Electronic Transactions on Numerical Analysis.
Volume 52, pp. 480–508, 2020.

ISSN 1068–9613.
DOI: 10.1553/etna_vol52s480

ETNA
Kent State University and
Johann Radon Institute (RICAM)

ON POLE-SWAPPING ALGORITHMS FOR THE EIGENVALUE PROBLEM*

DAAN CAMPS†, THOMAS MACH‡, RAF VANDEBRIL§, AND DAVID S. WATKINS¶

Abstract. Pole-swapping algorithms, which are generalizations of the QZ algorithm for the generalized eigenvalue problem, are studied. A new modular (and therefore more flexible) convergence theory that applies to all pole-swapping algorithms is developed. A key component of all such algorithms is a procedure that swaps two adjacent eigenvalues in a triangular pencil. An improved swapping routine is developed, and its superiority over existing methods is demonstrated by a backward error analysis and numerical tests. The modularity of the new convergence theory and the generality of the pole-swapping approach shed new light on bi-directional chasing algorithms, optimally packed shifts, and bulge pencils, and allow the design of novel algorithms.

Key words. eigenvalue, QZ algorithm, pole swapping, convergence

AMS subject classifications. 65F15, 15A18

1. Introduction.

The standard algorithm for computing the eigenvalues of a small- to medium-sized non-Hermitian matrix $A \in \mathbb{C}^{n\times n}$ is still Francis's implicitly-shifted QR algorithm [14, 33]. In many applications, eigenvalue problems arise naturally as generalized eigenvalue problems for a pencil $A - \lambda B$, and for these problems the Moler-Stewart variant of Francis's algorithm [24], commonly called the QZ algorithm, can be used. In this paper we may refer sometimes to a pencil $A - \lambda B$ and other times to a pair (A, B). Either way, we are talking about the same object.

A few years ago we published a generalization of the QZ algorithm [27]. More recently, an even more general algorithm, the *rational QZ (RQZ) algorithm*, was presented by Camps, Meerbergen, and Vandebril [13]. This arose from the study of rational Arnoldi methods and is related to work of Berljafa and Güttel [6].

In this paper we discuss the RQZ algorithm and introduce several variants. We develop a new modular (and therefore more flexible) convergence theory that can be applied immediately to all variants. We reinterpret the QZ algorithm and show that it can be viewed as a pole-swapping algorithm with poles at infinity. Moreover, we will show that the algorithm [20] for optimally packed chains of bulges is a disguised implementation of pole swapping. A key component of the RQZ and related algorithms is a procedure that swaps two adjacent eigenvalues in a triangular pencil. We present an improved swapping routine and demonstrate its superiority by numerical experiments and a backward error analysis.

Double-shift pole-swapping algorithms that can be applied to real matrix pencils exist [12, 25]. All of what is discussed in this paper for single shifts can be extended to the double-shift case, but we have not worked out every detail. The one item that will require further thought is the extension of the improved swapping routine of Section 8 to blocks larger than 1×1. A significant advantage of sticking to the complex single-shift case, as we have done here, is simplicity and clarity of presentation.

*Received January 8, 2020. Accepted July 6, 2020. Published online on September 18, 2020. Recommended by F. Dopico. This research was partially supported by the Research Council KU Leuven, project C14/16/056 (Inverse-free Rational Krylov Methods: Theory and Applications).
†Computational Research Division, Lawrence Berkeley National Laboratory, California (dcamps@lbl.gov).
‡Department of Mathematical Sciences, Kent State University, Ohio (tmach1@kent.edu).
§Department of Computer Science, KU Leuven, Belgium (raf.vandebril@cs.kuleuven.be).
¶Department of Mathematics, Washington State University (watkins@math.wsu.edu)

ETNA
Kent State University and
Johann Radon Institute (RICAM)

2. Hessenberg pairs. A pencil $A - \lambda B$ is called a *regular pencil* or *regular pair* if there is at least one complex μ such that $A - \mu B$ is invertible. Throughout this paper we make the blanket assumption of regularity.

A matrix $A \in \mathbb{C}^{n\times n}$ is in (upper) *Hessenberg form* if every entry below the first subdiagonal is zero. It is in *proper* Hessenberg form if every subdiagonal entry is nonzero, i.e., $a_{j+1,j} \neq 0$, for $j = 1, \ldots, n-1$. A preliminary step for the QZ algorithm is to reduce the pair (A, B) to Hessenberg-triangular form. That is, (A, B) is transformed by a unitary equivalence to a new pair $(\check{A}, \check{B})$ for which $\check{A}$ is upper Hessenberg and $\check{B}$ is upper triangular. Notice that if $\check{A}$ is not properly Hessenberg, then the eigenvalue problem can be split immediately into two or more independent subproblems. Thus, we can always assume that we are dealing with a matrix in proper Hessenberg form.

In the new theory we deal with a more general class of Hessenberg pencils. The pair (A, B) is called a *Hessenberg pair* if both A and B are Hessenberg matrices. If $a_{j+1,j} = 0 = b_{j+1,j}$ for some j, then we can immediately split the eigenvalue problem into two smaller problems. We therefore eliminate that case from further consideration. For reasons that will become apparent later, the ratios $a_{j+1,j}/b_{j+1,j}$, $j = 1, \ldots, n-1$, are called the *poles* of the Hessenberg pair (A, B). In the case $b_{j+1,j} = 0$, we have an infinite pole. The Hessenberg-triangular form is a special Hessenberg pair for which all of the poles are infinite.

Closely related to (A, B) is the *pole pair* (A_π, B_π) (or *pole pencil* $A_\pi - \lambda B_\pi$) obtained from (A, B) by deleting the first row and last column. The pole pencil is upper triangular, and its eigenvalues are obviously the poles of (A, B).

Operations on Hessenberg pairs. Introducing terminology that we have used in some of our recent work [1, 2, 3, 4], we define a *core transformation* (or *core* for short) to be a unitary matrix that acts only on two adjacent rows/columns, for example,

$$Q_3 = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & * & * & \\ & & * & * & \\ & & & & 1 \end{bmatrix},$$

where the four asterisks form a 2×2 unitary matrix. Givens rotations are examples of core transformations. Our core transformations always have subscripts that tell where the action is: Q_j acts on rows/columns j and $j + 1$.

Following [13] we introduce two types of operations, or *moves*, both of which manipulate the poles in the pair. Let $\sigma_1 = a_{21}/b_{21}, \ldots, \sigma_{n-1} = a_{n,n-1}/b_{n,n-1}$ denote the poles of the Hessenberg pair (A, B).

Changing a pole at the top or bottom. (Type I move). We can change the pole σ_1 to any value we want by applying a core transformation Q_1^* to the pencil on the left. Suppose we want to change σ_1 to ρ, say. Noting that only the first two entries of $(A - \rho B)e_1$ can be nonzero, we deduce that there is a Q_1 such that the second entry of $Q_1^*(A - \rho B)e_1$ is zero. In other words,

$$Q_1^*(A - \rho B)e_1 = \gamma e_1 \tag{2.1}$$

for some γ. If we then define $\hat{A} = Q_1^* A$ and $\hat{B} = Q_1^* B$, then $(\hat{A} - \rho\hat{B})e_1 = \gamma e_1$, which implies that $\hat{a}_{21} - \rho\hat{b}_{21} = 0$. This means that $\rho = \hat{a}_{21}/\hat{b}_{21}$ is the new first pole of $(\hat{A}, \hat{B})$. The other poles remain fixed, as they are untouched by the transformation.

This operation fails only if $\hat{a}_{21} = 0 = \hat{b}_{21}$, yielding $\rho = 0/0$. This happens exactly when the first columns of A and B are proportional. But this is not such a failure after all, as it

ETNA
Kent State University and
Johann Radon Institute (RICAM)

exposes $\hat{a}_{11}/\hat{b}_{11}$ as an eigenvalue of the pencil and allows us to deflate to a smaller problem by deleting the first row and column.

In summary, if we want to replace the pole σ_1 by ρ, we will either succeed in doing so or get a deflation of an eigenvalue.

REMARK 2.1. When we write something like $A - \rho B$ here and elsewhere, this should be viewed as shorthand for $\beta A - \alpha B$ where α and β are any scalars for which $\rho = \alpha/\beta$. As a practical matter this allows us to use modest-sized α and β even when ρ is very large, and in particular it allows us to implement the case $\rho = \infty$ by taking $\beta = 0$.

The pole σ_{n-1} at the bottom can also be replaced by any other pole, say τ, by a similar procedure. We want to transform the pencil $A - \lambda B$ to $\hat{A} - \lambda \hat{B} = (A - \lambda B)Z_{n-1}$ with $\hat{a}_{n,n-1}/\hat{b}_{n,n-1} = \tau$. Noting that the row vector $e_n^T(A - \tau B)$ has nonzero entries only in its last two positions, we see that there must be a core transformation Z_{n-1} that maps it to a multiple of e_n^T, i.e., $e_n^T(A - \tau B)Z_{n-1} = \gamma e_n^T$ for some γ. This is the desired transformation since it implies $e_n^T(\hat{A} - \tau \hat{B}) = \gamma e_n^T$, which is equivalent to $\hat{a}_{n,n-1}/\hat{b}_{n,n-1} = \tau$.

This fails only if $\hat{a}_{n,n-1} = 0 = \hat{b}_{n,n-1}$, yielding $\tau = 0/0$, which happens exactly when the nth rows of A and B are proportional. But again this is not really a failure at all, since it allows $\hat{a}_{nn}/\hat{b}_{nn}$ to be extracted as an eigenvalue and the problem to be deflated to a smaller one.

This discussion helps motivate the following definition. A Hessenberg pair is called a *proper Hessenberg pair* if three conditions hold: (i) $|a_{j+1,j}| + |b_{j+1,j}| > 0$, for $j = 1, \ldots, n-1$, (ii) the first columns of A and B are not proportional, (iii) the last rows of A and B are not proportional. The first condition just says that for each j, at least one of $a_{j+1,j}$ and $b_{j+1,j}$ is nonzero. If this condition is not satisfied, then we can immediately reduce the pencil to two smaller pencils. If either of conditions (ii) and (iii) is not satisfied, then we can also reduce the problem as we know from the discussion immediately above. Therefore, we can always assume, without loss of generality, that we are working with a proper Hessenberg pair.

PROPOSITION 2.2 ([13]). *In a proper Hessenberg pair, the core transformation Q_1 that replaces pole σ_1 by ρ satisfies*

$$Q_1 e_1 = \delta\,(A - \rho B)(A - \sigma_1 B)^{-1} e_1$$

for some nonzero δ.

Proof. From our construction we have $Q_1 e_1 = \gamma^{-1}(A - \rho B)e_1$. Since σ_1 is the first pole of the pair (A, B), we have $(A - \sigma_1 B)e_1 = \check{\gamma} e_1$ for some $\check{\gamma}$. The properness assumption guarantees that both γ and $\check{\gamma}$ are nonzero. Therefore, $Q_1 e_1 = \delta(A - \rho B)(A - \sigma_1 B)^{-1} e_1$, where $\delta = (\gamma\check{\gamma})^{-1}$. ∎

REMARK 2.3. The insertion of the extra factor $(A - \sigma_1 B)^{-1}$ may seem mysterious. As we shall see later, this is just what is needed for a consistent convergence theory. In the product $(A - \rho B)(A - \sigma_1 B)^{-1}$, the factor $A - \rho B$ signals that the pole ρ is entering the pencil, while the factor $(A - \sigma_1 B)^{-1}$ signals that the pole σ_1 is leaving.

PROPOSITION 2.4 ([13]). *In a proper Hessenberg pair, the core transformation Z_{n-1} that replaces pole σ_{n-1} by τ satisfies*

$$e_n^T Z_{n-1}^* = \delta\, e_n^T (A - \sigma_{n-1} B)^{-1}(A - \tau B)$$

for some nonzero δ.

Proof. From our construction we have $e_n^T Z_{n-1}^* = \gamma^{-1} e_n^T (A - \tau B)$. Since σ_{n-1} is the last pole of the pair (A, B), we have $e_n^T (A - \sigma_{n-1} B) = \check{\gamma} e_n^T$ for some nonzero $\check{\gamma}$. Therefore, $e_n^T Z_{n-1}^* = \delta e_n^T (A - \sigma_{n-1} B)^{-1}(A - \tau B)$, where $\delta = (\gamma\check{\gamma})^{-1}$. ∎

ETNA
Kent State University and
Johann Radon Institute (RICAM)

The arithmetic cost of a move of type I is just the cost of multiplying A and B by a single core transformation, Q_1^* or Z_{n-1}. If the cores are Givens rotations applied in the conventional way, then the cost is about $8n$ multiplications and $4n$ additions, or $12n$ (complex) flops. Different implementations could yield slightly different flop counts, but regardless of the details, the cost will be $O(n)$. Standard backward error analysis [34] shows that moves of type I are backward stable.

Interchanging two poles. (Type II move). The second of the two allowed operations is to interchange two adjacent poles by a unitary equivalence $\hat{A} - \lambda\hat{B} = Q_j^*(A - \lambda B)Z_{j-1}$. To understand this, consider the pole pencil $A_\pi - \lambda B_\pi$ obtained by discarding the first row and last column from $A - \lambda B$. This pencil is upper triangular and has $\sigma_1, \ldots, \sigma_{n-1}$ as its eigenvalues. There are standard techniques [5, 18, 19, 26], [31, §§ 4.8, 6.6] for interchanging any two adjacent eigenvalues σ_{j-1} and σ_j. We will describe an improved method in Section 8. Each of these requires only an equivalence transformation $\tilde{Q}_{j-1}^*(A_\pi - \lambda B_\pi)\tilde{Z}_{j-1}$ by two core transformations $\tilde{Q}_{j-1}$ and $\tilde{Z}_{j-1}$ of dimension $n-1$. We then enlarge these matrices by adjoining a row and column to the top of $\tilde{Q}_{j-1}$ and the bottom of $\tilde{Z}_{j-1}$:

$$Q_j = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{Q}_{j-1} \end{bmatrix} \qquad Z_{j-1} = \begin{bmatrix} \tilde{Z}_{j-1} & 0 \\ 0 & 1 \end{bmatrix}.$$

Then $\hat{A} - \lambda\hat{B} = Q_j^*(A - \lambda B)Z_{j-1}$ is the desired transformation.

