilon |\tilde{Q}_E||D|,$$

$$= \tilde{Q}_E^*(D + \Delta D), \quad \Delta D = \delta D + \tilde{Q}_E^*(\delta Q_E)^* D + \tilde{Q}_E^* (\delta Q_E)^* \delta D, \quad \epsilon \leq O(n \epsilon);$$

$$\text{computed}(\tilde{Q}_E^* C) = \tilde{Q}_E^* C + \delta_0 C = \tilde{Q}_E^* (C + \delta C), \quad \delta C = \tilde{Q}_E^* \delta_0 C, \quad |\delta_0 C| \leq \epsilon |\tilde{Q}_E||C|,$$

$$= \tilde{Q}_E^*(C + \Delta C), \quad \Delta C = \delta C + \tilde{Q}_E^* (\delta Q_E)^* C + \tilde{Q}_E^* (\delta Q_E) \delta C.$$

On the other hand, the unit blocks $I_n \oplus \Pi_{rE}$ in $\bar{A}$ and $I_{n+rE}$ in $\bar{B}$ assume exact orthogonality of $\tilde{Q}_E$, which is not feasible in finite precision arithmetic. If we set $\Delta \Sigma E = \delta_1 E + \Delta E$, then we can represent the computed linearization (2.7) as

$$\begin{pmatrix} I_n & 0 & 0 & 0 \\ 0 & \tilde{Q}_E & 0 & 0 \\ 0 & 0 & I_n & 0 \\ 0 & 0 & 0 & \tilde{Q}_E^{-1} \end{pmatrix} \begin{pmatrix} B + \Delta B & 0 & -I_n & 0 \\ 0 & D + \Delta D & 0 & -I_n \\ -I_n & 0 & E + \Delta E & 0 \\ 0 & -I_n & 0 & E \end{pmatrix} \begin{pmatrix} I_n & 0 & 0 & 0 \\ 0 & \tilde{Q}_E & 0 & 0 \\ 0 & 0 & I_n & 0 \\ 0 & 0 & 0 & \tilde{Q}_E \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \ 0 \ 0 \ 0 \end{pmatrix} - \lambda \begin{pmatrix} 0 & 0 & 0 \ 0 & 0 & 0 \ 0 & 0 & 0 \ 0 & 0 & 0 \end{pmatrix}.$$

Now we see at the block position (2,4) in the left matrix, $\tilde{Q}_E^*(-I_n)\tilde{Q}_E = -I_n + \Xi \neq -I_n$. Hence, (1.1) can be justified by a mixed stability scenario – if the computed pencil is changed by $\|\Xi\|_2 \leq \epsilon_{qr}$ to restore identity at the (2,4) position in the left matrix, then it can be interpreted as an exact transformation of a slightly changed initial pencil.

Alternatively, we can set $\Delta \Sigma E = \delta E + \Delta E$ and model (2.7) as

$$\begin{pmatrix} I_n & 0 & 0 & 0 \\ 0 & \tilde{Q}_E & 0 & 0 \\ 0 & 0 & I_n & 0 \\ 0 & 0 & 0 & \tilde{Q}_E \end{pmatrix} \begin{pmatrix} B + \Delta B & 0 & -I_n & 0 \\ 0 & D + \Delta D & 0 & -I_n \\ -I_n & 0 & E + \Delta E & 0 \\ 0 & -I_n & 0 & E \end{pmatrix} \begin{pmatrix} I_n & 0 & 0 & 0 \\ 0 & \tilde{Q}_E & 0 & 0 \\ 0 & 0 & I_n & 0 \\ 0 & 0 & 0 & \tilde{Q}_E \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \ 0 \ 0 \ 0 \end{pmatrix} - \lambda \begin{pmatrix} 0 & 0 & 0 \ 0 & 0 & 0 \ 0 & 0 & 0 \ 0 & 0 & 0 \end{pmatrix}.$$

In this case, the (2,2) block in the right matrix in (1.1) should be changed from $-\hat{Q}_E^*$ to $-\tilde{Q}_E^*$, by adding $\delta Q_E^*$, to establish exact equivalence with a slightly perturbed initial pencil.

\begin{remark}
The forward error introduced in (4.4) (thus making the model of the analysis of mixed forward-backward type) is due to the fact that in finite precision computation unitarity/orthogonality cannot be guaranteed\footnote{For that reason the QR factorization can only be mixed stable, and in general it is not backward stable.}. Note that this error is localized to one block of the linearization; its structure can be easily seen from the backward analysis of the e.g. Householder QR factorization.

