SIAM J. NUMER. ANAL.
Vol. 12, No. 6, December 1975

A GENERAL MATRIX EIGENVALUE ALGORITHM*

CHARLES F. VAN LOAN†

Abstract. A general QR-type process called the VZ algorithm is presented for the solution of the general matrix eigenvalue problem $ACx = \lambda BDx$. The matrices involved may be rectangular. For appropriate choices of A, B, C and D, we have some of the more familiar types of eigenproblems, and this is reflected in the fact that the QR, QZ and SVD algorithms are all special cases of the VZ algorithm. The main emphasis is upon the algorithm's generality as well as its bearing upon the generalized singular value problem $A^TAx = \mu^2 B^TBx$.

1. Introduction and notation. The QR algorithm [2] for the standard real eigenvalue problem

$$\text{(I)} \qquad Ax = \lambda x, \qquad A \in R^{n\times n}, \qquad 0 \neq x \in \mathbb{C}^n, \quad \lambda \in \mathbb{C}$$

has two important derivatives in the QZ algorithm [5], which solves the generalized eigenvalue problem

$$\text{(II)} \qquad Ax = \lambda Bx, \qquad A, B \in R^{n\times n}, \qquad 0 \neq x \in \mathbb{C}^n, \quad \lambda \in \mathbb{C},$$

and the SVD algorithm [3], which solves the singular value problem

$$A^TAx = \mu^2 x, \qquad A \in R^{m\times n}, \qquad m \geqq n,$$
$$0 \neq x \in R^n, \qquad \mu \geqq 0.$$

This paper is about another QR-type process called the VZ algorithm which was devised in conjunction with the generalized singular value problem

$$\text{(IV)} \qquad A^TAx = \mu^2 B^TBx, \qquad A, B \in R^{m\times n}, \quad m \geqq n,$$
$$0 \neq x \in R^n, \qquad \mu \geqq 0.$$

(A discussion of these problems can be found in [7].) This new routine solves a problem even more general than (IV), namely,

$$\text{(V)} \qquad ACx = \lambda BDx, \qquad A, B \in R^{n\times m}, \quad m \geqq n,$$
$$C, D \in R^{m\times n}, \qquad 0 \neq x \in \mathbb{C}^n, \quad \lambda \in \mathbb{C}.$$

The problems (I)–(IV) are clearly special cases of the problem (V), and hence VZ is capable of solving all of the various eigenproblems alluded to thus far. We shall in fact show that VZ is equivalent to QR, QZ and SVD when it solves the problems (I), (II) and (III), respectively. However, our purpose is not to advocate the blanket use of this general algorithm but rather to advance it as a useful device for unifying the theoretical and computational aspects of the entire QR family of algorithms.

We begin in § 2 by reviewing the numerical dangers of transforming the problems (II)–(V) into standard form (e.g., $ACx = \lambda BDx \Leftrightarrow Yx = \lambda x$, where

* Received by the editors October 4, 1974.

† Mathematics Department, University of Manchester, Manchester, England, M13 9PL. This work was supported by the Science Research Council under Grant B/RG/4071.

$Y=(BD)^{-1}(AC)$). This leads naturally to the decomposition theorems of § 3 which the various members of the QR family compute. Next, we examine the features which these algorithms have in common—an underlying equivalence relation, a finite step initial reduction and a Francis-type iteration. These topics are explored in § 4, § 5 and § 6. In § 7 we conclude with a numerical example and a comment upon the relevance of the VZ algorithm to the generalized singular value problem.

Our notation is as follows: if A is a matrix, then a_{ij} is its (i, j) entry, A^T is its transpose, A^* is its complex conjugate transpose, $\|A\|$ is its 2-norm, and $\lambda(A)$ is its set of eigenvalues (multiplicities included). For a pair of square matrices A and B, we let $\lambda(A, B)$ denote the zeros of $\det(A-\lambda B)$.

$A=\triangledown$ means A is upper triangular ($a_{ij}=0$, $i>j$); $A=\triangle$ means that $A^T=\triangledown$; $A=\diagdown$ means that A is diagonal ($a_{ij}=0$, $i\neq j$); $A=\diagdown\!\!\diagdown$ means that A is upper bidiagonal ($a_{ij}=0$, $i\neq j, j-1$); and $A=\boxtimes$ means that A is upper Hessenberg ($a_{ij}=0$, $i>j+1$). I_{mn} denotes the $A\in R^{m\times n}$ for which $a_{ij}=\delta_{ij}$. As a special case, $I_n=I_{nn}$.

To indicate computed quantities, we use the notation fl[exp] where exp is any algebraic expression. For a given computing machine, ε will denote the largest floating point number for which $\mathrm{fl}[1+\varepsilon]$ equals 1. If Q and A are stored matrices, $A:=QA$ means (a) form QA and (b) store the result in A.