If the swap is done as described by Van Dooren [26], then the procedure always succeeds and is backward stable in a sense. Our new swapping procedure will be shown to have improved stability. In order not to interrupt the flow of the paper, we defer the description of the new procedure, as well as a discussion of backward errors, to Section 8.

The flop count for a move of type II is about the same as for a move of type I, namely $12n$ if the core transformations are implemented as Givens rotations. In any event, the flop counts for moves of type I and II are about the same, and each move costs $O(n)$ flops.[1]

REMARK 2.5. We have one type of move that is able to change a pole at one end or the other and another type that swaps poles in the middle. It is natural to ask whether we can devise a move that changes a single pole in the middle. The answer is *no*. Consider a transformation

$$\hat{A} - \lambda\hat{B} = Q^*(A - \lambda B)Z, \tag{2.2}$$

where Q does not touch the first row and Z does not touch the last column. That is,

$$Q = \begin{bmatrix} 1 & \\ & \tilde{Q} \end{bmatrix} \qquad \text{and} \qquad Z = \begin{bmatrix} \tilde{Z} & \\ & 1 \end{bmatrix}.$$

Under any such transformation, the poles must remain invariant. This is so because the transformation (2.2) is equivalent to a transformation $\tilde{Q}^*(A_\pi - \lambda B_\pi)\tilde{Z}$ on the pole pencil. Since the poles of $A - \lambda B$ are the eigenvalues of the pole pencil, they must remain fixed.

Thus, any transformation meant to change a pole must touch either the first row or the last column. That's what the moves of type I do.

[1] This is the correct count for the case when only eigenvalues are being computed. If eigenvectors or some deflating subspaces are wanted as well, then the transforming matrices Q and Z also need to be updated on each move. This adds about $6n$ (complex) flops for a type I move and $12n$ flops for a type II, but the total is still $O(n)$.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

3. Building an algorithm from the pieces. Suppose we want to find the eigenvalues of some regular pair (A, B). As usual, there are two steps to the process. The first is a direct method that transforms (A, B) to a condensed form, in our case a Hessenberg pencil. The second step is an iterative process that uncovers the eigenvalues of the condensed form.

In some contexts the reduction phase can be skipped. As a notable example, the rational Arnoldi process [6] applied to a large matrix naturally generates, after k steps, a $k \times k$ Hessenberg pencil. The ith pole of the pencil is equal to the shift that was used in the ith step of the process. We can obtain estimates of the eigenvalues of the large matrix by computing the eigenvalues of the pencil. This requires no reduction; we can go directly to the iterative phase.

Reduction to a Hessenberg pencil. Moler and Stewart [24] showed how to reduce (A, B) to Hessenberg-triangular form by a direct method in $O(n^3)$ flops. The reduction is also described in [15, 31, 32] and elsewhere. If the resulting pair is not proper, then we can split it into smaller proper pairs, so let us assume it is proper. This is a Hessenberg pencil with all poles equal to ∞. If the user is happy to start from this configuration, s/he can move directly to the iterative phase.

If the user wants to set certain prescribed poles $\sigma_1, \ldots, \sigma_{n-1}$ before beginning the iterations, then this is also possible. One obvious procedure is to begin by introducing σ_{n-1} at the top of the pencil by a move of type I. Then σ_{n-1} can be swapped with each of the remaining infinite poles by moves of type II until it arrives at its desired position at the bottom. The total number of moves is $n-1$. Then σ_{n-2} can be introduced at the top by a move of type I. It can then be swapped with each of the remaining infinite poles until it arrives at its desired position just above σ_{n-1}. The total number of moves for this step is $n-2$. Then σ_{n-3} can be introduced, and so on. Eventually, we get each of $\sigma_1, \ldots, \sigma_{n-1}$ into its desired position. The total number of moves for this phase is about $n^2/2$, and the total flop count is $O(n^3)$.

One can equally well introduce the poles at the bottom and swap them upward, starting with σ_1, then σ_2, and so on. The amount of work is exactly the same, about $n^2/2$ moves. Better yet, one can take $k \approx (n-1)/2$ and introduce $\sigma_1, \ldots, \sigma_k$ (in reverse order) at the top and $\sigma_{k+1}, \ldots, \sigma_{n-1}$ at the bottom. This cuts the number of moves in half. However one does it, the cost is $O(n^3)$.

Camps, Meerbergen, and Vandebril [13] describe a procedure that introduces the poles during the reduction to Hessenberg form. They also present an example where a good choice of poles induces a deflation in the middle of the pencil.

The iterative phase (basic algorithm). During the discussion of moves of type I in Section 2, we defined *proper* Hessenberg pairs and noted that if a Hessenberg pair is not proper, then it can be reduced to smaller pairs that are. We therefore assume, without loss of generality, that we have a proper Hessenberg pair (A, B) with poles $\sigma_1, \ldots, \sigma_{n-1}$. We now describe an iteration of the RQZ algorithm proposed in [13]. We will call this *the basic algorithm*.

First a shift ρ is chosen. Any of the usual shifting strategies can be employed here. The simplest is the Rayleigh-quotient shift $\rho = a_{nn}/b_{nn}$. Then, ρ is introduced as a pole at the top of the pencil, replacing σ_1, by a move of type I. Next ρ is swapped with σ_2 by a move of type II. Then another move of type II is used to swap ρ with σ_3, and so on. After $n-2$ moves of type II, ρ arrives at the bottom of the pencil. The poles are now $\sigma_2, \ldots, \sigma_{n-1}$, and ρ. Finally, a move of type I is used to remove the pole ρ from the bottom, replacing it by a new pole σ_n. This completes the iteration. The user has complete flexibility in the choice of σ_n. One possibility is $\sigma_n = \infty$. Another, which might be called a *Rayleigh-quotient pole*, is $\sigma_n = a_{11}/b_{11}$.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

The cost of one iteration of the basic algorithm is n moves or $O(n^2)$ flops. With any of the standard shifting strategies, e.g., Rayleigh-quotient shift, repeated iterations will normally cause rapid convergence of an eigenvalue at the bottom of the pencil. Typically $a_{n,n-1} \to 0$ and $b_{n,n-1} \to 0$ quadratically, leaving a_{nn}/b_{nn} as an eigenvalue and allowing deflation of the problem. After $n-1$ deflations, all of the eigenvalues will have been found.

There are numerous variations on the basic algorithm. For example, it can be turned upside down. We can pick a shift, say $\rho = a_{11}/b_{11}$, insert it at the bottom of the pencil, and chase it to the top. Since we can do this, then why not chase shifts in both directions at once? Some possibilities along these lines will be discussed in Section 6.

Relationship to the QZ algorithm. We now show that when the basic algorithm is applied to a pair that has all poles infinity, it reduces to the single-shift version of the Moler/Stewart QZ algorithm. Consider a Hessenberg-triangular pair

$$\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \end{bmatrix} \qquad \begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix}$$

which has poles ∞, ∞, and ∞. An iteration of the basic algorithm begins by choosing a shift ρ and inserting it into the pair at the top by a move of type I. The transformation is $A \to Q_1^* A$, $B \to Q_1^* B$, where Q_1 satisfies (2.1). This is exactly the same as the transformation that starts single-shift QZ [32, p. 537]. It alters the first two rows of the matrices, so the transformed matrices have the form

$$\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \end{bmatrix} \qquad \begin{bmatrix} \times & \times & \times & \times \\ + & \times & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix} . \tag{3.1}$$

The triangular form of B has been disturbed, but this is still a Hessenberg pair. Its poles are ρ, ∞, ∞. (We will continue to refer to the matrices as "A" and "B", even though they change in the course of the iteration.) The next step of the basic algorithm is a move of type II that interchanges the pole ρ with the adjacent pole ∞, resulting in

$$\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \end{bmatrix} \qquad \begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & + & \times & \times \\ & & & \times \end{bmatrix} , \tag{3.2}$$

a Hessenberg pair with poles ∞, ρ, ∞. The transformation has the form $A \to Q_2^* A Z_1$, $B \to Q_2^* B Z_1$, with appropriately chosen core transformations Z_1 and Q_2. Let us consider now how things look if we apply the cores one at a time. Starting from the configuration shown in (3.1), first apply Z_1 on the right. This acts on columns one and two of each matrix and produces

$$\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ + & \times & \times & \times \\ & & \times & \times \end{bmatrix} \qquad \begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix} .$$

The entry b_{21} must now be zero. This is so because, as we know, after the application of Q_2^* on the left, b_{21} must be zero, as shown in (3.2). The left multiplication by Q_2^* cannot do this job, so it must have been done by Z_1. At the same time, Z_1 must produce a bulge at a_{31}. This proves that Z_1 is exactly the same transformation as is used at this point in the QZ bulge chase.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Now, when we apply Q_2^* on the left, it operates on rows two and three. It must set a_{31} to zero and create a new bulge at b_{32} to arrive at (3.2). Thus, Q_2 is exactly the same transformation as is used at this point in the QZ bulge chase.

The next step is a move of type II that transforms (3.2) to

(3.3)
$$\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \end{bmatrix} \qquad \begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & + & \times \end{bmatrix},$$

a Hessenberg pair with poles ∞, ∞, ρ. The transformation has the form $A \to Q_3^* A Z_2$, $B \to Q_3^* B Z_2$. Again we could look at what happens if we apply the cores one at a time, first Z_2, then Q_3^*, and we would find as before that these are exactly the same transformations as in a QZ bulge chase.

In our little example, we have now reached the bottom. In a larger example, we would continue moves of type II, pushing the pole ρ downward, and at each step we would have the same situation. The final step is a move of type I that removes ρ from the bottom of the pencil, replacing it by a pole ∞. This is exactly the transform, acting on columns $n-1$ and n, that sets $b_{n,n-1}$ (the entry b_{43} entry in (3.3)) to zero. Again this is exactly the same as the transformation that completes the QZ bulge chase. The pair is now in Hessenberg-triangular form.

We have demonstrated that the basic algorithm reduces to the single-shift QZ algorithm in the case when all of the poles are infinite.

4. Convergence theory. In the convergence theorems in this paper we make the blanket (and generically valid) assumption that none of the poles or shifts that are mentioned are eigenvalues of the pencil. We often find it convenient to assume that B is nonsingular.

The mechanism that drives all variants of Francis's algorithm is nested subspace iteration with changes of the coordinate system [32, p. 431], [33, p. 399], [1, Thm 2.2.3]. As a specific example, let us consider a single step of the QZ algorithm with shift ρ applied to a Hessenberg-triangular pencil $A - \lambda B$, yielding a new pencil $\hat{A} - \lambda \hat{B}$ with

(4.1)
$$\hat{A} - \lambda \hat{B} = Q^*(A - \lambda B)Z.$$

First we define some nested sequences of subspaces. For $k = 1, \ldots, n$, define

$$\mathcal{E}_k = \text{span}\{e_1, \ldots, e_k\},$$

where $e_1, \ldots, e_n$ are the standard basis vectors. Then define

$$\mathcal{Q}_k = Q\mathcal{E}_k \qquad \text{and} \qquad \mathcal{Z}_k = Z\mathcal{E}_k.$$

Thus, $\mathcal{Q}_k$ (resp. $\mathcal{Z}_k$) is the space spanned by the first k columns of Q (resp. Z).

THEOREM 4.1. *A single step of the QZ algorithm with shift ρ effects the nested subspace iterations*

$$\mathcal{Q}_k = (AB^{-1} - \rho I)\mathcal{E}_k, \qquad \mathcal{Z}_k = (B^{-1}A - \rho I)\mathcal{E}_k, \qquad k = 1, \ldots, n-1.$$

The change of the coordinate system (4.1) transforms both $\mathcal{Q}_k$ and $\mathcal{Z}_k$ back to $\mathcal{E}_k$.

We call this a *convergence theorem* even though it makes no mention of convergence. Theorems like this can be used together with the convergence theory of subspace iteration to draw conclusions about the convergence of the algorithm as explained in [31, 32, 33] and elsewhere.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Camps, Meerbergen, and Vandebril [13, Thm. 6.1] proved a result like Theorem 4.1 for the basic algorithm. The scenario is similar. The iteration begins with a proper Hessenberg pair (A, B) with poles $\sigma_1, \ldots, \sigma_{n-1}$, employs a shift ρ, and ends with a new proper Hessenberg pair $(\hat{A}, \hat{B})$ with poles $\sigma_2, \ldots, \sigma_n$. The old and new pairs are related by a unitary equivalence transformation of the form (4.1).

THEOREM 4.2. *A single step of the basic algorithm with shift ρ, starting with a proper Hessenberg pair (A, B) with poles $\sigma_1, \ldots, \sigma_{n-1}$ and ending with $(\hat{A}, \hat{B})$ with poles $\sigma_2, \ldots, \sigma_n$ effects the nested subspace iterations*

$$\mathcal{Q}_k = (A-\rho B)(A-\sigma_k B)^{-1}\mathcal{E}_k, \quad \mathcal{Z}_k = (A-\sigma_{k+1}B)^{-1}(A-\rho B)\mathcal{E}_k, \quad k = 1, \ldots, n-1.$$

The change of the coordinate system (4.1) transforms both $\mathcal{Q}_k$ and $\mathcal{Z}_k$ back to $\mathcal{E}_k$.

This theorem was proved in [13], but we will also provide a proof based on our new theory in Section 5. Comparing this with Theorem 4.1, we see that the inclusion of poles gives extra freedom that might be used to improve convergence.