We now consider the first two steps and show that the algorithm remains mixed stable. The proof is technically more involved, but it is important to see how the reduced linear pencil after small forward modification exactly corresponds to a quartic pencil with backward errors in the initial coefficient matrices. Also, the proof nicely illustrates the benefits of well scaled data.
\end{remark}
Theorem 4.1. Assume the notation of Proposition 4.1, and let

\[
\tilde{P}_2A_{22} = \begin{bmatrix}
B & 0 & -I_n \\
Q^*_{E,1}(D + \Delta D) & 0 & -I_{nE} \\
0 & 0 & 0 \\
R_{A_{22}} \Pi^T_{E} & 0 & 0 \\
0 & 0 & 0 \\
\end{bmatrix},
\]

\[
\tilde{P}_2B_{22} = \begin{bmatrix}
-A & 0 & 0 \\
-Q^*_{E,1}(C + \Delta C) & -Q^*_{E,1} & 0 \\
0_n & 0 & 0 \\
-\Delta N_1 & -\Delta N_2 & 0 \\
\end{bmatrix},
\]

be the computed version of (2.12). There exist small structured forward perturbation

\[
\mathcal{F}_{22} = \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & -\delta Q^*_{E,1} & 0 & 0 \\
0 & 0 & 0 & 0 \\
\Delta N_1 & \Delta N_2 & 0 & 0 \\
\end{bmatrix},
\]

of \(\tilde{P}_2B_{22}\), and backward errors \(\Delta C, \Delta_2 D, \Delta_2 E\) such that \(\tilde{P}_2A_{22} - \lambda(\tilde{P}_2B_{22} + \mathcal{F}_{22})\) corresponds to an exact reduced quadratification of a quartic pencil

\[
\lambda^4 A + \lambda^3 B + \lambda^2 (C + \Delta C) + \lambda (D + \Delta_2 D) + (E + \Delta_2 E),
\]

with the exact transformation given in (4.7) and (4.10) below. Under mild technical assumption (on the size of \(n\)), \(\|\Delta N_1\|_2\), \(\|\Delta N_2\|_2\) are small relative to \(\|\tilde{N}_1\|_2\) and \(\|\tilde{N}_2\|_2\), respectively, see (4.13) below. Further, \(\Delta C\) is as in Proposition 4.1, and \(\Delta_2 D\) and \(\Delta_2 E\) are small perturbations of \(D, E\), respectively, if the matrices are so scaled that the norms of \(D\) and \(E\) are of same order.

Proof. We continue based on the details and the notation from the proof of Proposition 4.1. The next step is computation of the rank revealing factorization of the block matrix \(\begin{bmatrix}
Q^*_{E,1}(D + \delta D) \\
0 \\
0 \\
R_{E} \Pi^T_{E} \\
\end{bmatrix}\).

It is convenient to consider the left matrix in (4.11) with the relevant blocks already swapped (see (2.12))

\[
\begin{bmatrix}
B & 0 & -I_{n} \\
Q^*_{E,1}(D + \delta D) & 0 & -I_{nE} \\
0 & 0 & 0 \\
0 & 0 & 0 \\
R_{E} \Pi^T_{E} & 0 & 0 \\
\end{bmatrix} \rightarrow \begin{bmatrix}
B & 0 & -I_{n} \\
Q^*_{E,1}(D + \delta D) & 0 & -I_{nE} \\
0 & 0 & 0 \\
0 & 0 & 0 \\
Q^*_{E,2}(D + \delta D) & 0 & 0 \\
R_{E} \Pi^T_{E} & 0 & 0 \\
\end{bmatrix}.
\]

For the computed factors \(\tilde{\Pi}_{A_{22}}, \tilde{\Pi}_{E_{22}}, \tilde{R}_{A_{22}}\) it holds that

\[
\left[\begin{bmatrix}
Q^*_{E,2}(D + \delta D) \\
\end{bmatrix} + \begin{bmatrix}
\Gamma_1 \\
\Gamma_2 \\
\end{bmatrix}\right] \tilde{\Pi}_{A_{22}} \equiv \left[\begin{bmatrix}
Q^*_{E,2}(D + \delta D) \\
\end{bmatrix} + \begin{bmatrix}
\Gamma_1 \\
\Gamma_2 \\
\end{bmatrix}\right] \tilde{\Pi}_{A_{22}} = \tilde{Q}_{A_{22}} \left[\begin{bmatrix}
\Gamma_1 \\
\Gamma_2 \\
\end{bmatrix}\right],
\]

where \(\tilde{Q}_{A_{22}}\) is exactly unitary and \(\tilde{Q}_{A_{22}} \approx \tilde{Q}_{A_{22}}\), \(\tilde{R}_{A_{22}}\) is \(\tilde{r}_2 \times n\) of full row rank \(^5\) and \(\Delta = \begin{bmatrix}
\Gamma_1 \\
\Gamma_2 \\
\end{bmatrix}\) is the backward error of the QR factorization.

\[
\left\|\begin{bmatrix}
\Gamma_1 \\
\Gamma_2 \\
\end{bmatrix} \right\|_2 \leq \epsilon_{qr} \left\|\begin{bmatrix}
Q^*_{E,2}(D + \delta D) \\
\end{bmatrix} \right\|_2.
\]

