SIAM J. Matrix Anal. Appl.
Vol. 23, No. 4, pp. 948–973

THE MULTISHIFT *QR* ALGORITHM. PART II: AGGRESSIVE EARLY DEFLATION[*]

KAREN BRAMAN[†], RALPH BYERS[†], AND ROY MATHIAS[‡]

Abstract. Aggressive early deflation is a QR algorithm deflation strategy that takes advantage of matrix perturbations outside of the subdiagonal entries of the Hessenberg QR iterate. It identifies and deflates converged eigenvalues long before the classic small-subdiagonal strategy would. The new deflation strategy enhances the performance of conventional large-bulge multishift QR algorithms, but it is particularly effective in combination with the small-bulge multishift QR algorithm. The small-bulge multishift QR sweep with aggressive early deflation maintains a high rate of execution of floating point operations while significantly reducing the number of operations required.

Key words. QR algorithm, deflation, implicit shifts, eigenvalues, eigenvectors

AMS subject classifications. 65F15, 15A18

PII. S0895479801384585

1. Introduction.

An underappreciated consideration in the QR algorithm is the problem of detecting deflation. In conventional Hessenberg QR algorithms convergence is recognized when one of the Hessenberg subdiagonal entries becomes small enough to be safely set to zero. The eigenvalue problem then decouples into two smaller problems. This is the earliest deflation strategy used with the QR algorithm [24, 25, 45], and it has changed little since the early 1960s.

The small-subdiagonal convergence criterion sometimes does not recognize and deflate converged eigenvalues. Consider, for example, the 6-by-6 Hessenberg matrix

$$S_6 = \begin{bmatrix} 6 & 5 & 4 & 3 & 2 & 1 \\ 0.001 & 1 & 0 & 0 & 0 & 0 \\ & 0.001 & 2 & 0 & 0 & 0 \\ & & 0.001 & 3 & 0 & 0 \\ & & & 0.001 & 4 & 0 \\ & & & & 0.001 & 5 \end{bmatrix}. \tag{1.1}$$

By ordinary standards, the subdiagonal entries of S_6 are not particularly small and certainly not negligible. If shifts are taken to be the eigenvalues of a trailing principal submatrix, then a multishift QR algorithm would use 5, 4, 3, etc., as the shifts for the next QR sweep. Table 1 lists the distances between S_6 and a matrix with eigenvalues equal to these likely choices of shifts. (The distances were estimated by the method described in this paper.) In particular, there is a perturbation $E \in \mathbf{R}^{6\times 6}$,

[*]Received by the editors February 5, 2001; accepted for publication (in revised form) by D. Boley May 15, 2001; published electronically March 27, 2002. This work was partially supported by the National Computational Science Alliance under award DMS990005N and utilized the NCSA SGI/Cray Origin2000.

http://www.siam.org/journals/simax/23-4/38458.html

[†]Department of Mathematics, University of Kansas, 405 Snow Hall, Lawrence, KS 66045 (braman@math.ukans.edu, byers@math.ukans.edu). The research of these authors was partially supported by National Science Foundation awards CCR-9732671, DMS-9628626, CCR-9404425, and MRI-9977352 and by the NSF EPSCoR/K*STAR program through the Center for Advanced Scientific Computing.

[‡]Department of Mathematics, College of William & Mary, Williamsburg, VA 23187 (mathias@math.wm.edu). The research of this author was partially supported by NSF grants DMS-9504795 and DMS-9704534.

Table 1

Estimates of the distances between S_6 *in* (1.1) *and a matrix with eigenvalues equal to eigenvalues of a trailing principal submatrix.*

A matrix with these eigenvalues...	...is within this spectral norm distance of S_6.
1 2, 3, 4, and 5	1×10^{-3}
2, 3, 4, and 5	1×10^{-6}
3, 4, and 5	5×10^{-10}
4 and 5	2×10^{-13}
5	4×10^{-17}

$\|E\|_2 \approx 4 \times 10^{-17}$ for which 5 is an eigenvalue of $H + E$. In typical finite precision computation, the unit roundoff is roughly 2×10^{-16}. In that context, arguably, the shift 5 is a converged eigenvalue that should be detected and deflated before the next QR sweep. Possibly, both 4 and 5 can be accepted as eigenvalues and be deflated. If the unit roundoff were, say, 1×10^{-7}, then 5, 4, 3, and possibly even 2 might be acceptable eigenvalues.

The undetected but essentially converged shift 5 would ordinarily be used as a shift in the QR algorithm and after a QR sweep or two the corresponding subdiagonal will become small enough to set to zero. The extra sweep or two represents unnecessary arithmetic work, but it is benign. It does not introduce numerical instability. However, the converged eigenvalues do occupy shifts that could be used to start work on other, unconverged eigenvalues. In the multishift QR setting, the small-subdiagonal convergence criterion usually remains unsatisfied until all or nearly all of the simultaneous shifts have converged to eigenvalues to full precision. (The QR algorithm tends to deflate submatrices of roughly the same order as the number of shifts.) However, it is typical for some of the shifts to converge before others. In our experience with the two-tone small-bulge QR algorithm [10] and small-subdiagonal deflation, it is commonly the case that, up to rounding error, *half or more of the shifts are converged eigenvalues!* Unable to recognize and deflate the converged eigenvalues, the algorithm must reuse them in the next sweep. It cannot work on new, unconverged eigenvalues until a small subdiagonal appears when all or nearly all the current shifts have converged. Much of the potential performance of the multishift QR algorithm may be lost.

Readers of this paper need to be familiar with the double implicit shift QR algorithm [24, 25, 32]. (See, for example, any of the textbooks [15, 29, 38, 41, 45].) Familiarity with the large-bulge multishift QR algorithm [3] as implemented in LAPACK [1] and/or the small-bulge multishift QR algorithm [10] is helpful.

1.1. Notation.

Throughout this paper we use the following notation and definitions.

1. We will use the "colon notation" to denote submatrices: $H_{i:j,k:l}$ is the submatrix of matrix H in rows i–j and columns k–l inclusively. The notation $H_{:,k:l}$ indicates the submatrix in columns k–l inclusively (and all rows). The notation $H_{i:j,:}$ indicates the submatrix in rows i–j inclusively (and all columns).

2. The notation $\sigma_k = \sigma_k(M)$ denotes the kth largest magnitude singular value of the matrix M. The singular value of smallest magnitude will also be written as $\sigma_{\min} = \sigma_{\min}(M)$.

3. The spectral norm is denoted $\|M\|_2 = \sigma_1(M)$. The Frobenius norm is $\|M\|_F = \sqrt{\operatorname{trace}(M^T M)}$.

4. The jth column of the identity matrix is denoted e_j. The matrix formed from the first k columns of I, $[e_1, e_2, e_3, \ldots, e_k]$, is E_k. The subspace $\mathcal{E}_k \subset \mathbf{C}^n$ is the subspace spanned by the first k columns of I, i.e., $\mathcal{E}_k = \mathrm{span}(e_1, e_2, e_3, \ldots, e_k)$. The orthogonal complement of $\mathcal{E}_k$ is $\mathrm{span}(e_{k+1}, e_{k+2}, e_{k+3}, \ldots, e_n)$ and is denoted $\mathcal{E}_k^\perp$.

5. A 1-*unitary* matrix is a unitary matrix $Q \in \mathbf{C}^{n\times n}$ for which $Qe_1 = e_1$ and $e_1^T Q = e_1^T$, i.e., the first row and column of Q is the first row and column of I. A real 1-unitary matrix is called 1-orthogonal. Householder reduction to Hessenberg form [29, Algorithm 7.4.2] produces a 1-unitary matrix in the form of a product of Householder reflections.

6. A *quasi-triangular* matrix is a real, block triangular matrix with 1-by-1 and 2-by-2 blocks along the diagonal.

7. The orthogonal projection of a vector $x \in \mathbf{R}^n$ onto a subspace $\mathcal{Q} \subset \mathbf{C}^n$ is denoted by $\mathrm{Proj}_{\mathcal{Q}}(x)$. If the subspace $\mathcal{Q}$ is spanned by the columns of a matrix Q, we will abbreviate $\mathrm{Proj}_{\mathrm{Range}(Q)}(x)$ by $\mathrm{Proj}_Q(x)$.

8. A matrix $H \in \mathbf{R}^{n\times n}$ is in Hessenberg form if $h_{ij} = 0$ whenever $i > j+1$. The matrix H is said to be unreduced if, in addition, the subdiagonal entries are nonzero, i.e., $h_{ij} \neq 0$ whenever $i = j+1$.

9. Following [29, p. 19], we define a "flop" as a single floating point operation, i.e., either a floating point addition or a floating point multiplication together with its associated subscripting. The Fortran statement

$$\mathtt{C(I,J) = C(I,J) + A(I,K) * B(K,J)}$$

involves two flops.

In some of the examples in section 3, we report an automatic hardware count of the number of floating point instructions executed. Note that a trinary multiply-add instruction counts as just one executed instruction in the hardware count even though it executes two flops. Thus, depending on the compiler and optimization level, the above Fortran statement could be executed using either one floating point instruction or two floating point instructions (perhaps along with some integer subscripting calculations) even though it involves two flops.

10. The (i,j)th minor of a matrix $M \in \mathbf{R}^{n\times n}$ is represented by $M(i|j)$. (The (i,j)th minor is the determinant of the matrix obtained by deleting the ith row and jth column.)

11. The classical adjoint matrix or adjugate of a matrix $M \in \mathbf{R}^{n\times n}$ is written $\mathrm{adj}(M)$. Its (i,j)th entry is the (j,i)th cofactor, $(-1)^{i+j} M(i|j)$.

2. Aggressive early deflation.

In broad outline, the aggressive early deflation procedure derived and analyzed below works as follows. Partition an unreduced Hessenberg matrix H as

$$H = \begin{array}{c} \\ n-k-1 \\ 1 \\ k \end{array} \begin{array}{c} \begin{array}{ccc} n-k-1 & 1 & k \end{array} \\ \left[\begin{array}{ccc} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ 0 & H_{32} & H_{33} \end{array}\right] \end{array}, \tag{2.1}$$

where k is an integer $1 \le k < n$. Let $H_{33} = VTV^H$ be a Schur decomposition of H_{33}. Consider the similarity transformation to a Hessenberg-plus-spike form

$$\begin{bmatrix} I & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & V \end{bmatrix}^H \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ 0 & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & V \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13}V \\ H_{21} & H_{22} & H_{23}V \\ 0 & s & T \end{bmatrix}. \tag{2.2}$$

We explain below in section 2.8 that it is often the case that the last several components of s are tiny even when no subdiagonal entry of H is particularly small. The right-hand column in Table 1 is the spike s obtained from (1.1) with $k = 5$.

If the trailing m components of s are set to zero, then (2.2) takes the form

$$\begin{array}{ll} & \begin{array}{cccc} n-k-1 & 1 & k-m & m \end{array} \\ \begin{array}{l} n-k-1 \\ 1 \\ k-m \\ m \end{array} & \left[\begin{array}{cccc} H_{11} & H_{12} & \tilde{H}_{13} & \tilde{H}_{14} \\ H_{21} & H_{22} & \tilde{H}_{23} & \tilde{H}_{24} \\ 0 & \tilde{s} & T_{11} & T_{12} \\ 0 & 0 & 0 & T_{22} \end{array}\right]. \end{array}$$

The eigenvalues of T_{22} are deflated! Ordinarily, many more QR sweeps would be needed to reduce the subdiagonals of H to the point that these eigenvalues deflate.

If $k \ll n$, then the work needed to compute the Schur decomposition, form s, and return to Hessenberg form is small compared to the work of the QR sweeps that are saved. Approximately $4nk^2 + 31k^3$ flops are needed for this. See [10, Appendix B] for a more detailed description of the algorithm and a derivation of the flop count. Of these, if k is big enough, $4nk^2$ will be level 3 BLAS operations.

Incidentally, the eigenvalues of T_{11} make good shifts for the next multishift QR sweep.

2.1. Reducing perturbations. Let $H \in \mathbf{C}^{n\times n}$ be an unreduced Hessenberg matrix, let $P \in \mathbf{C}^{n\times n}$ be a perturbation matrix, and let $Q \in \mathbf{C}^{n\times n}$ be a 1-unitary matrix such that $\hat{H} \equiv Q^H(H+P)Q$ is Hessenberg. We call P a *reducing perturbation* if $\hat{H}$ is a reduced Hessenberg matrix. If P is a reducing perturbation of negligible magnitude, then the problem of finding the eigenvalues of H splits into the smaller problems of finding the eigenvalues of the two or more diagonal blocks of the block triangular matrix $\hat{H} = Q^H(H+P)Q$. At least in principle, aggressive early deflation consists of finding a reducing perturbation P of negligible magnitude, if possible, and using it to deflate the eigenvalue problem into two or more smaller problems.

Reducing perturbations are easily characterized in terms of the zero structure of a left eigenvector.