Now consider Theorem 4.2 in the case when all of the poles are infinite. When $\sigma_k = \infty$, the operator $(A-\rho B)(A-\sigma_k B)^{-1}$ becomes (when appropriately rescaled) $(A-\rho B)B^{-1} = AB^{-1} - \rho I$. Similarly, $(A-\sigma_{k+1}B)^{-1}(A-\rho B)$ becomes $B^{-1}(A-\rho B) = B^{-1}A - \rho I$. These operators are exactly the ones that appear in Theorem 4.1, just as we would expect.

Although the QZ algorithm is a special case of the basic algorithm, there is an important difference in their implementation. The QZ algorithm acts on proper Hessenberg-triangular pencils. It is a bulge-chasing algorithm. The initial equivalence transformation of each iteration creates a bulge in the Hessenberg-triangular form. The rest of the iteration consists of equivalence transformations that chase the bulge back and forth between A and B until it finally disappears off the bottom of the pencil. At that point, the Hessenberg-triangular form has been restored, and the iteration is complete. The QZ algorithm can also be implemented as a *core-chasing* algorithm as is shown in [1] and [3], but the situation is the same: The Hessenberg-triangular form is disturbed at the beginning of the iteration and not restored until the very end.

Now let us contrast this with what happens in the basic algorithm (with infinite poles or otherwise). The basic algorithm operates on proper Hessenberg pairs, in which neither matrix is required to be triangular. Each iteration starts with a move of type I, performs a sequence of moves of type II, and ends with a move of type I. These moves do not disturb the Hessenberg form; it is preserved throughout. This implies that we can think of each move as a "mini iteration" and ask whether we can obtain a result like Theorem 4.1 or 4.2 for each individual move of type I or II. It turns out that we can.

Each move of either type is an equivalence transform of the form

$$\hat{A} = Q_j^* A Z_{j-1} \qquad \hat{B} = Q_j^* B Z_{j-1}.$$

The case $j = 1$ denotes a move of type I, and we have $Z_0 = I$. The case $j = n$ also denotes a type I move, and in this case $Q_n = I$. The cases $j = 2, \ldots, n-1$ are of type II. Suppose (A, B) has poles $\sigma_1, \ldots, \sigma_{n-1}$. A move of type II interchanges the poles σ_{j-1} and σ_j. For the moves of type I, in the case $j = 1$, suppose the pole σ_1 is replaced by a new pole σ_0; in the case $j = n$, suppose σ_{n-1} is replaced by a new pole σ_n. With this notation we can cover both types of move by a single theorem.

As above we define sequences of nested subspaces $(\mathcal{Q}_k)$ and $(\mathcal{Z}_k)$, where $\mathcal{Q}_k$ (resp. $\mathcal{Z}_k$) is the space spanned by the first k columns of Q_j (resp. Z_{j-1}). But note that, because Q_j and

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Z_{j-1} are core transformations, these spaces are mostly trivial in this setting: $\mathcal{Q}_k = \mathcal{E}_k$ except when $k = j$, and $\mathcal{Z}_k = \mathcal{E}_k$ except when $k = j-1$.

THEOREM 4.3. *Using notation and terminology established directly above, the move*

$$\hat{A} - \lambda\hat{B} = Q_j^*(A - \lambda B)Z_{j-1} \tag{4.2}$$

effects nested subspace iterations that are, however, mostly trivial. The nontrivial actions are

$$\mathcal{Q}_j = (A - \sigma_{j-1}B)(A - \sigma_j B)^{-1}\mathcal{E}_j$$

and

$$\mathcal{Z}_{j-1} = (A - \sigma_j B)^{-1}(A - \sigma_{j-1}B)\mathcal{E}_{j-1}.$$

The change of the coordinate system (4.2) *transforms* $\mathcal{Q}_j$ *back to* $\mathcal{E}_j$ *and* $\mathcal{Z}_{j-1}$ *back to* $\mathcal{E}_{j-1}$.

The proof of Theorem 4.3 makes use of rational Krylov subspaces. Given $C \in \mathbb{C}^{n\times n}$ and $v \in \mathbb{C}^n$, the standard *Krylov subspaces* $\mathcal{K}_j(C, v)$ are defined by

$$\mathcal{K}_j(C, v) = \text{span}\big\{v, Cv, C^2v, \ldots, C^{j-1}v\big\}, \qquad j = 1, 2, \ldots, n.$$

Given an ordered set of poles $[\sigma_1, \sigma_2, \ldots, \sigma_{n-1}]$, none in the spectrum of C, the *rational Krylov subspaces* $\mathcal{K}_j(C, v, [\sigma_1, \ldots, \sigma_{j-1}])$ are defined by

$$\begin{aligned}
\mathcal{K}_1(C, v, [\,]) &= \text{span}\{v\},\\
\mathcal{K}_2(C, v, [\sigma_1]) &= \text{span}\big\{v, (C - \sigma_1 I)^{-1}v\big\},\\
\mathcal{K}_3(C, v, [\sigma_1, \sigma_2]) &= \text{span}\big\{v, (C - \sigma_1 I)^{-1}v, (C - \sigma_2 I)^{-1}(C - \sigma_1 I)^{-1}v\big\},
\end{aligned}$$

and in general

$$\mathcal{K}_j(C, v, [\sigma_1, \ldots, \sigma_{j-1}]) = \text{span}\Big\{v, (C - \sigma_1 I)^{-1}v, \ldots, \Big(\textstyle\prod_{i=1}^{j-1}(C - \sigma_i I)^{-1}\Big)\, v\Big\}.$$

Making the abbreviation $C(\sigma) = C - \sigma I$, we can rewrite this as

$$\mathcal{K}_j(C, v, [\sigma_1, \ldots, \sigma_{j-1}]) = \prod_{i=1}^{j-1} C(\sigma_i)^{-1}\, \text{span}\Big\{\textstyle\prod_{i=1}^{j-1} C(\sigma_i)v, \prod_{i=2}^{j-1} C(\sigma_i)v, \ldots, v\Big\}.$$

The span on the right-hand side involves only positive powers of C, so the shifts are irrelevant; it is just the standard Krylov subspace $\mathcal{K}_j(C, v)$. Therefore,

$$\mathcal{K}_j(C, v, [\sigma_1, \ldots, \sigma_{j-1}]) = \left(\prod_{i=1}^{j-1}(C - \sigma_i I)^{-1}\right)\mathcal{K}_j(C, v). \tag{4.3}$$

Given a pair (A, B) with B nonsingular, we define *rational Krylov subspaces*

$$\mathcal{K}_j(A, B, v, [\sigma_1, \ldots, \sigma_{j-1}]) \qquad \text{and} \qquad \mathcal{L}_j(A, B, v, [\sigma_1, \ldots, \sigma_{j-1}])$$

associated with the pair by

$$\mathcal{K}_j(A, B, v, [\sigma_1, \ldots, \sigma_{j-1}]) = \mathcal{K}_j(AB^{-1}, v, [\sigma_1, \ldots, \sigma_{j-1}])$$

and

$$\mathcal{L}_j(A, B, v, [\sigma_1, \ldots, \sigma_{j-1}]) = \mathcal{K}_j(B^{-1}A, v, [\sigma_1, \ldots, \sigma_{j-1}]),$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

$j = 1, \ldots, n$. We have assumed for convenience that B is nonsingular. See [13] for a definition of these spaces that does not require this assumption. We are using the symbol $\mathcal{K}_j$ to denote several different types of Krylov subspaces. The meaning in each case is uniquely determined by the number and type of arguments.

We will make use of the following result, which is Theorem 5.6 in [13].

PROPOSITION 4.4. *Let (A, B) be a proper upper Hessenberg pair with poles $[\sigma_1, \ldots, \sigma_{n-1}]$. Let $\mathcal{E}_j = \operatorname{span}\{e_1, \ldots, e_j\}$ as before. Then, for $j = 1, \ldots, n-1$,*

$$\mathcal{E}_j = \mathcal{K}_j(A, B, e_1, [\sigma_1, \ldots, \sigma_{j-1}]) = \mathcal{L}_j(A, B, e_1, [\sigma_2, \ldots, \sigma_j]).$$

See [13] for the proof. Notice that in the $\mathcal{L}_j$-spaces, the poles are $[\sigma_2, \ldots, \sigma_j]$, starting from σ_2. With Proposition 4.4 in hand, we can prove Theorem 4.3.

Proof of Theorem 4.3. Proposition 2.2 shows that $Q_1 e_1 = \delta\,(A - \sigma_0 B)(A - \sigma_1 B)^{-1} e_1$ for some nonzero δ. This establishes the case $j = 1$ of Theorem 4.3.

Now consider $j > 1$. The transformation $\hat{A} - \lambda\hat{B} = Q_j^*(A - \lambda B)Z_{j-1}$ interchanges the poles σ_{j-1} and σ_j, so the ordered pole set of $(\hat{A}, \hat{B})$ is

$$[\sigma_1, \ldots, \sigma_{j-2}, \sigma_j, \sigma_{j-1}, \sigma_{j+1}, \ldots, \sigma_{n-1}].$$

Applying Proposition 4.4 to $(\hat{A}, \hat{B})$ we have

$$\mathcal{E}_j = \mathcal{K}_j(\hat{A}, \hat{B}, e_1, [\sigma_1, \ldots, \sigma_{j-2}, \sigma_j]).$$

Therefore,

$$\begin{aligned} \mathcal{Q}_j = Q_j \mathcal{E}_j &= Q_j \mathcal{K}_j(\hat{A}, \hat{B}, e_1, [\sigma_1, \ldots, \sigma_{j-2}, \sigma_j]) \\ &= \mathcal{K}_j(A, B, Q_j e_1, [\sigma_1, \ldots, \sigma_{j-2}, \sigma_j]). \end{aligned}$$

Noting that $Q_j e_1 = e_1$, using the abbreviations $C = AB^{-1}$ and $C(\sigma) = AB^{-1} - \sigma I$, and using (4.3) twice, we obtain

$$\begin{aligned} \mathcal{Q}_j &= \mathcal{K}_j(AB^{-1}, e_1, [\sigma_1, \ldots, \sigma_{j-2}, \sigma_j]) \\ &= C(\sigma_j)^{-1} \left(\prod_{i=1}^{j-2} C(\sigma_i)^{-1} \right) \mathcal{K}_j(C, e_1) \\ &= C(\sigma_j)^{-1} C(\sigma_{j-1}) \left(\prod_{i=1}^{j-1} C(\sigma_i)^{-1} \right) \mathcal{K}_j(C, e_1) \\ &= C(\sigma_j)^{-1} C(\sigma_{j-1}) \mathcal{K}_j(A, B, e_1, [\sigma_1, \ldots, \sigma_{j-1}]) \\ &= C(\sigma_j)^{-1} C(\sigma_{j-1}) \mathcal{E}_j. \end{aligned}$$

In the final step we used Proposition 4.4 again. Since

$$\begin{aligned} C(\sigma_j)^{-1} C(\sigma_{j-1}) = C(\sigma_{j-1}) C(\sigma_j)^{-1} &= (AB^{-1} - \sigma_{j-1} I)(AB^{-1} - \sigma_j I)^{-1} \\ &= (A - \sigma_{j-1} B)(A - \sigma_j B)^{-1}, \end{aligned}$$

we get the desired result $\mathcal{Q}_j = (A - \sigma_{j-1} B)(A - \sigma_j B)^{-1} \mathcal{E}_j$.

In this argument we have assumed that B^{-1} exists. However, the result also holds for singular B by a continuity argument.

Now consider the spaces $\mathcal{Z}_{j-1}$. In the case $j = 2$ we have $\hat{A} - \lambda\hat{B} = Q_2^*(A - \lambda B)Z_1$. Substituting $\lambda = \sigma_2$ and solving for Z_1, we have $Z_1 = (A - \sigma_2 B)^{-1} Q_2(\hat{A} - \sigma_2 \hat{B})$. The

ETNA
Kent State University and
Johann Radon Institute (RICAM)

ordered pole set for $(\hat{A}, \hat{B})$ is $[\sigma_2, \sigma_1, \sigma_3, \ldots, \sigma_{n-1}]$, so $(\hat{A}-\sigma_2\hat{B})e_1 = \gamma e_1$ for some nonzero γ. Similarly, $(A - \sigma_1 B)e_1 = \delta e_1$ for some nonzero δ. Therefore,

$$Z_1 e_1 = \gamma(A - \sigma_2 B)^{-1} Q_2 e_1 = \gamma(A - \sigma_2 B)^{-1} e_1 = \gamma\delta^{-1}(A - \sigma_2 B)^{-1}(A - \sigma_1 B)e_1.$$

This proves that

$$\mathcal{Z}_1 = (A - \sigma_2 B)^{-1}(A - \sigma_1 B)\mathcal{E}_1,$$

as desired.