\(^5\)The zero block beneath of \(\tilde{R}_{A_{22}}\) may be void.
We can push $\Gamma_1$ and $\Gamma_2$ backward in $D$ and $E$, respectively, as follows. First, $\tilde{Q}_{E,2}(D + \Delta D) + \Gamma_1 = \tilde{Q}_{E,2}^{*}(D + \Delta D)$ and
\[
\tilde{Q}_{E}^{*}(D + \Delta D + \tilde{Q}_{E,2}\Gamma_1) = \begin{pmatrix}
\tilde{Q}_{E,1}^{*}(D + \Delta D) \\
\tilde{Q}_{E,2}(D + \Delta D) + \Gamma_1
\end{pmatrix}.
\]
If $D$ and $E$ are so scaled that their norms are nearly of the same order, then $\Gamma_1$, $\Gamma_2$ will be, respectively, their relatively small perturbations. Further, an analogous conclusion holds also column-wise, which motivates scaling the initial data by diagonal matrices to equilibrate on the matrix elements level\(^6\). Hence, if the additive perturbation $\Delta D$ is replaced with $\Delta E, \tilde{Q}_{E,2}\Gamma_1$, $\tilde{X}_1$ remains unchanged, and $\tilde{X}_2$ is precisely as in (4.6). (Here $\tilde{X} = (\tilde{x}_1 \tilde{x}_2)$.) Similarly,
\[
\tilde{Q}_{E}^{*}(E + \delta E + \Delta E + \tilde{Q}_{E,1}\Gamma_2) = \begin{pmatrix}
\tilde{R}_E\Pi_E^{T} + \Gamma_2 \\
0
\end{pmatrix}.
\]

Now define $\tilde{P}_2 = (\mathbb{I}_{2n+\tilde{r}_E} \oplus \tilde{Q}_{A_{22}}^{*})\Pi$ analogously to (2.11). The left matrix in (4.13) can be interpreted as an exact transformation of type (3.4), followed by the transformation of type (2.12) with $\tilde{P}_2$, but with initial matrices that are changed as $D \sim D + \Delta_{\Sigma} D; E \sim E + \Delta_{\Sigma} E$, $\Delta_{\Sigma} E = \delta E + \Delta E + \tilde{Q}_{E,1}\Gamma_2$.

\[
\begin{pmatrix}
\tilde{Q}_{E,1}(C + \Delta C) \\
\tilde{Q}_{E,2}(C + \Delta C) \\
0_{(n+\tilde{r}_E)\times 2n}
\end{pmatrix}
\begin{pmatrix}
\mathbb{I}_{n} & 0 & 0 \\
0 & \tilde{Q}_{E,1}^{*} & 0 \\
0 & 0 & \tilde{Q}_{E,2}^{*}
\end{pmatrix}
\rightarrow
\begin{pmatrix}
-A \\
0 \\
0_{(n+\tilde{r}_E)\times 2n}
\end{pmatrix}
\begin{pmatrix}
-A \\
0 \\
0_{(n+\tilde{r}_E)\times 2n}
\end{pmatrix}
\begin{pmatrix}
Q_{E,1}^{*}(C + \Delta C) \\
Q_{E,2}^{*}(C + \Delta C) \\
0_{n}
\end{pmatrix}
\begin{pmatrix}
-Q_{E,1}^{*} \\
-Q_{E,2}^{*} \\
0
\end{pmatrix}
\begin{pmatrix}
0_{n+\tilde{r}_E} \\
0_{n+\tilde{r}_E}
\end{pmatrix}.
\]

Recall, $\tilde{Y} = \text{computed}(\tilde{Q}_{E}^{*}C) = \tilde{Q}_{E}^{*}(C + \Delta C)$; introduce block-row partition $\tilde{Y} = \par{\tilde{Y}_1 \tilde{Y}_2}$ with $\tilde{Y}_2 = \tilde{Q}_{E,2}(C + \Delta C)$. Similarly, introduce block-column partitions $\tilde{Q}_{A_{22}} = (\tilde{\Omega}_1 \tilde{\Omega}_2)$, $\tilde{Q}_{A_{22}}^{*} = (\tilde{\Omega}_1^{*} \tilde{\Omega}_2^{*})$. The last column block $\tilde{N}_{[4]}$ in the matrix
\[
\text{computed}(\tilde{Q}_{A_{22}}^{*}) = \begin{pmatrix}
-\tilde{Q}_{E,2}^{*}(C + \Delta C) \\
0_{n}\tilde{r}_E \times n \\
0_{n}\tilde{r}_E \times n
\end{pmatrix}
\text{computed}(\tilde{Q}_{A_{22}}^{*}) = \begin{pmatrix}
-\tilde{N}_{[1]} \\
-\tilde{N}_{[2]} \tilde{N}_{[3]} \\
\tilde{N}_{[4]}
\end{pmatrix}
\]

is simply $-\tilde{\Omega}_2$. Since we used $\tilde{Q}_{A_{22}}$ in the backward error analysis of the left-hand matrix, here we will have to use a mixed error analysis: $-\tilde{\Omega}_2$ will be changed by a forward error into $-\tilde{\Omega}_2$. Recall that our model of the analysis (using exactly unitary instead of the computed numerically unitary matrices) will also require small forward perturbation to change $-\tilde{Q}_{E,1}^{*}$ into $-\tilde{Q}_{E,1}$.

