A MULTISHIFT, MULTIPOLE RATIONAL QZ METHOD WITH AGGRESSIVE EARLY DEFLATION*

THIJS STEEL†, DAAN CAMPS‡, KARL MEERBERGEN†, AND RAF VANDEBRIL†

Abstract. The rational QZ method generalizes the QZ method by implicitly supporting rational subspace iteration. In this paper we extend the rational QZ method by introducing shifts and poles of higher multiplicity in the Hessenberg pencil, which is a pencil consisting of two Hessenberg matrices. The result is a multishift, multipole iteration on block Hessenberg pencils which allows one to stick to real arithmetic for a real input pencil. In combination with optimally packed shifts and aggressive early deflation as an advanced deflation technique we obtain an efficient method for the dense generalized eigenvalue problem. In the numerical experiments we compare the results with state-of-the-art routines for the generalized eigenvalue problem and show that we are competitive in terms of speed and accuracy.

Key words. generalized eigenvalues, rational QZ, rational Krylov, multishift, multipole, aggressive early deflation, Fortran implementation

AMS subject classifications. 65F15, 15A18

1. Introduction.

The rational QZ method (RQZ) [10] generalizes the standard QZ method of Moler & Stewart [21]. Both are methods for the numerical solution of the dense, unsymmetric generalized eigenvalue problem for a pencil of matrices $A, B \in \mathbb{F}^{n\times n}$, $\mathbb{F} \in \{\mathbb{C}, \mathbb{R}\}$. The set of eigenvalues of the pencil (A, B) is denoted as Λ and defined by,

$$\Lambda = \{\lambda = \alpha/\beta \in \bar{\mathbb{C}} : \det(\beta A - \alpha B) = 0\}, \tag{1}$$

with $\bar{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$. Eigenvalues with $\beta = 0$ are located at ∞. We assume throughout this paper that the pencil (A, B) is *regular* which means that its characteristic polynomial differs from zero. This implies that there are exactly n eigenvalues including those at ∞.

The RQZ method acts on pencils in Hessenberg, Hessenberg form instead of the Hessenberg, triangular form used in the QZ method. For simplicity we will write *Hessenberg pencil* when we mean a pencil with two matrices in Hessenberg form. This is an advantage if, e.g., the input is already in this form, like in rational Krylov methods. It relies on *pole swapping* instead of *bulge chasing*. Pole swapping was introduced by Berljafa and Güttel [3] to reorder the poles in rational functions used to generate Krylov subspaces, this is done by acting on the associated Hessenberg pencil. This technique generalizes the implicitly restarted Arnoldi approach of Sorensen [22] to the rational Krylov setting [10]. Bulge chasing can be considered as a special instance of pole swapping: all poles are taken equal to infinity, allowing for an alternative implementation where bulges are chased rather than poles are swapped. A thorough analysis of the differences between bulge chasing and pole swapping, and the advantages and increased flexibility of the pole swapping approach can be found in

*Submitted to the editors March 12, 2019.

Funding: The research was partially supported by the Research Council KU Leuven, projects C14/16/056 (Inverse-free Rational Krylov Methods: Theory and Applications), OT/14/074 (Numerical algorithms for large scale matrices with uncertain coefficients).

†Department of Computer Science, KU Leuven, University of Leuven, 3001 Leuven, Belgium. (thijs.steel@cs.kuleuven.be, karl.meerbergen@kuleuven.be, raf.vandebril@kuleuven.be)

‡Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA, USA (dcamps@lbl.gov)

Camps, Watkins, Vandebril, and Mach [11]. We elude a little on the similarities and differences in Section 2. It is, for instance, also illustrated by Camps et al. [11], that the natural manner of getting optimally packed poles in the pole swapping setting is identical to the technique of Karlsson, Kressner, and Lang [18] of delaying some transformations to pack bulges as close as possible together.

The RQZ method [10] computes the generalized Schur form of (A, B), namely

$$(S, T) = Q^* (A, B) Z, \tag{2}$$

such that (S, T) is a triangular, triangular pencil unitarily equivalent to (A, B). The eigenvalues of (A, B) are readily available as the ratios s_{ii}/t_{ii} of the diagonal elements of S and T. Both the single shift RQZ method and the RQZ method with optimally packed shifts, as formulated in [10], are applicable to real- and complex-valued pencils. However, it requires complex arithmetic for real-valued pencils having complex conjugate eigenvalues.

In this paper we introduce the multishift, multipole RQZ method which acts on pencils where both matrices are in *block Hessenberg* form. Though, the theory, as presented here holds for multishifts of any size, we have only implemented a Fortran version for multishifts and multipoles of size 2. We do so because larger numbers of shifts typically lead to shift blurring [25], with delayed convergence as a consequence. The main benefit of using shifts and poles of higher multiplicity (2 is sufficient) is that complex conjugate pairs of shifts and poles can be represented in real arithmetic for real-valued pencils. This is similar to the well-known implicit double-shift QR step introduced by Francis [14] and the double-shift QZ step [21]. The main advantages of the results in this paper are thus for the case $\mathbb{F} = \mathbb{R}$. The multishift, multipole RQZ method no longer converges to a triangular, triangular pencil (2); Instead, for $A, B \in \mathbb{R}^{n\times n}$, it will converge to the real generalized Schur form,

$$(S, T) = Q^T (A, B) Z = \left(\begin{bmatrix} S_{11} & S_{12} & \dots & S_{1m} \\ 0 & S_{22} & \ddots & S_{2m} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & S_{mm} \end{bmatrix}, \begin{bmatrix} T_{11} & T_{12} & \dots & T_{1m} \\ 0 & T_{22} & \ddots & T_{2m} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & T_{mm} \end{bmatrix} \right), \tag{3}$$

where the diagonal subpencils (S_{ii}, T_{ii}), $i = 1, \dots, m$ are of dimension 1×1 and 2×2 and correspond to respectively the real and complex conjugate eigenvalues of (A, B).

The paper is organized as follows. In Section 2 we briefly refresh the rational QZ algorithm. In Section 3 we define the block Hessenberg pencils on which we will be running our algorithm. We will also characterize essential uniqueness of these block Hessenberg pencils in Theorem 3.8 by linking block Hessenberg pencils via unitary equivalences to Hessenberg pencils. The basic algorithm, where blocks containing the shifts or poles are being swapped along the diagonal, is introduced in Section 4. Section 5 discusses aggressive early deflation for the multishift algorithm. Some particular properties, such as numerically reliable swapping, deflation heuristics, some parameters for the blocking, and exploiting initial sparsity of the algorithm are presented in Section 6, after which we present the full algorithm. Numerical experiments are shown in Section 7. The algorithm is implemented as part of the Fortran package `libRQZ` which is made publicly available at numa.cs.kuleuven.be/software/rqz and github.com/thijssteel/multishift-multipole-rqz. The convergence and the link with rational Krylov is examined in Section 8. We conclude the paper in Section 9.

Notation. With capital letters $A, B, \ldots$ we denote matrices, $\boldsymbol{a}, \boldsymbol{b}, \ldots$ are vectors, and $\alpha, \beta, \ldots$ scalars. Subspaces are denoted with calligraphic letters. For example, $\mathcal{R}(A) = \mathcal{R}(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n)$ is the column space of $A = [\boldsymbol{a}_1 \ldots \boldsymbol{a}_n]$, $\mathcal{E}_k = \mathcal{R}(\boldsymbol{e}_1, \ldots, \boldsymbol{e}_k)$ with $\boldsymbol{e}_j$ the jth canonical basis vector of appropriate dimension. The kth order polynomial Krylov subspace generated by A from starting vector $\boldsymbol{v}$ is $\mathcal{K}_k(A, \boldsymbol{v}) = \mathcal{R}(\boldsymbol{v}, A\boldsymbol{v}, \ldots, A^{k-1}\boldsymbol{v})$. The tuple (α, β) is ordered and $\{\alpha, \beta\}$ denotes an unordered multiset with repetition; we need this to identify the poles in the block Hessenberg pencils. We have already used the complex plane extended with the point at infinity, $\bar{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$ and division of a nonzero $\alpha \in \mathbb{C}$ by 0 results in infinity.

2. The single shift RQZ algorithm. We summarize the flow of the QZ - [26] and the single shift rational QZ algorithm [10]; identify similarities and differences.

The conventional QZ algorithm is based on bulge chasing. To initiate a QZ step, a cleverly chosen unitary transformation Q is applied to A and B. This transformation perturbs the Hessenberg, triangular structure creating what is called the bulge [26]. The next steps of the algorithm consist of restoring the structure by applying unitary equivalence transformations that chase the bulge downwards. When the bulge slides off the pencil, a step is completed. If the shifts are chosen well, a deflatable eigenvalue appears in the bottom-right corner after a few QZ steps. In understanding the QZ algorithm we exploit the link between Hessenberg, triangular pencils and polynomial Krylov subspaces.

The RQZ algorithm is an extension of the QZ algorithm that acts on Hessenberg pencils instead; here there are rational Krylov subspaces lurking behind. To define a rational Krylov subspace a rational function with a prescribed set of poles is required; these poles are encoded as ratios of the subdiagonal elements in the Hessenberg pencil[1]. More details on the connection with Krylov and an analysis of the theory for the multishift multipole setting discussed in this paper are found in Section 8; The single shift rational QZ is discussed by Camps et al. [10].

Consider now a Hessenberg pencil. Instead of introducing a bulge, we replace the pole (ratio of first subdiagonal elements) by another pole; if the context requires it, we name this pole the shift-pole to emphasize where it comes from. Next, we chase the shift-pole downward by continuously swapping the shift-pole with the pole next to it thereby moving it to the bottom of the pencil[2]. Once at the bottom of the pencil, the shift-pole is replaced with another pole and we are done. Again, well chosen shift-poles lead to convergence.

An advantage over classical bulge chasing algorithms is that by chasing the shift-pole to the bottom, all the other poles will have moved up one position. Although it can take many iterations, the poles we set at the bottom will eventually arrive at the top. This upward movement also drives convergence at the top of the pencil governed by the poles. Special cases, such as the possibility to stop in the middle of a chase (we still end up with a Hessenberg pencil) and intriguing connections between tightly packed poles and bulges are discussed by Camps et al. [11]; Mach et al. [20] use bidirectional chasing to solve palindromic eigenvalue problems by pole swapping.

3. Block Hessenberg pencils. In this section we define block Hessenberg pencils. We also specify properness, since proper block Hessenberg pencils serve as input to the rational multishift QZ algorithm. We do not impose constraints on the sizes

[1] Reconsidering a Hessenberg, triangular pencil, we note that all ratios will be infinity, hence the rational function becomes polynomial and we are in the QZ case.

[2] This technique is equivalent to reordering a generalized Schur factorization.

of the blocks, but the accompanying software operates on blocks of size two at most, as this allows us to stick to real arithmetic for real pencils. We can, however, not exclude that some applications might lead to a block Hessenberg pencil in a more general form, suitable for the rational multishift QZ algorithm instead of running an initial reduction.