For $j > 2$ we have $\hat{A} - \lambda\hat{B} = Q_j^*(A - \lambda B)Z_{j-1}$. Arguing just as we did for $\mathcal{Q}_j$, we have

$$\begin{aligned}\mathcal{Z}_{j-1} = Z_{j-1}\mathcal{E}_{j-1} &= Z_{j-1}\mathcal{L}_{j-1}(\hat{A}, \hat{B}, e_1, [\sigma_2, \ldots, \sigma_{j-2}, \sigma_j]) \\ &= \mathcal{L}_{j-1}(A, B, Z_{j-1}e_1, [\sigma_2, \ldots, \sigma_{j-2}, \sigma_j]).\end{aligned}$$

Using $Z_{j-1}e_1 = e_1$ and making the abbreviations $D = B^{-1}A$ and $D(\sigma) = B^{-1}A - \sigma I$, we have

$$\begin{aligned}\mathcal{Z}_{j-1} &= \mathcal{L}_{j-1}(A, B, e_1, [\sigma_2, \ldots, \sigma_{j-2}, \sigma_j]) \\ &= \mathcal{K}_{j-1}(D, e_1, [\sigma_2, \ldots, \sigma_{j-2}, \sigma_j]) \\ &= D(\sigma_j)^{-1}\left(\prod_{i=2}^{j-2} D(\sigma_i)^{-1}\right)\mathcal{K}_{j-1}(D, e_1) \\ &= D(\sigma_j)^{-1}D(\sigma_{j-1})\left(\prod_{i=2}^{j-1} D(\sigma_i)^{-1}\right)\mathcal{K}_{j-1}(D, e_1) \\ &= D(\sigma_j)^{-1}D(\sigma_{j-1})\mathcal{K}_{j-1}(D, e_1, [\sigma_2, \ldots, \sigma_{j-1}]) \\ &= (A - \sigma_j B)^{-1}(A - \sigma_{j-1}B)\mathcal{L}_{j-1}(A, B, e_1, [\sigma_2, \ldots, \sigma_{j-1}]) \\ &= (A - \sigma_j B)^{-1}(A - \sigma_{j-1}B)\mathcal{E}_{j-1}. \qquad \square\end{aligned}$$

REMARK 4.5. We used Proposition 2.2 to prove the case $j = 1$, but we did not use Proposition 2.4. In connection with this, we remark that Theorem 4.3 immediately implies the dual results

$$\mathcal{Q}_j^\perp = (A^* - \overline{\sigma}_{j-1}B^*)^{-1}(A^* - \overline{\sigma}_j B^*)\mathcal{E}_j^\perp$$

and

$$\mathcal{Z}_{j-1}^\perp = (A^* - \overline{\sigma}_j B^*)(A^* - \overline{\sigma}_{j-1}B^*)^{-1}\mathcal{E}_{j-1}^\perp,$$

obtained by noting that $\mathcal{U} = C\mathcal{S}$ if and only if $\mathcal{U}^\perp = (C^*)^{-1}\mathcal{S}^\perp$. We could equally well have derived the dual results first and then deduced Theorem 4.3. In that case we would use Proposition 2.4 to prove the case $j = n$ and not use Proposition 2.2 at all. From Proposition 2.4 with $\tau = \sigma_n$, we have immediately

$$Z_{n-1}e_n = \overline{\delta}\,(A^* - \overline{\sigma}_n B^*)(A^* - \overline{\sigma}_{n-1}B^*)^{-1}e_n,$$

which implies

$$\mathcal{Z}_{n-1}^\perp = (A^* - \overline{\sigma}_n B^*)(A^* - \overline{\sigma}_{n-1}B^*)^{-1}\mathcal{E}_{n-1}^\perp,$$

the case $j = n$ of the dual result.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

5. Using Theorem 4.3. In all of the convergence theorems of the previous section, we have actions of the form $\mathcal{Q}_k = r(AB^{-1})\mathcal{E}_k$ and $\mathcal{Z}_k = r(B^{-1}A)\mathcal{E}_k$, where r is a rational function, e.g., $r(z) = (z-\sigma_{j-1})/(z-\sigma_j)$. In the following lemma, the functions r and s can be any functions defined on the spectrum of the pencil $A-\lambda B$, but in our applications they will always be rational. In this case, being defined on the spectrum of $A-\lambda B$ just means that none of the poles are eigenvalues.

LEMMA 5.1. *Consider two successive changes of the coordinate system*

$$\tilde{A}-\lambda\tilde{B} = \tilde{Q}^*(A-\lambda B)\tilde{Z} \qquad \textit{and} \qquad \hat{A}-\lambda\hat{B} = \hat{Q}^*(\tilde{A}-\lambda\tilde{B})\hat{Z},$$

so that

$$\hat{A}-\lambda\hat{B} = Q^*(A-\lambda B)Z, \qquad \textit{where} \qquad Q=\tilde{Q}\hat{Q} \quad \textit{and} \quad Z=\tilde{Z}\hat{Z}.$$

For $k = 1, \dots, n-1$, if

$$\tilde{Q}\mathcal{E}_k = r(AB^{-1})\mathcal{E}_k \qquad \textit{and} \qquad \hat{Q}\mathcal{E}_k = s(\tilde{A}\tilde{B}^{-1})\mathcal{E}_k,$$

then

$$Q\mathcal{E}_k = sr(AB^{-1})\mathcal{E}_k,$$

where sr is the pointwise product of s and r. If

$$\tilde{Z}\mathcal{E}_k = r(B^{-1}A)\mathcal{E}_k \qquad \textit{and} \qquad \hat{Z}\mathcal{E}_k = s(\tilde{B}^{-1}\tilde{A})\mathcal{E}_k,$$

then

$$Z\mathcal{E}_k = sr(B^{-1}A)\mathcal{E}_k.$$

Proof. Noting that $\tilde{Q}\,s(\tilde{A}\tilde{B}^{-1}) = s(AB^{-1})\tilde{Q}$, we have

$$Q\mathcal{E}_k = \tilde{Q}\hat{Q}\mathcal{E}_k = \tilde{Q}\,s(\tilde{A}\tilde{B}^{-1})\mathcal{E}_k = s(AB^{-1})\tilde{Q}\mathcal{E}_k = s(AB^{-1})r(AB^{-1})\mathcal{E}_k,$$

so $Q\mathcal{E}_k = sr(AB^{-1})\mathcal{E}_k$. The result for $Z\mathcal{E}_k$ is proved similarly, using the identity $\tilde{Z}\,s(\tilde{B}^{-1}\tilde{A}) = s(B^{-1}A)\tilde{Z}$. ∎

Clearly this lemma can be extended by induction to three or more successive changes of the coordinate system, and that's how we are going to use it.

Proof of Theorem 4.2. As a first application of Theorem 4.3, we show that it can be used to prove Theorem 4.2.

Proof of Theorem 4.2. According to Theorem 4.2, for each k the basic algorithm effects a transformation

$$\mathcal{Q}_k = (A-\rho B)(A-\sigma_k B)^{-1}\mathcal{E}_k. \tag{5.1}$$

Let us see why this is so. Recall that the basic algorithm begins with a move of type I that introduces the shift ρ as a pole at the top of the pencil. It then does a sequence of moves of type II that swap ρ with the other poles one by one. For a given k, most of these moves have no effect on $\mathcal{E}_k$. The only exception is the kth move, the case $j=k$ in Theorem 4.3. This is where we need to focus.

One iteration of the basic algorithm performs the equivalence

$$\hat{A}-\lambda\hat{B} = Q^*(A-\lambda B)Z,$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

where Q and Z are products of core transformations:

$$Q = Q_1 Q_2 \cdots Q_{n-1}, \qquad Z = Z_1 Z_2 \cdots Z_{n-1}.$$

The core Q_1 is the one that replaces pole σ_1 with the shift ρ. Q_2 (together with Z_1) swaps ρ with σ_2, Q_3 (together with Z_2) swaps ρ with σ_3, and so on. Z_{n-1} removes ρ and installs a new pole σ_n. We are interested in the action of Q_k (together with Z_{k-1}), which swaps ρ with σ_k. Thus we factor Q and Z as

$$Q = \tilde{Q} Q_k \hat{Q}, \qquad Z = \tilde{Z} Z_{k-1} \hat{Z},$$

where $\tilde{Q} = Q_1 \cdots Q_{k-1}$, and so on. Now we break the transformation into three parts:

$$\begin{aligned} \tilde{A} - \lambda \tilde{B} &= \tilde{Q}^* (A - \lambda B) \tilde{Z}, \\ \check{A} - \lambda \check{B} &= Q_k^* (\tilde{A} - \lambda \tilde{B}) Z_{k-1}, \\ \hat{A} - \lambda \hat{B} &= \hat{Q}^* (\check{A} - \lambda \check{B}) \hat{Z}. \end{aligned} \tag{5.2}$$

Because each of the cores $Q_1, \ldots, Q_{k-1}$ leaves $\mathcal{E}_k$ invariant, we have

$$\tilde{Q} \mathcal{E}_k = \mathcal{E}_k = r(AB^{-1}) \mathcal{E}_k, \qquad \text{where } r(z) = 1.$$

We can apply Theorem 4.3 with $j = k$ to the transformation (5.2), taking into account that the poles that are swapped in the kth move are ρ and σ_k, to get

$$Q_k \mathcal{E}_k = (\tilde{A} - \rho \tilde{B})(\tilde{A} - \sigma_k \tilde{B})^{-1} \mathcal{E}_k = s(\tilde{A} \tilde{B}^{-1}) \mathcal{E}_k, \qquad \text{where } s(z) = (z - \rho)/(z - \sigma_k).$$

Finally, noting that $Q_{k+1}, \ldots, Q_{n-1}$ all leave $\mathcal{E}_k$ invariant, we have

$$\hat{Q} \mathcal{E}_k = \mathcal{E}_k = t(\check{A} \check{B}^{-1}) \mathcal{E}_k, \qquad \text{where } t(z) = 1.$$

Now, applying Lemma 5.1 to the product $Q = \tilde{Q} Q_k \hat{Q}$, we get

$$Q \mathcal{E}_k = tsr(AB^{-1}) \mathcal{E}_k = s(AB^{-1}) \mathcal{E}_k,$$

which is exactly (5.1).

We can prove the Z-part of Theorem 4.2 in exactly the same way. We have

$$\tilde{Z} \mathcal{E}_{k-1} = \mathcal{E}_{k-1} = 1(B^{-1} A) \mathcal{E}_{k-1},$$

and by Theorem 4.3 with $j = k$,

$$Z_{k-1} \mathcal{E}_{k-1} = (\tilde{A} - \sigma_k \tilde{B})^{-1} (\tilde{A} - \rho \tilde{B}) \mathcal{E}_{k-1} = s(\tilde{B}^{-1} \tilde{A}) \mathcal{E}_{k-1},$$

and finally,

$$\hat{Z} \mathcal{E}_{k-1} = \mathcal{E}_{k-1} = 1(\check{B}^{-1} \check{A}) \mathcal{E}_{k-1}.$$

Therefore, by Lemma 5.1,

$$Z \mathcal{E}_{k-1} = s(B^{-1} A) \mathcal{E}_{k-1} = (A - \sigma_k B)^{-1} (A - \rho B) \mathcal{E}_{k-1}.$$

Adding one to the index k, we arrive at the Z-part of Theorem 4.2, thereby completing the proof. □

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Generalization of the proof. The basic algorithm is just one of many possible algorithms that make use of moves of types I and II on proper Hessenberg forms. We have already pointed out that one could run the algorithm in the opposite direction or in both directions at once. There are lots of other possibilities, and we will look at some in what follows.

From our proof of Theorem 4.2 it should now be clear that we will be able to use Theorem 4.3, together with Lemma 5.1, to analyze the action of any algorithm that acts on a proper Hessenberg pencil by moves of types I and II. Consider a transformation

$$\hat{A} - \lambda \hat{B} = Q^*(A - \lambda B)Z, \tag{5.3}$$

where Q and Z are products of core transformations generated by any sequence of moves of type I and II. If we want to find the action of Q on $\mathcal{E}_k$ for some k, then we only need to look at the core transformations of the form Q_k, i.e., the ones that act in the $(k, k+1)$ plane. Thus we factor Q into a product of the form

$$Q = \tilde{Q}Q_{1,k}\check{Q}Q_{2,k}\hat{Q}Q_{3,k}\cdots, \tag{5.4}$$

where $\tilde{Q}$, $\check{Q}$, $\dots$ are products of core transformations that do not act in the $(k, k+1)$ plane and therefore satisfy $\tilde{Q}\mathcal{E}_k = \mathcal{E}_k$, $\check{Q}\mathcal{E}_k = \mathcal{E}_k$, and so on, and $Q_{1,k}$, $Q_{2,k}$, $\dots$ are cores that do act in the $(k, k+1)$ plane. Let us say there are m such cores $Q_{1,k}, \dots, Q_{m,k}$.

The transforming matrix Z has a fully analogous factorization

$$Z = \tilde{Z}Z_{1,k-1}\check{Z}Z_{2,k-1}\hat{Z}Z_{3,k-1}\cdots, \tag{5.5}$$

assuming that we use the convention that moves of type I have the form $Q_1^*(A - \lambda B)Z_0$ with $Z_0 = I$ or $Q_n^*(A - \lambda B)Z_{n-1}$ with $Q_n = I$. We have $\tilde{Z}\mathcal{E}_{k-1} = \mathcal{E}_{k-1}$, $\check{Z}\mathcal{E}_{k-1} = \mathcal{E}_{k-1}$, et cetera. The transformations that act nontrivially on $\mathcal{E}_{k-1}$ are $Z_{1,k-1}, \dots, Z_{m,k-1}$.

Suppose that at the move corresponding to the transformations $Q_{j,k}$ and $Z_{j,k-1}$, the poles that get swapped are $\sigma_{j,k-1}$ and $\sigma_{j,k}$. Then, according to Theorem 4.3, the function associated with this swap is $r_j(z) = (z - \sigma_{j,k-1})/(z - \sigma_{j,k})$. Let r denote the product of these functions:

$$r(z) = r_1(z)\cdots r_m(z) = \prod_{j=1}^{m} \frac{z - \sigma_{j,k-1}}{z - \sigma_{j,k}}. \tag{5.6}$$

Then, applying Lemma 5.1 to the long product of transformations defined by (5.4) and (5.5), we find that the action of Q on $\mathcal{E}_k$ and of Z on $\mathcal{E}_{k-1}$ is given by

$$\mathcal{Q}_k = Q\mathcal{E}_k = r(AB^{-1})\mathcal{E}_k \qquad \text{and} \qquad \mathcal{Z}_{k-1} = Z\mathcal{E}_{k-1} = r(B^{-1}A)\mathcal{E}_{k-1}. \tag{5.7}$$

We summarize these findings as a theorem.