\(^6\text{See Example 2.4}\)
Consider now the first two blocks in $\widetilde{N}$.

$$\widetilde{N}_{[1]} = \text{computed}(\tilde{\Omega}_1 \tilde{Y}_2) = \tilde{\Omega}_1 \tilde{Y}_2 + \delta \tilde{N}_{[1]} = \tilde{\Omega}_1 \tilde{Y}_2 + \delta \tilde{\Omega}_1 \tilde{Y}_2 + \delta \tilde{N}_{[1]}, \quad |\delta \tilde{N}_{[1]}| \leq \epsilon |\tilde{\Omega}_1| |\tilde{Y}_2|$$  \hspace{1cm} (4.11)

In this block too we will commit a forward error and replace it with

$$\tilde{Q}^*_{A_{22}} \left( \tilde{Q}^*_{E,2} (C + \Delta C) \right)_{0_{\tilde{F}_E \times n}} = \tilde{\Omega}_1 \tilde{Y}_2 = \tilde{N}_{[1]} - \Delta \tilde{N}_{[1]}, \quad \Delta \tilde{N}_{[1]} = \delta \tilde{\Omega}_1 \tilde{Y}_2 + \delta \tilde{N}_{[1]}$$  \hspace{1cm} (4.12)

To estimate this forward change we first note that

$$\|\tilde{Y}_2(:,i)\|_2 \leq \frac{\|\tilde{N}_{[1]}(:,i)\|_2}{1 - \|\delta \tilde{\Omega}_1\|_2 - \epsilon \|\tilde{\Omega}_1\|_2}$$

Hence

$$\|\delta \tilde{\Omega}_1 \tilde{Y}_2(:,i)\|_2 \leq \frac{\|\delta \tilde{\Omega}_1\|_2 \|\tilde{N}_{[1]}(:,i)\|_2}{1 - \|\delta \tilde{\Omega}_1\|_2 - \epsilon \|\tilde{\Omega}_1\|_2}, \quad \|\delta \tilde{N}_{[1]}(:,i)\|_2 \leq \frac{\epsilon \|\tilde{\Omega}_1\|_2 \|\tilde{N}_{[1]}(:,i)\|_2}{1 - \|\delta \tilde{\Omega}_1\|_2 - \epsilon \|\tilde{\Omega}_1\|_2}$$  \hspace{1cm} (4.13)

and we conclude that $\Delta \tilde{N}_{[1]}$ is a column-wise small perturbation of $\tilde{N}_{[1]}$. Computation of $\tilde{N}_{[2]} = \text{computed}(\tilde{\Omega}_1 \tilde{Q}^*_{E,2})$ is analogous, but for the purpose of mixed stability interpretation, $\tilde{Q}^*_{E,2}$ has to be replaced with $\tilde{Q}^*_{E,2} = \tilde{Q}^*_{E,2} - \delta \tilde{Q}^*_{E,2}$, which yields

$$\tilde{N}_{[2]} = \tilde{\Omega}_1 \tilde{Q}^*_{E,2} + \tilde{\Omega}_1 \delta \tilde{Q}^*_{E,2} + \underbrace{\delta \tilde{\Omega}_1 \tilde{Q}^*_{E,2}}_{\delta \tilde{\Omega}_1 \tilde{Q}^*_{E,2}} + \delta \tilde{N}_{[2]}, \quad |\delta \tilde{N}_{[2]}| \leq \epsilon |\tilde{\Omega}_1| |\tilde{Q}^*_{E,2}|$$  \hspace{1cm} (4.14)

Hence, if the computed right-hand matrix is changed by a forward perturbation as

$$\begin{pmatrix}
-A & 0 & 0 & 0_n + \tilde{r}_E \\
-\tilde{Q}^*_{E,1} (C + \Delta C) & -\tilde{Q}^*_{E,1} & 0 & 0_{\tilde{F}_E \times (n + \tilde{r}_E)} \\
0_n & 0_n & -I_{n} & 0_{n \times \tilde{r}_E} \\
-\tilde{N}_{[1]} & -\tilde{N}_{[2]} & N_{[3]} & N_{[4]} \\
\tilde{Q}^*_{A_{22}} (C + \Delta C) /_{\tilde{F}_E \times n} & \tilde{Q}^*_{A_{22}} (C + \Delta C) /_{\tilde{F}_E \times n} & \tilde{Q}^*_{A_{22}} (C + \Delta C) /_{\tilde{F}_E \times n} & \tilde{Q}^*_{A_{22}} (C + \Delta C) /_{\tilde{F}_E \times n}
\end{pmatrix}
+ \begin{pmatrix}
0 & 0 & 0_n + \tilde{r}_E \\
0 & -\delta \tilde{Q}^*_{E,1} & 0_n & 0_{\tilde{F}_E \times (n + \tilde{r}_E)} \\
0_n & 0_n & -I_n & 0_{n \times \tilde{r}_E} \\
\tilde{N}_{[3]} & \tilde{N}_{[2]} & 0 & \Delta \tilde{N}_{[4]}
\end{pmatrix}, \quad (4.15)
$$