3.1. Definitions and elementary results.

DEFINITION 3.1 (Block triangular matrix). *A matrix $R \in \mathbb{F}^{n\times n}$ is called a block triangular matrix*[3] *with partition $\boldsymbol{s} = (s_1, \ldots, s_m)$, $s_1 + \ldots + s_m = n$, if it admits the form,*

$$\begin{bmatrix} R_{11} & R_{12} & \ldots & R_{1m} \\ & R_{22} & \ldots & R_{2m} \\ & & \ddots & \vdots \\ & & & R_{mm} \end{bmatrix}, \tag{4}$$

with block R_{jk} of size $s_j \times s_k$ for $1 \leq j \leq k \leq m$. The vector $\boldsymbol{s}$ defines the sizes of the blocks and is called the partition vector. The partition can be explicitly denoted as $R_{\boldsymbol{s}}$.

We also use the notation $D_{\boldsymbol{s}} = \operatorname{diag}(D_{11}, D_{22}, \ldots, D_{mm})$ for block diagonal matrices. Further, notice that if $R_{\boldsymbol{s}}$ is a nonsingular block triangular matrix, $\hat{R}_{\boldsymbol{s}} = R_{\boldsymbol{s}}^{-1}$ is also block triangular with an identical partition $\boldsymbol{s}$.

The partition of a block Hessenberg matrix is defined by the partition of the block triangular matrix in its lower left corner.

DEFINITION 3.2 (Block Hessenberg matrix). *A matrix $H \in \mathbb{F}^{n\times n}$ is called a block Hessenberg matrix with partition $\boldsymbol{s} = (s_1, \ldots, s_m)$, $s_1 + \ldots + s_m = n-1$, if it admits the form,*

$$H_{\boldsymbol{s}} = \begin{bmatrix} \boldsymbol{h}_{11}^T & h_{12} \\ H_{21} & \boldsymbol{h}_{22} \end{bmatrix}, \tag{5}$$

with H_{21} an $(n-1)\times(n-1)$ block triangular matrix with partition $\boldsymbol{s}$, $\boldsymbol{h}_{11}$ and $\boldsymbol{h}_{22}$ vectors of length $n-1$, and h_{12} a scalar.

DEFINITION 3.3 (Block Hessenberg pencil). *The $n\times n$ pencil (A, B) is a block Hessenberg pencil with partition $\boldsymbol{s} = (s_1, \ldots, s_m)$ if both*

$$A = \begin{bmatrix} \boldsymbol{a}_{11}^T & a_{12} \\ A_{21} & \boldsymbol{a}_{22} \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} \boldsymbol{b}_{11}^T & b_{12} \\ B_{21} & \boldsymbol{b}_{22} \end{bmatrix}, \tag{6}$$

are block Hessenberg matrices with a coinciding partition. The block triangular pencil (A_{21}, B_{21}) in (6) is called the pole pencil *of (A, B). If the pole pencil is regular, the poles $\Xi(A, B)$ are defined as the eigenvalues of the pole pencil $\Lambda(A_{21}, B_{21})$.*

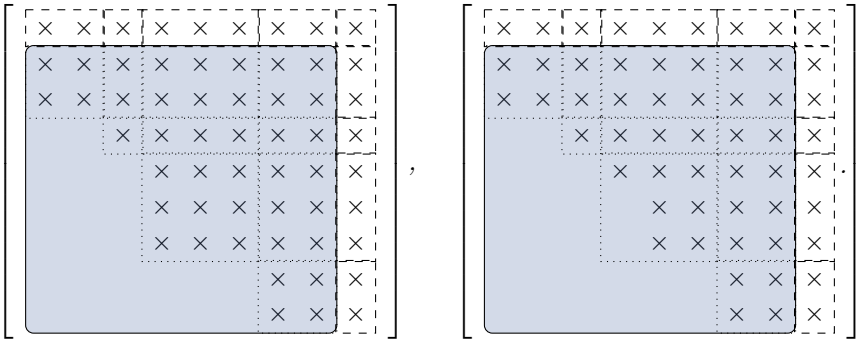
The ordering of the poles plays an important role in the convergence theory (see Section 8). Hence, for (A_{21}, B_{21}) from (6) we define the pole tuple as

$$\Xi(A, B) = \Lambda(A_{21}, B_{21}) = (\Xi^1, \ldots, \Xi^m) = (\{\xi_1^1, \ldots, \xi_{s_1}^1\}, \ldots, \{\xi_1^m, \ldots, \xi_{s_m}^m\}). \tag{7}$$

This imposes no specific ordering of the poles within a block, but the mutual ordering of the multisets of poles corresponds to the ordering of the blocks in (A_{21}, B_{21}).

[3] As we will only use upper triangular matrices, we omit the word *upper* for brevity.

EXAMPLE 3.4. *The pencil (A, B) is a 9×9 block Hessenberg pencil with partition $\boldsymbol{s} = (2, 1, 3, 2)$ if it has the form:*



The shaded part of the matrices is the pole pencil which is in the desired block triangular form. Note that matrix B also allows a partition $(2, 1, 1, 2, 2)$; but, we require a matching partition for both A and B and hence the third block (1×1) and the fourth block (2×2) are merged to form a single block of size 3×3. The pole tuple is $\Xi(A, B) = (\{\xi_1^1, \xi_2^1\}, \{\xi_1^2\}, \{\xi_1^3, \xi_2^3, \xi_3^3\}, \{\xi_1^4, \xi_2^4\})$, where ξ_1^1 and ξ_2^1 are the eigenvalues of the principal 2×2 block, and so forth.

Following Example 3.4, we note that block Hessenberg matrices and pencils can admit more than one partition. If (A, B) is a block Hessenberg pencil with partition $\boldsymbol{s} = (s_1, \ldots, s_k, s_{k+1}, \ldots, s_m)$, then it also admits the partition $\hat{\boldsymbol{s}} = (s_1, \ldots, s_k + s_{k+1}, \ldots, s_m)$; consecutive blocks can be grouped together.

We will prove in Section 8 that convergence will only take place in-between blocks. It is therefore better to have as many blocks as possible, meaning that none of the blocks should be splittable into smaller blocks. To stick to real arithmetic in the real case, we can restrict ourselves to blocks of sizes 1×1 and 2×2.

3.2. Properness and uniqueness. The final definition generalizes *properness* or *irreducibility* to a block Hessenberg pencil. Being proper guarantees that there are no obvious options for a deflation that splits the problem into smaller, independent problems.

DEFINITION 3.5. *A regular block Hessenberg pencil (A, B) with partition $\boldsymbol{s} = (s_1, \ldots, s_m)$ is said to be proper (or irreducible) if:*

I. Its pole pencil is regular;

II. The first block column of (A, B) of size $(s_1+1)\times s_1$,

$$\begin{bmatrix} \boldsymbol{a}_{1,1}^T \\ A_{2,1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{a}_1 & \ldots & \boldsymbol{a}_{s_1} \end{bmatrix}, \quad \begin{bmatrix} \boldsymbol{b}_{1,1}^T \\ B_{2,1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 & \ldots & \boldsymbol{b}_{s_1} \end{bmatrix}, \quad \boldsymbol{a}_i, \boldsymbol{b}_i \in \mathbb{F}^{s_1+1},$$

satisfies for $i = 1, \ldots, s_1$,

$$\mathcal{R}(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_i) \neq \mathcal{R}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_i);$$

III. The last block row of (A, B) of size $s_m\times(s_m+1)$,

$$\begin{bmatrix} A_{m+1,m} & \boldsymbol{a}_{m+1,m+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{a}_{s_m}^T \\ \vdots \\ \boldsymbol{a}_1^T \end{bmatrix}, \quad \begin{bmatrix} B_{m+1,m} & \boldsymbol{b}_{m+1,m+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_{s_m}^T \\ \vdots \\ \boldsymbol{b}_1^T \end{bmatrix}, \quad \boldsymbol{a}_i, \boldsymbol{b}_i \in \mathbb{F}^{s_m+1},$$

satisfies for $i = 1, \ldots, s_m$,

$$\mathcal{R}(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_i) \neq \mathcal{R}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_i).$$

Violating conditions II or III implies a deflation at either the top-left or bottom-right corner of the pencil. Condition I ensures there are no obvious deflations anywhere in the interior of the pencil. We illustrate Definition 3.5 by an example.

EXAMPLE 3.6. *Consider the* 4×4 *real-valued block Hessenberg pencil* (A, B) *with partition* $(2, 1)$ *given by:*

$$\begin{bmatrix} -0.300 & 0.075 & 0.500 & 0.250 \\ 0.395 & 0.520 & -0.350 & 2.000 \\ -0.140 & 0.860 & 1.350 & -0.800 \\ & & 1.000 & 0.850 \end{bmatrix}, \quad \begin{bmatrix} -0.150 & -0.600 & 0.150 & -1.500 \\ 0.160 & 0.940 & -5.000 & 1.350 \\ -0.120 & -0.080 & -2.400 & -1.000 \\ & & 0.200 & 1.800 \end{bmatrix}.$$

Condition I of Definition 3.5 is satisfied, the pole pencil is regular and the pole tuple of (A, B) *is given by:*

$$\Xi = (\{1.5 + i\sqrt{15/8}, 1.5 - i\sqrt{15/8}\}, 5). \tag{8}$$

Condition III of Definition 3.5 is also satisfied. For the last block row of (A, B)*, we clearly have that* $\mathcal{R}(\begin{bmatrix}1 & 0.85\end{bmatrix}^T) \neq \mathcal{R}(\begin{bmatrix}0.2 & 1.8\end{bmatrix}^T)$*. Notice that this implies that we cannot simultaneously create a zero in position* $(4, 3)$ *of both* A *and* B *by rotating the last two columns. The pencil is, however,* improper *as Condition II of Definition 3.5 is violated. We have* $\mathcal{R}(\boldsymbol{a}_1) \neq \mathcal{R}(\boldsymbol{b}_1)$*, but* $\mathcal{R}(\boldsymbol{a}_1, \boldsymbol{a}_2) = \mathcal{R}(\boldsymbol{b}_1, \boldsymbol{b}_2)$*. If we compute an orthonormal basis* Q_1 *of* $\mathcal{R}(\boldsymbol{a}_1, \boldsymbol{a}_2)$ *and extend this upto an orthonormal matrix* $Q = \begin{bmatrix}Q_1 & \boldsymbol{q}_2\end{bmatrix}$*, then* $(\hat{A}, \hat{B}) = Q^T(A, B)$ *has zero elements in positions* $(3, 1)$ *and* $(3, 2)$*. This allows us to split the problem into two submatrices and deflate the eigenvalues of the leading* 2×2 *subpencil, which are the complex conjugate poles in* (8)*.*

The next theorem generalizes the implicit Q theorem to block Hessenberg pencils. We define first what *essentially unique* means in the block formulation.