THEOREM 5.2. *Consider a transformation* (5.3)*, where Q and Z are products of core transformations generated by any sequence of moves of types I and II. For some k, suppose that m of the moves acted at the kth position, swapping poles $\sigma_{j,k-1}$ and $\sigma_{j,k}$ for $j = 1, \dots, m$. Define a rational function r by* (5.6)*. Then the action of Q on $\mathcal{E}_k$ and of Z on $\mathcal{E}_{k-1}$ is given by* (5.7)*. The transformation* (5.3) *transforms $\mathcal{Q}_k$ back to $\mathcal{E}_k$ and $\mathcal{Z}_{k-1}$ back to $\mathcal{E}_{k-1}$.*

6. Variations on the basic algorithm. In this section we consider algorithms built exclusively from moves of types I and II. Since the moves are backward stable, the resulting algorithms are also backward stable. We do not claim that all of the ideas presented here will result in practical algorithms; some of them are quite speculative.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

The basic algorithm (like the single-shift bulge-chasing and core-chasing algorithms) takes a single shift, inserts it into the top of the pencil, and chases it to the bottom. This algorithm suffers from inefficient use of cache memory and negligible potential for parallelism. In the case of bulge-chasing algorithms, the problem was remedied by selecting a large number of shifts at once, creating many small bulges one after the other, and chasing this chain of bulges together to the bottom of the matrix or pencil [8, 22, 23]. This allows the use of Level 3 BLAS and therefore efficient cache use. It also provides an opportunity for parallelism [16].

Chasing multiple shifts at once. The same remedy works for pole-swapping algorithms, as was already mentioned in [12, 13, 25]. We can choose m shifts $\rho_1, \ldots, \rho_m$, where typically $1 \ll m \ll n$.[2] Suppose the poles of $A - \lambda B$ are

$$\sigma_1, \ldots, \sigma_m, \sigma_{m+1}, \ldots, \sigma_n.$$

By a sequence of moves of types I and II, we can replace $\sigma_1, \ldots, \sigma_m$ by $\rho_1, \ldots, \rho_m$, so that the poles of the new pencil are

$$\rho_1, \ldots, \rho_m, \sigma_{m+1}, \ldots, \sigma_n.$$

Then we can chase these m shifts together to the bottom, creating enough arithmetic to make efficient use of cache. To be precise, in the first step we would swap σ_{m+1} with ρ_m, then σ_{m+1} with ρ_{m-1}, and so on. Eventually we swap σ_{m+1} with ρ_1, putting σ_{m+1} at the top. Then we go on to the next step.

We can pass a chain of shifts from top to bottom, and we can equally well pass a chain from bottom to top. If we wish, we can pass chains in both directions at once. Suppose we have shifts $\rho_1, \ldots, \rho_m$ that we wish to chase from top to bottom and shifts $\tau_1, \ldots, \tau_m$ that we wish to chase from bottom to top. Using moves of types I and II, we can introduce them:

$$\rho_1, \ldots, \rho_m, \sigma_{m+1}, \ldots \sigma_{n-m-1}, \tau_1, \ldots, \tau_m.$$

We then chase the ρ's downward and the τ's upward. The two chains pass through each other, and eventually we get to the position

$$\tau_1, \ldots, \tau_m, \sigma_{m+1}, \ldots \sigma_{n-m-1}, \rho_1, \ldots, \rho_m.$$

The reader can verify that the poles in the middle, $\sigma_{m+1}, \ldots, \sigma_{n-m-1}$, get moved around in the process, but they end up exactly where they started. At this point we can regard the iteration as complete, or we can "complete" the iteration by removing the τ_i and ρ_i from the pencil and replacing them with new sets of shifts.

Let's see what Theorem 5.2 tells us about this bi-directional procedure. Let

$$r(z) = \prod_{i=1}^{m} \frac{z - \rho_i}{z - \tau_i}.$$

Then, for $k = m + 1, \ldots, n - m$, we have the action

$$\mathcal{Q}_k = Q\mathcal{E}_k = r(AB^{-1})\mathcal{E}_k \qquad \text{and} \qquad \mathcal{Z}_{k-1} = Z\mathcal{E}_{k-1} = r(B^{-1}A)\mathcal{E}_{k-1}.$$

The reason for this is that each of the ρ_i passes downward through the kth position, causing a factor $z - \rho_i$, and each of the τ_i passes upward, causing a factor $(z - \tau_i)^{-1}$. This isn't

[2] One way to obtain m shifts is to use an auxiliary routine to compute the eigenvalues of the lower-right-hand $m \times m$ subpencil of $A - \lambda B$ and use these as the shifts.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

all that happens at position k, but it's all that matters. To see this, consider, for example, a position k at which all of the ρ_i pass through before any of the τ_i get there. Passing each ρ_i downward requires also passing a σ_j upward, causing a factor $(z-\sigma_j)^{-1}$. Later on, when the τ_i are being passed upward, each σ_j that was previously passed upward gets passed downward through the kth position, causing a factor $z-\sigma_j$. The factors $(z-\sigma_j)^{-1}$ and $z-\sigma_j$ cancel each other out. We know that this must happen for each σ_j because each σ_j starts and ends in the same position.

An optimistic scenario. Consider a situation in which we have in hand the information that we need to split the problem. Suppose we know a k (with $m+1 \leq k \leq n-m-1$) where (we think) we can split the pencil, and suppose that we have in mind an (m,m) rational function

$$r(z) = \prod_{i=1}^{m} \frac{z-\rho_i}{z-\tau_i}$$

that can (nearly) split it. By this we mean that $r(AB^{-1})\mathcal{E}_k$ is (nearly) invariant under AB^{-1} and $r(B^{-1}A)\mathcal{E}_k$ is (nearly) invariant under $B^{-1}A$. If we then take the ρ_i as shifts to be passed downward and the τ_i as shifts to be passed upward, then we will get both

$$\mathcal{Q}_k = Q\mathcal{E}_k = r(AB^{-1})\mathcal{E}_k \qquad \text{and} \qquad \mathcal{Z}_k = Z\mathcal{E}_k = r(B^{-1}A)\mathcal{E}_k.$$

The change of variables $\hat{A} - \lambda\hat{B} = Q^*(A-\lambda B)Z$ maps both of these spaces back to $\mathcal{E}_k$. Thus, $\mathcal{E}_k$ is (nearly) invariant under both $\hat{A}\hat{B}^{-1}$ and $\hat{B}^{-1}\hat{A}$, which implies that $(\mathcal{E}_k, \mathcal{E}_k)$ is (nearly) a deflating subspace for $(\hat{A},\hat{B})$. If the pencil does not quite split apart, then another step with the same (or improved?) shifts may get the job done. Notice that to achieve the desired spaces $\mathcal{Q}_k = r(AB^{-1})\mathcal{E}_k$ and $\mathcal{Z}_k = r(B^{-1}A)\mathcal{E}_k$, it is not necessary to pass the shifts all the way through the pencil. All that is needed is that $\rho_1, \ldots, \rho_m$ are pushed downward past position $k+1$ and $\tau_1, \ldots, \tau_m$ are passed upward past position k.

Of course this is a very optimistic scenario. (Where do we get these special shifts?) We include it here just to indicate what might be possible and to illustrate the use of Theorem 5.2.

Practical shift strategies. A more realistic plan is to take (for example) $\rho_1, \ldots, \rho_m$ to be the eigenvalues of the lower-right-hand $m \times m$ subpencil and $\tau_1, \ldots, \tau_m$ the eigenvalues of the upper-left-hand $m \times m$ subpencil, which will have the effect of causing deflations near the ends of the pencil.[3] An even better idea is to include *aggressive early deflation* [9], which is easy to implement in this context. This was already discussed in detail in [12, 13], so we will not dwell on it.

Steady streams of shifts. We conclude this section with one more interesting but fanciful idea. Imagine that we introduce steady streams of shifts at the top and the bottom. Eventually the streams start to pass through each other. How do we move the streams in their respective directions in an expeditious way? To answer this question, let us first look at the small case $n=8$, for which we have seven poles. Suppose we have at some point the poles

$$\tau_1,\ \rho_3,\ \tau_2,\ \rho_2,\ \tau_3,\ \rho_1,\ \tau_4,$$

where the shifts ρ_i are moving downward and the τ_i upward. We can introduce a new shift ρ_4 at the top by a move of type I that removes τ_1. At the same time we can do three moves of

[3] Notice, however, that a strategy like this should also include some provision to ensure that the upward-moving shifts are well separated from the downward-moving shifts. If some ρ_j is (nearly) equal to one of the τ_i, then they will (nearly) cancel each other out.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

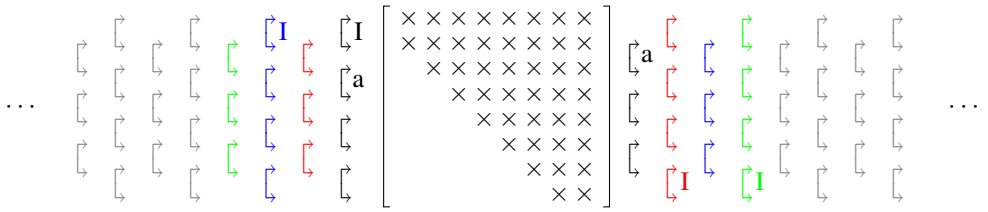
type II to interchange ρ_3 with τ_2, ρ_2 with τ_3, and ρ_1 with τ_4. The result is

$$\rho_4,\ \tau_2,\ \rho_3,\ \tau_3,\ \rho_2,\ \tau_4,\ \rho_1.$$

This is one step. For the next step we use a move of type I to introduce a new shift τ_5 at the bottom, removing ρ_1. At the same time we do three moves of type II to interchange τ_4 with ρ_2, τ_3 with ρ_3, and τ_2 with ρ_4. The result is

$$\tau_2,\ \rho_4,\ \tau_3,\ \rho_3,\ \tau_4,\ \rho_2,\ \tau_5.$$

The third step is like the first, the fourth step is like the second, and so on. We can illustrate these steps schematically with a diagram.



The matrix in the middle can be either A or B, since the same core transformations are applied to both. The cores of the first step are in black, with the move of type I marked accordingly. Each move of type II requires two cores, one on the left and one on the right. For example, the two cores marked with the symbol "a" belong to a single move. The second step is in red, with the core of type I marked accordingly on the right. The third and fourth steps are marked in blue and green, respectively. Four subsequent steps are shown in grey. We are illustrating the case $n = 8$, which is typical of even n. The odd case, which is slightly different, is left to the reader.

Before we get too excited about this elegant scheme, we must acknowledge that there are some challenges in the way of a competitive implementation. Thinking now of larger n, we see that each step is rich in arithmetic and highly parallel. Each step consists of about $n/2$ moves or $O(n^2)$ flops. To move a shift from one end of the pencil to the other requires about n steps, or $O(n^3)$ flops. Therefore, any competitive implementation must exploit the parallelism well. Another, possibly larger, issue is this: How do we get a steady stream of good shifts to feed in at the two ends?

7. Connections to earlier work.

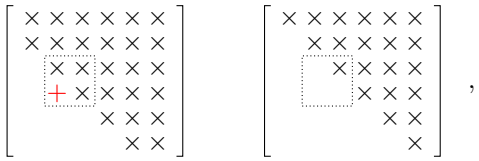
Bulge pencils. The purpose of shifting is to accelerate convergence. In the standard Francis bulge-chasing algorithm, the shifts are inserted at the top. That is, the shifts are used to help determine the initial transformation that creates the bulge. Then the shifts are forgotten, and the bulge is chased downward until it disappears off the bottom. Well-chosen shifts, inserted at the top, lead to rapid emergence of eigenvalues at the bottom of the matrix or pencil. Thus, the information about the shifts is somehow transmitted in the bulge from top to bottom.

A bit more than twenty years ago, one of the authors began to study the mechanism by which the shift information is conveyed in bulge-chasing algorithms. This study took some time, it seemed to be nontrivial, and it led to the discovery of the *bulge pencil* [28, 29, 31].

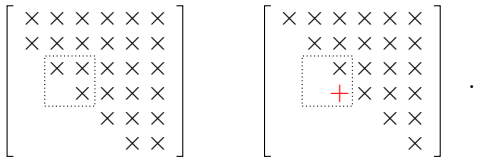
Now let's take a fresh look at the bulge pencil in light of what we now know about pole swapping. Suppose we pick a single shift ρ and begin chasing a bulge downward in a pair

ETNA
Kent State University and
Johann Radon Institute (RICAM)

(A, B). After a couple of steps we have

(7.1) 

with the bulge located at position $(4, 2)$. The 2×2 subpencil outlined in (7.1) is the bulge pencil. Its eigenvalues are ρ and ∞ [31, Chap. 7].[4] If we now do one more transformation on the left, moving the bulge from A to B, we obtain



This is a Hessenberg pair, and the eigenvalues of the bulge pencil are now in plain sight. In the $(3, 2)$-position we have the pole ∞, and in the $(4, 3)$-position we have a finite pole, which we know to be the shift ρ. What was opaque before is now transparent.

Certain structured problems require algorithms that chase bulges in both directions in order to preserve the structure. The first example of such an algorithm was the Hamiltonian QR algorithm of Byers [10, 11]. Some more recent examples are algorithms for the palindromic and even eigenvalue problems discussed in [21]. Our understanding of the bulge pencil made it possible to explain completely how to pass bulges (and the shifts that they contain) through each other in general in both structured and unstructured cases [30]. It took time and effort to figure this out, but now, in light of what we know about pole swapping, we can see that passing shifts through each other is simple. It's just a matter of swapping two eigenvalues of the pole pencil. Once again, what was opaque before is now transparent.

Tightly and optimally packed shifts. The schemes discussed in Section 6 insert not just one shift but long chains of shifts $\rho_1, \ldots, \rho_m$ into the pencil as poles and then chase them downward (or upward) in a bunch. In such a scheme it is important for efficiency to have the shifts packed as tightly together as possible. It is clear that in our current scenario we achieve this; the shifts $\rho_1, \ldots, \rho_m$ appear as adjacent poles in the Hessenberg pair, and there is no way that they could be packed any closer. (The same result is achieved effortlessly when this methodology is applied to core-chasing algorithms [1].) In contrast, in the bulge-chasing scenario, the packing of bulges is not naturally optimal, and it is not obvious how to fix the problem. However, with some effort, a remedy was eventually found [20]. In hindsight we can show that the remedy is a disguised implementation of a pole-swapping algorithm.