the resulting matrix is the $(3n + \tilde{r}_E) \times (3n + \tilde{r}_E)$ main submatrix of

$$\begin{pmatrix}
1_{2n + \tilde{r}_E} & 0 & 0 & 0 & 0 & 0_{\tilde{N}_{[1]} - \tilde{r}_E} \\
0 & \tilde{Q}^*_{A_{22}} & 0 & 0 & 0 & 0_{\tilde{N}_{[1]} - \tilde{r}_E} \\
0 & 0 & 1_{n - \tilde{r}_E} & 0 & 0 & 0_{\tilde{N}_{[1]} - \tilde{r}_E}
\end{pmatrix} \begin{pmatrix}
\tilde{I}_n & 0 & 0 & 0 & 0 & 0_{\tilde{N}_{[1]} - \tilde{r}_E} \\
0 & \tilde{Q}^*_{E,1} & 0 & 0 & 0 & 0_{\tilde{N}_{[1]} - \tilde{r}_E} \\
0 & 0 & \tilde{Q}^*_{E,1} & 0 & 0 & 0_{\tilde{N}_{[1]} - \tilde{r}_E} \\
0 & 0 & 0 & -\tilde{A} & 0_n & 0_n \\
0 & 0 & 0 & 0_n & -\tilde{A} & 0_n \\
0 & 0 & 0 & 0_n & 0_n & -\tilde{A}
\end{pmatrix}
\begin{pmatrix}
\tilde{I}_n & 0 & 0 & 0 & 0 & 0_{\tilde{N}_{[1]} - \tilde{r}_E} \\
0 & \tilde{Q}^*_{E,1} & 0 & 0 & 0 & 0_{\tilde{N}_{[1]} - \tilde{r}_E} \\
0 & 0 & \tilde{Q}^*_{E,1} & 0 & 0 & 0_{\tilde{N}_{[1]} - \tilde{r}_E} \\
0 & 0 & 0 & -\tilde{A} & 0_n & 0_n \\
0 & 0 & 0 & 0_n & -\tilde{A} & 0_n \\
0 & 0 & 0 & 0_n & 0_n & -\tilde{A}
\end{pmatrix} \end{pmatrix}, \quad (4.16)
5 Numerical examples

In this section, we provide numerical examples and compare our new algorithm KVARTeig with polyeig from MATLAB, and quadeig and KVADeig applied to the quadratification (1.4). Our goal is to illustrate the potential of the techniques introduced in KVADeig and KVARTeig, and to motivate further development. All test examples are taken from the NLEVP benchmark collection [2].

Example 5.1. We first test KVARTeig on three examples with the default input values: butterfly (n = 64); orr_sommerfeld (n = 64); planar waveguide (n = 129). The results are tested using the norm-wise backward error [15, §2.2] (the component-wise backward errors

\[ \eta(\lambda, x) = \frac{\| (\lambda^4 A + \lambda^3 B + \lambda^2 C + \lambda D + E)x \|_2}{\| \lambda^4 A \|_2 + \| \lambda^3 B \|_2 + \| \lambda^2 C \|_2 + \| \lambda D \|_2 + \| E \|_2 \| x \|_2} ; \eta(\infty, x) = \frac{\| Ax \|_2}{\| A \|_2 \| x \|_2}. \]

(5.1)

The extreme values of \( \eta(\lambda, x) \) over all right eigenpairs are summarized in Table 1.

| Algorithm         | butterfly | orr_sommerfeld | planar waveguide |
|-------------------|-----------|----------------|------------------|
| polyeig           | min \( \eta \) | max \( \eta \) | min \( \eta \) | max \( \eta \) | min \( \eta \) | max \( \eta \) |
| quadeig           | 2.0432e-016 | 8.6189e-016 | 1.3618e-017 | 8.0176e-006 | 1.6060e-016 | 3.0879e-012 |
| KVADeig           | 6.5690e-017 | 2.0389e-015 | 6.1163e-015 | 4.0733e-004 | 4.9977e-016 | 2.0389e-015 |
| balanced KVADeig  | 6.5690e-017 | 2.0389e-015 | 6.2547e-021 | 2.1200e-007 | 4.7557e-016 | 1.4994e-016 |
| KVARTeig          | 5.8416e-017 | 1.5776e-015 | 6.3789e-021 | 1.7600e-015 | 6.3789e-021 | 1.7600e-015 |

(KVADeig and balanced_KVADeig refer to the modification of quadeig in [7].)

In this experiment, only KVARTeig has received scaled matrix coefficients, with the parameter scaling as described in [2,4] polyeig, quadeig and both variants of KVADeig worked with raw data A, B, C, D, E, and the parameter scaling is applied in quadeig, KVADeig only to the quadratic pencil \( \lambda^2 M + \lambda C + K \) from (1.4). This may serve as a simulation of a genuine quadratic problem in which the coefficients are composed of blocks with different parameter dependencies, possibly on different scales – then parameter scaling cannot resolve different scales inside \( M, C, K \). Hence, this example is primarily a test of the quadratic solvers as potential tools for quadratification based solution of quartic problems.