DEFINITION 3.7. *Two block Hessenberg pencils* $(\hat{A}, \hat{B})$ *and* $(\check{A}, \check{B})$ *with partitions* $\hat{\boldsymbol{s}}$ *and* $\check{\boldsymbol{s}}$ *respectively are said to be essentially identical if there exist equivalences with unitary block diagonal matrices such that*

$$\hat{D}_1^*(\hat{A}, \hat{B})\hat{D}_2 = \check{D}_1^*(\check{A}, \check{B})\check{D}_2,$$

where $\hat{D}_1$ *has partition* $(1, \hat{\boldsymbol{s}})$ *and* $\hat{D}_2$ *has partition* $(\hat{\boldsymbol{s}}, 1)$*; and* $\check{D}_1$ *and* $\check{D}_2$ *have partitions* $(1, \check{\boldsymbol{s}})$ *and* $(\check{\boldsymbol{s}}, 1)$*.*

Though not mentioned explicitly in the definition, we will see in the proof of Theorem 3.8 that one can transform two essentially identical block Hessenberg pencils to the same Hessenberg pencil. Moreover, suppose that we have a real block Hessenberg pencil then the pencil will always be essentially identical to a real block Hessenberg pencil having the B matrix of Hessenberg form; this is the form we will use in our implementation.

To simplify the formulation of the next theorem we clarify the wording *extracting a list of poles*. The pole tuple is a list of multisets; with extracting a list of poles out of the pole tuple we mean assigning an ordering to elements in the multisets. Suppose, as an example, that we have a pole tuple $(\{3, 2, 1\}, 4, \{6, 5\}, \{8, 7\})$, then we can extract various lists out of it, such as, e.g., $(3, 2, 1, 4, 6, 5, 8, 7)$, but also $(1, 3, 2, 4, 5, 6, 7, 8)$.

There are typically many possibilities of extracting a list of poles out of a pole tuple. Suppose the pencil $(\hat{A}, \hat{B})$ has pole set $(\{3,2,1\}, 4, \{6,5\}, \{8,7\})$ and $(\check{A}, \check{B})$ has pole set $(1, \{5,4,3,2\}, \{7,6\}, 8)$. Then we can extract a list of poles $(1,2,3,4,5,6,7,8)$ that is feasible for both pencils, this is required in the next theorem.

THEOREM 3.8. *Let (A,B) be a proper matrix pencil and let $\hat{Q}, \check{Q}, \hat{Z}, \check{Z}$ be unitary matrices with $\hat{Q}\boldsymbol{e}_1 = \sigma\check{Q}\boldsymbol{e}_1$, $|\sigma| = 1$, such that,*

$$(\hat{A}, \hat{B}) = \hat{Q}^*(A,B)\hat{Z} \quad \textit{and} \quad (\check{A}, \check{B}) = \check{Q}^*(A,B)\check{Z}$$

are block Hessenberg pencils. If we can extract an identical list of poles out of the pole tuples of both pencils, then the pencils $(\hat{A}, \hat{B})$ and $(\check{A}, \check{B})$ are essentially unique.

Proof. Consider the generalized Schur factorization of each individual diagonal block in the pole pencils of $(\hat{A}, \hat{B})$ and $(\check{A}, \check{B})$, where we have ordered the eigenvalues in the resulting Schur forms in the same manner for both pencils. We know, because of the assumptions on the poles in the theorem, that such an ordering must exist.

Since the pole pencils are of size $(n-1) \times (n-1)$, we must embed the unitary transformations in an $n \times n$ identity matrix before we can apply them on the pencils directly. As a result we get a multiplication with block diagonal unitary matrices resulting in two Hessenberg pencils with identical pole tuples

$$\hat{D}_1^*(\hat{A}, \hat{B})\hat{D}_2 \quad \text{and} \quad \check{D}_1^*(\check{A}, \check{B})\check{D}_2,$$

where, because of the embedding $\hat{D}_1^*\boldsymbol{e}_1 = |\hat{\sigma}|\boldsymbol{e}_1$, $\check{D}_1^*\boldsymbol{e}_1 = |\check{\sigma}|\boldsymbol{e}_1$, $\hat{D}_2\boldsymbol{e}_n = |\hat{\gamma}|\boldsymbol{e}_n$, and $\check{D}_2\boldsymbol{e}_n = |\check{\gamma}|\boldsymbol{e}_n$, and $\hat{\sigma}$, $\check{\sigma}$, $\hat{\gamma}$, and $\check{\gamma}$ are unimodular.

It is easy to verify that both pencils will be proper and looking back at the original pencil (A,B) we obtain $\hat{D}_1^*\hat{Q}^*(A,B)\hat{Z}\hat{D}_2$ and $\check{D}_1^*\check{Q}^*(A,B)\check{Z}\check{D}_2$ on which we can apply the rational implicit Q theorem [10, Theorem 5.1]. As a result both Hessenberg pencils are essentially unique, thereby proving the theorem. ∎

This theorem does not only justify our definition of essential uniqueness, it also provides a manner of mapping block Hessenberg pencils to Hessenberg pencils. This mapping is not unique, it is essentially unique in the block sense. It should thus be clear that results that hold for the Hessenberg case, such as convergence, still hold, but only in-between blocks. We can not make claims of what is happening within a block.

4. The basic algorithm. The algorithm operates on proper pencils, otherwise it will break down; if the pencil is improper, deflation is possible, and the problem should be split into independent subproblems. We assume to be operating on a proper block Hessenberg pencil (A,B) with partition $\boldsymbol{s} = (s_1, \ldots, s_m)$ and pole tuple $\Xi = (\Xi^1, \ldots, \Xi^m)$, where Ξ^j is a multiset of poles. All poles are assumed different from the eigenvalues. We review two different operations to change the pole tuple Ξ. The first operation replaces the first or last blocks of pole of the pencil, the second operation swaps two adjacent pole blocks.

Given a matrix pencil $(A,B) \in \mathbb{F}^{n\times n}$ (generic, not necessarily block Hessenberg), with $\varrho = \mu/\nu \in \bar{\mathbb{C}}$ and $\xi = \alpha/\beta \in \bar{\mathbb{C}} \setminus \Lambda$, we define the following elementary rational matrices,

$$\begin{aligned} M(\varrho,\xi) &= (\nu A - \mu B)(\beta A - \alpha B)^{-1}, \\ N(\varrho,\xi) &= (\beta A - \alpha B)^{-1}(\nu A - \mu B). \end{aligned} \tag{9}$$

Notice that the matrices $M(\varrho,\xi)$ and $N(\varrho,\xi)$ represent an entire class of matrices that are all nonzero scalar multiple of each other. Every representative is fine.

Changing poles at the boundary. The first $\ell = s_1+\ldots+s_i$ poles in the first i pole blocks $\Xi^1,\ldots,\Xi^i$ can be replaced by ℓ new poles $\mathrm{P} = \{\varrho_1,\ldots,\varrho_\ell\}$, which are considered different from the original poles. These new poles are classically called the shifts. In fact they are nothing else than poles, but when it is necessary to emphasize their origin, we call them shift-poles. To introduce the shift-poles consider the vector,

$$\boldsymbol{x} = \gamma \prod_{j=1}^{\ell} M(\varrho_j,\xi_j)\, \boldsymbol{e}_1, \tag{10}$$

with $\xi_1,\ldots,\xi_\ell$ the poles of $\Xi^1,\ldots,\Xi^i$. Now compute a unitary matrix Q such that,

$$Q^*\boldsymbol{x} = \alpha \boldsymbol{e}_1. \tag{11}$$

We will prove in Section 8 that the new poles P are introduced in the block Hessenberg pencil by updating $(\hat{A},\hat{B}) = Q^*(A,B)$. More precisely we end up with a block Hessenberg pencil $(\hat{A},\hat{B})$ with partition $\hat{\boldsymbol{s}} = (\ell, s_{i+1},\ldots,s_m)$ and poles $\hat{\Xi} = (\mathrm{P},\Xi^{i+1},\ldots,\Xi^m)$.

The last ℓ poles, say ξ_j for $j = m-\ell+1,\ldots m$, the last i blocks $\Xi^{m-i+1},\ldots,\Xi^m$ of (A,B) can be changed to $\mathrm{P} = \{\varrho_1,\ldots,\varrho_\ell\}$ in a similar fashion. We compute first the row vector,

$$\boldsymbol{x}^T = \gamma \boldsymbol{e}_n^T \prod_{j=m-\ell+1}^{m} N(\varrho_j,\xi_j), \tag{12}$$

and then a unitary matrix $Z = \mathrm{diag}(I, Z_{\ell+1})$ such that $\boldsymbol{x}^T Z = \alpha \boldsymbol{e}_n^T$. The pencil $(\hat{A},\hat{B}) = (A,B)Z$ then becomes block Hessenberg with pole tuple (Ξ^1, …, Ξ^{m-i}, P).

In order to compute the vector $\boldsymbol{x}$ of Equation (10), ℓ shifted linear systems need to be solved as $M(\varrho_i,\xi_i) = (\nu_i A-\mu_i B)(\beta_i A-\alpha_i B)^{-1}$. These linear systems are essentially of size ℓ because $(\beta_\ell A-\alpha_\ell B)^{-1}$ is a block triangular matrix with a leading block of size $\ell\times\ell$. This limits the computational cost of computing $\boldsymbol{x}$ to $O(\ell^3)$, which is small as long as $\ell \ll n$. It also follows that the vector $\boldsymbol{x}$ can be computed even when poles in Ξ^1,…, Ξ^i are equal to eigenvalues of the pencil not present in the leading block, since properness ensures this.

We remark that if (A,B) is a real-valued pencil and the poles and shift-poles considered in (10) and (12) are both closed under complex conjugation, then the vectors $\boldsymbol{x}$ and $\boldsymbol{x}^T$ and consequently the matrices Q and Z are also real-valued. This follows from the commutativity of the elementary rational matrices $M(\varrho,\xi)$ and $M(\bar{\varrho},\bar{\xi})$ (see [10]) in combination with the property that $M(\bar{\varrho},\bar{\xi}) = \overline{M(\varrho,\xi)}$ for real-valued pencils. We have,

$$\overline{M(\varrho,\xi)M(\bar{\varrho},\bar{\xi})} = \overline{M(\bar{\varrho},\bar{\xi})M(\varrho,\xi)} = M(\varrho,\xi)M(\bar{\varrho},\bar{\xi}) \tag{13}$$

so $M(\varrho,\xi)M(\bar{\varrho},\bar{\xi})$ is a real-valued matrix if A and B are real-valued.