LEMMA 2.1. *A matrix $P \in \mathbf{C}^{n\times n}$ is a reducing perturbation for a matrix $H \in \mathbf{C}^{n\times n}$ if and only if $H + P$ has a left eigenvector $v \in \mathbf{C}^n$ with zero first component (i.e., $v_1 = 0$).*

Proof. Let $Q \in \mathbf{C}^{n\times n}$ be a 1-unitary matrix such that $Q^H(H+P)Q = \hat{H}$ is Hessenberg.

Suppose that P is a reducing perturbation, and, consequently, $\hat{H}$ is a reduced Hessenberg matrix. In particular, $\hat{H}$ is block triangular, so $\hat{H}$ has a left eigenvector $\hat{v}$ with $\hat{v}_1 = 0$. It follows that $v = Q^H\hat{v}$ is a left eigenvector of H and $v_1 = \hat{v}_1 = 0$.

Conversely, suppose that $(H + P)$ has a left eigenvector v with corresponding eigenvalue $\lambda \in \mathbf{C}$ and $v_1 = 0$. So, $\hat{v} = Q^H v$ is a left eigenvector of $\hat{H}$ and $\hat{v}_1 = 0$. Let k be the smallest index for which $\hat{v}_k \neq 0$. The Hessenberg structure of $\hat{H}$ implies that the $(k-1)$st component of $\lambda\hat{v} = \hat{v}^H\hat{H}$ is $0 = \lambda\bar{\hat{v}}_{k-1} = \bar{\hat{v}}_k\hat{h}_{k,k-1}$. Hence, $h_{k,k-1} = 0$ and $\hat{H}$ is a reduced Hessenberg matrix. □

Of course, in the context of aggressive deflation, it is not sufficient to find a small norm reducing perturbation. Returning a dense n-by-n matrix $H + P$ to Hessenberg form and accumulating the orthogonal similarity transformations would cost roughly $7n^3 + O(n^2)$ flops [29, Algorithm 7.4.2]—more arithmetic work than needed by many QR sweeps. A useful reducing perturbation P needs to have enough zero structure so that relatively little work is needed to return $H + P$ to Hessenberg form.

If the reducing perturbation P is restricted to be Hessenberg, then no extra work is needed, because $H+P$ is already Hessenberg. It is easy to show that a Hessenberg reducing perturbation of minimal Frobenius norm is zero except for some subdiagonal entry. Thus, restricting P to be Hessenberg leads to the small-subdiagonal deflation strategy.

A more aggressive deflation strategy must consider reducing perturbations that are not necessarily Hessenberg. Consider searching for a deflating perturbation $P \in \mathbf{C}^{n\times n}$ which may be nonzero only its last k rows and $k+1$ columns. With this choice, $H+P$ is Hessenberg in its initial $n-k-1$ columns. If $k \ll n$, then returning to Hessenberg form would need an acceptably small amount of arithmetic.

The next lemma identifies deflating perturbations of this form that have minimal norm.

LEMMA 2.2. *Let $H \in \mathbf{C}^{n\times n}$ be an unreduced Hessenberg matrix partitioned as in* (2.1)*. Let $\mu_* \in \mathbf{C}$ be a minimizer of $f(\mu) = \sigma_k([H_{32},\, H_{33} - \mu I])$ with corresponding left singular vector $u_* \in \mathbf{C}^k$ and right singular vector $v_* \in \mathbf{C}^{k+1}$. If $P \in \mathbf{C}^{n\times n}$ is given by*

$$P = \begin{array}{c} \\ n-k-1 \\ 1 \\ k \end{array} \begin{array}{c} \begin{array}{ccc} n-k-1 & 1 & k \end{array} \\ \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & P_{32} & P_{33} \end{array}\right] \end{array}, \tag{2.3}$$

where $[P_{32},\, P_{33}] = -f(\mu_*)u_*v_*^H \in \mathbf{C}^{k\times(k+1)}$, then $\|P\|_2 = f(\mu_*) = \sigma_k([H_{32},\, H_{33} - \mu_* I])$, and P is a reducing perturbation of minimal spectral and Frobenius norm with the zero structure of (2.3).

Proof. By construction,

$$[0,\, 0,\, u_*^H] \left[\begin{array}{ccc} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ 0 & H_{32}+P_{32} & H_{33}+P_{33} \end{array}\right] = \mu_*[0,\, 0,\, u_*^H].$$

Hence, μ_* is an eigenvalue of $H_{33} + P_{33}$ and $H + P$ with left eigenvectors u_* and $[0,\, 0,\, u_*^H]$, respectively. Hence, by Lemma 2.1, P is a reducing perturbation.

Suppose that $u \in \mathbf{C}^n$ is a left eigenvector of $H+P$ with eigenvalue $\mu \in \mathbf{C}$ and that $u_1 = 0$. Because the first $n-k-1$ columns of $H+P$ are in unreduced Hessenberg form, u must have zeros in its first $n-k$ components. The trailing k components form a left null vector of $[H_{32},\, H_{33} - \mu I] + [P_{32},\, P_{33}]$. It is an application of the singular value decomposition that $[P_{32},\, P_{33}]$ is the perturbation of smallest spectral and Frobenius norm for which $\operatorname{rank}([H_{32},\, H_{33} - \mu I] + [P_{32},\, P_{33}]) < k$. Hence, P is a perturbation of smallest spectral and Frobenius norm for which $H+P$ admits a left eigenvector whose first $n-k$ components are zeros. □

If $\|P\|_2$ is tiny enough to be neglected, then μ_* is essentially an eigenvalue of H with left eigenvector $[0,\, 0,\, u_*^H]$.

The problem of minimizing $f(\mu) = \sigma_k([H_{32},\, H_{33} - \mu I])$ has been extensively studied in the form of the controllability radius problem. A pair of matrices (A, B) is said to be controllable if for all $\lambda \in \mathbf{C}$ the matrix $[B,\, A - \lambda I]$ has full row rank [30]. Controllability is a sort of nonsingularity or well-definedness property required in many contexts in control theory [33, 34]. It is also a factor in the numerical condition of computational control problems [11, 16, 14, 34]. The minimum value of $f(\mu)$ is $\nu(H_{33}, H_{32})$, the Frobenius norm distance and spectral norm distance from the pair $(H_{33},\, H_{32})$ to the nearest uncontrollable pair [21, 34, 35].

A remarkably eclectic collection of mathematical tools and techniques have been used to attack the problem of calculating or estimating the controllability radius. See, for example, [6, 8, 7, 12, 13, 16, 21, 22, 23, 26, 27, 44]. By Lemma 2.2, all of them are potentially applicable to shift selection and deflation.

A dissatisfying aspect of Lemma 2.2 is that the optimal deflating perturbation P usually deflates only a single eigenvalue. Possibly, a perturbation of slightly larger magnitude would deflate several eigenvalues. Another dissatisfying aspect of Lemma 2.2 is that even when H is a real matrix, the perturbation matrix P may be complex. The extra work and storage required by complex arithmetic makes its use unattractive, but the staircase form and eigenvalue clustering methods for approximating the controllability radius [5, 16, 18, 17, 31, 35] can be easily and naturally adapted to find a "small norm" real deflating perturbation and so avoid complex arithmetic.

A simple heuristic way to use Lemma 2.2 is to approximate the minimizing μ_* by one of the eigenvalues of H_{33}. The corresponding left eigenvector u_* serves as an approximate left singular vector, and e_1, the first column of the $(k+1)$-by-$(k+1)$ identity, serves as an approximate right singular vector. This leads to a reducing perturbation of the form

$$P = \begin{array}{c} \\ n-k-1 \\ 1 \\ k \end{array} \begin{array}{c} \begin{array}{ccc} n-k-1 & 1 & k \end{array} \\ \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & P_3 & 0 \end{array}\right] \end{array}. \tag{2.4}$$

We call perturbation matrices with this sparsity pattern *k-spike perturbations.*

2.2. k-spike reducing perturbations. Spike reducing perturbations can be naturally adapted to avoid complex arithmetic when H is real and to deflate several eigenvalues at a time.

We will say that a k-spike reducing perturbation $P \in \mathbf{C}^{n\times n}$ (2.4) *deflates m eigenvalues and their corresponding left-invariant subspace* from a matrix $H \in \mathbf{C}^{n\times n}$ if there is a 1-unitary matrix Q for which $\hat{H} = Q^H(H+P)Q$ is a reduced Hessenberg matrix with $\hat{h}_{n-m+1,n-m} = 0$. (If several subdiagonal entries of $\hat{H}$ are zero, then there are several different values of m for which a perturbation P may be said to deflate m eigenvalues.) The matrix $\hat{H}$ takes the form

$$\hat{H} = Q^H(H+P)Q = \begin{array}{c} \\ n-m \\ m \end{array} \begin{array}{c} \begin{array}{cc} n-m & m \end{array} \\ \left[\begin{array}{cc} \hat{T}_{11} & \hat{T}_{12} \\ 0 & \hat{T}_{22} \end{array}\right] \end{array},$$

where $\hat{T}_{11}$ and $\hat{T}_{22}$ are Hessenberg. The last m columns of Q span an m-dimensional left-invariant subspace of $H+P$ corresponding to the eigenvalues of $\hat{T}_{22}$. The 1-unitary matrix Q is not unique, but if $\hat{T}_{11}$ is unreduced Hessenberg, then the implicit Q theorem [29, Theorem 7.4.3] implies that the first $n-m$ columns of Q are unique up to column scaling by numbers of modulus 1. There is enough freedom in the choice of the remaining m columns of Q to make $\hat{T}_{22}$ triangular. (If H and Q are restricted to be real, then there is only enough freedom to make $\hat{T}_{22}$ be quasi-triangular.) In this case m eigenvalues are displayed along the diagonal of $\hat{T}_{22}$. If P is of negligible magnitude, then P deflates the m eigenvalues of $\hat{T}_{22}$ and the left-invariant subspace spanned by the last m columns of Q. The smaller problem of calculating the remaining $n-m$ eigenvalues of $\hat{T}_{11}$ remains.

The following lemma and theorem characterize multieigenvalue deflating k-spike reducing perturbations in terms of the zero structure of the left-invariant subspaces they deflate. In the statement of the lemma, recall that $\mathcal{E}_k$ is the space spanned by the first k columns of I.

Lemma 2.3. *A matrix $P \in \mathbf{C}^{n\times n}$ is a reducing perturbation for $H \in \mathbf{C}^{n\times n}$ that deflates at least m eigenvalues if and only if $H+P$ has a left-invariant subspace $\mathcal{Q}$ of dimension m or greater for which $\mathcal{Q} \subset \mathcal{E}_1^{\perp}$.*

Proof. Let $Q \in \mathbf{C}^{n\times n}$ be a 1-unitary matrix such that $Q^H(H+P)Q = \hat{H}$ is Hessenberg.

Suppose that P is a reducing perturbation that deflates $\hat{m} \geq m$ eigenvalues. The reduced Hessenberg matrix $\hat{H}$ is block triangular with an $\hat{m}$-by-$\hat{m}$ $(2,2)$ block. It follows that the last $\hat{m}$ columns of Q span an $\hat{m}$-dimensional left-invariant subspace, $\mathcal{V}$ of $H+P$. The 1-unitary structure of Q implies that the first entry in each of the last $\hat{m}$ columns of Q is zero. Hence, $\mathcal{V} \subset \mathcal{E}_1^{\perp}$.

Conversely, if $(H+P)$ has a left-invariant subspace $\mathcal{V} \subset \mathcal{E}_1^{\perp}$ of dimension $\hat{m} \geq m$, then $\hat{\mathcal{V}} = Q^H\mathcal{V}$ is an $\hat{m}$-dimensional left-invariant subspace of $\hat{H}$. It follows from the 1-unitary structure of Q that $\hat{\mathcal{V}} \subset \mathcal{E}_1^{\perp}$. Let k be the largest integer for which $\hat{\mathcal{V}} \subset \mathcal{E}_k^{\perp}$. The first k entries of all members of $\hat{\mathcal{V}}$ are zero, but there is a vector $\hat{v} \in \hat{\mathcal{V}}$ for which $\hat{v}_{k+1} \neq 0$. Note that $k \leq n - \hat{m} \leq n - m$, because $\hat{\mathcal{V}}$ has dimension $\hat{m} \geq m$. The matrix $\hat{H}$ is a Hessenberg matrix for which $\hat{v}^H\hat{H} \subset \hat{\mathcal{V}}$. In particular, the kth entry of $\hat{v}^H\hat{H}$ is zero, i.e., $\hat{v}_{k+1}\hat{h}_{k+1,k} = 0$. Hence, $\hat{h}_{k+1,k} = 0$ and P deflates at least m eigenvalues. □

The next theorem characterizes k-spike reducing perturbations that deflate several eigenvalues.