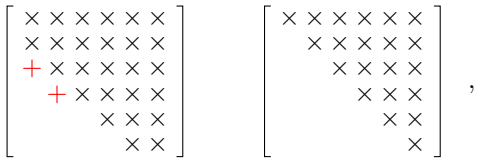
We have explained already that pole swapping reduces to bulge chasing if all poles that are not shifts are set to infinity. The philosophy is, however, different. Bulge chasing executes in each step an equivalence where the transforms on the left and right act on columns and rows having the same indices, say i and $i + 1$. Pole swapping, on the other hand, has the transformation on the left acting on rows i and $i + 1$, while the transformation on the right acts on columns $i - 1$ and i. Pole swapping is half an equivalence off compared to bulge chasing. This lag is natural in the pole-swapping setting and appears to be the foundational strategy to get optimally packed bulges.

[4] In [31] we considered (large-bulge) multishift algorithms with k shifts $\rho_1, \ldots, \rho_k$. Then the bulge pencil is $(k + 1) \times (k + 1)$ and has eigenvalues $\rho_1, \ldots, \rho_k$, and ∞. Here we are considering only the case $k = 1$.

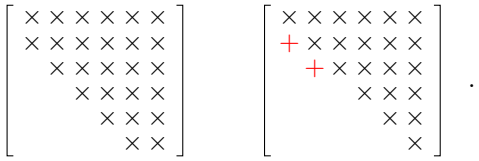
ETNA
Kent State University and
Johann Radon Institute (RICAM)

An optimally packed chain of two single shifts in the bulge chasing setting would, ideally, look like

(7.2)

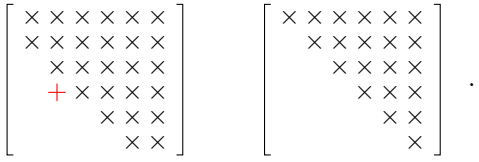


whereas in the pole-swapping setting it would resemble

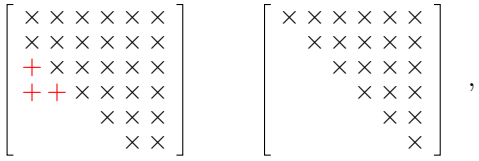


For simplicity, and without loss of generality, we restrict ourselves to two single shifts.

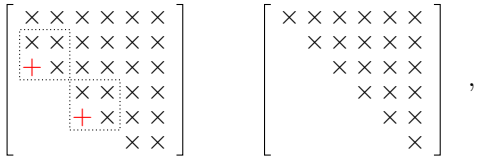
We have seen that getting an optimally packed chain of shifts in the pole-swapping setting is trivial. In the bulge chasing case, however, it is impossible to achieve (7.2). Introducing the first shift and chasing it down a row results in



Introducing the second shift does not work. We end up with



and both single shifts have been combined into a 2×2 multishift bulge. The scheme introduced by Braman, Byers, and Mathias [8] delays the introduction of the second shift until the first has been moved two spots down. We get

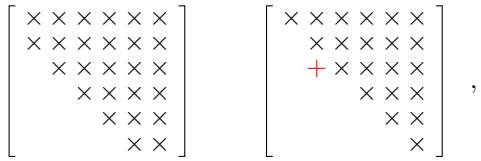


which are so-called tightly packed shifts. It is impossible to pack them any closer; otherwise the two 2×2 bulge pencils (marked in the figure) would overlap.

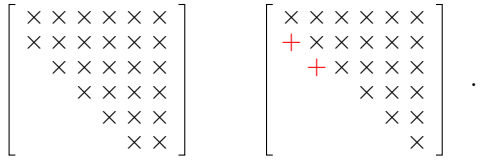
A solution to pack the bulges as tight as in (7.2) was proposed by Karlsson, Kressner, and Lang [20]. The trick is to defer some transformations from the right. Suppose that the first bulge is introduced and we would like to move it down a row; instead of executing an entire

ETNA
Kent State University and
Johann Radon Institute (RICAM)

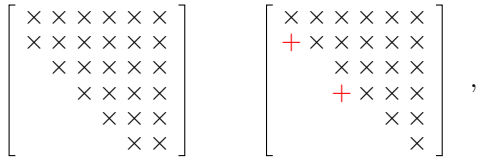
bulge-chasing step, we only execute the transformation from the left, while the transformation on the right is postponed. We end up with



which is nothing else than having moved the first pole down a position. Next we introduce the second shift, but we do not execute the transformation from the right. We get



To start the chasing, one now brings the first shift to the right, creating a bulge, and then annihilates the bulge. Thus, one does not execute an entire bulge-chase step, but again the transformation from the right is delayed. We end up with



after which we can do the same with the second shift. Obviously, this is just pole swapping, but the description in terms of bulges and delayed transformations conceals this fact.

Karlsson et al. [20] discussed the optimal packing of the bulges in terms of double-shift bulges. Since we have not discussed double-shift pole-swapping algorithms here, we do not explore this. The principles are, however, identical. The algorithm of Karlsson et al. [20] is a pole-swapping algorithm (with poles at infinity) avant-la-lettre.

8. The new pole-swapping procedure. We now describe the new swapping procedure that was promised at the beginning. The process of swapping two adjacent poles is equivalent to swapping two adjacent eigenvalues in the upper-triangular pole pencil. For the description it suffices to look at a 2×2 subpencil. Consider therefore a 2×2 upper-triangular pencil

$$A - \lambda B = \begin{bmatrix} \alpha_1 & a \\ 0 & \alpha_2 \end{bmatrix} - \lambda \begin{bmatrix} \beta_1 & b \\ 0 & \beta_2 \end{bmatrix} \tag{8.1}$$

with eigenvalues $\sigma_1 = \alpha_1/\beta_1$ and $\sigma_2 = \alpha_2/\beta_2$. We want to swap the eigenvalues. That is, we want to find core transformations Q and Z such that

$$Q^*(A - \lambda B)Z = \hat{A} - \lambda \hat{B} = \begin{bmatrix} \hat{\alpha}_1 & \hat{a} \\ & \hat{\alpha}_2 \end{bmatrix} - \lambda \begin{bmatrix} \hat{\beta}_1 & \hat{b} \\ & \hat{\beta}_2 \end{bmatrix},$$

with $\hat{\alpha}_1/\hat{\beta}_1 = \sigma_2$ and $\hat{\alpha}_2/\hat{\beta}_2 = \sigma_1$.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Solution in exact arithmetic.

Exact method 1. This method "grabs σ_2" and pulls it upward. Substituting $\lambda = \alpha_2/\beta_2$ in the pencil, we have

$$\beta_2 A - \alpha_2 B = \begin{bmatrix} \beta_2\alpha_1 - \alpha_2\beta_1 & \beta_2 a - \alpha_2 b \\ 0 & 0 \end{bmatrix},$$

from which we deduce that the vector

$$x = \begin{bmatrix} \alpha_2 b - \beta_2 a \\ \beta_2\alpha_1 - \alpha_2\beta_1 \end{bmatrix} \tag{8.2}$$

is a right eigenvector of the pencil associated with the eigenvalue $\sigma_2 = \alpha_2/\beta_2$. Let

$$y = \begin{bmatrix} \alpha_1 b - \beta_1 a \\ \beta_2\alpha_1 - \alpha_2\beta_1 \end{bmatrix}. \tag{8.3}$$

Direct computation shows that

$$Ax = \alpha_2 y \qquad \text{and} \qquad Bx = \beta_2 y.$$

Thus, the spaces spanned by x and y form a one-dimensional deflating pair for (A, B) associated with the eigenvalue α_2/β_2.

Let Q and Z be cores such that

$$Z^* x = \gamma e_1 \qquad \text{and} \qquad Q^* y = \zeta e_1,$$

and define

$$\hat{A} - \lambda\hat{B} = Q^* A Z - \lambda\, Q^* B Z.$$

Then we claim that $\hat{A} - \lambda\hat{B}$ is an upper triangular pencil with the eigenvalue α_2/β_2 on top. This is verified by the calculations

$$\hat{A}e_1 = Q^* A Z e_1 = \gamma^{-1} Q^* A x = \alpha_2\gamma^{-1} Q^* y = \alpha_2\gamma^{-1}\zeta\, e_1$$

and

$$\hat{B}e_1 = Q^* B Z e_1 = \gamma^{-1} Q^* B x = \beta_2\gamma^{-1} Q^* y = \beta_2\gamma^{-1}\zeta\, e_1.$$

This procedure fails if and only if $x = 0$, which happens whenever $A = B$, for example. The condition $x = 0$ implies that the eigenvalues are equal, so in this case the swap can be skipped.

Exact method 2. This method, which is the dual of the previous method, "grabs σ_1" and pushes it downward. Substituting $\lambda = \alpha_1/\beta_1$ in the pencil, we have

$$\beta_1 A - \alpha_1 B = \begin{bmatrix} 0 & \beta_1 a - \alpha_1 b \\ 0 & \beta_1\alpha_2 - \alpha_1\beta_2 \end{bmatrix},$$

from which we deduce that the vector

$$v^T = \begin{bmatrix} \beta_1\alpha_2 - \alpha_1\beta_2 & \alpha_1 b - \beta_1 a \end{bmatrix} \tag{8.4}$$

is a left eigenvector of the pencil associated with the eigenvalue $\sigma_1 = \alpha_1/\beta_1$. Let

$$w^T = \begin{bmatrix} \beta_1\alpha_2 - \alpha_1\beta_2 & \alpha_2 b - \beta_2 a \end{bmatrix}. \tag{8.5}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Direct computation shows that

$$v^T A = \alpha_1 w^T \qquad \text{and} \qquad v^T B = \beta_1 w^T.$$

Let Q and Z be cores such that

$$v^T Q = \zeta e_2^T \qquad \text{and} \qquad w^T Z = \gamma e_2^T,$$

and define

$$\hat{A} - \lambda \hat{B} = Q^* A Z - \lambda\, Q^* B Z.$$

Then we claim that $\hat{A} - \lambda\hat{B}$ is an upper triangular pencil with the eigenvalue α_1/β_1 on the bottom. This is verified by the calculations

$$e_2^T \hat{A} = e_2^T Q^* A Z = \zeta^{-1} v^T A Z = \alpha_1 \zeta^{-1} w^T Z = \alpha_1 \zeta^{-1} \gamma\, e_2^T$$

and

$$e_2^T \hat{B} = e_2^T Q^* B Z = \zeta^{-1} v^T B Z = \beta_1 \zeta^{-1} w^T Z = \beta_1 \zeta^{-1} \gamma\, e_2^T.$$

This procedure fails if and only if $v^T = 0$, in which case $\sigma_1 = \sigma_2$, and the swap can be skipped. The reader can easily verify that the two methods produce exactly the same Q and Z.

Solution in floating point arithmetic. In the interest of stability one should not implement either of the above procedures in practice. There are several alternatives.

Case 1. We will demonstrate below that the following procedure, which is based on exact method 1, is stable in the case $|\,\sigma_1\,| \geq |\,\sigma_2\,|$. Compute x as in (8.2). Then compute Z such that $Z^* x = \gamma e_1$, where $\gamma = \|\,x\,\|$. (Here and in what follows, the norm symbol refers to either the vector 2-norm or matrix 2-norm, depending on the context.) Then compute BZ. Since $Ze_1 = \gamma^{-1} x$, the first column of BZ is $\gamma^{-1}\beta_2\, y$. Do not compute Q using the vector y as defined in (8.3). Instead compute Q so that $Q^*(BZe_1) = \beta_2\gamma^{-1}\zeta\, e_1$. Then let

$$\hat{A} = Q^* A Z \qquad \text{and} \qquad \hat{B} = Q^* B Z.$$

Case 2. For the case when $|\,\sigma_1\,| < |\,\sigma_2\,|$ we need a different procedure. There are multiple possibilities, the simplest of which is to apply the above procedure with the roles of A and B reversed. We compute Z as before, then use AZ instead of BZ to determine Q. Specifically, since $Ze_1 = \gamma^{-1} x$, the first column of AZ is $\gamma^{-1}\alpha_2\, y$. Thus, we can compute Q so that $Q^*(AZe_1) = \alpha_2\gamma^{-1}\zeta\, e_1$.

This procedure is similar to that of Van Dooren [26]. His method always computes Z first, then uses either AZ or BZ to compute Q. The only difference is that our criterion for switching between BZ and AZ is different from that in [26]. This makes a difference in the backward error.

Another procedure, which is based on exact method 2, computes Q first. Compute the vector v^T as in (8.4), then compute Q such that $v^T Q = \zeta e_2^T$, where $\zeta = \|\,v\,\|$. Then compute $Q^* B$. Since $e_2^T Q^* = \zeta^{-1} v^T$, the second row of $Q^* B$ is $\zeta^{-1}\beta_1\, w^T$. Do not compute Z using w^T as defined in (8.5). Instead compute Z so that $(e_2^T Q^* B)Z = \beta_1\zeta^{-1}\gamma\, e_2^T$. Then let

$$\hat{A} = Q^* A Z \qquad \text{and} \qquad \hat{B} = Q^* B Z.$$

This is exactly equivalent to the procedure from Case 1 applied to a "flipped" pencil. Let $F = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$, the *flip* matrix, and consider the pencil

$$F A^T F - \lambda F B^T F = \begin{bmatrix} \alpha_2 & a \\ & \alpha_1 \end{bmatrix} - \lambda \begin{bmatrix} \beta_2 & b \\ & \beta_1 \end{bmatrix}.$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

This has the eigenvalues reversed. The condition $|\sigma_2| > |\sigma_1|$ implies that we can stably apply the method from Case 1, and then "unflip" the result. The equation $\hat{A} - \lambda\hat{B} = Q^*(A - \lambda B)Z$ implies

$$F\hat{A}^T F - \lambda F\hat{B}^T F = (FZ^T F)(FA^T F - \lambda FB^T F)(F\overline{Q}F),$$

which shows that the roles of Q and Z are reversed in the flipped procedure. (Of course Q and $F\overline{Q}F$ are not exactly the same, but they contain the same information.) The "compute Q first"-procedure that we have just outlined is a way of implementing the "flipped"-procedure without actually doing the flips.