Swapping adjacent pole blocks. Swapping block i with block $i+1$ requires the computation of a unitary equivalence of size $(s_i+s_{i+1})\times(s_i+s_{i+1})$ that updates the pencil $(\hat{A},\hat{B}) = Q^*(A,B)Z$ in such a way that the new pole tuple and partition vector are given by,

$$\begin{aligned}\hat{\Xi} &= (\Xi^1,\ldots,\Xi^{i-1},\Xi^{i+1},\Xi^i,\Xi^{i+2},\ldots,\Xi^m),\\ \hat{\boldsymbol{s}} &= (s_1,\ldots,s_{i-1},s_{i+1},s_i,s_{i+2},\ldots,s_m).\end{aligned}$$

This problem is equivalent to reordering eigenvalues in the generalized Schur form. Two different approaches to solve this problem have been proposed in the literature. The first approach, studied by Kågström [15, 17], requires the solution of a coupled Sylvester equation. This method is applicable for general blocksizes. The second approach, studied by Van Dooren [23], is a direct method that relies on the computation of a right eigenvector of a pole in block $i{+}1$. This method has been studied for swapping a block of dimension $1{\times}1$ or $2{\times}2$ with a block of dimension $1{\times}1$, or vice versa. In our implementation we combine both techniques and use iterative refinement. More details are discussed in Section 6.

Multishift, multipole RQZ step. We propose the following three step procedure as the generic multishift, multipole RQZ step.

I. Starting from a proper block Hessenberg pencil with pole tuple $\Xi = (\Xi^1, \ldots, \Xi^m)$ and partition $\boldsymbol{s} = (s_1, \ldots, s_m)$. Select or compute ℓ shifts P. Introduce the shift-poles in the block Hessenberg pencil. The pencil now has pole tuple $\Xi = (\mathrm{P}, \Xi^{i+1}, \ldots, \Xi^m)$ and partition vector $\boldsymbol{s} = (\ell, s_{i+1}, \ldots, s_m)$.

II. Repeatedly use the swapping procedure to construct unitary equivalences that swap the block carrying the shift-poles with the next pole-block in line. In the end the shift-poles reach the last block and the pole tuple equals $\Xi = (\Xi^{i+1}, \ldots, \Xi^m, \mathrm{P})$ and the partition vector $\boldsymbol{s} = (s_{i+1}, \ldots, s_m, \ell)$.

III. Compute or select ℓ new poles Ξ^{m+1} and introduce them at the end of the pencil to change the pole tuple to $\Xi = (\Xi^{i+1}, \ldots, \Xi^m, \Xi^{m+1})$.

These three steps constitute a single multishift, multipole RQZ sweep. After every sweep, the properness of the pencil is checked and the problem is split into independent subproblems wherever possible. We will prove in Section 8 that continuously executing RQZ sweeps with well-chosen shifts will lead to deflations and eventually converges to the Schur form. Moreover, we stress that this is the basic form, an enhanced form of the algorithm, with aggressive deflation (see Section 5) is discussed in Section 6.

The multishift QZ method is a special case of this algorithm where the pencil initially has pole tuple $(\infty, \ldots, \infty)$ and partition $(1, \ldots, 1)$ and where this form is always restored in step III of the algorithm by only allowing the introduction of poles at infinity.

The shift-poles will create convergence in the lower-right corner of the matrix, the poles move up slowly and will lead to deflations in the upper-left corner of the matrix. In the classical QZ algorithm, it is not possible to stear convergence in the upper-left corner, the poles moving up, will always be equal to ∞. Again, we defer the theoretical analysis of the convergence to Section 8.

5. Aggressive early deflation. Aggressive early deflation (AED) significantly speeds up the convergence of the QR [7] and QZ [16] methods by identifying deflatable eigenvalues before classical deflation criteria are able to detect them: We will check whether the eigenvalues of leading and trailing principal subpencils are eigenvalues of the original pencil.

Because the shift-poles lead to convergence in the bottom-right corner of the pencil and the poles cause convergence in the upper-left corner, AED can be performed at both sides of the pencil. We present the description of the AED process only for the upper-left sides of the pencil, the bottom-right proceeds similarly.

The deflation window sizes are w_e for the bottom-right and w_s for the upper-left side of the pencil, they are chosen to cover an integer number of blocks, avoiding thereby subdivision of blocks. The deflation windows are shown in Pane I of Figure 1. The window sizes used in our practical implementation are found in Section 6.

The pencil (A, B) is subdivided as follows

$$
\begin{array}{c|ccc|c|c}
 & w_s & & & w_e & \\
\hline
w_s & A_{11} & A_{12} & A_{13} & A_{14} & w_s \\
\hline
2 & A_{21} & A_{22} & A_{23} & A_{24} & \\
 & & A_{32} & A_{33} & A_{34} & \\
\hline
w_e & & & A_{43} & A_{44} & w_e
\end{array}
\quad,\quad
\begin{array}{c|ccc|c|c}
 & w_s & & & w_e & \\
\hline
w_s & B_{11} & B_{12} & B_{13} & B_{14} & w_s \\
\hline
1 & B_{21} & B_{22} & B_{23} & B_{24} & \\
 & & B_{32} & B_{33} & B_{34} & \\
\hline
w_e & & & B_{43} & B_{44} & w_e
\end{array}
\quad, \tag{14}
$$

where we assume A_{21} to have two rows, indicating that we have a 2×2 block just after the deflation window. We restrict the size of A_{21} and of the other blocks to be at most 2×2 since it simplifies the presentation and it is sufficient for the real double-shift rational QZ algorithm. Recall from the discussion following Theorem 3.8 that we have taken the design decision to have B in Hessenberg form, as a consequence B_{21} has only one row. The subpencils (A_{11}, B_{11}) and (A_{44}, B_{44}) are the upper-left and bottom-right deflation windows.

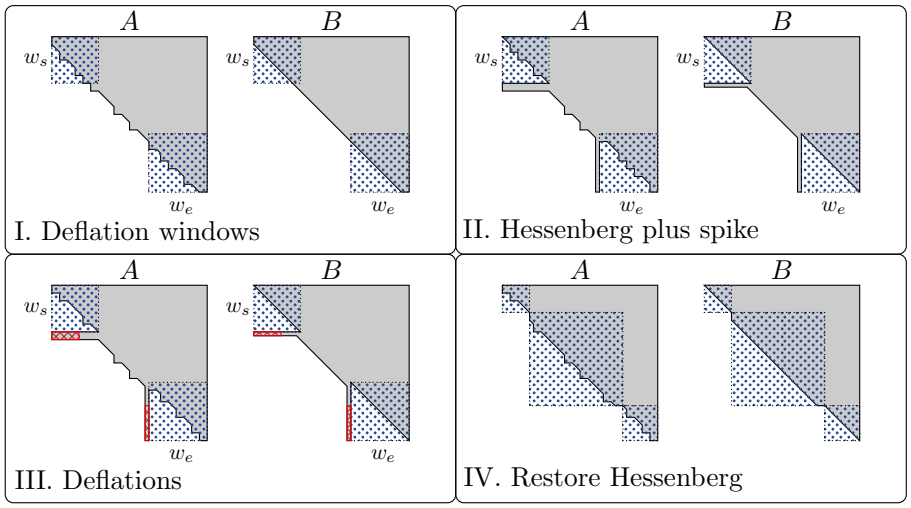


FIG. 1. *Visualization of the three stages of aggressive early deflation.*

In the first phase, shown in pane II of Figure 1, the parts of the pencil within the deflation windows are reduced to quasi Schur form (block upper triangular with blocks of size at most 2):

$$(S_{11}, T_{11}) = Q_s^T (A_{11}, B_{11}) Z_s, \quad \text{and} \quad (S_{44}, T_{44}) = Q_e^T (A_{44}, B_{44}) Z_e. \tag{15}$$

This can be done with the RQZ method as all subpencils in the deflation windows are in block Hessenberg form. Applying these transformations as an equivalence to (A, B) results in the pencil $(\check{A}, \check{B})$:

$$
\begin{array}{|ccc|c|}
\hline
S_{11} & Q_s^T A_{12} & Q_s^T A_{13} & Q_s^T A_{14} Z_e \\
\hline
A_{21} Z_s & A_{22} & A_{23} & A_{24} Z_e \\
 & A_{32} & A_{33} & A_{34} Z_e \\
\hline
 & & Q_e^T A_{43} & S_{44} \\
\hline
\end{array}
\quad,\quad
\begin{array}{|ccc|c|}
\hline
T_{11} & Q_s^T B_{12} & Q_s^T B_{13} & Q_s^T B_{14} Z_e \\
\hline
B_{21} Z_s & B_{22} & B_{23} & B_{24} Z_e \\
 & B_{32} & B_{33} & B_{34} Z_e \\
\hline
 & & Q_e^T B_{43} & T_{44} \\
\hline
\end{array}
\quad. \tag{16}
$$

The spikes shown in pane II of Figure 1 correspond to the matrices $A_{21}Z_s$, $B_{21}Z_s$, $Q_e^T A_{43}$, and $Q_e^T B_{43}$, with $B_{21}Z_s$ of dimension $1\times w_s$ and $A_{21}Z_s$ of dimension $2\times w_s$. The two rows of $A_{21}Z_s$ are scalar multiples of each other and also a multiple of $B_{21}Z_s$. We denote with $\boldsymbol{p}_s^B = b_{w_s+1,w_s}\boldsymbol{e}_{w_s}^T Z_s = b_{w_s+1,w_s}\boldsymbol{p}_s$ the spike at the upper-left deflation window of B. Similarly, $\boldsymbol{p}_s^A = \zeta \boldsymbol{e}_{w_s}^T Z_s = \zeta \boldsymbol{p}_s$, with ζ equal to $|a_{w_s+1,w_s}| + |a_{w_s+2,w_s}|$. Taking the sum makes the deflation check, discussed further on, easier.

The second phase in the AED process is illustrated in Pane III of Figure 1 and entails testing for deflatable eigenvalues inside the deflation windows. The deflation test starts at the left-side of the spikes $\boldsymbol{p}_s^A$ and $\boldsymbol{p}_s^B$. If there is a 1×1 eigenvalue located at this position, we test if:

$$|\boldsymbol{p}_{s,1}^A| < c\epsilon_m(|a_{1,1}| + |a_{2,2}|) \quad \text{and} \quad |\boldsymbol{p}_{s,1}^B| < c\epsilon_m(|b_{1,1}| + |b_{2,2}|), \tag{17}$$

with c a modest constant and ϵ_m the machine precision. If there is a 2×2 block at this position (which means a complex conjugate pair of eigenvalues in the real case) at this position, we test if:

$$|\boldsymbol{p}_{s,1}^A| + |\boldsymbol{p}_{s,2}^A| < c\epsilon_m\|A_{1:2,1:2}\|_F \quad \text{and} \quad |\boldsymbol{p}_{s,1}^B| + |\boldsymbol{p}_{s,2}^B| < c\epsilon_m\|B_{1:2,1:2}\|_F. \tag{18}$$

If the first eigenvalue is deflatable according to (17) or (18), the corresponding spike elements in $\boldsymbol{p}_s^A$ and $\boldsymbol{p}_s^B$ are set to zero and the next eigenvalue is tested according to the same criterion. If the first eigenvalue is not deflatable, another eigenvalue that has not yet been tested, is swapped to the top-left corner, this also changes the values of the spikes. Then it is checked if this is deflatable according to (17) or (18). This procedure is continued until all deflatable eigenvalues inside the deflation window are identified. The swapping of eigenvalues within the deflation window does not change the form of (15) but of course S_{11} and T_{11} change, just like the vector $\boldsymbol{p}_s$. The same strategy is used for AED at the bottom-right side of the pencil. In pane III of Figure 1 all spike elements that signal a deflation are marked in red.