The noticeable difference in the worst case of the backward errors in the orr_sommerfeld example motivated a closer look and the next experiment. Recall, the function from the NLEVP library for generating this quartic eigenvalue problem has three optional input arguments, \( n, \omega, R \): \( n \) represents the dimension of the problem, \( \omega \) is the frequency, and \( R \) is Reynolds number. The default values are: \( n = 64, \omega = 0.26943 \) and \( R = 5772 \) (these values are used in Table 1).

Example 5.2. Structured backward errors provide a higher resolution insight into the numerical quality of the computed solutions. For the data of orr_sommerfeld in Example 5.1, we compute the component-wise backward errors

\[ \omega(\lambda, x) = \min \{ \epsilon : (\lambda^4 \tilde{A} + \lambda^3 \tilde{B} + \lambda^2 \tilde{C} + \lambda \tilde{D} + \tilde{E})x = 0, |\delta A| \leq \epsilon |A|, \ldots, |\delta E| \leq \epsilon |E| \} = \max_{i=1,3} \frac{\| (\lambda^4 A + \lambda^3 B + \lambda^2 C + \lambda D + E)x_i \|_2}{\| (\lambda^4 |A| + \lambda^3 |B| + \lambda^2 |C| + |D| + |E|)|x_i|}$

(5.2)

Planar waveguide

We used polyeig as provided in Matlab.
The values of $\omega(\lambda, x)$ are shown in Figure 2. The advantage of KVARTeig over the other methods is now apparent. On the left panel, we can also clearly see the benefits of our modification KVADeig of quadeig, introduced recently in [7]; however the right panel shows the clear advantage of KVARTeig.

![Figure 2](image2.png)

Figure 2: (Example 5.2) Component-wise backward errors for all eigenpairs.

**Example 5.3.** Now, for a stress test, we increase the Reynolds number to $R = 10000$. The norm-wise and the component-wise backward errors for all eigenvalues (listed increasingly in modulus), are shown in Figure 3 and Figure 4, respectively. Note how the backward error for KVARTeig in Figure 3 remains nearly flat at the roundoff level, and how KVADeig also performs well (even with structured backward error for the right eigenpairs), despite being oblivious of the underlying structure of the quadratification.

![Figure 3](image3.png)

Figure 3: (Example 5.3) The norm-wise backward errors for all eigenpairs.
Remark 5.1. The results of this experiment, with the computed backward errors shown in Figure 3 and Figure 4, are instructive. First, in this example quadeig deflated 64 infinite eigenvalues of the quadratic pencil $\lambda^2 M + \lambda C + K$ (see (1.4)), because in the preprocessing stage of the algorithm, the numerical rank of the matrix $M = \begin{pmatrix} A & 0 \\ C & I_n \end{pmatrix}$ of order 128 is computed as 64. On the other hand, the existence of infinite eigenvalues in the original quartic eigenvalue problem depends on the rank of the leading coefficient matrix $A$. If we inspect the singular values of $M$ and $A$, then, as clearly shown in Figure 5, $A$ is numerically of full rank (its condition number is below $10^6$, so KVARTeig safely removed the possibility of infinite eigenvalues) and $M$ is indeed numerically rank deficient (relative to a tolerance of the order of the roundoff unit). Note that parameter scaling of the quadratic pencil (1.4) cannot remove this problem.

However, KVADeig (applied to $\lambda^2 M + \lambda C + K$) declared the matrix $M$ nonsingular, and thus no infinite eigenvalues where deflated nor found by the QZ algorithm. This is because KVADeig uses more local truncation strategy, see Remark 2.2. Good results by balanced_KVADeig are due to balancing [7, §4.2], and this example once more justifies our approach in KVADeig (using local drop-off truncation strategy and balancing in combination with parameter scaling).

Example 5.4. This example illustrates the well known importance of parameter scaling. In particular, it shows how the performance of polyeig can be substantially improved. We use the same benchmark problem as in Example 5.3, but initially we scale the matrices as described in [21] so that all algorithms start with scaled data.
Example 5.5. We continue experimenting with the *orr_sommerfeld* example; we choose the default values of the Reynolds number $R$ and the frequency $\omega$, but increase the dimension to $n = 1000$, and compute all 4000 eigenpairs.

Without parameter scaling of the initial data, the Matlab function `polyeig` failed completely – all computed eigenvalues were of the form $\pm \text{Inf} \pm \text{Inf}$. An application of `quadeig` to the quadratification (1.4) returned 1144 infinite eigenvalues. With `KVADeig` and the same quadratification, 2912 infinite eigenvalues are detected. However, if we use balancing (`balanced_KVADeig`), the leading coefficient matrix is declared regular, and no infinite eigenvalues are detected.