Theorem 2.4. *Let $H \in \mathbf{C}^{n\times n}$ be an unreduced Hessenberg matrix partitioned as in* (2.1)*, and let P be a k-spike perturbation as in* (2.4)*. The k-spike perturbation P deflates at least $m \leq k$ eigenvalues if and only if for some m-dimensional left-invariant subspace $\tilde{\mathcal{V}}$ of H_{33}, $Proj_{\tilde{\mathcal{V}}}(P_3) = -Proj_{\tilde{\mathcal{V}}}(H_{32})$.*

The k-spike reducing perturbation of minimal Frobenius norm corresponds to the choice $P_3 = -Proj_{\tilde{\mathcal{V}}}(H_{32})$ for some m-dimensional left-invariant subspace $\tilde{\mathcal{V}}$ of H_{33}.

Proof. Suppose that the columns of $\tilde{V} \in \mathbf{C}^{k\times m}$ form an orthonormal basis of an m-dimensional left-invariant subspace $\tilde{\mathcal{V}}$ of H_{33}. Let $\Lambda \in \mathbf{C}^{m\times m}$ satisfy $\tilde{V}^H H_{33} = \Lambda \tilde{V}^H$. Define $V \in \mathbf{C}^{n\times m}$ by

$$V^H = \begin{array}{cccc} & n-k-1 & 1 & k \\ m & \big[\quad 0 & 0 & \tilde{V}^H \big] \end{array}. \tag{2.5}$$

If $\mathrm{Proj}_{\tilde{\mathcal{V}}}(P_3) = -\mathrm{Proj}_{\tilde{\mathcal{V}}}(H_{32})$, then $\tilde{V}^H P_3 = -\tilde{V}^H H_{32}$ and, by direct calculation, $V^H(H+P) = \Lambda V^H$. Hence, $\mathcal{V} \equiv \mathrm{Range}(V)$ is an m-dimensional left-invariant subspace of $H+P$. It follows from (2.5) that $\mathcal{V} \subset \mathcal{E}_1^{\perp}$. Hence, by Lemma 2.3, P is a deflating perturbation that deflates at least m eigenvalues and the left-invariant subspace $\mathcal{V}$.

Conversely, let P be a k-spike perturbation that deflates $\hat{m}$ eigenvalues, with $m \leq \hat{m} \leq k$. Let Q be a 1-unitary matrix chosen so that $\hat{H} = Q^H(H+P)Q$ is Hessenberg. The first k columns of $H+P$ are unreduced Hessenberg, because the first k columns of H are. Without loss of generality, we may choose Q so that $\hat{h}_{n-m+1,n-m} = 0$. The implicit Q theorem [29, Theorem 7.4.3] implies that Q takes the form

$$Q = \begin{array}{l} \\ n-k-1 \\ 1 \\ k-m \\ m \end{array} \begin{array}{c} \begin{array}{ccc} n-k-1 & 1 & k \end{array} \\ \left[\begin{array}{ccc} D_1 & 0 & 0 \\ 0 & D_2 & 0 \\ 0 & 0 & \hat{V} \\ 0 & 0 & \tilde{V} \end{array}\right] \end{array},$$

where D_1 and D_2 are diagonal and unitary. With this choice, the last m columns of Q span an m-dimensional left-invariant subspace $H+P$. In the notation of (2.1) and (2.4), there exists a matrix $\Lambda \in \mathbf{C}^{m\times m}$ for which

$$\left[\begin{array}{ccc} 0 & 0 & \tilde{V} \end{array}\right]^H \left[\begin{array}{ccc} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ 0 & H_{32}+P_3 & H_{33} \end{array}\right] = \Lambda \left[\begin{array}{ccc} 0 & 0 & \tilde{V} \end{array}\right]^H.$$

In particular, $\tilde{V}^H(H_{32}+P_3) = 0$ and $\tilde{V}^H H_{33} = \Lambda\tilde{V}^H$. So, $\mathcal{V} = \mathrm{Range}(\tilde{V}) \subset \mathbf{C}^{k\times k}$ is an m-dimensional left-invariant subspace of H_{33} and $\mathrm{Proj}_{\tilde{\mathcal{V}}}(P_3) = -\mathrm{Proj}_{\tilde{\mathcal{V}}}(H_{32})$.

An elementary application of linear least squares shows that for any subspace $\mathcal{V} \subset \mathbf{C}^k$, $P_3 = -\mathrm{Proj}_{\tilde{\mathcal{V}}}(H_{32})$ is the minimum Frobenius norm solution to $\mathrm{Proj}_{\tilde{\mathcal{V}}}(P_3) = -\mathrm{Proj}_{\tilde{\mathcal{V}}}(H_{32})$. □

2.3. Implementing aggressive early deflation. Theorem 2.4 shows how to calculate (hopefully) small norm k-spike perturbations P that deflate several eigenvalues:

1. Select a “deflation window” consisting of the trailing k-by-k principal submatrix H_{33}.
2. Select an m-dimensional left-invariant subspace $\mathcal{V} \subset \mathbf{C}^k$ of H_{33} such that $\|\mathrm{Proj}_{\mathcal{V}}(H_{32})\|_2$ is “small.”
3. Compute $P_3 = -\mathrm{Proj}_{\mathcal{V}}(H_{32})$. (The spike perturbation P is given by (2.4).)

If $\|P\|_F = \|P_3\|_2$ is negligible, then, in principle, it may be used to deflate m eigenvalues by returning $H+P$ to Hessenberg form with a 1-unitary similarity transformation $\hat{H} = Q^H(H+P)Q$.

There are several practical considerations. First, the calculation must be organized so that the computed version of $\hat{H} = Q^H(H+P)Q$ is indeed *reduced* Hessenberg despite rounding errors. Second, some flexible but effective heuristic must be used to select the invariant subspace. (Including the trivial space $\{0\}$, there are typically 2^k invariant subspaces of H_{33}. Computing bases of all of them is impractical.) Finally, if H is real, it would be best to restrict all calculations to real arithmetic.

The following procedure takes these practical considerations into account. Let $H \in \mathbf{R}^{n\times n}$ be a real, unreduced Hessenberg matrix. Make an a priori choice of the deflation window size k. (The success of aggressive early deflation is relatively insensitive to the deflation window size. However, it is best to choose k to be larger than the number of simultaneous shifts in the multishift QR algorithm.) Partition H as in (2.1). Using the QR algorithm (what else?), compute a real Schur decomposition of H_{33}, $V^T H_{33} V = T$, where $V \in \mathbf{R}^{k\times k}$ is orthogonal and $T \in \mathbf{R}^{k\times k}$ is quasi-triangular. (This might be implemented as a recursive subroutine.) The eigenvalues of H_{33} are arranged along the diagonal of T as the 1-by-1 and 2-by-2 diagonal blocks. For each integer $m < k$, for which $t_{k-m+1,k-m} = 0$, $\mathcal{V}_m = \mathrm{Range}(V_{k-m+1:k,:})$ is an m-dimensional left-invariant subspace of H_{33}. If $s = V^T H_{32} \in \mathbf{R}^k$, then $\|s_{k-m+1:k}\|_2 = \|-\mathrm{Proj}_{\mathcal{V}_m}(H_{32})\|_F$. By Theorem 2.4, $\|s_{k-m+1:k}\|_2$ is the magnitude of the minimal norm k-spike perturbation that deflates the trailing m eigenvalues of T. If $\|s_{k-m+1:k}\|_2$ is so tiny that it may be safely set to zero, then we may use the corresponding k-spike perturbation to deflate m eigenvalues. These m eigenvalues are *deflatable.*

For each possible ordering of the eigenvalues of T (keeping complex conjugate pairs adjacent), there is a real Schur decomposition which achieves that ordering along the diagonal of T. Software for efficiently updating one real Schur decomposition to another by unitary similarity transformation is widely available. (See, for example, [4, 39] or [1, `DTREXC`].) By reordering the eigenvalues along the diagonal of T, a deflation procedure may augment the set of deflatable eigenvalues. Suppose, for example, that the trailing m eigenvalues along the diagonal of T are deflatable, but $\lambda = t_{k-m,k-m}$ is a simple eigenvalue that cannot be classified as deflatable because $|s_{k-m}|$ is not "small enough." Use a unitary similarity transformation to rotate λ up to t_{11} while leaving the m eigenvalues already classified as deflatable in their original positions. The new value $t_{k-m,k-m}$ (or, in case of a complex conjugate pair, the 2-by-2 block $T_{k-m-1:k-m,k-m-1:k-m}$) now may be deflatable. In this way, each eigenvalue may be examined and classified as deflatable or nondeflatable. The deflatable eigenvalues collect in the trailing diagonal entries of T.

Having calculated a suitable, reordered, real Schur decomposition $H_{33} = VTV^T$ and $s = V^T H_{32}$, and having classified the trailing m eigenvalues as deflatable as described above, compute the similarity transformation

$$\begin{bmatrix} I & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & V \end{bmatrix}^T \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ 0 & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & V \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13}V \\ H_{21} & H_{22} & H_{23}V \\ 0 & s & T \end{bmatrix} \tag{2.6}$$

by multiplying H_{13} and H_{23} from the right by V. The trailing m entries of s are tiny—small enough to be set to zero. Set the trailing m entries of s to zero to obtain a perturbed vector $[\tilde{s}^T,\, 0]^T \equiv [s_{1:k-m}^T,\, 0]$. The matrix

$$\tilde{H} = \begin{array}{l} \\ n-k-1 \\ 1 \\ k-m \\ m \end{array} \begin{array}{c} \begin{array}{cccc} n-k-1 & 1 & k-m & m \end{array} \\ \begin{bmatrix} H_{11} & H_{12} & \tilde{H}_{13} & \tilde{H}_{14} \\ H_{21} & H_{22} & \tilde{H}_{23} & \tilde{H}_{24} \\ 0 & \tilde{s} & T_{11} & T_{12} \\ 0 & 0 & 0 & T_{22} \end{bmatrix} \end{array}$$

is block triangular with m-by-m quasi-triangular, trailing principal submatrix T_{22}. The m eigenvalues of T_{22} are eigenvalues of $\tilde{H}$.

It remains to return $\tilde{H}$ to reduced Hessenberg form. The first $n-k-1$ columns of $\tilde{H}$ are still in unreduced Hessenberg form, so the first $n-k-1$ steps of Householder's method [29, Algorithm 7.4.2] do not modify $\tilde{H}$. Hence, when returning $\tilde{H}$ to Hessenberg form, the first $n-k-1$ steps of Householder's method [29, Algorithm 7.4.2] are unnecessary and may be skipped. In addition Householder's method [29, Algorithm 7.4.2] does not modify $T_{22} = H_{n-m:n,n-m:n}$, so the last $m-1$ steps can be skipped as well. If $k^2 \ll n$, then the work required to return $H + P$ back to Hessenberg form is small compared to the cost of a QR sweep.

2.4. When are spike components negligible?

When an entry in the spike is "small enough," it may be safely set to zero without significant adverse effect on accuracy. Considerable care is needed to make this decision without unnecessarily sacrificing accuracy. A conservative stopping criterion was all that was needed to greatly increase the accuracy of the Jacobi method [19]. This subsection presents a short discussion of the question in the context of aggressive early deflation but does not give a definitive answer.

Setting the last m components of the spike to zero is equivalent to adding a k-spike perturbation of norm $\|s_{k-m+1:k}\|_2$. Rounding errors in the reduction to Hessenberg form and during the QR sweep are equivalent to perturbing H by an additive perturbation matrix of norm $p(n)\mu\|H\|_2$, where μ is the unit roundoff and $p(n)$ is a low degree polynomial that depends on the the norm, the details of the algorithm, and the details of the finite precision arithmetic. (See [45, Chap. 3].) It is natural to require that $\|s_{k-m+1:k}\|_2$ be at least roughly as small as the other unavoidable rounding errors before it can be set to zero. This suggests what might be called *norm-stable deflation.*

Norm-stable deflation:

The last m components of the spike may be set to zero if, for some rounding error small number ε, $\|s_{k-m+1:k}\|_2 \leq \varepsilon\|H\|_F$.

Requiring spike components to satisfy this criterion before being set to zero is a minimum requirement. If H is a graded matrix, even norm-stable deflation may deflate approximate eigenvalues before they have converged to a limiting accuracy. The more conservative deflation strategies described below are empirically reliable and, in some cases, yield more accurate computed eigenvalues.

In terms of the partitioning (2.1), aggressive early deflation depends only on H_{32} and H_{33}. It is natural to use a deflation criterion that depends only upon them. This suggests what we call window-norm-stable deflation.

Window-norm-stable deflation:

The last m components of the spike may be set to zero if, for some rounding error small number ε, $\|s_{k-m+1:k}\|_2 \leq \varepsilon\|[H_{32},\ H_{33}]\|_F$.

Window-norm-stable deflation compromises between norm-stable deflation and the EISPACK and LAPACK compare-to-nearby-diagonal-entries strategy. Note that the deflation window size is likely to be substantial. (In the numerical examples reported in section 3, window sizes ranged from $k = 48$ to $k = 450$.) Hence, window-norm-stable deflation resembles norm-stable deflation and suffers many of the same drawbacks.