In the third and last phase, the nonzero spike elements are removed in such a way that the (block) Hessenberg form is restored. This form is shown in pane IV of Figure 1: the larger block in the middle is in block Hessenberg form and the smaller blocks at the upper-left and bottom-right side of the pencil are in quasi Schur form. The block Hessenberg restoration is achieved by a sequence of rotations as follows. Assume that after all the reordering the new spike equals $\hat{\boldsymbol{p}}_s$ and we have the Schur form $(\hat{S}_{11}, \hat{T}_{11})$, where the first i entries of $\hat{\boldsymbol{p}}_s$ are zero. This means that the upper-left $i\times i$ block of $(\hat{S}_{11}, \hat{T}_{11})$ is deflatable. We then compute rotations $G_{i+1}, \ldots, G_{w_s-1}$ such that, $\hat{\boldsymbol{p}}_s G_{i+1} \cdots G_{w_s-1}$ becomes a multiple of $\boldsymbol{e}_{w_s}^T$. Applying those transformations to the left of $(\hat{S}_{11}, \hat{T}_{11})$ restores the block Hessenberg form and removes the spikes.

At the end of the AED phase, we can re-initiate the chasing and use some of the undeflated eigenvalues of $(\hat{S}_{11}, \hat{T}_{11})$ as poles, to be inserted in the bottom-right in the upcoming rational multishift QZ step. In case that AED was so succesful that many converged eigenvalues have been deflated, one could do another step of AED before proceeding to the chase. This technique also used in our final algorithm in Section 6.

6. The implemented algorithm and heuristics. We have discussed the basic algorithm in Section 4 and AED in Section 5. Here we propose the full algorithm including additional add-ons in the implementation such as blocking and iterative refinement for inaccurate swaps. Also the decisions on heuristics such as deflations, block sizes, aggressive deflation windows and the number of tightly packed poles are presented.

6.1. Deflation. We always take B in Hessenberg form (see the comments following Definition 3.7); This simplifies the deflation criteria based on Definition 3.5.

Checking Condition I of Definition 3.5. The ith pole along the subdiagonal is considered deflatable if,

$$|a_{i+1,i}| < c\epsilon_m(|a_{i,i}| + |a_{i+1,i+1}|), \quad \text{and,} \quad |b_{i+1,i}| < c\epsilon_m(|b_{i,i}| + |b_{i+1,i+1}|), \tag{19}$$

in the case of a single pole. If the ith pole is a double-pole we consider it deflatable if either there are small elements in the leading column of the block,

$$\begin{aligned} |a_{i+1,i}| + |a_{i+2,i}| < c\epsilon_m(|a_{i,i}| + |a_{i+1,i+1}|), \quad \text{and,} \\ |b_{i+1,i}| < c\epsilon_m(|b_{i,i}| + |b_{i+1,i+1}|), \end{aligned} \tag{20}$$

or there are sufficiently small elements in the trailing row of the block,

$$\begin{aligned} |a_{i+2,i}| + |a_{i+2,i+1}| < c\epsilon_m(|a_{i+1,i+1}| + |a_{i+2,i+2}|), \quad \text{and,} \\ |b_{i+2,i+1}| < c\epsilon_m(|b_{i+1,i+1}| + |b_{i+2,i+2}|). \end{aligned} \tag{21}$$

Checking Condition II and III of Definition 3.5. The first pole block of size $s_1 = 1$ or 2 can be deflated whenever there exists an $(s_1+1)\times(s_1+1)$ orthogonal matrix Q such that,

$$Q^T \left(\begin{bmatrix} \boldsymbol{a}_{1,1}^T \\ A_{2,1} \end{bmatrix}, \begin{bmatrix} \boldsymbol{b}_{1,1}^T \\ B_{2,1} \end{bmatrix} \right) = \left(\begin{bmatrix} A_{1,1} \\ \mathbf{0}^T \end{bmatrix}, \begin{bmatrix} B_{1,1} \\ \mathbf{0}^T \end{bmatrix} \right) \tag{22}$$

Here, the last row is considered numerically zero according to a relative tolerance similar to (19)–(21). The matrix Q is constructed such to create a desired zero in the column with largest norm; if the pole block is deflatable, this should create zeros in the other positions as well. A similar approach is used to check for deflations in the last block row.

6.2. Accuracy of swapping pole blocks. The equivalence between swapping eigenvalues in the generalized Schur form and swapping poles allows us to use the `LAPACK` routine `DTGEXC`. However, some modifications can greatly increase the effectiveness of this routine.

When one of the blocks to be swapped is of size 1, the method described in [23] can be used. We note that a modest modification proposed by Camps et al. [11] of Van Dooren's method [23] improves the theoretical bound on the backward error of the swap. The computed matrices $\hat{Q},\hat{Z},\hat{A},\hat{B}$ are proven to satisfy

$$\hat{Q}^*(A + E_A, B + E_B)\hat{Z} = (\hat{A}, \hat{B}), \tag{23}$$

with $\|E_A\|_2 \leq c\epsilon\|A\|_2$ and $\|E_B\|_2 \leq c\epsilon\|B\|_2$ [8]. When one of the blocks is of size two, numerical evidence suggest that the same strategy can be applied; a theoretical backward error analysis is lacking, however.

If both of the blocks are of size two, the transformations are computed according to [15, 17] relying on solving the Sylvester equation. However, this method is not norm-wise backward stable. It leads to occasional non-negligible off-diagonal blocks when an ill-conditioned 2×2 block close to convergence is involved. In this case, a step of iterative refinement takes place [9] to decrease the norm of the off-diagonal block. Our numerical experiments indicated that iterative refinement is required in about 5% of all 2×2 with 2×2 swaps during a typical RQZ iteration. In Section 7 the numerics illustrate that these swaps lead to a backward stable algorithm.

6.3. Blocking and packing poles. To avoid shift blurring, we avoid large multiplicities. However, only chasing a few at a time does not perform well on modern computer architectures. Braman, Byers, and Mathias [6] proposed using a train of shifts. Instead of introducing a shift and chasing it all the way to the bottom of the pencil, a shift is chased just far enough to make room for another shift to be introduced. As a result, we end up with a train of shifts that can be chased to the bottom efficiently. The same idea was applied to the QZ algorithm by Kågström and Kressner [16].

We also make use of blocking, which tries to make optimal use of the computer architecture by enhancing cache efficiency as a result of reducing the cost of moving data in and out of memory. The direct updates are therefore limited to a moving computational window (a block on the diagonal of the pencil) in which the train of shifts is chased from the top-left corner of the window to the bottom-right. During this chase all executed transformations are accumulated and the rest of the pencil is updated at once via matrix-matrix multiplications. To execute these matrix-matrix multiplications we rely on highly optimized BLAS implementations [4, 24].

The computational efficiency will benefit from getting as many shifts as possible in a single window. Camps et al. [11] illustrated that the rational QZ method allows to pack shift-poles optimally in a straightforward way. In the bulge chasing setting one can easily pack shifts tightly [6], but more advanced techniques are required to pack them optimally [18].

The sliding window size $n_w = n_s + k$, where n_s stands for the number of optimally packed shift(-pole)s that are present in a window and k denotes the number of positions that each shift-pole can move down. To select the window size Karlsson et al. [18] chose to minimize the amount of flops during a full sweep. This selection of the window size is optimal in a sense, but it neglects other factors that can be important in practice. While the computations inside the window are usually negligable when purely counting flops, they are in essence computationally more demanding than the highly optimized matrix-matrix multiplications (BLAS) to update the off-diagonal part of the pencil. As a result the calculated optimum will be an overestimate when using a large number of shifts. Secondly, most BLAS implementations, and especially parallel variants of BLAS, become more efficient when doing larger multiplications. This indicates that the calculated optimum will be an underestimate when using a small number of shifts.

To account for the overestimation we propose to select a blocksize $n_w = n_s + k$ that minimizes the function

$$\text{cost}(k) = \frac{2cn(k+n_s)^2 + 4n_s k(k+n_s)}{k}, \tag{24}$$

where c is the inverse of the flop rate of the BLAS calls relative to the window update. This minimum is attained for

$$k = n_s\left(1 + \frac{2n_s}{cn}\right)^{-\frac{1}{2}}. \tag{25}$$

For very large values of c or n, this will tend towards n_s, the optimum obtained in [18]. This value will still be an underestimate for small values of n_s. Therefore we impose a minimal blocksize based on the size of the pencil. In our implementation, c is set to 0.1.

6.4. Aggressive deflation windows and number of shifts. The parameters n_s, which stands for the number of shifts, and w_e and w_s, which are the windows for

aggressive deflation at the top and at the bottom, are harder to select. These parameters have a significant impact on both convergence speed and the overall execution time. We have selected the parameters empirically. A summary is found in Table 1. The first column lists the size of the pencil. The second column lists the batch size n_s of shifts that are handled in one iteration. The third column lists the window size w_e for aggressive early deflation at the bottom-right side of the pencil. Finally, the fourth column lists the window size w_s for aggressive early deflation at the upper-left side of the pencil. If, in the AED window 8%, w.r.t. w_e or w_s, of the eigenvalues are found, we do another step of AED and keep repeating this procedure before starting the RQZ sweep.

TABLE 1
Settings: n problem size, m step multiplicity, w_e AED window size at the bottom-right side of the pencil, w_s AED window size at the upper-left side of the pencil.

n	n_s	w_e	w_s
$[1; 80[$	1—2	1—2	1—2
$[80; 150[$	4	8	4
$[150; 590[$	32	48	32
$[590; 3000[$	40	96	40
$[3000; \infty[$	64	96	64

Due to the nature of the algorithm the poles move slower to the top than shift-poles to the bottom, hence $w_s \leq w_e$. Alternative versions of swapping algorithms where we get equally fast convergence to the top are proposed by Camps et al. [11] and use bidirectional chasing.

6.5. Full algorithm. A double-shift real rational QZ step (DRRQZ) step with all add-ons proceeds as follows. The heuristics were discussed in the previous sections.