Backward error analysis. It suffices to prove backward stability in Case 1, since the options in Case 2 are both variants of Case 1.

The swapping operation is a unitary equivalence, and such transformations generally are stable [17], but there is one thing we have to verify. The core Q is designed so that $Q^*(BZ)$ has a zero in the $(2,1)$-position. This automatically creates a zero in the $(2,1)$-position of $Q^*(AZ)$ because the first columns of AZ and BZ are both proportional to y. This is true in exact arithmetic. We just need to verify that in floating-point arithmetic the entry that is created in the $(2,1)$-position of Q^*AZ is small enough that backward stability is not compromised by setting it to zero. For this it suffices that its magnitude be no bigger than a modest multiple of $u\|A\|$, where u is the unit roundoff.

The swapping operation begins with the computation of x in (8.2). In floating-point arithmetic we get

$$\text{fl}(x) = \begin{bmatrix} \alpha_2 b(1+\epsilon_1) - \beta_2 a(1+\epsilon_2) \\ \beta_2\alpha_1(1+\epsilon_3) - \alpha_2\beta_1(1+\epsilon_4) \end{bmatrix}, \tag{8.6}$$

where each ϵ_i is the result of two roundoff errors, a multiplication and a subtraction mapped back to the product terms, and therefore satisfies $|\epsilon_i| \leq 2u + O(u^2)$. We will use the abbreviation $|\epsilon_i| \lesssim u$ to mean that $|\epsilon_i|$ is no bigger than a modest constant times u.

The next step is to compute Z. In practice we do this using $\text{fl}(x)$ and make additional roundoff errors in the computation. We get $\tilde{Z} = \text{fl}(Z)$ satisfying

$$\tilde{Z}e_1 = \tilde{x} = \tilde{\gamma}^{-1} \begin{bmatrix} \text{fl}(x_1)(1+\epsilon_5) \\ \text{fl}(x_2)(1+\epsilon_6) \end{bmatrix}. \tag{8.7}$$

Here $\tilde{\gamma} = \|\,\text{fl}(x)\,\|$. A tiny relative error is made during this norm computation, and another tiny error is made when $\text{fl}(x_1)$ is divided by $\tilde{\gamma}$. These are the causes of the error ϵ_5, and we have $|\epsilon_5| \lesssim u$. Similarly $|\epsilon_6| \lesssim u$. For more details about this computation, see [1, § 1.4].

The vector $\tilde{x}$ defined by (8.7) is our computed (and normalized) version of a right eigenvector associated with the eigenvalue σ_2. For later use we wish to show that $\tilde{x}$ is exactly an eigenvector of a slightly perturbed pencil. Thus we seek $\tilde{\alpha}_1$, $\tilde{\alpha}_2$, $\tilde{\beta}_1$, and $\tilde{\beta}_2$ such that

$$\left(\tilde{\beta}_2 \begin{bmatrix} \tilde{\alpha}_1 & a \\ & \tilde{\alpha}_2 \end{bmatrix} - \tilde{\alpha}_2 \begin{bmatrix} \tilde{\beta}_1 & b \\ & \tilde{\beta}_2 \end{bmatrix}\right) \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{8.8}$$

Notice that we are not going to back any of the error onto a or b. This equation is equivalent to

$$(\tilde{\beta}_2\tilde{\alpha}_1 - \tilde{\alpha}_2\tilde{\beta}_1)\tilde{x}_1 + (\tilde{\beta}_2 a - \tilde{\alpha}_2 b)\tilde{x}_2 = 0.$$

Filling in the values of $\tilde{x}_1$ and $\tilde{x}_2$ from (8.7) and (8.6), we can demonstrate that this equation holds if we make the assignments

$$\tilde{\alpha}_1 = \alpha_1 \frac{(1+\epsilon_3)(1+\epsilon_6)}{(1+\epsilon_2)(1+\epsilon_5)}, \qquad \tilde{\alpha}_2 = \alpha_2(1+\epsilon_1)(1+\epsilon_5),$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

$$\tilde{\beta}_1 = \beta_1 \frac{(1+\epsilon_4)(1+\epsilon_6)}{(1+\epsilon_1)(1+\epsilon_5)}, \qquad \tilde{\beta}_2 = \beta_2(1+\epsilon_2)(1+\epsilon_5).$$

Clearly $|\tilde{\alpha}_i - \alpha_i| \lesssim u|\alpha_i|$ and $|\tilde{\beta}_i - \beta_i| \lesssim u|\beta_i|$ for $i = 1, 2$. Equation (8.8) can be written more compactly as

$$\tilde{\beta}_2 \tilde{A}\tilde{x} = \tilde{\alpha}_2 \tilde{B}\tilde{x}. \tag{8.9}$$

Thus, $\tilde{x}$ is an eigenvector of the perturbed pencil $\tilde{A} - \lambda\tilde{B}$ associated with the eigenvalue $\tilde{\sigma}_2 = \tilde{\alpha}_2/\tilde{\beta}_2$. We also write

$$\tilde{A} = A + \delta A \qquad \text{and} \qquad \tilde{B} = B + \delta B_1,$$

with δA and δB_1 diagonal matrices satisfying $\|\delta A\| \lesssim u\|A\|$ and $\|\delta B_1\| \lesssim u\|B\|$.

Finally, we compute Q. In exact arithmetic, Q is constructed so that $Q^*(BZe_1) = \eta e_1$, for some η, so the first column of Q must be proportional to BZe_1. In practice, instead of BZe_1, we use

$$\check{y} = \text{fl}(B\tilde{Z}e_1) = \text{fl}(B\tilde{x}) = \tilde{\gamma}^{-1} \begin{bmatrix} \beta_1\tilde{x}_1(1+\epsilon_1') + b\tilde{x}_2(1+\epsilon_2') \\ \beta_2\tilde{x}_2(1+\epsilon_3') \end{bmatrix},$$

where $|\epsilon_i'| \lesssim u$ for $i = 1, 2, 3$. The computed version of Q is $\tilde{Q} = \text{fl}(Q)$ satisfying

$$\tilde{Q}e_1 = \check{\zeta}^{-1} \begin{bmatrix} \check{y}_1(1+\epsilon_4') \\ \check{y}_2(1+\epsilon_5') \end{bmatrix},$$

where $\check{\zeta} = \|\check{y}\|$ and ϵ_4' and ϵ_5' are due to the tiny roundoff errors in the calculation.

For our analysis we need to establish that there is a slightly perturbed matrix

$$\hat{B} = B + \delta B_2 = \begin{bmatrix} \hat{\beta}_1 & b \\ & \hat{\beta}_2 \end{bmatrix}$$

such that $\tilde{Q}^*\hat{B}\tilde{Z}$ has an exact zero in the $(2,1)$-position. This just means that $\tilde{y} = \tilde{Q}e_1$ is exactly proportional to $\hat{B}\tilde{Z}e_1 = \hat{B}\tilde{x}$. It is easy to verify that the choice

$$\hat{\beta}_1 = \beta_1 \frac{(1+\epsilon_1')}{(1+\epsilon_2')}, \qquad \hat{\beta}_2 = \beta_2 \frac{(1+\epsilon_3')(1+\epsilon_5')}{(1+\epsilon_2')(1+\epsilon_4')}$$

does the trick. Clearly $|\hat{\beta}_1 - \beta_1| \lesssim u\,|\beta_1|$ and $|\hat{\beta}_2 - \beta_2| \lesssim u\,|\beta_2|$, and δB_2 is a diagonal matrix satisfying $\|\delta B_2\| \lesssim u\,\|B\|$.

Our final computed results are $\text{fl}(\tilde{Q}^*A\tilde{Z})$ and $\text{fl}(\tilde{Q}^*B\tilde{Z})$. We have to show that the $(2,1)$-entries of these matrices are small enough that we can set them to zero without compromising backward stability. The "B" part is routine. Focusing on the $(2,1)$-entry, we have

$$e_2^T\,\text{fl}(\tilde{Q}^*B\tilde{Z})e_1 = e_2^T\tilde{Q}^*B\tilde{Z}e_1 + e_2^TE_1e_1,$$

where E_1 is the matrix of roundoff errors incurred in multiplying the three matrices together and satisfies $\|E_1\| \lesssim u\,\|\tilde{Q}\|\,\|B\|\,\|\tilde{Z}\|$, i.e., $\|E_1\| \lesssim u\,\|B\|$. The remaining term is

$$e_2^T\tilde{Q}^*B\tilde{Z}e_1 = e_2^T\tilde{Q}^*\hat{B}\tilde{Z}e_1 - e_2^T\tilde{Q}^*\delta B_2\tilde{Z}e_1.$$

The first term on the right-hand side is exactly zero by construction. The second is bounded above by $\|\delta B_2\| \lesssim u\,\|B\|$. This takes care of the "B" part.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

The "A" part (the important part) is more delicate. We have

$$e_2^T \mathrm{fl}(\tilde{Q}^* A \tilde{Z}) e_1 = e_2^T \tilde{Q}^* A \tilde{Z} e_1 + e_2^T E_2 e_1,$$

where E_2 is the matrix of roundoff errors incurred in multiplying the three matrices together and satisfies $\| E_2 \| \lesssim u \| A \|$. The remaining term is

$$e_2^T \tilde{Q}^* A \tilde{Z} e_1 = e_2^T \tilde{Q}^* \tilde{A} \tilde{Z} e_1 - e_2^T \tilde{Q}^* \delta A \tilde{Z} e_1.$$

The second term on the right-hand side is bounded above by $\| \delta A \| \lesssim u \| A \|$, so now we can just focus on the other term. Here we make use of (8.9), which can be written as $\tilde{A}\tilde{Z}e_1 = (\tilde{\alpha}_2/\tilde{\beta}_2)\tilde{B}\tilde{Z}e_1$.

$$e_2^T \tilde{Q}^* \tilde{A} \tilde{Z} e_1 = \frac{\tilde{\alpha}_2}{\tilde{\beta}_2} e_2^T \tilde{Q}^* \tilde{B} \tilde{Z} e_1 = \frac{\tilde{\alpha}_2}{\tilde{\beta}_2} e_2^T \tilde{Q}^* \hat{B} \tilde{Z} e_1 + \frac{\tilde{\alpha}_2}{\tilde{\beta}_2} e_2^T \tilde{Q}^* (\delta B_1 - \delta B_2) \tilde{Z} e_1.$$

The term containing $\hat{B}$ is zero by construction, so now we just need to concentrate on the other term. Let $\delta B = \delta B_1 - \delta B_2$. From the definitions of δB_1 and δB_2, we see that

$$\delta B = \begin{bmatrix} \epsilon_1'' \beta_1 & 0 \\ 0 & \epsilon_2'' \beta_2 \end{bmatrix},$$

where $|\epsilon_i''| \lesssim u$ for $i = 1, 2$. Moreover $\dfrac{\tilde{\alpha}_2}{\tilde{\beta}_2} = \dfrac{\alpha_2}{\beta_2}(1 + \epsilon_3'')$ for some tiny ϵ_3''. We also use our assumption $|\sigma_1| \geq |\sigma_2|$ to deduce that $|\beta_1 \alpha_2 / \beta_2| \leq |\alpha_1|$. Thus,

$$|(\tilde{\alpha}_2/\tilde{\beta}_2)\delta B| = (1 + \epsilon_3'') \begin{bmatrix} |\epsilon_1'' \beta_1 \alpha_2/\beta_2| & \\ & |\epsilon_2'' \alpha_2| \end{bmatrix} \leq (1 + \epsilon_3'') \begin{bmatrix} |\epsilon_1'' \alpha_1| & \\ & |\epsilon_2'' \alpha_2| \end{bmatrix},$$

so

$$\| (\tilde{\alpha}_2/\tilde{\beta}_2)\delta B \| \lesssim u \| A \|.$$

We conclude that our one remaining term, which is $(\tilde{\alpha}_2/\tilde{\beta}_2) e_2^T \tilde{Q}^* (\delta B) \tilde{Z} e_1$, satisfies

$$|(\tilde{\alpha}_2/\tilde{\beta}_2) e_2^T \tilde{Q}^* (\delta B) \tilde{Z} e_1| \lesssim u \| A \|.$$

We have demonstrated that

$$|e_2^T \mathrm{fl}(\tilde{Q}^* A \tilde{Z}) e_1| \lesssim u \| A \| \qquad \text{and} \qquad |e_2^T \mathrm{fl}(\tilde{Q}^* B \tilde{Z}) e_1| \lesssim u \| B \|,$$

so we can set these numbers to zero without compromising backward stability. The $\lesssim$ symbols hide constants, but these constants are not too large due to the small total number of operations required by the swap.

Our procedure improves on that of Van Dooren [26] in that the latter only guarantees that the two entries are bounded above by $u \max\{\| A \|, \| B \|\}$ instead of $u \| A \|$ and $u \| B \|$ separately. It follows that our procedure produces better results in cases where A and B have vastly different norms. We remind the reader that the A and B referred to here are the small matrices defined in (8.1) and not the larger matrices in which they are embedded. Therefore, we cannot solve the problem of different norms by a simple rescaling of the large matrices at the outset as this does not guarantee equal norms in all of the little submatrices in which the swaps take place.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 8.1
Distribution of errors $|\hat{a}_{21}|/\|A\|$ and $|\hat{b}_{21}|/\|B\|$ for our method, Van Dooren's method, and the Sylvester method.

$\lvert\hat{x}_{21}\rvert/\lVert X\rVert$		$[0, 10^{-16}]$	$(10^{-16}, 10^{-15}]$	$(10^{-15}, 10^{-10}]$	$(10^{-10}, 10^{-5}]$	$(10^{-5}, 10^{0}]$
Our method	A	99.71%	0.29%	0%	0%	0%
	B	99.85%	0.15%	0%	0%	0%
Van Dooren	A	98.19%	0.55%	0.93%	0.27%	0.06%
	B	98.19%	0.55%	0.93%	0.27%	0.06%
Sylvester	A	93.34%	5.88%	0.57%	0.17%	0.04%
	B	93.34%	5.88%	0.57%	0.17%	0.04%

TABLE 8.2
Distribution of errors $|\hat{a}_{21}|/\Delta$ and $|\hat{b}_{21}|/\Delta$ for our method, Van Dooren's method, and the Sylvester method.