EISPACK [28, 37] and LAPACK [1] use small-subdiagonal deflation. Subdiagonal entries are considered small enough to deflate only if they are tiny compared to *nearby* matrix entries. EISPACK [28, 37] subroutines `HQR/HQR2` and LAPACK [1] subroutines `DHSEQR/DLAHQR` ordinarily set a subdiagonal entry $h_{i+1,i}$ to zero only if $|h_{i+1,i}| \leq \mu(|h_{ii}| + |h_{i+1,i+1}|)$, where μ is the unit roundoff. (If both $h_{ii} = 0$ and $h_{i+1,i+1} = 0$, then EISPACK falls back on norm-stable deflation. LAPACK falls back on a modified norm-stable deflation criterion. LAPACK also sets $h_{i+1,i}$ to zero when it is near the underflow threshold.) This more conservative deflation strategy respects the local scale of graded matrices and sometimes significantly improves the accuracy of the computed eigenvalues.

A deflation strategy in the spirit of the EISPACK/LAPACK compare-to-nearby-diagonal-entries strategy might be to compare the spike component s_k to the diagonal block of T in row k. For example, if $t_{k,k}$ is a 1-by-1 diagonal block in T, i.e., a real eigenvalue of T, then s_k might be set to zero if it is tiny compared to t_{kk}. Once s_k has been set to zero, the same deflation strategy may be applied to s_{k-1} and then to s_{k-2} and so on until a nondeflatable eigenvalue is encountered. If $T_{k-1:k,k-1:k}$ is a 2-by-2 diagonal block with complex conjugate eigenvalues λ and $\bar{\lambda}$, then a similar deflation strategy may be applied to the two spike components s_{k-1} and s_k by comparing them to $\|T_{k-1:k,k-1:k}\|_2$ or $|\lambda| = \sqrt{\det(T_{k-1:k,k-1:k})}$.

Nearby-diagonal deflation:

If t_{jj} is a 1-by-1 diagonal block of T, then s_j may be set to zero if, for some rounding error small number ε, $|s_j| \le \varepsilon\, |t_{jj}|$.

If $T_{j:j+1,j:j+1}$ is a 2-by-2 diagonal block of T, then s_j and s_{j+1} may be set to zero if, for some rounding error small number ε, $\max(s_j, s_{j+1}) \le \varepsilon\sqrt{\det(T_{j:j+1,j:j+1})}$.

Recall that the spike is $s = V^T H_{32} = h_{n-k+1,n-k} V_{1,:}$, where $V^T H_{33} V = T$ is a real Schur decomposition. Setting $s_{k-m+1:k}$ to zero is equivalent to perturbing V to a nearby matrix $\tilde{V}$ by setting $V_{1,k-m+1:k}$ to zero. This corresponds to replacing the Schur decomposition $H_{33} = VTV^T$ by the multiplicatively perturbed factorization $(I+E)^{-1} H_{33} (I+E) = ((I+E)^{-1}V)T((V^T(I+E))) = \tilde{V}^{-T} T \tilde{V}^T$, where E satisfies $V^T(I+E) = \tilde{V}^T$. The matrix E may be chosen to be a 1-spike perturbation with $\|E\|_2 = \|E\|_F = \|V_{1,k-m+1:k}\|_2 = \|s_{k-m+1:k}\|_2 / h_{n-k+1,n-k}$.

Finite precision arithmetic nearly always prevents the computed version of V from being exactly orthogonal, but it can be shown that the computed V is of the form $\hat{V}(I+\hat{E})$, where $\hat{V}$ is exactly orthogonal and $\|\hat{E}\|_2$ is rounding error small [45, Chap. 3]. Hence, if $\|E\|_2$ is rounding error small, then replacing V by $\tilde{V} = V(I+E)$ is a perturbation of similar character to and similar or smaller magnitude to the unavoidable rounding errors already contaminating V. Similarly, rounding errors in the computation of the Schur decomposition of H_{33} are equivalent to a perturbation of H_{33} that is typically at least as large as (and probably *less* structured than) $(I+E)^{-1}H_{33}(I+E)$ [29, p. 381]. Therefore, the components of $V_{1,k-m+1:k}$ may not be distinguished from zero within rounding error generated uncertainties. This suggests the next deflation strategy.

Window-Schur deflation:

The last m components of the spike may be set to zero if, for some rounding error small number ε, $\|s_{k-m+1:k}\|_2 \le \varepsilon \|H_{32}\|_2$.

The numerical examples reported in this paper use both nearby-diagonal deflation and window-Schur deflation by setting small spike components to zero if either criterion is satisfied.

At this point it is safe to remark that aggressive early deflation equipped with any of the above strategies is a "normwise" backward numerically stable procedure. It uses only orthogonal matrix computations well known to be backward numerically stable in the presence of rounding errors [45], along with tiny perturbations that are equivalent to normwise tiny perturbations of the original matrix A.

2.5. Combining aggressive early deflation with small-subdiagonal deflation. In our experience, aggressive early deflation is more powerful than small-subdiagonal deflation. However, it does not replace small-subdiagonal deflation entirely. Occasionally, a tiny subdiagonal entry may appear outside of the deflation window. Such an opportunity for deflation goes undetected by aggressive early deflation. Even within the deflation window, aggressive early deflation may miss an opportunity to deflate that small-subdiagonal deflation does not. For example, let

$$H = \begin{bmatrix} 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & \varepsilon & 2 & 0 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix},$$

where $0 < \varepsilon \ll 1$. Suppose that ε is negligible (i.e., small enough to be safely set to zero), but $\sqrt{\varepsilon}/2$ is not. The small-subdiagonal strategy would deflate by setting ε to zero.

With a deflation window size of $k = 4$, H_{33} is the trailing principal 4-by-4 submatrix of H, and H_{32} is the first column of the 4-by-4 identity matrix. The eigenvalues of H_{33} are $2 \pm \varepsilon^{1/4}$ and $2 \pm i\varepsilon^{1/4}$ with corresponding almost-normalized right eigenvectors $[\pm\varepsilon^{1/2}, \varepsilon^{3/4}, \pm 1, \varepsilon^{1/4}]$ and $[\pm i\varepsilon^{1/2}, -\varepsilon^{3/4}, \mp i, \varepsilon^{1/4}]$. For any eigenvalue ordering in the Schur decomposition of H_{33}, the tip of the spike has magnitude $|s_4| = \sqrt{\varepsilon}\left(1 + \varepsilon^{1/2} + \varepsilon + \varepsilon^{3/2}\right)^{-1/2} > \sqrt{\varepsilon}/2$, which is not negligible.

It is inexpensive to monitor the subdiagonal entries during the course of a QR sweep and take advantage of a small-subdiagonal deflation should one occur. Note that this includes taking advantage of any small-subdiagonal entries that may appear between the bulges during a two-tone multishift QR sweep [9, 10]. In [42], this is called "vigilant deflation."

Ordinarily, the bulges above a vigilant deflation collapse as they encounter the newly created subdiagonal zero. This may block those shifts so that, for the current QR sweep, the benefit of using many simultaneous shifts may be lost. However, the bulges can be reintroduced, in the row in which the new zero subdiagonal appears, using the same methods that are used to introduce bulges at the upper left-hand corner [20, 24, 25], [29, p. 377], [45, p. 530]. In this way, the shift information passes through a zero subdiagonal and the two-tone QR sweep continues with all its shifts.

2.6. Choice of shifts. It is well known that a good choice of shifts in the QR algorithm leads to rapid convergence. Shifts selected to be the eigenvalues of a trailing principal submatrix give local quadratic convergence [45, 43]. In the symmetric case, convergence is cubic [46]. Deferred shifts retard convergence [40]. The implementation of the large-bulge multishift QR algorithm in LAPACK selects its shifts to be the eigenvalues of a trailing principal submatrix.

With aggressive early deflation, it is natural to select the shifts from among the nondeflatable eigenvalues. This saves an extra small eigenvalue calculation and incorporates more information from the matrix into the shifts. In the numerical examples reported in this paper, we arbitrarily select the shifts to be the nondeflatable eigenvalues that appear lowest along the diagonal of T in (2.6).

2.7. Iterated early deflation. In our experience, aggressive early deflation is so effective that it is often better to skip a multishift QR sweep and immediately apply the aggressive early deflation strategy again to the remaining, undeflated Hessenberg matrix. Aggressive early deflation can be applied over and over again, sometimes deflating a great many eigenvalues without the cost of a QR sweep. We find that it is best to skip the next multishift QR sweep whenever aggressive early deflation isolates more than a few eigenvalues.

2.8. Analysis of early deflation. This subsection gives a partial explanation of the success of aggressive early deflation.

Let $H_{33} = VTV^T$ be the real Schur decomposition in (2.6). Suppose that $t_{nn} = \lambda \in \mathbf{R}$ is a 1-by-1 diagonal block in the quasi-triangular structure of T, i.e., $\lambda = t_{nn}$ is a real eigenvalue of H_{33}. The early deflation strategy finds at least one deflatable eigenvalue if the tip of the spike, $s_k = h_{n-k+1,n-k}v_{1k}$, is small enough. (The tip of the spike is s_k, the last component of s in (2.6).)

Let $QR = (H_{33} - \lambda I)$ be a QR factorization. Because $H_{33} - \lambda I$ is a singular, unreduced Hessenberg matrix, $Q \in \mathbf{R}^{k \times k}$ is also unreduced Hessenberg and the last row of the triangular factor R is zero. The last column of Q is a normalized left eigenvector of H_{33} corresponding to the eigenvalue λ. The unreduced structure of H_{33} also implies that λ has geometric multiplicity one and that its corresponding real, normalized left eigenvector is unique up to scaling by a complex number of modulus one. The last column of the matrix of Schur vectors V is also a normalized left eigenvector corresponding to λ, and, in particular, $|q_{1k}| = |v_{1k}|$. The tip of the spike may be written as $|s_k| = |h_{n-k+1,n-k}v_{1k}| = |h_{n-k+1,n-k}q_{1k}|$.

The matrix Q is Hessenberg, so the $(1,k)$th cofactor is

$$(-1)^{k+1}Q(1|k) = (-1)^{1+k}\prod_{i=1}^{k-1} q_{i+1,i}.$$

However, Q is also orthogonal, so

$$Q^{-1} = Q^T = \operatorname{adj}(Q)/\det(Q).$$

(Here, $\operatorname{adj}(Q)$ is the classical adjoint matrix or adjugate of Q.) Hence,

$$q_{1k} = \frac{(-1)^{k+1}\prod_{i=1}^{k-1} q_{i+1,i}}{\det(Q)}.$$

If $\bar{q}$ is the geometric mean

$$\bar{q} = \left|\prod_{i=1}^{k-1} q_{i+1,i}\right|^{1/(k-1)},$$

then, because $|\det(Q)| = 1$, the tip of the spike has modulus

$$|s_k| = |h_{n-k+1,n-k}q_{1k}| = |h_{n-k+1,n-k}|\,\bar{q}^{k-1}.$$

The $q_{i+1,i}$'s are entries in an orthogonal matrix, so for each i, $|q_{i+1,i}| \le 1$ and, hence, $\bar{q} \le 1$. Even when $\bar{q}$ is only moderately smaller than one, $\bar{q}^{k-1}$ may be tiny. For example, if $\bar{q} \le 1/2$ and the deflation window size is, say, $k > 50$, then $\bar{q}^{k-1} \le 9\times 10^{-16}$ and the tip of the spike, $|s_k| \le 9 \times 10^{-16}\,|h_{n-k+1,n-k}|$, may well be small enough to set to zero. Note that this can occur even when no subdiagonal of Q is particularly small and when many have modulus one or close to one.