I. Check for interior deflations to select the active part of the matrix.
II. AED at the top.
  (a) Check for deflations at the upper-left side of the pencil using AED. If sufficient eigenvalues are found redo the AED.
  (b) Take n_s undeflated eigenvalues to be introduced as poles at the end of the pencil in III(c).
III. AED at the bottom.
  (a) Check for deflations at the bottom-right side of the pencil using AED, and repeat if necessary.
  (b) Store n_s undeflated eigenvalues to be introduced as shift-poles.
  (c) The undeflated eigenvalues from II(b) are introduced as poles at the bottom and moved to the top of the window[4].
IV. Initiate the sweep by introducing n_s shift-poles from step III(b). The involved transformations are accumulated and the pencil is updated by level 3 BLAS matrix-matrix multiplication.
V. Chase the batch of n_s shift-poles to the last n_s positions on the subdiagonal of the block Hessenberg pencil. The chasing is performed by repeatedly swapping the n_s shift-poles with the next k poles thereby using blocking.

[4]They must be moved to the top of the window, otherwise the next AED step removes them again.

7. Numerical experiments. In this section we will demonstrate the accuracy and efficiency of our implementation, which is a double-shift and double-pole version for handling real pencils, and which sticks to real arithmetic. We name our algorithm `DRRQZ`. We will show that it is much faster than our complex single shift rational QZ algorithm, named `ZRQZ`, and we will compare with the QZ algorithm of `LAPACK` and `PDHGEQZ` [2]. `PDHGEQZ` is a library designed to be used on large parallel machines on very large problems. For a fair comparison with our code, `PDHGEQZ` was not run in parallel, but linked to the same parallel BLAS implementation. Even run serially, `PDHGEQZ` performs very well [1].

The QZ algorithm based on bulge chasing and the `DRRQZ` implementation have some essential differences.

- In the QZ case we always swap (implemented as a bulge chase) the shift-pole block which is of size 2×2 with a 1×1 pole block, since the poles moving up always equal ∞. In the `DRRQZ` we can also encounter 2×2 pole blocks to be swapped with the shift-pole block, so some additional branches in the code are required.
- The swapping procedure is implemented completely differently. The QZ setting allows us to execute all operations on one side of the pencil, before handling the other side, in the `DRRQZ` setting this is not true, so we pay a price in cache efficiency.

There are also some particularities of the codes we will compare with.

- `PDHGEQZ` has a special built in feature to deflate infinite eigenvalues, that appear as zeroes on the diagonal of B, while executing the algorithm; this feature is unexisting in the rational QZ algorithm and appears to be hard to implement generic Hessenberg pencils.
- The `DRRQZ` setting allows us to trivially deal with optimally packed shift-poles [11]. They are more tightly packed than the ones in `PDHGEQZ` and `LAPACK`.
- `LAPACK` doesn't use AED, `PHDGEQZ` and `DRRQZ` do, `DRRQZ` also executes AED on the top of the pencil

The numerical tests have been performed on an Intel Xeon E5-2697 v3 CPU with 14 cores and 128GB of RAM. Our `DRRQZ` implementation with aggressive deflation is compiled with *gfortran* version 7.4.0 using compilation flag `-Ofast`. The code is linked with release 88 of `MKL` and the link to the git repository is available at numa.cs.kuleuven.be/software/rqz and github.com/thijssteel/multishift-multipole-rqz.

7.1. Simple test problems. For our first numerical experiment, we will compare the implementations using two different pencils. The first pencil is in Hessenberg, triangular form, with the nonzero entries drawn from a uniform distribution between 0 and 1. This is the Hessrand2 test problem from [1]. Because of the random nature of this pencils, the results are averaged over 10 runs[5]. AED is so effective on this pencil that barely any sweeps are required. Most of the computation time is therefore spent in AED. The second pencil is also in Hessenberg, triangular form, with the nonzero entries given by $A_{i,j} = i + j$ and $B_{i,j} = 2i + 3j$. This pencil was constructed explicitly such that AED is less effective. As a consequence we expect the sweep to have a bigger role in the execution time.

Figure 2 shows the execution time of `DRRQZ`, `DHGEQZ` and `PDHGEQZ` for problems of size 1000 up to 8000 on a loglog scale. The left part shows the results for the randomly generated pencil. `libRQZ` is faster than `PDHGEQZ` and both methods show

[5]Except for `LAPACK`, there we considered a single run due to high computational cost.

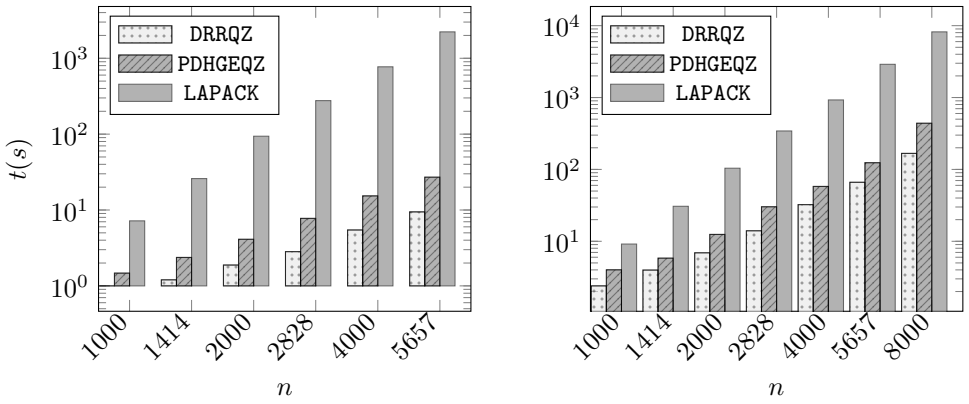


FIG. 2. *Execution time of* `DHGEQZ` *of* `LAPACK`*,* `libRQZ` *and* `PDHGEQZ` *on randomly generated real-valued matrix pencils (*left*) and the '$i+j$' pencil (*right*).*

large speedups over `LAPACK` (which does not use AED). The right part shows the results for the second '$i+j$' pencil. The larger amount of sweeps results in a smaller speedup over both competitors.

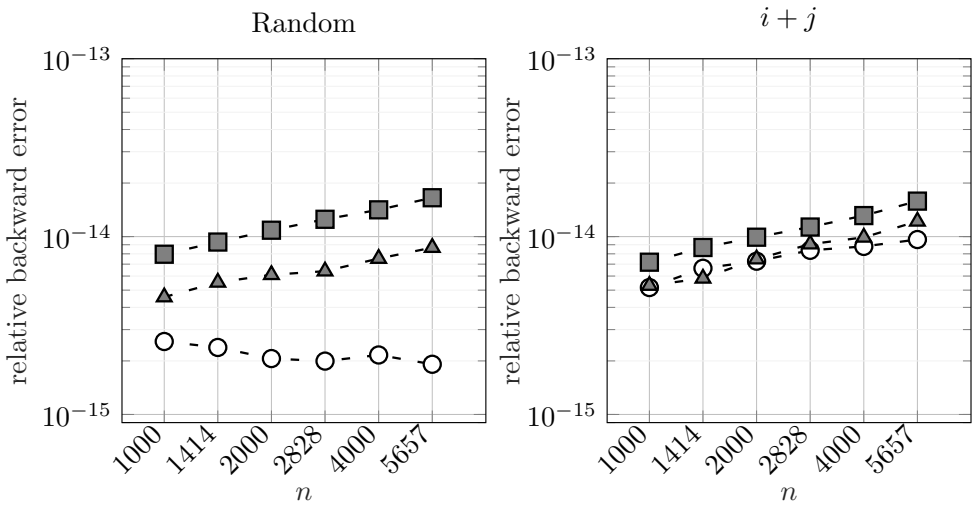


FIG. 3. *Maximum of the relative backward error on A and B of Schur decomposition computed with* `LAPACK` *(*squares*),* `PDHGEQZ` *(*triangles*) and* `DRRQZ` *from* `libRQZ` *(*circles*). Both for randomly generated pencils (*left*) and the '$i+j$' pencil (*right*).*

Figure 3 shows the relative backward errors,

$$\|S - Q^T A Z\|_F / \|A\|_F, \quad \text{and,} \quad \|T - Q^T B Z\|_F / \|B\|_F,$$

on the generalized real Schur decompositions obtained with `libRQZ`, `PDHGEQZ` and `DHGEQZ`. We observe that the relative backward errors of `libRQZ` are almost always the smallest.

7.2. Complex code and real double-pole code. We compare the `DRRQZ` code with the RQZ code, the latter will just ignore the fact that the matrix is real, and use complex shift-poles to find the complex conjugate pairs of eigenvalues. Complex arithmetic is between 2 and 6 times more expensive than real arithmetic. However, the swapping procedure is significantly simpler because all the pole blocks are of size 1 so we do not expect the real code to achieve speedups as large as 6. The numerical experiments reveal that around 5000, `DRRQZ` becomes twice as fast.

We solved the '$i + j$' pencil using the real and complex RQZ code. The results are displayed in Table 2. The real code is always faster with speedups of up to 2.4 for the larger pencils.

TABLE 2
Execution time and relative backward error of real and complex RQZ code on the 'i + j' pencil.

	`RQZ`		`DRRQZ`		
n	$t(s)$	$\max(\Delta A, \Delta B)$	$t(s)$	$\max(\Delta A, \Delta B)$	Speedup
1000	2.56	$5.48 \cdot 10^{-15}$	2.12	$5.17 \cdot 10^{-15}$	1.21
1414	4.96	$5.93 \cdot 10^{-15}$	3.64	$6.64 \cdot 10^{-15}$	1.36
2000	10.68	$6.35 \cdot 10^{-15}$	7.03	$7.27 \cdot 10^{-15}$	1.52
2828	24.05	$6.9 \cdot 10^{-15}$	14.19	$8.37 \cdot 10^{-15}$	1.69
4000	57.59	$7.7 \cdot 10^{-15}$	31.24	$8.57 \cdot 10^{-15}$	1.84
5657	144.27	$8.2 \cdot 10^{-15}$	65.17	$9.62 \cdot 10^{-15}$	2.21
8000	375.75	$9.1 \cdot 10^{-15}$	157.11	$9.89 \cdot 10^{-15}$	2.39

7.3. Problems from applications. In this section we test `libRQZ` on five pencils originating from applications. We study the *cavity* and *obstacle flow* pencils generated with IFISS [12, 13]. The same pencils were studied in our initial paper on the RQZ method [10]. Besides these pencils, we have selected two pencils from Matrix market [5] originating from the MHD collection and the *rail* pencil from the Oberwolfach benchmark collection [19].

Before being able to run the eigenvalue solvers. The pencils need to be reduced to Hessenberg, triangular form. At this moment there is no competitive direct method to reduce a pencil to Hessenberg, Hessenberg form, with a prescribed set of poles. The lack of such a reduction is an important aspect of future research. We do note, however, that the RQZ method accepts a more general input form, and for instance pencils resulting from rational Krylov methods [3] would no longer need a reduction step.