The result of `KVARTeig` depends on the truncation strategy. If the truncation of the pivoted QR factorization is done relative to the norm of $A$, the numerical rank is 988, meaning that 12 infinite eigenvalues are deflated immediately in the preprocessing phase. In the case of drop-off strategy, the matrix $A$ is not numerically rank deficient. The singular values of the matrix $A$ and the absolute values of the diagonal entries of the triangular factor (from the rank revealing factorization (2.1)) are presented in Figure 8. The norm-wise and component-wise backward errors for the computed right and left eigenpairs are present in the Figures 9, 10, 11, 12.
Figure 8: (Example 5.5) Left panel: The singular values of $A$ and the absolute values of the rank revealing triangular factor. Right panel: The ratios $\sigma_1(A)/\sigma_i(A)$ and $1/\varepsilon$, where $\varepsilon$ is the machine precision ($\text{eps}$) in Matlab.

Figure 9: (Example 5.5) Norm-wise backward errors for the right eigenpairs.
Figure 10: (Example 5.5) Norm-wise backward errors for the left eigenpairs.

Figure 11: (Example 5.5) Component-wise backward errors for the right eigenpairs.

In the case of scaled coefficient matrices, \texttt{quadeig} detected 282 infinite eigenvalues, \texttt{KVADeig} 31, and balanced \texttt{KVADeig} 18.
Example 5.6. In this example, we use transposed matrices from the mirror example and
calculate the number of zero and infinite eigenvalues found by the four algorithms were:
polyeig (no zeros and 5 infinities); quadeig (7 zeros and 9 infinities); KVADeig and
KVARTeig 9 zeros and 9 infinities. The component-wise backward errors are given
in Figure 13.

Example 5.7. In this example we illustrate potential benefits of equilibrating the coefficient
matrices on the element level, mentioned in Remark 2.7. Such diagonal scalings balance the
absolute values of nonzero entries over all matrices.

We take the butterfly example and pre-multiply its coefficient matrices with diagonal ma-
trix ∆ with randomly permuted powers $2^i$, $i = 1, \ldots, n = 64$ on the diagonal. This is an entirely
artificial step to simulate a situation with ill-conditioning caused by removable scaling (that may

---

Figure 12: (Example 5.5) Component-wise backward errors for the left eigenpairs.

Figure 13: (Example 5.6) Left panel: Backward error for the transposed mirror example. Right
panel: Backward error for the original mirror example (This is the right panel from Figure 1
here given for comparison.)
originating in an inappropriate scale of physical units. We obtain an equivalent problem, but numerical algorithms may be more or less sensitive to this change of representation.

Then, we compute balancing matrices $\Delta_L$, $\Delta_R$ (see Remark 2.1) and examine how this preprocessing $((A, B, C, D, E) \rightsquigarrow \Delta_L(A, B, C, D, E)\Delta_R)$ influences the numerical accuracy of the algorithms under study. The results are shown in Figures 14 and 15.

**Figure 14:** (Example 5.7, butterfly.) Component-wise backward errors for the modified butterfly example, where the coefficients are premultiplied by a diagonal matrix $\Delta$, $(A, B, C, D, E) \rightsquigarrow \Delta(A, B, C, D, E)$.

The graphs clearly demonstrate the impact of the balancing $(A, B, C, D, E) \rightsquigarrow \Delta_L(A, B, C, D, E)\Delta_R$. Note also that without balancing $KVADeig$ still performs well, much better than $polyeig$ and $quadeig$ under the same conditions.

**Figure 15:** (Example 5.7, modified butterfly.) Left panel: The computed spectrum of the original butterfly example. Right panel: The spectrum of an equivalent representation, where the coefficients are premultiplied by a diagonal matrix $\Delta$, $(A, B, C, D, E) \rightsquigarrow \Delta(A, B, C, D, E)$. Note the spurious eigenvalues computed by $polyeig$ and $quadeig$.

**Example 5.8.** In our last example, we checked the least squares approach to recovering the eigenvectors, as described in §3.1.2. The computed backward error in all tested cases was comparable with the method of selecting the vector with smallest residual.
6 Concluding remarks

We have shown that our algorithm KVARTeig for solving quartic eigenvalue problems is a useful contribution that fills the gap in the toolbox for the polynomial eigenvalue problems, both for the full solution of medium size non-structured problems and for solving the projected problems in subspace based methods for large scale structured/sparse problems. Numerical experiments with the benchmark examples from the NLEVP collection show that KVARTeig is superior to polyeig from Matlab, or quadeig applied to a quadratification of the original quartic problem. Further, the numerical performances of KVADeig on the quadratification of the quartic problem additionally justify the modifications that underpinned the development both in [7] and in this paper.

Given the wide spectrum of applications of the quartic eigenvalue problem, we are certain that our proposed algorithm will prove useful in many computational tasks in applied sciences and engineering. LAPACK–style implementations of both new algorithms are in progress.