The geometric mean of the subdiagonal entries of H_{33}, and $h_{n-k+1,n-k}$,

$$\bar{h} = \left|\prod_{j=n-k}^{n-1} h_{j+1,j}\right|^{1/k},$$

is proportional to $\bar{q}$. To see this, note that the Hessenberg, orthogonal structure of Q, the upper triangular structure of R, $QR = (H_{33} - \lambda I)$, implies that for $i = 1, 2, 3, \ldots, n-1$,

$$q_{i+1,i}r_{ii} = h_{n-k+i+1,n-k+i}$$

and

$$(H_{33} - \lambda I)(k|k) = Q(k|k)R(k|k).$$

Also, because $Q^T = Q^{-1} = \operatorname{adj}(Q)/\det(Q)$, $|q_{kk}| = |Q(k|k)|$. The tip of the spike can then be expressed as

$$\begin{aligned} |s_k| &= |h_{n-k+1,n-k}|\,\bar{q}^{k-1} \\ &= |h_{n-k+1,n-k}| \prod_{i=1}^{k-1} \left| \frac{h_{n-k+i+1,n-k+i}}{r_{ii}} \right| \\ &= \frac{\bar{h}^k}{|R(k|k)|} \qquad (2.7) \\ &= \frac{|q_{kk}|\,\bar{h}^k}{|(H_{33} - \lambda I)(k|k)|}. \qquad (2.8) \end{aligned}$$

This expression can be simplified under the assumption that λ has algebraic multiplicity one. In that case, let $H_{33} = XJX^{-1}$ be the Jordan canonical form of H_{33} ordered so that the 1-by-1 Jordan block corresponding to λ appears in the lower right-hand corner of J. For notational convenience, set $Z = X^{-1}$, and partition $H_{33} - \lambda I = X(J - \lambda I)Z$ as

$$H_{33} - \lambda I = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} \begin{bmatrix} (J_{11} - \lambda I) & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix},$$

where the $(1,1)$ blocks are $(k-1)$-by-$(k-1)$, the $(1,2)$ blocks are $(k-1)$-by-1, the $(2,1)$ blocks are 1-by-$(k-1)$, and the $(2,2)$ blocks are 1-by-1. It follows that $(H_{33} - \lambda I)(k|k) = \det(X_{11}(J_{11} - \lambda I)Z_{11})$. Now, $Z = X^{-1} = \operatorname{adj}(X)/\det(X)$ and, in particular, $Z_{22} = \det(X_{11})/\det(X)$. Similarly, $X_{22} = \det(Z_{11})\det(X)$. Without loss of generality, we may choose X and $Z = X^{-1}$ such that those rows of Z which are left eigenvectors have 2-norm equal to one. The last row of Z is a normalized left eigenvector of H_{33} corresponding to the eigenvalue λ, so $|q_{kk}| = |Z_{22}| = |\det X_{11}/\det X|$. Hence, (2.8) becomes

$$\begin{aligned} |s_k| &= \frac{|q_{kk}|\,\bar{h}^k}{|(H_{33} - \lambda I)(k|k)|} \\ &= \frac{|q_{kk}|\,\bar{h}^k}{|\det(X_{11}(J_{11} - \lambda I)Z_{11})|} \\ &= \frac{|\bar{h}^k|}{|\det(J_{11} - \lambda I)\det(X)\det(Z_{11})|} \\ &= \frac{|\bar{h}^k|}{|\det(J_{11} - \lambda I)|\,|X_{22}|}. \end{aligned}$$

Finally, if $\bar{\mu}$ is the geometric mean of the differences between λ and the other eigenvalues of H_{33},

$$\bar{\mu} = |\det(J_{11} - \lambda I)|^{1/(k-1)},$$

then

$$|s_k| = \frac{\bar{h}^k}{\bar{\mu}^{k-1}\,|X_{22}|},$$

where X_{22} is the last component of a right eigenvector corresponding to eigenvalue λ.

Like $\bar{q}^{k-1}$, $\bar{h}^k$ may be tiny even when no subdiagonal $h_{j+1,j}$ is particularly small. If λ is well separated from the other eigenvalues H_{33} in the sense that $\bar{\mu}^{k-1}\,|X_{22}|$ is not "too small," then even a moderately small geometric mean $\bar{h}$ may make λ a deflatable eigenvalue.

The above observations apply equally well to all normalized left eigenvectors, but, of course, not all of them will have tiny first components (except, perhaps, in the case of a highly ill-conditioned eigenvalue). Suppose H_{33} is diagonalizable and that m of the normalized left eigenvectors of H_{33} have tiny first components. Stack the left eigenvectors as the rows of a matrix Y with the distinguished set of m vectors on the bottom. Now, $H_{33} = Y^{-1}DY$, where D is the diagonal matrix of eigenvalues. If $H_{33} = VTV^H$ is a complex Schur decomposition with eigenvalues ordered along the diagonal of T in the same order as along the diagonal of D, then $R = YV$ is upper triangular and $R^{-1}Y_{:,1} = V^H_{1,:}$. In particular, the trailing m components of the first row of V can be bounded in terms of the corresponding components of Y and the trailing m-by-m principal submatrix of R as

$$\begin{aligned}\|V^H_{k-m+1:k,1}\|_2 &= \|R^{-1}_{k-m+1:k,k-m+1:k}Y_{k-m+1:k,1}\|_2\\ &\le \|R^{-1}_{k-m+1:k,k-m+1:k}\|_2\|Y_{k-m+1:k,1}\|_2.\end{aligned}$$

By assumption, $\|Y_{k-m+1:k,1}\|_2$ is tiny. So is $\|V_{1,k-m+1:k}\|_2$, if $\|R^{-1}_{k-m+1:k,k-m+1:k}\|_2$ is not "too big," i.e., if no eigenvalue corresponding to the selected m left eigenvectors is "too ill-conditioned."

3. Numerical examples.

We compared the performance of the large-bulge multishift QR algorithm [3] and the small-bulge multishift QR algorithm [9, 10] with and without aggressive early deflation. The test matrices included ad hoc and pseudorandom Hessenberg matrices of order 500-by-500 to 10,000-by-10,000 and nonrandom matrices of similar order taken from a variety of applications in science and engineering [2].

Computational environment. The numerical examples were run on an Origin2000 computer equipped with 400MHz IP27 R12000 processors and 16 gigabytes of memory. Each processor uses 32 kilobytes of level 1 instruction cache, 32 kilobytes of level 1 data cache, and 8 megabytes of level 2 combined instruction and data cache. For serial execution, the experimental Fortran implementation of the small-bulge multishift QR algorithm was compiled with version 7.30 of the MIPSpro Fortran 77 compiler with options `-64 -TARG:platform=ip27 -Ofast=ip27 -LNO`. The same options were used to compile `DHSEQR` from LAPACK version 2. For parallel execution the `-mp` and `-pfa` options were added. The programs called optimized LAPACK and BLAS subroutines from the SGI/Cray Scientific Library version 1.2.0.0.

In our computational environment, we observed that the measured serial "cpu time" of any particular program with its particular data might vary by at most a few percent. We were fortunate to get exclusive use of several processors for the purpose of timing parallel benchmark runs.

Except where otherwise mentioned, n-by-n matrices were stored in n-by-n arrays.

We report the floating point execution rate in millions of floating point instructions per second or "mega-flops" for short. (A trinary multiply-add operation counts as one instruction although it executes two flops.) For comparison purposes, we measured the floating point execution rate of the level 3 BLAS matrix-matrix multiply subroutine `DGEMM` and triangular matrix multiply subroutine `DTRMM` from the SGI/Cray Scientific Library version 1.2.0.0 applied to matrix products similar to those that dominate the small-bulge multishift QR algorithm [9, 10]. In serial execution, in the Origin2000 computational environment described above, DGEMM computes the product of the transpose of a 200-by-200 matrix times a 200-by-10,000 slab embedded a 10,000-by-10,000 array at roughly 330 mega-flops. It computes the product of a 10,000-by-200 slab times a 200-by-200 matrix at roughly 325 mega-flops. DTRMM computes the product of the transpose of a triangular 200-by-200 matrix times a 200-by-10,000 slab at roughly 305 mega-flops if it is an upper triangular matrix and at roughly 260 mega-flops if it is a lower triangular matrix. DTRMM computes the product of a 10,000-by-200 slab times a 200-by-200 triangular matrix at roughly 275 mega-flops if it is an upper triangular matrix and at roughly 255 mega-flops if it is a lower triangular matrix.

Implementation details. We call our experimental Fortran implementation of the small-bulge multishift QR algorithm *without* aggressive early deflation TTQR and *with* aggressive early deflation TTQRE. As described in [9, 10], the small-bulge multishift QR algorithm avoids the phenomenon of shift-blurring by chasing a tightly packed chain of m small bulges [9, 10]. Both TTQR and TTQRE use vigilant small-subdiagonal deflation [42]. Following EISPACK [36] and LAPACK [1], a Hessenberg subdiagonal entry $h_{i+1,j}$ is set to zero when $|h_{i+1,i}| \leq \varepsilon\,(|h_{ii}| + |h_{i+1,i+1}|)$ with ε equal to the unit roundoff.

The experimental implementation of TTQRE uses both nearby-diagonal deflation and window-Schur deflation by setting small spike components to zero if either criterion is satisfied. If the early deflation procedure with a k-by-k deflation window isolates $15k/100$ or more eigenvalues, then TTQRE skips the next QR sweep and immediately applies the early deflation procedure again to the remaining unreduced Hessenberg submatrix. In this way, TTQRE is sometimes able to isolate a great many eigenvalues without the expense of a QR sweep.

Both TTQR and TTQRE use LAPACK subroutine DHSEQR [1], the conventional large-bulge QR algorithm, to reduce diagonal subblocks of order no greater than 1.5 times the number of simultaneous shifts.

For the large-bulge multishift QR algorithm, we use subroutine DHSEQR from LAPACK version 2 which is widely recognized to be an excellent implementation. For the large-bulge multishift QR algorithm with aggressive early deflation, we modified DHSEQR by inserting aggressive early deflation following the search for small subdiagonals. The modified program performs a large-bulge QR sweep only if no deflations are found. We call the resulting program DHSEQRE.

For reference, Table 2 lists names and short descriptions of the four algorithms.

Choosing shift multiplicity and size of the deflation window. As of this writing, it is not well understood how best to choose the number of simultaneous shifts and the size of the deflation window. The relationship between shift multiplicity, deflation window size, and execution time is a complex interaction between the character of the Hessenberg matrix, the hardware architecture of the computational environment, and the not-yet-well-understood convergence behavior of aggressive early deflation. Cache size, cache strategy, and placement of data in machine memory can have a strong effect on execution time.

In Example 1, we did extensive preliminary experiments to determine good choices of these parameters for each algorithm. However, these choices may or may not perform well on a computer with a different architecture or when applied to Hessenberg matrices of different character or different order. In Example 2, we did extensive

TABLE 2

Names and descriptions of the algorithms in the numerical experiments described in section 3.

Name	Description
`DHSEQR`:	The large-bulge multishift QR algorithm [3] using only small-subdiagonal deflation as implemented in LAPACK 2.0 [1].
`DHSEQRE`:	The large-bulge multishift QR algorithm [3] using both small-subdiagonal and aggressive early deflation.
`TTQR`:	The two-tone, small-bulge multishift QR algorithm [9, 10] using only small-subdiagonal deflation.
`TTQRE`:	The two-tone, small-bulge multishift QR algorithm [9, 10] using both small-subdiagonal deflation and aggressive early deflation.

preliminary experiments on the order $n = 5{,}000$ example only. For the other examples, the choices are based upon a few preliminary experiments and ad hoc educated guesses. Consequently, no particular significance should be attached to the shift multiplicities and deflation window sizes used in this paper.

In the examples reported here, we use a fixed number of simultaneous shifts and a fixed deflation window size throughout. However, there may be an advantage in choosing these parameters dynamically as the algorithm progresses.

Example 1. We ran `DHSEQR`, `DHSEQRE`, `TTQR`, and `TTQRE` on pseudorandomly generated Hessenberg matrices of various sizes from 500-by-500 to 1,000-by-1,000. We selected the entries on the diagonal and upper triangle to be normally distributed pseudorandom numbers with mean zero and variance one. We set the subdiagonal entry $h_{j,j+1} = \sqrt{\chi^2_{n-j}}$, where χ^2_{n-j} is selected from a Chi-squared distribution with $n - j$ degrees of freedom. These pseudorandom Hessenberg matrices have essentially the same distribution as if a matrix of normally distributed, mean zero, variance one pseudorandom variables had been reduced to Hessenberg form using [29, Algorithm 7.4.2].

Figure 1 displays the serial execution time, rate of floating point instruction execution, and hardware count of executed floating point instructions for pseudorandom Hessenberg test matrices of orders $n = 500$ to $n = 1{,}000$ using the Origin2000 computer described above.