Our code uses a basic input format, whereas `PDHGEQZ` makes use of `ScaLAPACK`'s data format. As a consequence our current implementation can not make direct use of the superior built-in reduction to Hessenberg, triangular form in `PDHGEQZ`. Instead we have to rely on `DGEQRFP` and `DGGHD3` of `LAPACK`.

The results of the numerical tests are summarized in Table 3. The table lists the execution time of both the reduction and iterative steps and the maximum of the relative backward errors of A and B for the generalized real Schur form. `libRQZ` is better in terms of execution time of the iterative part and the backward error. We stress, that for both methods, the reduction to Hessenberg, triangular form requires much more computation time than the iterative part. Work and research is required to either adapt our code to operate on the `ScaLAPACK` data format, or to devise a fast reduction to Hessenberg, Hessenberg form.

TABLE 3
Execution times of the reduction ($t_{HT}(s)$) and iterative ($t_{IT}(s)$) step and maximum relative backward error on the generalized quasi Schur form computed with `DRRQZ` *and* `PDHGEQZ` *for pencils originating from applications.*

		`DRRQZ`			`PDHGEQZ`		
Problem	n	$t_{\mathrm{HT}}(s)$	$t_{\mathrm{IT}}(s)$	$\max(\Delta A, \Delta B)$	$t_{\mathrm{HT}}(s)$	$t_{\mathrm{IT}}(s)$	$\max(\Delta A, \Delta B)$
Obstacle Flow	2488	104.95	5.3	$8.2 \cdot 10^{-15}$	24.4	7.3	$1.0 \cdot 10^{-14}$
Cavity Flow	2467	105.4	4.4	$6.3 \cdot 10^{-15}$	26.5	6.3	$9.0 \cdot 10^{-15}$
MHD3200	3200	230.1	5.0	$2.9 \cdot 10^{-15}$	54.6	9.1	$9.2 \cdot 10^{-15}$
MHD4800	4800	915.1	15.2	$4.0 \cdot 10^{-15}$	184.3	22.7	$1.5 \cdot 10^{-14}$
RAIL	5177	1215.1	28.5	$1.1 \cdot 10^{-15}$	258.7	47.3	$3.5 \cdot 10^{-14}$

8. Theoretical analysis of the rational QZ algorithm. In this section we will prove that the rational QZ algorithm is equivalent to subspace iteration driven by a rational function. We will rely heavily on the connection with rational Krylov. The proof and analysis proceeds similarly to the work of Watkins [26]. A more detailed analysis focussing of the single shift rational QZ case can be found in Camps et al. [10] and the references therein.

8.1. Rational Krylov and block Hessenberg pencils. We study the structure of *rational Krylov subspaces* generated by proper block Hessenberg pencils. These results prove the correctness of the *pole introduction* operation (see Section 4) and *essential uniqueness* of a multishift, multipole RQZ step.

The elementary rational matrices M and N (9) are used to construct rational Krylov matrices generated by a regular matrix pencil, a starting vector, and a tuple of poles. They satisfy some basic properties we need later on, for proofs see [10, Lemma 5.3]). The inverse $M(\varrho, \xi)^{-1}$ is defined if $\varrho \notin \Lambda$ and is equal to $M(\xi, \varrho)$. They are commutative, $M(\varrho_1, \xi_1)M(\varrho_2, \xi_2) = M(\varrho_2, \xi_2)M(\varrho_1, \xi_1)$, and they can be merged together, $M(\varrho, \xi_1)M(\xi_1, \xi_2) = M(\varrho, \xi_2)$, if a pole and shift are equal. Analogous results hold for $N(\varrho, \xi)$.

DEFINITION 8.1 (Rational Krylov matrices). *Let $A, B \in \mathbb{F}^{n\times n}$ form a regular matrix pencil, $\boldsymbol{v} \in \mathbb{F}^n$ a nonzero vector, $k \leq n$, $\Xi = (\xi_1, \ldots, \xi_{k-1})$ a tuple of poles distinct from the eigenvalues, and $P = (\varrho_1, \ldots, \varrho_{k-1}) \subset \bar{\mathbb{C}}$ a tuple of shifts distinct from the poles. The corresponding rational Krylov matrices are defined as:*

$$
\begin{aligned}
K_k^{rat}(A, B, \boldsymbol{v}, \Xi, P) &= \left[\boldsymbol{v}, M(\varrho_1, \xi_1)\boldsymbol{v}, M(\varrho_2, \xi_2)M(\varrho_1, \xi_1)\boldsymbol{v}, \ldots, \left(\prod_{i=1}^{k-1} M(\varrho_i, \xi_i)\right)\boldsymbol{v}\right], \\
L_k^{rat}(A, B, \boldsymbol{v}, \Xi, P) &= \left[\boldsymbol{v}, N(\varrho_1, \xi_1)\boldsymbol{v}, N(\varrho_2, \xi_2)N(\varrho_1, \xi_1)\boldsymbol{v}, \ldots, \left(\prod_{i=1}^{k-1} N(\varrho_i, \xi_i)\right)\boldsymbol{v}\right].
\end{aligned} \tag{26}
$$

The column spaces of these matrices span the *rational Krylov subspaces.*

DEFINITION 8.2 (Rational Krylov subspaces). *The rational Krylov subspaces $\mathcal{K}_k^{rat}$ and $\mathcal{L}_k^{rat}$, $k \leq n$, associated with the $n\times n$ regular pencil (A, B), a nonzero vector $\boldsymbol{v} \in \mathbb{F}^n$, and pole tuple $\Xi = (\xi_1, \ldots, \xi_{k-1})$ distinct from the eigenvalues, are defined*

as,

$$\begin{aligned}\mathcal{K}_k^{rat}(A,B,\boldsymbol{v},\Xi) &\equiv \mathcal{R}(K_k^{rat}(A,B,\boldsymbol{v},\Xi,\mathrm{P})) = \prod_{i=1}^{k-1} M(\hat{\varrho},\xi_i)\cdot \mathcal{K}_k(M(\check{\varrho},\hat{\varrho}),\boldsymbol{v}),\\ \mathcal{L}_k^{rat}(A,B,\boldsymbol{v},\Xi) &\equiv \mathcal{R}(L_k^{rat}(A,B,\boldsymbol{v},\Xi,\mathrm{P})) = \prod_{i=1}^{k-1} N(\hat{\varrho},\xi_i)\cdot \mathcal{K}_k(N(\check{\varrho},\hat{\varrho}),\boldsymbol{v}),\end{aligned} \tag{27}$$

where the shift tuple P *is freely chosen in agreement with Definition 8.1, $\hat{\varrho}$ is a shift different from the eigenvalues and poles, and $\check{\varrho}$ is an alternative shift different from $\hat{\varrho}$.*

The first equality in (27) defines the rational Krylov subspaces, the second equality repeats [10, Lemma 5.6.II]. This result shows that rational Krylov subspaces are *shift invariant* as they are independent of the choice of shifts P.

The following theorem generalizes [10, Theorem 5.6] and shows that the rational Krylov subspaces $\mathcal{K}^{\mathrm{rat}}$ and $\mathcal{L}^{\mathrm{rat}}$ stemming from proper block Hessenberg pencils link closely to subspaces built from the standard basis vectors. This theorem is essential in proving the link with subspace iteration in Subsection 8.3.

THEOREM 8.3. *Given an $n\times n$ proper block Hessenberg pencil (A,B) with partition $\boldsymbol{s} = (s_1, \ldots, s_m)$, poles $\Xi = (\Xi^1,\ldots,\Xi^m)$ with $\Xi^i = \{\xi_1^i,\ldots\xi_{s_i}^i\}$ that are all different from the eigenvalues. Then for $j = 0,1,\ldots,m$,*

$$\mathcal{K}_{s_1+\cdots s_j+1}^{rat}(A,B,\boldsymbol{e}_1,(\Xi^1,\ldots,\Xi^j)) = \mathcal{E}_{s_1+\cdots+s_j+1}. \tag{28}$$

While for $j = 1,\ldots,m$,

$$\mathcal{L}_{s_1+\cdots+s_j}^{rat}(A,B,\boldsymbol{z}_1,(\breve{\Xi}^1,\Xi^2,\ldots,\Xi^j)) = \mathcal{E}_{s_1+\cdots+s_j}, \tag{29}$$

with $\breve{\Xi}^1 = \{\xi_1^1,\ldots,\xi_{s_1-1}^1\}$, and $\boldsymbol{z}_1$ the right eigenvector of the pole pencil corresponding to pole $\xi_{s_1}^1$. Here $\xi_{s_1}^1$ can be any of the poles in Ξ^1.

Proof. We rely on the transformation $(\hat{A},\hat{B}) = Q^*(A,B)Z$ from proper block Hessenberg pencil (A,B) to proper Hessenberg pencil $(\hat{A},\hat{B})$ as proposed in Theorem 3.8, relying on block diagonal unitary matrices. We know that Theorem 5.6 of Camps et al. [10] links rational Krylov subspaces built from Hessenberg pencils to the subspaces $\mathcal{E}_j$; the block diagonal unitary equivalence transformation messes this up a little, the equality is now only valid for particular values of j, such as proposed in (28).

Equation (29) is slightly more involved. The transformation to Hessenberg pencil requires to use a new elementary rational matrix $\hat{N}(\varrho,\xi) = Z^*N(\varrho,\xi)Z$. With this $\hat{N}$ we can fall back on [10, Theorem 5.6] to prove the theorem. ∎

Again we see that operating with block Hessenberg pencils destroys the properties in the blocks, but in between blocks theorems as in [10] remain true.