Acknowledgement

This research was part of the second authors thesis [18] and of the project IP-11-2013-9345 (Mathematical Modelling, Analysis and Computing With Applications to Complex Mechanical Systems) supported by the Croatian Science Foundation. The authors thank Serkan Gugercin (Virginia Tech), Luka Grubišić and Zvonimir Bujanović (University of Zagreb) for valuable comments.

References

[1] Timo Betcke. Optimal scaling of generalized and polynomial eigenvalue problems. SIAM Journal on Matrix Analysis and Applications, 30(4):1320–1338, 2008.

[2] Timo Betcke, Nicholas J. Higham, Volker Mehrmann, Christian Schröder, and Françoise Tisseur. Nlevp: A collection of nonlinear eigenvalue problems. ACM Transactions on Mathematical Software (TOMS), 39(2):7, 2013.

[3] Carmen Campos and Jose E. Roman. Parallel krylov solvers for the polynomial eigenvalue problem in slepc. SIAM Journal on Scientific Computing, 38(5):S385–S411, 2016.

[4] Hongjia Chen, Akira Imakura, and Tetsuya Sakurai. Improving backward stability of sakurai-sugiura method with balancing technique in polynomial eigenvalue problem. Applications of Mathematics, 62(4):357–375, Aug 2017.

[5] Gökhan Danabasoglu and Sedat Biringen. A chebyshev matrix method for the spatial modes of the orrÅŞsommerfeld equation. International Journal for Numerical Methods in Fluids, 11:1033 – 1037, 11 1990.

[6] Fernando De Terán, Froilán M Dopico, and D Steven Mackey. Spectral equivalence of matrix polynomials and the index sum theorem. Linear Algebra and its Applications, 459:264–333, 2014.

[7] Zlatko Drmač and Ivana Šain Glišić. New numerical algorithm for deflation of infinite and zero eigenvalues and full solution of quadratic eigenvalue problems. arXiv e-prints, page arXiv:1904.05418, Apr 2019.
[8] Hung-Yuan Fan, Wen-Wei Lin, and Paul Van Dooren. Normwise scaling of second order polynomial matrices. *SIAM journal on matrix analysis and applications*, 26(1):252–256, 2004.

[9] Brendan Gavin, Agnieszka Miedlar, and Eric Polizzi. FEAST eigensolver for nonlinear eigenvalue problems. *Journal of Computational Science*, 27:107–117, jul 2018.

[10] Sven Hammarling, Christopher J. Munro, and Françoise Tisseur. An algorithm for the complete solution of quadratic eigenvalue problems. *ACM Transactions on Mathematical Software (TOMS)*, 39(3):18, 2013.

[11] Nicholas J. Higham, Ren-Cang Li, and Françoise Tisseur. Backward error of polynomial eigenproblems solved by linearization. *SIAM journal on matrix analysis and applications*, 29(4):1218–1241, 2007.

[12] Volker Mehrmann and David S Watkins. Polynomial eigenvalue problems with Hamiltonian structure. *Electronic Transactions on Numerical Analysis*, 13:106–118, 2002.

[13] Tetsuya Sakurai and Hiroshi Sugiura. A projection method for generalized eigenvalue problems using numerical integration. *Journal of Computational and Applied Mathematics*, 159(1):119 – 128, 2003. 6th Japan-China Joint Seminar on Numerical Mathematics; In Search for the Frontier of Computational and Applied Mathematics toward the 21st Century.

[14] David Stowell and Johannes Tausch. Variational formulation for guided and leaky modes in multilayer dielectric waveguides. *Communications in Computational Physics*, 7(3), 01 2009.

[15] Françoise Tisseur. Backward error and condition of polynomial eigenvalue problems. *Linear Algebra and its Applications*, 309(1-3):339–361, 2000.

[16] Istvan A. Veres, Thomas Berer, and Osamu Matsuda. Complex band structures of two dimensional phononic crystals: Analysis by the finite element method. *Journal of Applied Physics*, 114(8):083519, 2013.

[17] R.F. Vieira, F.B. Virtuoso, and E.B.R. Pereira. A higher order model for thin-walled structures with deformable cross-sections. *International Journal of Solids and Structures*, 51(3):575 – 598, 2014.

[18] I. Šain Glišić. *Robust numerical methods for nonlinear eigenvalue problems*. PhD thesis, University of Zagreb, Faculty of Science, Department of Mathematics, 12 2018.

[19] Yong Xiao, Jihong Wen, Lingzhi Huang, and Xisen Wen. Analysis and experimental realization of locally resonant phononic plates carrying a periodic array of beam-like resonators. *Journal of Physics D: Applied Physics*, 47:045307, 01 2014.

[20] Shinnosuke Yokota and Tetsuya Sakurai. A projection method for nonlinear eigenvalue problems using contour integrals. *JSIAM Letters*, 5:41–44, 2013.

[21] H. Zha. The restricted singular value decomposition of matrix triplets. *SIAM Journal on Matrix Analysis and Applications*, 12(1):172–194, 1991.
[22] B. Zhang and Y. F. Li. A method for calibrating the central catadioptric camera via homographic matrix. In 2008 International Conference on Information and Automation, pages 972–977, June 2008.