To choose the number of simultaneous shifts and the deflation window size, we ran preliminary experiments using a wide variety of simultaneous shifts and deflation window sizes. Good choices of the parameters were usually different for each of the different algorithms. Even for individual algorithms no single choice minimizes execution time over the whole range of orders $n = 500$ to $n = 1{,}000$. However, the following choices result in execution times that are no more than 10% longer than the minimum that we observed. In Figure 1, `DHSEQR` uses 6 simultaneous shifts, `DHSEQRE` uses 22 simultaneous shifts, and both `TTQR` and `TTQRE` use 60 simultaneous shifts. `DHSEQRE` uses a 48-by-48 deflation window and `TTQRE` uses a 90-by-90 deflation window. (The execution time of no algorithm would be reduced by changing the number of simultaneous shifts or the deflation window size to the choice used by one of the other algorithms. For example, reducing the number of simultaneous shifts used by `DHSEQRE` to 6 increases its execution time between 10% and 50%. Increasing the number of simultaneous shifts used by `DHSEQR` to 22 increases its execution time

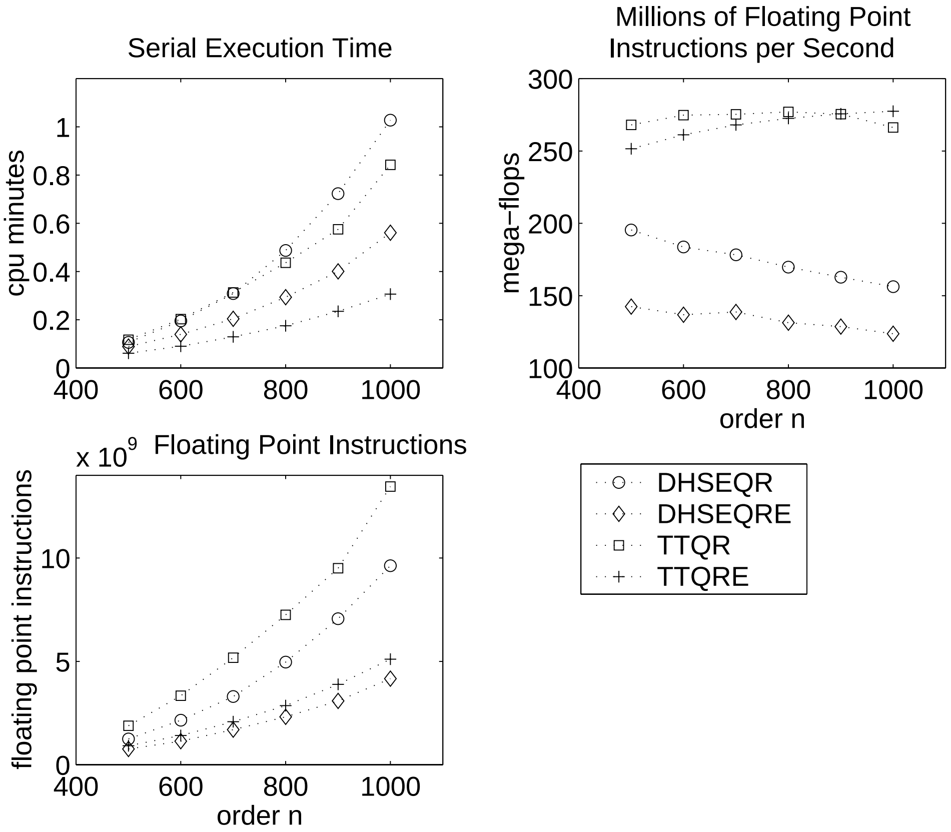

FIG. 1. *Serial cpu execution time, rate of floating point instruction execution, and hardware count of floating point instructions executed by* DHSEQR*,* DHSEQRE*,* TTQR*, and* TTQRE *computing the quasi-triangular and orthogonal Schur factors of pseudorandom Hessenberg matrices. In this figure,* DHSEQR *uses* 6 *simultaneous shifts,* DHSEQRE *uses* 22 *simultaneous shifts, and both* TTQR *and* TTQRE *use* 60 *simultaneous shifts.* DHSEQRE *uses a* 48*-by-*48 *deflation window and* TTQRE *uses a* 90*-by-*90 *deflation window.*

between 35% and 95%.)

To compare the effects of rounding errors in each of the algorithms, we computed the relative residual $\|A\tilde{Q}-\tilde{Q}\tilde{T}\|_F/\|A\|$ and the departure from orthogonality $\|\tilde{Q}^T\tilde{Q}-I\|_F/\sqrt{n}$, where $\tilde{Q}$ is the computed nearly orthogonal factor and $\tilde{T}$ is the computed quasi-triangular factor. (The matrix $\tilde{Q}$ would be orthogonal were it not for rounding errors.) For all tested matrix orders between 500 and 1,000, and for all four algorithms, the relative residuals and departure from orthogonality lie between $\frac{1}{2}\times 10^{-14}$ and 2×10^{-14}. This compares well with the unit roundoff of the finite precision arithmetic of 2.22×10^{-16}. As measured by "normwise backward error," the four algorithms are empirically numerically stable and of roughly equal accuracy.

Figure 1 demonstrates that aggressive early deflation is effective at reducing both the execution time and the number of floating point instructions executed by both the large- and small-bulge multishift QR algorithm. In addition, the level 3 BLAS based small-bulge multishift QR algorithm used by `TTQR` and `TTQRE` [9, 10] maintains a relatively high rate of execution of floating point instructions compared to the level 2 BLAS based large-bulge QR algorithm used by `DHSEQR` and `DHSEQRE`.

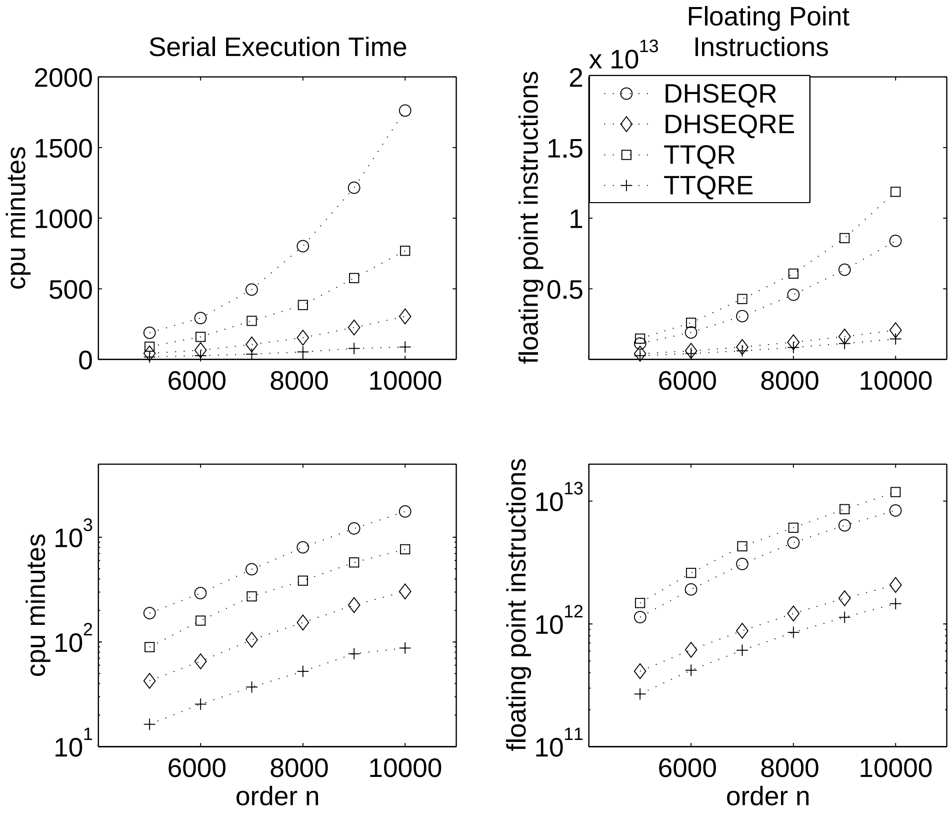

Fig. 2. *Serial cpu execution time of and floating point instructions executed by* DHSEQR, DHSEQRE, TTQR, *and* TTQRE *computing both the orthogonal and quasi-triangular factor of the Schur decomposition of pseudorandom Hessenberg matrices. The same data is displayed on linear and semilog plots. In this figure, both* DHSEQR *and* DHSEQRE *use* 8 *simultaneous shifts.* DHSEQRE *uses a* 150*-by-*150 *deflation window.* TTQR *uses* 150 *simultaneous shifts per QR sweep.* TTQRE *uses* 200 *simultaneous shifts with a* 450*-by-*450 *deflation window. Computational costs for the reduction to Hessenberg form are not included.*

Example 2. We repeated the previous numerical experiment using pseudorandom Hessenberg matrices of orders between $n = 5{,}000$ and $n = 10{,}000$.

To choose the number of simultaneous shifts and the deflation window size, we ran preliminary experiments on pseudorandom matrices of order $n = 5{,}000$ using a wide variety of choices of these parameters. The following choices yield the minimum execution times that we observed for the order $n = 5{,}000$ test matrix. In Figure 2, both DHSEQR and DHSEQRE use 8 simultaneous shifts. TTQR uses 150 simultaneous shifts, and TTQRE uses 200 simultaneous shifts. DHSEQRE used a 150-by-150 deflation window, and TTQRE used a 450-by-450 deflation window.

In this example, TTQR and TTQRE execute between 240 and 280 million floating point instructions per second (counting a trinary multiply-add operation as one instruction) with an average of 260 and 270, respectively. There is no obvious trend in the execution rates. The rate of floating point instruction execution by DHSEQR gradually declines from 100 million floating point instructions per second for the 5,000-by-5,000 example down to 80 million floating point instructions per second for the 10,000-by-10,000 example. The rate of floating point instruction execution by

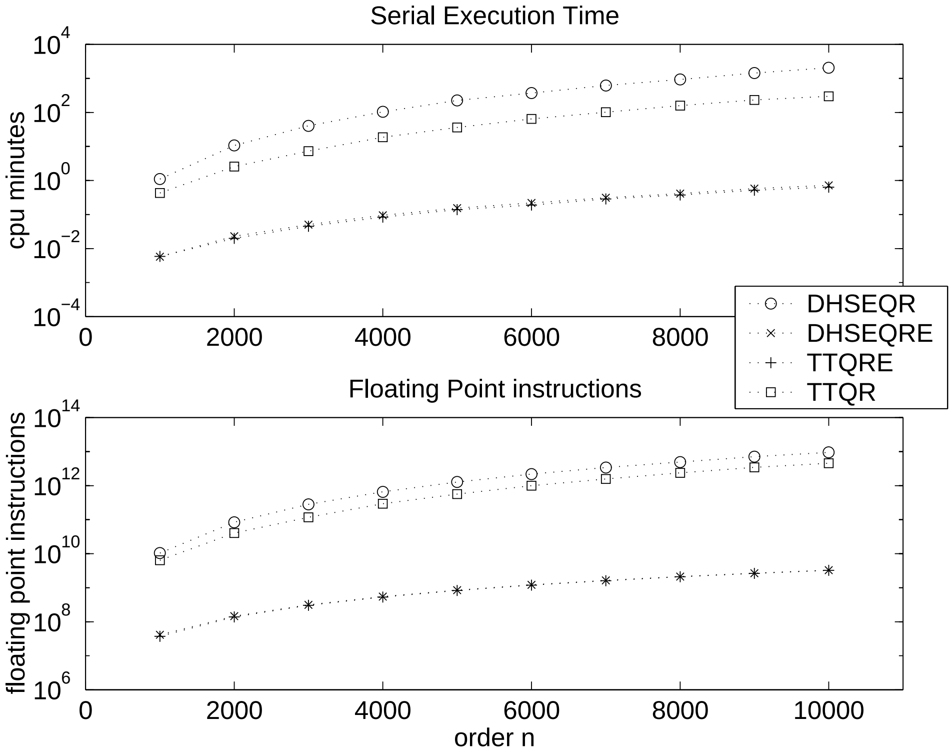

FIG. 3. *Serial execution time, number of floating point instructions executed, and floating point execution rate of* DHSEQR*,* DHSEQRE*,* TTQR*, and* TTQRE *computing both the orthogonal and quasi-triangular factor of the Schur decomposition of matrices of the form of* (1.1)*. The two algorithms* DHSEQRE *and* TTQRE *perform essentially identical operations, so their graphs coincide.* DHSEQR *uses* 6 *simultaneous shifts.* TTQR *uses* 150 *simultaneous shifts. Both* DHSEQRE *and* TTQRE *use a tiny* 10*-by-*10 *deflation window. (Similar results are obtained using larger deflation windows.)*

DHSEQRE gradually declines from roughly 160 down to 113 million floating point instructions per second.

As n increases from 5,000 to 10,000, DHSEQR took 12 to 20 times longer, DHSEQRE took 2.6 to 3.5 times longer, and TTQR took 5.5 to 8.8 times longer than TTQRE. In addition, DHSEQR executed 4.2 to 5.7 times as many, DHSEQRE executed 1.4 to 1.5 times as many, and TTQR executed 5.5 to 8.1 times as many floating point instructions as TTQRE.

Example 3. This example shows aggressive early deflation at its best. Figure 3 displays serial execution time and hardware count of executed floating point instructions for DHSEQR, DHSEQRE, TTQR, and TTQRE applied to matrices of the form of (1.1) of orders $n = 1{,}000$ to $n = 10{,}000$. DHSEQR uses 6 simultaneous shifts. TTQR uses 150 simultaneous shifts. Both DHSEQRE and TTQRE use a tiny 10-by-10 deflation window. (Similar results are obtained using larger deflation windows.)