$\lvert\hat{x}_{21}\rvert/\Delta$		$[0, 10^{-16}]$	$(10^{-16}, 10^{-15}]$	$(10^{-15}, 10^{-10}]$	$(10^{-10}, 10^{-5}]$	$(10^{-5}, 10^{0}]$
Our method	A	99.87%	0.13%	0%	0%	0%
	B	99.93%	0.07%	0%	0%	0%
Van Dooren	A	99.94%	0.06%	0%	0%	0%
	B	99.94%	0.06%	0%	0%	0%
Sylvester	A	97.26%	2.74%	0%	0%	0%
	B	97.26%	2.74%	0%	0%	0%

Numerical experiments. In most cases it does not matter which swapping procedure is used; they all perform well. In order to see a difference, they must be stress-tested on pencils that have elements that vary widely in magnitude. Therefore, in the two experiments reported here, we used pencils whose nonzero entries are randomly generated complex numbers with magnitudes distributed logarithmically in the range from 10^{-12} to 10^{12}.

In our first test we generated sixty-four million random 2×2 upper-triangular pencils and computed the swapping transformations using three different algorithms: our method, the method of Van Dooren [26], and a method that solves the generalized Sylvester equation explicitly to determine Q and Z [7]. The computations were done in IEEE standard double-precision arithmetic, for which $u \approx 10^{-16}$. Table 8.1 shows that our method always produces residuals $|a_{21}|/\|A\|$ and $|b_{21}|/\|B\|$ that are under 10^{-15}, and more than 99.7% of them are under 10^{-16}. In contrast, the Van Dooren and Sylvester methods sometimes produce much larger residuals, approaching 10^{0} in a few cases. If we change the criterion and consider the residuals $|a_{21}|/\Delta$ and $|b_{21}|/\Delta$, where $\Delta = \max\{\|A\|, \|B\|\}$, then all methods perform well, as Table 8.2 shows. By this criterion all residuals are under 10^{-15}. Our method and Van Dooren's method perform about equally well, and the Sylvester method is almost as good. We conclude that if $\|A\|$ and $\|B\|$ are roughly the same, then it doesn't matter which method is used. However, in problems for which there can be large differences in magnitude between $\|A\|$ and $\|B\|$, our method is better.

It is natural to ask whether improved backward stability of the swapping transformations actually results in more accurately computed eigenvalues of the larger pencils. To test this, we considered ten thousand randomly generated 3×3 upper Hessenberg pencils with logarithmically distributed entries with magnitudes varying from 10^{-12} to 10^{12}. Since we do not know the exact eigenvalues of these pencils, we used MATLAB with the ADVANPIX

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Multiprecision Computing Toolbox[5] to compute "exact" eigenvalues in quadruple precision arithmetic. We compared these with the approximate eigenvalues computed using our method and Van Dooren's.

Before we look at that comparison, we note that we didn't just compute the eigenvalues; in fact we computed the Schur form $A_T - \lambda B_T = Q^*(A - \lambda B)Z$, where A_T and B_T are upper triangular. This allowed us to compute residuals

$$r_A = \| A - QA_TZ^* \|/\| A \| \qquad \text{and} \qquad r_B = \| B - QB_TZ^* \|/\| B \|, \tag{8.10}$$

which are measures of the backward error. When the computation was done using our method, the residuals were always tiny, never exceeding 10^{-14}, verifying normwise backward stability. When Van Dooren's criterion was used, the residuals (8.10) were usually just as small but occasionally larger. If the denominators $\| A \|$ and $\| B \|$ in the residuals r_A and r_B are replaced by $\Delta = \max\{\| A \|, \| B \|\}$, then the Van Dooren residuals also become uniformly small, never exceeding 10^{-14}.

Of course tiny backward errors do not guarantee accurately computed eigenvalues, as some of them may be ill conditioned. Moreover, decreasing the backward error does not necessarily guarantee improved eigenvalue accuracy, so we must make the comparison. Let λ_i, $i = 1, 2, 3$, denote the "exact" eigenvalues produced in quadruple precision, let $\lambda_i^{(o)}$ denote the approximate eigenvalues computed by our method, and let

$$e^{(o)} = \max_i |\, \lambda_i^{(o)} - \lambda_i \,|/|\, \lambda_i \,|, \tag{8.11}$$

be the maximum relative error. Let $\lambda_i^{(v)}$ denote the eigenvalues computed by Van Dooren's method, and let $e^{(v)}$ denote the maximum relative error, defined analogously to $e^{(o)}$ as in (8.11).

We examined the ratios $e^{(v)}/e^{(0)}$ and found that just over 98% of our trials resulted in $0.1 < e^{(v)}/e^{(o)} < 10$, indicating that neither method was significantly more accurate than the other. (In fact there were many cases where $e^{(v)}/e^{(o)} = 1$ since it often happens that our criterion and Van Dooren's criterion make exactly the same decisions.) Of the remaining trials, which numbered 181, there were 145 in which our method did significantly better than Van Dooren's, i.e., $e^{(v)}/e^{(o)} > 10$, and 36 in which $e^{(v)}/e^{(o)} < 0.1$. Thus, our new method obtained more accurate eigenvalues in about 80% of the significant cases. More details are given in histogram form in Figure 8.1. The gap in the center of the figure is due to having left out the many cases for which $e^{(v)}/e^{(o)}$ is close to 1. In the interest of compactness and clarity, the figure also leaves out one "off the charts" case for which $e^{(v)}/e^{(o)} \approx 10^{15}$.

9. Conclusions. We have discussed the RQZ algorithm and a number of variants, which we refer to generally as pole-swapping algorithms. We have made two main contributions: 1) We have developed a flexible, modular convergence theory that can be applied to any pole-swapping algorithm. 2) We have presented a new, more accurate, swapping procedure. A backward error analysis and numerical experiments demonstrate the superiority of the new procedure.

Acknowledgment. We thank the anonymous referees for carefully reading the paper and suggesting several improvements.

[5] https://www.advanpix.com

ETNA
Kent State University and
Johann Radon Institute (RICAM)

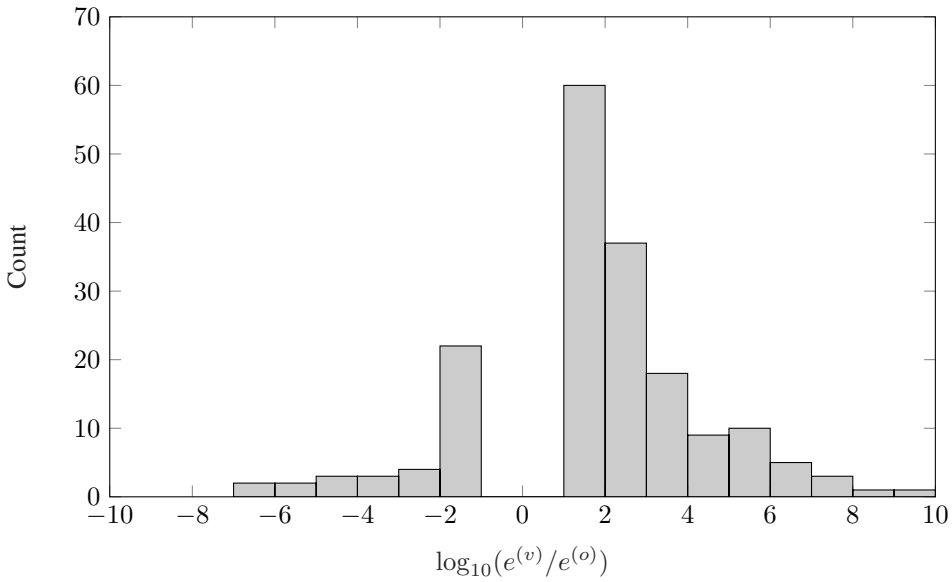


FIG. 8.1. *Histogram of the logarithm of $e^{(v)}/e^{(0)}$ in significant cases.*

REFERENCES

[1] J. L. AURENTZ, T. MACH, L. ROBOL, R. VANDEBRIL, AND D. S. WATKINS, *Core-Chasing Algorithms for the Eigenvalue Problem*, SIAM, Philadelphia, 2018.
[2] ———, *Fast and backward stable computation of roots of polynomials, Part II: Backward error analysis; companion matrix and companion pencil*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 1245–1269.
[3] ———, *Fast and backward stable computation of the eigenvalues and eigenvectors of matrix polynomials*, Math. Comp., 88 (2019), pp. 313–347.
[4] J. L. AURENTZ, T. MACH, R. VANDEBRIL, AND D. S. WATKINS, *Fast and backward stable computation of roots of polynomials*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 942–973.
[5] Z. BAI AND J. W. DEMMEL, *On swapping diagonal blocks in real Schur form*, Linear Algebra Appl., 186 (1993), pp. 73–95.
[6] M. BERLJAFA AND S. GÜTTEL, *Generalized rational Krylov decompositions with an application to rational approximation*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 894–916.
[7] A. BOJANCZYK AND P. VAN DOOREN, *Reordering diagonal blocks in the real Schur form*, in Linear Algebra for Large Scale and Real-Time Applications, M. Moonen, G. Golub, and B. D. Moor, eds., NATO ASI Series E: Applied Sciences, Springer, Dordrecht, 1993, pp. 351–352.
[8] K. BRAMAN, R. BYERS, AND R. MATTHIAS, *The multishift QR algorithm, part I: Maintaining well focused shifts and level 3 performance*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 929–947.
[9] ———, *The multishift QR algorithm, part II: Aggressive early deflation*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 948–973.
[10] R. BYERS, *Hamiltonian and Symplectic Algorithms for the Algebraic Riccati Equation*, PhD. Thesis, Dept. Comp. Sci., Cornell University, Ithaca, 1983.
[11] ———, *A Hamiltonian QR algorithm*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 212–229.
[12] D. CAMPS, *Pole Swapping Methods for the Eigenvalue Problem: Rational QR Algorithms*, PhD. Thesis, Faculty of Engineering Science, KU Leuven, Leuven, 2019.
[13] D. CAMPS, K. MEERBERGEN, AND R. VANDEBRIL, *A rational QZ method*, SIAM J. Matrix Anal. Appl., 40 (2019), pp. 943–972.
[14] J. G. F. FRANCIS, *The QR transformation. II*, Comput. J., 4 (1961), pp. 332–345.
[15] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 4th ed., Johns Hopkins University Press, Baltimore, 2013.
[16] R. GRANAT, B. KÅGSTRÖM, AND D. KRESSNER, *A novel parallel QR algorithm for hybrid distributed memory HPC systems*, SIAM J. Sci. Comput., 32 (2010), pp. 2345–2378.
[17] N. J. HIGHAM, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.
[18] B. KÅGSTRÖM AND P. POROMAA, *Computing eigenspaces with specified eigenvalues of a regular matrix*

ETNA
Kent State University and
Johann Radon Institute (RICAM)

pair (A, B) *and condition estimation: theory, algorithms and software*, Numer. Algorithms, 12 (1996), pp. 369–407.

[19] ———, *LAPACK-style algorithms and software for solving the generalized Sylvester equation and estimating the separation between regular matrix pairs*, ACM Trans. Math. Software, 22 (1996), pp. 78–103.

[20] L. KARLSSON, D. KRESSNER, AND B. LANG, *Optimally packed chains of bulges in multishift QR algorithms*, ACM Trans. Math. Software, 40 (2014), Art. 12, 15 pages.

[21] D. KRESSNER, C. SCHRÖDER, AND D. S. WATKINS, *Implicit QR algorithms for palindromic and even eigenvalue problems*, Numer. Algorithms, 51 (2009), pp. 209–238.

[22] B. LANG, *Effiziente Orthogonaltransformationen bei der Eigen- und Singulärwertzerlegung*, Habilitationsschrift, Fachbereich Mathematik, Universität Wuppertal, Wuppertal, 1997.

[23] B. LANG, *Using level* 3 *BLAS in rotation-based algorithms*, SIAM J. Sci. Comput., 19 (1998), pp. 626–634.

[24] C. B. MOLER AND G. W. STEWART, *An algorithm for generalized matrix eigenvalue problems*, SIAM J. Numer. Anal., 10 (1973), pp. 241–256.

[25] T. STEEL, D. CAMPS, K. MEERBERGEN, AND R. VANDEBRIL, *A multishift, multipole rational QZ method with aggressive early deflation*, Preprint on arXiv, 2020. https://arxiv.org/abs/1902.10954

[26] P. VAN DOOREN, *A generalized eigenvalue approach for solving Riccati equations*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 121–135.

[27] R. VANDEBRIL AND D. S. WATKINS, *An extension of the QZ algorithm beyond the Hessenberg-upper triangular pencil*, Electron. Trans. Numer. Anal., 40 (2013), pp. 17–35. http://etna.ricam.oeaw.ac.at/vol.40.2013/pp17-35.dir/pp17-35.pdf

[28] D. S. WATKINS, *Forward stability and transmission of shifts in the* QR *algorithm*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 469–487.

[29] ———, *The transmission of shifts and shift blurring in the* QR *algorithm*, Linear Algebra Appl., 241/243 (1996), pp. 877–896.

[30] ———, *Bulge exchanges in algorithms of* QR *type*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 1074–1096.

[31] ———, *The Matrix Eigenvalue Problem.* GR *and Krylov Subspace Methods*, SIAM, Philadelphia, 2007.

[32] ———, *Fundamentals of Matrix Computations*, 3rd ed., Wiley, Hoboken, 2010.

[33] ———, *Francis's algorithm*, Amer. Math. Monthly, 118 (2011), pp. 387–403.

[34] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.