8.2. Correctness of the algorithm. We prove correctness of the procedure (see Section 4) to introduce the poles in the algorithm, that is, replacing the first ℓ poles by ℓ new ones, named the shift-poles. From (27) and Theorem 8.3 we have that,

$$\boldsymbol{x} \in \mathcal{K}_{\ell+1}^{\mathrm{rat}}(A,B,\boldsymbol{e}_1,\Xi) = \mathcal{E}_{\ell+1}. \tag{30}$$

This implies that Q in (11) is of the form $\mathrm{diag}(Q_{\ell+1}, I)$, with $Q_{\ell+1}$ an $(\ell+1)\times(\ell+1)$ unitary matrix. Furthermore, for $j = 0, 1, \ldots, m-i+1$,

$$
\begin{aligned}
&\mathcal{K}^{\mathrm{rat}}_{\hat{s}_1+\cdots+\hat{s}_j+1}(\hat{A}, \hat{B}, \boldsymbol{e}_1, (\mathrm{P}, \Xi^{i+1}, \ldots, \Xi^m)) \\
&= \prod_{k=1}^{\hat{s}_1+\cdots+\hat{s}_j} \hat{M}(\hat{\varrho}, \hat{\xi}_k) \cdot \mathcal{K}_{\hat{s}_1+\cdots+\hat{s}_j+1}(\hat{M}(\check{\varrho}, \hat{\varrho}), \boldsymbol{e}_1) \\
&= Q^* M(\hat{\varrho}, \varrho_1) \cdots M(\hat{\varrho}, \varrho_\ell) M(\hat{\varrho}, \xi_{\ell+1}) \cdots M(\hat{\varrho}, \xi_{\hat{s}_1+\cdots+\hat{s}_j}) \cdot \mathcal{K}_{\hat{s}_1+\cdots+\hat{s}_j+1}(M(\check{\varrho}, \hat{\varrho}), \boldsymbol{q}_1) \\
&= Q^* M(\hat{\varrho}, \varrho_1) \cdots M(\hat{\varrho}, \varrho_\ell) M(\hat{\varrho}, \xi_{\ell+1}) \cdots M(\hat{\varrho}, \xi_{\hat{s}_1+\cdots+\hat{s}_j}) \\
&\qquad \cdot \mathcal{K}_{\hat{s}_1+\cdots+\hat{s}_j+1}\left(M(\check{\varrho}, \hat{\varrho}), \prod_{k=1}^{\ell} M(\varrho_k, \xi_k) \right) \boldsymbol{e}_1) \\
&= Q^* M(\hat{\varrho}, \xi_1) \cdots M(\hat{\varrho}, \xi_\ell) M(\hat{\varrho}, \xi_{\ell+1}) \cdots M(\hat{\varrho}, \xi_{\hat{s}_1+\cdots+\hat{s}_j}) \cdot \mathcal{K}_{\hat{s}_1+\cdots+\hat{s}_j+1}(M(\check{\varrho}, \hat{\varrho}), \boldsymbol{e}_1) \\
&= Q^* \mathcal{K}^{\mathrm{rat}}_{\hat{s}_1+\cdots+\hat{s}_j+1}(A, B, \boldsymbol{e}_1, (\Xi^1, \ldots, \Xi^i, \Xi^{i+1}, \ldots, \Xi^m)) \\
&= Q^* \mathcal{E}_{\hat{s}_1+\cdots+\hat{s}_j+1} = \mathcal{E}_{\hat{s}_1+\cdots+\hat{s}_j+1}.
\end{aligned}
$$

In the first equality we used (27), we applied $\hat{M} = Q^* M Q$ in the second equality, and combined (10) and (11) to get $\boldsymbol{q}_1 = \prod_{k=1}^{\ell} M(\varrho_k, \xi_k)\boldsymbol{e}_1$ in the third equality. The fourth equality uses the commutativity of the M matrices and the property that $M(\hat{\varrho}, \varrho_k)M(\varrho_k, \xi_k)$ can be merged to $M(\hat{\varrho}, \xi_k)$. This results in the rational Krylov subspace of the original pencil with the original poles in the fifth equality and by Theorem 8.3 we know that this is equal to $\mathcal{E}_{\hat{s}_1+\cdots+\hat{s}_j+1}$. Finally, since Q has a block diagonal structure, it does not affect the $\mathcal{E}_{\hat{s}_1+\cdots+\hat{s}_j+1}$ for the given sizes. It is clear that $(\hat{A}, \hat{B})$ is a proper block Hessenberg pencil with partition $\hat{\boldsymbol{s}} = (\ell, s_{i+1}, \ldots, s_m)$ by construction. The last poles are unchanged by the block diagonal structure of Q and the first ℓ poles are changed to P which follows from the uniqueness of block Hessenberg pencils, see Theorem 3.8.

8.3. Convergence. In [10, Theorem 6.1] it is shown that an RQZ step with shift ϱ on a Hessenberg pencil with pole tuple $\Xi = (\xi_1, \ldots, \xi_{n-1})$ and new pole ξ_n performs nested subspace iteration accelerated by

$$
q_k^Q(z) = \frac{z-\varrho}{z-\xi_k}, \quad \text{and} \quad q_k^Z(z) = \frac{z-\varrho}{z-\xi_{k+1}}, \tag{31}
$$

for the kth column vector of respectively Q and Z. Based on Theorem 3.8 this can be extended to block Hessenberg pencils where in the multishift, multipole RQZ method convergence will now only hold for particular values of k, which correspond to the in-between block positions. Again shifts that have been swapped along the subdiagonal of the block Hessenberg pencil will lead to deflations at the end of the pencil, while poles that have been moved to the front of the pencil lead to convergence of eigenvalues at the beginning. This holds under the assumption that a good choice of poles and shifts is made.

9. Conclusion and future work. In this paper we have generalized the rational QZ method from Hessenberg to block Hessenberg pencils. This allows for the use of complex conjugate shifts and poles in real arithmetic. In the spirit of recent developments we used small shift and pole multiplicities, packed optimally together. We also implemented the aggressive early deflation strategy for block Hessenberg pencils. Numerical experiments indicated that this combination leads to an efficient

algorithm for the generalized eigenvalue problem that is competitive with state of the art implementations.

Important directions for future research include an efficient reduction to Hessenberg, Hessenberg form and investigating whether it is possible to streamline the swapping procedure so that only a single, particular swap, e.g. 2×2 with 1×1 would be used. This would simplify the code and create an additional speed up.

Acknowledgements. The authors are grateful to Paul Van Dooren and Nicola Mastronardi for their help with the iterative refinement procedure for 2×2 with 2×2 swaps [9] which was essential for handling 2×2 blocks accurately.

Reviewers commented on the article asking major changes with respect to organization, theory, and software; and we are grateful for these requests as they have significantly improved this paper.

REFERENCES

[1] B. Adlerborn, B. Kågström, and D. Kressner, *A parallel QZ algorithm for distributed memory HPC systems*, SIAM Journal on Scientific Computing, 36 (2014), pp. C480–C503.
[2] B. Adlerborn, B. Kågström, and D. Kressner, *PDHGEQZ user guide*, 2015.
[3] M. Berljafa and S. Güttel, *Generalized rational Krylov decompositions with an application to rational approximation*, SIAM Journal on Matrix Analysis and Applications, 36 (2015), pp. 894–916.
[4] L. S. Blackford, A. Petitet, R. Pozo, K. Remington, R. C. Whaley, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, et al., *An updated set of basic linear algebra subprograms (BLAS)*, ACM Transactions on Mathematical Software, 28 (2002), pp. 135–151.
[5] R. F. Boisvert, R. Pozo, K. Remington, R. F. Barrett, and J. J. Dongarra, *Matrix market: A web resource for test matrix collections*, in Proceedings of the IFIP TC2/WG2.5 Working Conference on Quality of Numerical Software: Assessment and Enhancement, London, UK, 1997, Chapman & Hall, Ltd., pp. 125–137.
[6] K. Braman, R. Byers, and R. Mathias, *The multishift QR algorithm. Part I: maintaining well-focused shifts and level 3 performance*, SIAM Journal on Matrix Analysis and Applications, 23 (2002), pp. 929–947.
[7] K. Braman, R. Byers, and R. Mathias, *The multishift QR algorithm. Part II: aggressive early deflation*, SIAM Journal on Matrix Analysis and Applications, 23 (2002), pp. 948–973.
[8] D. Camps, *Pole swapping methods for the eigenvalue problem*, PhD thesis, KU Leuven, 2019.
[9] D. Camps, N. Mastronardi, R. Vandebril, and P. Van Dooren, *Swapping 2×2 blocks in the Schur and generalized Schur form*, Journal of Computational and Applied Mathematics, 373 (2019), pp. 1–8.
[10] D. Camps, K. Meerbergen, and R. Vandebril, *A rational QZ method*, SIAM Journal on Matrix Analysis and Applications, 40 (2019), pp. 943–972.
[11] D. Camps, R. Vandebril, D. S. Watkins, and T. Mach, *On pole-swapping algorithms for the eigenvalue problem*. Submitted for publication, 2020.
[12] H. Elman, A. Ramage, and D. Silvester, *Algorithm 866: IFISS, a Matlab toolbox for modelling incompressible flow*, ACM Transactions on Mathematical Software, 33 (2007), pp. 2–14.
[13] H. Elman, A. Ramage, and D. Silvester, *IFISS: A computational laboratory for investigating incompressible flow problems*, SIAM Review, 56 (2014), pp. 261–273.
[14] J. G. F. Francis, *The QR Transformation—Part 2*, The Computer Journal, 4 (1962), pp. 332–345, https://doi.org/10.1093/comjnl/4.4.332.
[15] B. Kågström, *A direct method for reordering eigenvalues in the generalized real Schur form of a regular matrix pair (A, B)*, in Linear Algebra for Large Scale and Real-Time Applications, M. S. Moonen, G. H. Golub, and B. L. R. De Moor, eds., Springer Netherlands, Dordrecht, 1993, pp. 195–218, https://doi.org/10.1007/978-94-015-8196-7_11.
[16] B. Kågström and D. Kressner, *Multishift variants of the QZ algorithm with aggressive early deflation*, SIAM Journal on Matrix Analysis and Applications, 29 (2007), pp. 199–227.
[17] B. Kågström and P. Poromaa, *Computing eigenspaces with specified eigenvalues of a regular matrix pair (A, B) and condition estimation: theory, algorithms and software*, Numerical Algorithms, 12 (1996), pp. 369–407.

[18] L. Karlsson, D. Kressner, and B. Lang, *Optimally packed chains of bulges in multishift QR algorithms*, ACM Transactions on Mathematical Software, 40 (2014).
[19] J. G. Korvink and E. B. Rudnyi, *Oberwolfach benchmark collection*, in Dimension Reduction of Large-Scale Systems, P. Benner, D. C. Sorensen, and V. Mehrmann, eds., Berlin, Heidelberg, 2005, Springer Berlin Heidelberg, pp. 311–315.
[20] T. Mach, T. Steel, R. Vandebril, and D. S. Watkins, *Pole-swapping algorithms for alternating and palindromic eigenvalue problems*, Vietnam Journal of Mathematics, (2020).
[21] C. B. Moler and G. W. Stewart, *An algorithm for generalized matrix eigenvalue problems*, SIAM Journal on Numerical Analysis, 10 (1973), pp. 1–52.
[22] D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM Journal on Matrix Analysis and Applications, 13 (1992), pp. 357–385.
[23] P. Van Dooren, *A generalized eigenvalue approach for solving Riccati equations*, SIAM Journal on Statistical Computing, 2 (1981), pp. 121–135, https://doi.org/10.1137/0902010.
[24] E. Wang, Q. Zhang, B. Shen, G. Zhang, X. Lu, Q. Wu, and Y. Wang, *Intel math kernel library*, in High-Performance Computing on the Intel® Xeon Phi, Springer, 2014, pp. 167–188.
[25] D. S. Watkins, *The transmission of shifts and shift blurring in the QR algorithm*, Linear Algebra and its Applications, 241-243 (1996), pp. 877–896.
[26] D. S. Watkins, *The Matrix Eigenvalue Problem: GR and Krylov Subspace Methods*, SIAM, Philadelphia, USA, 2007.