In this example, DHSEQRE and TTQRE complete the Schur decomposition in 0.04% to 0.5% of the execution time used by DHSEQR and in 0.25% to 1% of the execution time used by TTQR. TTQR maintains a floating point execution rate of over 250 million floating point instructions per second throughout. Execution rates for the other three programs drop from roughly 120 million floating point instructions per second for the

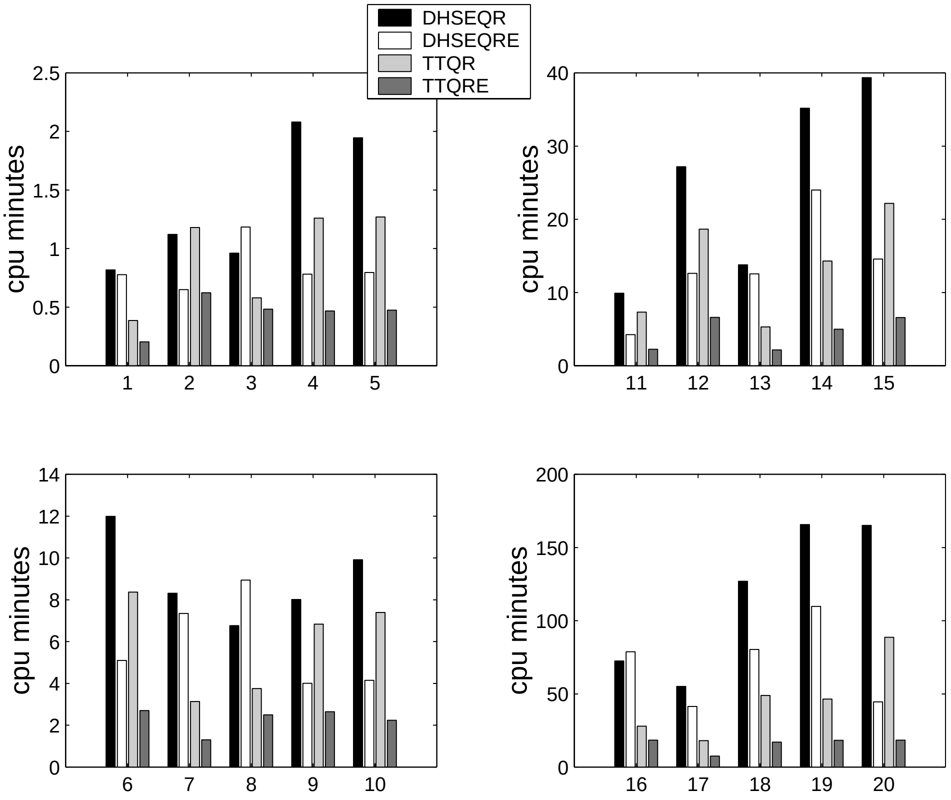

Fig. 4. *Serial execution times of* `DHSEQR`*,* `DHSEQRE`*,* `TTQR`*, and* `TTQRE` *computing the orthogonal and quasi-triangular factor of the Schur decompositions of* 20 *non-Hermitian eigenvalue selected from* [2]*. The numbers along the bottom edge of the bar graphs correspond to the numbers in the left-hand column of Table* 3*. Both* `DHSEQR` *and* `DHSEQRE` *use* 6 *simultaneous shifts.* `DHSEQRE` *uses a* 50*-by-*50*,* 100*-by-*100*, or* 150*-by-*150 *deflation window when the order* n *of the test matrix falls in the range* $1{,}000 \leq n < 2{,}000$*,* $2{,}000 \leq n < 4{,}000$*, and* $4{,}000 < n$*, respectively.* `TTQR` *uses* 60*,* 116*,* 150*, or* 180 *simultaneous shifts when the order* n *of the text matrix falls in the range* $1{,}000 \leq n < 2{,}000$*,* $2{,}000 \leq n < 2{,}500$*,* $2{,}500 \leq n < 4{,}000$*, and* $4{,}000 \leq n$*, respectively.* `TTQRE` *uses* 90*,* 120*,* 180*,* 240*, or* 270 *simultaneous shifts when the order* n *of the test matrix falls in the range* $1{,}000 \leq n < 2{,}000$*,* $2{,}000 \leq n < 2{,}500$*,* $2{,}500 \leq n < 4{,}500$*,* $4{,}500 \leq n < 7{,}000$*, and* $7{,}000 < n$*, respectively. As usual,* `TTQRE` *uses a deflation window of order* 1.5 *times the number of simultaneous shifts. (The execution time of the reduction to Hessenberg form is not included.)*

$n = 1{,}000$ example down to roughly 75 million floating point instructions per second for the $n = 10{,}000$ example.

The remarkable performance of `DHSEQRE` and `TTQRE` is due entirely to aggressive early deflation. In our experimental implementation of aggressive early deflation, if more than a few eigenvalues are isolated, then `TTQRE` and `DHSEQRE` skip the next QR sweep and immediately apply aggressive early deflation to the remaining unreduced Hessenberg submatrix. In this way, `TTQRE` and `DHSEQRE` complete the entire Schur decomposition of (1.1) without once performing a multishift QR sweep outside of a deflation window! This explains why `DHSEQRE` and `TTQRE` show nearly identical execution time and floating point instruction count. It also explains `TTQRE`'s uncharacteristically low floating point execution rate. Most of its level 3 BLAS operations lie in the unexecuted small-bulge multishift QR sweep.

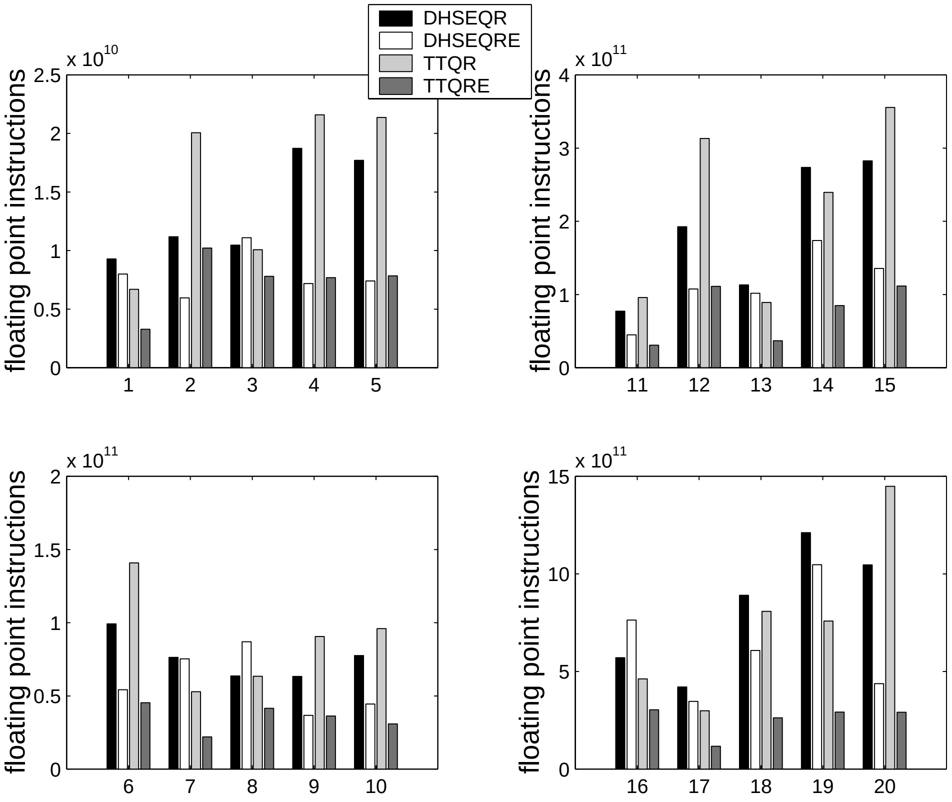

Fig. 5. *Hardware count of floating point instructions executed by* DHSEQR, DHSEQRE, TTQR, *and* TTQRE *computing the orthogonal and quasi-triangular factor of the Schur decompositions of* 20 *non-Hermitian eigenvalue problems selected from* [2]. *The numbers along the bottom edge of the bar graphs correspond to the numbers in the left-hand column of Table* 3. *Both* DHSEQR *and* DHSEQRE *use* 6 *simultaneous shifts.* DHSEQRE *uses a* 50*-by-*50*,* 100*-by-*100*, or* 150*-by-*150 *deflation window when the order n of the test matrix falls in the range* $1{,}000 \leq n < 2{,}000$, $2{,}000 \leq n < 4{,}000$, *and* $4{,}000 < n$, *respectively.* TTQR *uses* 60, 116, 150, *or* 180 *simultaneous shifts when the order n of the text matrix falls in the range* $1{,}000 \leq n < 2{,}000$, $2{,}000 \leq n < 2{,}500$, $2{,}500 \leq n < 4{,}000$, *and* $4{,}000 \leq n$, *respectively.* TTQRE *uses* 90, 120, 180, 240, *or* 270 *simultaneous shifts when the order n of the test matrix falls in the range* $1{,}000 \leq n < 2{,}000$, $2{,}000 \leq n < 2{,}500$, $2{,}500 \leq n < 4{,}500$, $4{,}500 \leq n < 7{,}000$, *and* $7{,}000 < n$, *respectively. As usual,* TTQRE *uses a deflation window of order* 1.5 *times the number of simultaneous shifts. (Floating point instructions executed during the reduction to Hessenberg form are not included.)*

It is easy to show that in examples like this, if aggressive early deflation eliminates the need for all or nearly all multishift QR sweeps, the amount of arithmetic work needed to compute the n-by-n Hessenberg Schur decomposition grows as $O(n^2)$.

Example 4. Figures 4 and 5 display the serial execution times and hardware count of executed floating point instructions of DHSEQR, DHSEQRE, TTQR, and TTQRE applied to the real non-Hermitian eigenvalue problems from the NEP collection [2] that are listed in Table 3. Each set of four bars in Figures 4 and 5 is labeled at the bottom by the number of the corresponding test matrix in Table 3. Summaries of the applications, descriptions of the matrices, and references can be found in [2].

To avoid cache conflicts, each n-by-n matrix is stored in an $(n + 7)$-by-$(n + 7)$ array. Both DHSEQR and DHSEQRE use 6 simultaneous shifts. DHSEQRE uses a 50-by-50,

Table 3

Matrices selected from the collection of non-Hermitian eigenvalue problems [2]. *The numbers along the bottom edge of the bar graphs in Figures* 4 *and* 5 *correspond to the numbers in the left-hand column.*

	Acronym	Order n	Discipline
1	OLM1000	1,000	Hydrodynamics
2	TUB1000	1,000	Computational fluid dynamics
3	TOLS1090	1,090	Aeroelasticity
4	RDB1250	1,250	Chemical engineering
5	RDB1250L	1,250	Chemical engineering
6	BWM2000	2,000	Chemical engineering
7	OLM2000	2,000	Hydrodynamics
8	TOLS2000	2,000	Aeroelasticity
9	DW2048	2,048	Electrical engineering
10	RDB2048	2,048	Chemical engineering
11	RDB2048L	2,048	Chemical engineering
12	PDE2961	2,961	Partial differential equations
13	MHD3200A	3,200	Plasma physics
14	MHD3200B	3,200	Plasma physics
15	RDB3200L	3,200	Chemical engineering
16	TOLS4000	4,000	Aeroelasticity
17	MHD4800A	4,800	Plasma physics
18	MHD4800B	4,800	Plasma physics
19	OLM5000	5,000	Hydrodynamics
20	RW5151	5,151	Probability
21	DW8192	8,192	Electrical engineering

100-by-100, or 150-by-150 deflation window when the order n of the test matrix falls in the range $1{,}000 \le n < 2{,}000$, $2{,}000 \le n < 4{,}000$, and $4{,}000 < n$, respectively. `TTQR` uses 60, 116, 150, or 180 simultaneous shifts when the order n of the text matrix falls in the range $1{,}000 \le n < 2{,}000$, $2{,}000 \le n < 2{,}500$, $2{,}500 \le n < 4{,}000$, and $4{,}000 \le n$, respectively. `TTQRE` uses 90, 120, 180, 240, or 270 simultaneous shifts when the order n of the test matrix falls in the range $1{,}000 \le n < 2{,}000$, $2{,}000 \le n < 2{,}500$, $2{,}500 \le n < 4{,}500$, $4{,}500 \le n < 7{,}000$, and $7{,}000 < n$, respectively. As usual, `TTQRE` uses a deflation window of order 1.5 times the number of simultaneous shifts.

In Figure 4 the median ratio of `DHSEQRE`'s execution time to `DHSEQR`'s execution time is .58. The median ratio of `TTQR`'s execution time to `DHSEQR`'s execution time is .56. The median ratio of `TTQRE`'s execution time to `DHSEQR`'s execution time is .23.

Matrix number 21 in Table 3, DW8192, is not reported in Figures 4 and 5 only because the execution times are out of scale with the other reported times. In the Origin2000 computational environment described above, `DHSEQR` calculates the Schur decomposition (including both quasi-triangular and orthogonal factors) of the Hessenberg matrix derived from DW8192 in 544 cpu minutes; `DHSEQRE` uses 300 minutes; `TTQR` uses 301 cpu minutes; and `TTQRE` uses 120 cpu minutes.

Example 5. Our experimental implementation of `TTQRE` is not well tuned for parallel computation. However, it does make heavy use of the level 3 BLAS (particularly matrix-matrix multiply), so it is not surprising to observe modest but not insignificant speedups when the experimental version of `TTQRE` is compiled for parallel execution and linked with parallel versions of the BLAS.

Figure 6 displays wall clock execution times and parallel speedups for `TTQRE` applied to the pseudorandom Hessenberg matrices described in Example 1. (Parallel speedup is the ratio T_1/T_p, where T_1 is the 1 processor wall clock execution time and T_p is the p-processor wall clock execution time.) We were fortunate to get exclusive use of

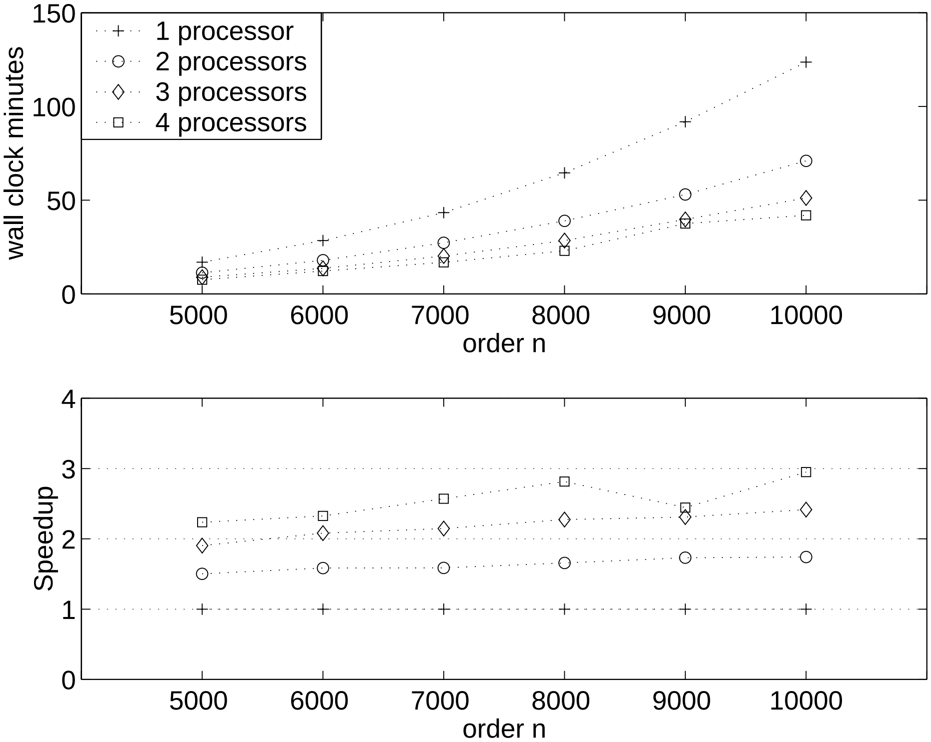

FIG. 6. *Wall clock execution time and parallel speedups of* `TTQRE` *computing both the orthogonal and quasi-triangular factor of the Schur decomposition of pseudorandom Hessenberg matrices. (The execution time of the reduction to Hessenberg form is not included.)*

several processors for the purpose of timing parallel benchmark runs. In this example, `TTQRE` uses 150 simultaneous shifts along with a 226-by-226 deflation window. To avoid cache conflicts, some n-by-n matrices were stored in $(n+1)$-by-$(n+1)$ arrays.

The parallel speedups in Figure 6 are modest. For a 5,000-by-5,000 pseudorandom Hessenberg matrix, `TTQRE` computes the quasi-triangular and orthogonal Schur factors in approximately 9% of the time needed by `DHSEQR`. With four processor parallelism execution time drops to only 4%. For a 10,000-by-10,000 pseudorandom Hessenberg matrix, `TTQRE` computes the quasi-triangular and orthogonal Schur factors in approximately 7% of the time needed by `DHSEQR`. With four processor parallelism execution time drops to only 2%.

Our experimental implementation of `TTQRE` has one or more serial bottle necks. The worst of these can be traced to the "near diagonal" portion of the small-bulge multishift QR algorithm [9, 10]. A tuned production version of `TTQRE` that is designed for parallel computation is in progress.

A similar numerical experiment using `TTQR` with a figure showing parallel execution times and speedups appears in [9].

4. Conclusion.

Aggressive early deflation recognizes converged eigenvalues before classical small-subdiagonal deflation would. In experiments with random Hessenberg matrices and with Hessenberg matrices from a variety of engineering and scientific applications [2], it significantly reduces both the number of floating point instructions and the execution time needed by both the large- and small-bulge multishift QR algorithm. Sometimes, aggressive early deflation reduces execution time from hours down to minutes. In experiments with n-by-n Hessenberg matrices of the form of (1.1), aggressive early deflation computes the Schur decomposition using only $O(n^2)$ flops.

Aggressive early deflation is both theoretically and empirically "normwise" backward numerically stable.

Although aggressive early deflation is effective in combination with conventional QR algorithms, the combination of aggressive early deflation with the two-tone, small-bulge multishift QR algorithm [9, 10] takes advantage of the capabilities of advanced architecture computers to sustain a high floating point instruction execution rate and attain at least modest parallel speedups.

Acknowledgment. The authors would like to thank David Watkins for helpful discussions and for bringing [42] to their attention.

REFERENCES

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK Users' Guide*, SIAM, Philadelphia, 1992.

[2] Z. Bai, D. Day, J. Demmel, and J. Dongarra, *A Test Matrix Collection for Non-Hermitian Eigenvalue Problems*, Tech. report, Department of Mathematics, University of Kentucky, Lexington, KY. Also available online from http://math.nist.gov/MatrixMarket.

[3] Z. Bai and J. Demmel, *On a block implementation of Hessenberg QR iteration*, Intl. J. of High Speed Comput., 1 (1989), pp. 97–112. Also available online as LAPACK Working Note 8 from http://www.netlib.org/lapack/lawns/lawn08.ps and http://www.netlib.org/lapack/lawnspdf/lawn08.pdf.

[4] Z. Bai and J. Demmel, *On swapping diagonal blocks in real Schur form*, Linear Algebra Appl., 186 (1993), pp. 73–95. Also available online as LAPACK Working Note 54 from http://www.netlib.org/lapack/lawns/lawn54.ps and http://www.netlib.org/lapack/lawnspdf/lawn54.pdf.

[5] T. Beelen and P. Van Dooren, *An improved algorithm for the computation of Kronecker's canonical form of a singular pencil*, Linear Algebra Appl., 105 (1988), pp. 9–65.

[6] D. Boley, *Computing rank-deficiency of rectangular matrix pencils*, Systems Control Lett., 9 (1987), pp. 207–214.

[7] D. Boley, *Estimating the sensitivity of the algebraic structure of pencils with simple eigenvalue estimates*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 632–643.

[8] D. Boley and W. Lu, *Measuring how far a controllable system is from an uncontrollable one*, IEEE Trans. Automat. Control, 31 (1986), pp. 249–251.

[9] K. Braman, R. Byers, and R. Mathias, *The multishift QR algorithm. Part* I: *Maintaining well-focused shifts and level* 3 *performance*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 929–947.

[10] K. Braman, R. Byers, and R. Mathias, *The Multi-Shift QR-Algorithm: Aggressive Deflation, Maintaining Well Focused Shifts, and Level* 3 *Performance*, Tech. Report 99-05-01, Department of Mathematics, University of Kansas, Lawrence, KS, 1999. Also available online from http://www.math.ukans.edu/~reports/1999.html.

[11] R. Byers, *Numerical condition of the algebraic Riccati equation*, Contemp. Math., 47 (1985), pp. 35–49.

[12] R. Byers, *Detecting nearly uncontrollable pairs*, in Signal Processing, Scattering and Operator Theory, and Numerical Methods, Proceedings of the International Symposium MTNS-89, Vol. 3, Amsterdam, M. A. Kaashoek, J. H. van Schuppen, and A. C. M. Ran, eds., Birkhäuser, Boston, 1990, pp. 447–457.

[13] J. Demmel, *A Lower Bound on the Distance to the Nearest Uncontrollable System*, Tech. report, Courant Institute, Computer Science Dept., New York University, New York, 1987.

[14] J. Demmel, *On condition numbers and the distance to the nearest ill-posed problem*, Numer. Math., 51 (1987), pp. 251–289.

[15] J. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[16] J. Demmel and B. Kågstrom, *Accurate solutions of ill-posed problems in control theory*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 126–145.

[17] J. Demmel and B. Kågstrom, *Stably computing the Kronecker structure and reducing subspaces of singular pencils $A - \lambda B$ for uncertain data*, in Large Scale Eigenvalue Problems, J. Cullum and R. A. Willoughby, eds., North-Holland, Amsterdam, 1986.

[18] J. Demmel and B. Kågstrom, *Stable eigendecompositions of matrix pencils*, Linear Algebra Appl., 88/89 (1987), pp. 139–186.

[19] J. Demmel and K. Veselić, *Jacobi's method is more accurate than QR*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1204–1245.

[20] A. A. Dubrulle and G. H. Golub, *A multishift QR iteration without computation of the shifts*, Numer. Algorithms, 7 (1994), pp. 173–181.

[21] R. Eising, *Between controllable and uncontrollable*, Systems Control Lett., 4 (1984), pp. 263–264.

[22] R. Eising, *The distance between a system and the set of uncontrollable systems*, in Proceedings of the Mathematical Theory of Networks and Systems, Beer-Sheva, P. A. Fuhrmann, ed., Springer-Verlag, London, 1984, pp. 303–314.

[23] L. Elsner and C. He, *An algorithm for computing the distance to uncontrollability*, Systems Control Lett., 17 (1992), pp. 453–464.

[24] J. G. F. Francis, *The QR transformation: A unitary analogue to the LR transformation.* I, Comput. J., 4 (1961/1962), pp. 265–271.

[25] J. G. F. Francis, *The QR transformation.* II, Comput. J., 4 (1961/1962), pp. 332–345.

[26] P. Gahinet and A. J. Laub, *Algebraic Riccati equations and the distance to the nearest uncontrollable pair*, SIAM J. Control Optim., 30 (1992), pp. 765–786.

[27] M. Gao and M. Neumann, *A global minimum search algorithm for estimating the distance to uncontrollability*, Linear Algebra Appl., 188/189 (1993), pp. 305–350.

[28] B. S. Garbow, J. M. Boyle, J. J. Dongarra, and C. B. Moler, *Matrix Eigensystem Routines: EISPACK Guide Extension*, Springer-Verlag, New York, 1972.

[29] G. Golub and C. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[30] M. L. J. Hautus, *Controllability and observability conditions of linear autonomous systems*, Nederl. Akad. Wetensch. Proc. Ser. A 72, 1969, pp. 443–448.

[31] B. Kågstrom, *RGSVD—an algorithm for computing the Kronecker structure and reducing subspaces of singular $A - \lambda B$ pencils*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 185–211.

[32] V. N. Kublanovskaya, *On some algorithms for the solution of the complete eigenvalue problem*, U.S.S.R. Comput. Math. and Math. Phys., 3 (1961), pp. 637–657.

[33] H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems*, Wiley-Interscience, New York, 1972.

[34] A. Laub, *Numerical linear algebra aspects of control design computations*, IEEE Trans. Automat. Control, 30 (1985), pp. 97–108.

[35] C. C. Paige, *Properties of numerical algorithms related to computing controllability*, IEEE Trans. Automat. Control, 26 (1981), pp. 130–139.

[36] B. T. Smith, J. M. Boyle, J. J. Dongarra, B. S. Garbow, Y. Ikebe, V. C. Klema, and C. B. Moler, *Matrix Eigensystem Routines—EISPACK Guide*, Lecture Notes in Comput. Sci. 6, Springer-Verlag, New York, 1976.

[37] B. T. Smith, J. M. Boyle, Y. Ikebe, V. C. Klema, and C. B. Moler, *Matrix Eigensystem Routines: EISPACK Guide*, 2nd ed., Springer-Verlag, New York, 1970.

[38] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, New York, 1973.

[39] G. W. Stewart, *Algorithm* 406 *HQR*3 *and EXCHNG: Fortran subroutines for calculating and ordering and eigenvalues of a real upper Hessenberg matrix*, ACM Trans. Math. Software, 2 (1976), pp. 275–280.

[40] R. A. van de Geijn, *Deferred shifting schemes for parallel QR methods*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 180–194.

[41] D. S. Watkins, *Fundamentals of matrix computations*, John Wiley, New York, 1991.

[42] D. S. Watkins, *Shifting strategies for the parallel QR algorithm*, SIAM J. Sci. Comput., 15 (1994), pp. 953–958.

[43] D. S. Watkins and L. Elsner, *Convergence of algorithms of decomposition type for the eigenvalue problem*, Linear Algebra Appl., 143 (1991), pp. 19–47.

[44] M. Wicks and R. DeCarlo, *Computing the Distance to an Uncontrollable System*, IEEE Trans. Automat. Control, 36 (1991), pp. 39–49.

[45] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, UK, 1965.

[46] J. H. Wilkinson, *Global convergence of tridiagonal QR algorithm with origin shifts*, Linear Algebra Appl., 1 (1968), pp. 409–420.