2. The numerical dangers of putting general eigenproblems into standard form. We shall see that the computed eigenvalues λ of the problem (V) which the VZ algorithm returns satisfy

$$\det[(A+E_A)(C+E_C)-\lambda(B+E_B)(D+E_D)]=0, \tag{2.1}$$

where E_A, E_B, E_C and E_D are matrices whose norms have orders $\varepsilon\|A\|$, $\varepsilon\|B\|$, $\varepsilon\|C\|$ and $\varepsilon\|D\|$, respectively. Similar statements can be made about the computed results of the QR, QZ and SVD algorithms. The result (2.1) is made possible because the VZ algorithm has incorporated certain features of the SVD and QZ routines. First, there is no attempt to form the matrix products AC or BD. (This corresponds to the avoidance of A^TA in SVD.) Second, there is no attempt to work with $(BD)^{-1}(AC)$. (This corresponds to the avoidance of $B^{-1}A$ in QZ.)

To show how matrix products can cause undue inaccuracy, we consider the singular value problem (III) and borrow an example from [3]. Let A be given by

$$A=\begin{bmatrix}1 & 1\\ \sqrt{\varepsilon} & 0\\ 0 & \sqrt{\varepsilon}\end{bmatrix}.$$

Now, the singular values of any matrix A are the nonnegative square roots of the eigenvalues of A^TA. This immediately suggests the following method for our 3×2 example:

(a) Compute $Y=\mathrm{fl}[A^TA]$;
(b) Calculate $\lambda(Y)=\{\lambda_1, \lambda_2\}$ (say, with QR);
(c) Take $\sqrt{\lambda_1}$ and $\sqrt{\lambda_2}$ as our computed singular values.

Since $\mathrm{fl}[1+\varepsilon]=1$, we find that

$$Y=\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix},$$

giving, at best, $\lambda_1=2$ and $\lambda_2=0$. Thus, by the above technique, we are led to believe that A has singular values $\sqrt{2}$ and 0.

We now think of $\sqrt{2}$ and 0 as the exact singular values of some matrix $A+E$ where E is hopefully small. Using the Wielandt–Hoffman theorem for singular values, we shall show that this is not exactly the case. This theorem states that if A has singular values $\{\mu_i\}$ with $\mu_1 \geqq \mu_2 \geqq \cdots \geqq \mu_n$ and $A+E$ has singular values $\{\tilde{\mu}_i\}$ with $\tilde{\mu}_1 \geqq \cdots \geqq \tilde{\mu}_n$, then

$$\sum_1^n (\mu_i-\tilde{\mu}_i)^2 \leqq \|E\|_F^2 = \operatorname{trace}(E^TE) \leqq n\|E\|^2. \tag{2.2}$$

In our example, A has exact singular values $\mu_1=\sqrt{2+\varepsilon}$ and $\mu_2=\sqrt{\varepsilon}$, while $A+E$ has singular values $\tilde{\mu}_1=\sqrt{2}$ and $\tilde{\mu}_2=0$. Hence from (2.2), we obtain

$$2\|E\|^2 \geqq (\sqrt{2+\varepsilon}-\sqrt{2})^2+(\sqrt{\varepsilon}-0)^2 > \varepsilon,$$

which shows that $\|E\|$ has order $\sqrt{\varepsilon}\|A\|$. Thus $\|E\|$ is not small *relative to the machine precision* ε. This shows why the formation of the products AC and BD in the problem (V) can lead to displeasing and unnecessary errors.

A second numerical difficulty arises in connection with the $ACx=\lambda BDx$ problem if we attempt to form $(BD)^{-1}$ with the intention of solving the theoretically equivalent standard problem $(BD)^{-1}ACx=\lambda x$. Of course, we reach an impasse with this technique if BD is singular, but practical difficulties can arise long before singularity in BD sets in. To see why, consider the $Ax=\lambda Bx$ problem with

$$A=\begin{bmatrix}\sqrt{\varepsilon} & \sqrt{\varepsilon}\\ 1 & -1\end{bmatrix}, \qquad B=\begin{bmatrix}1 & 1\\ \sqrt{\varepsilon} & -\sqrt{\varepsilon}\end{bmatrix}.$$

A and B have identical singular values, the smaller of which is $\sqrt{2\varepsilon}$. Now suppose we have computed B^{-1} exactly,

$$B^{-1}=\frac{1}{2}\begin{bmatrix}1 & \varepsilon^{-1/2}\\ 1 & -\varepsilon^{-1/2}\end{bmatrix}$$

and then compute $B^{-1}A$. Since $\mathrm{fl}[\varepsilon^{-1/2}\pm\varepsilon^{1/2}]=\varepsilon^{-1/2}$, we find that

$$\mathrm{fl}[B^{-1}A]=\frac{1}{2}\begin{bmatrix}\varepsilon^{-1/2} & -\varepsilon^{-1/2}\\ -\varepsilon^{-1/2} & \varepsilon^{-1/2}\end{bmatrix},$$

which has eigenvalues 0 and $\varepsilon^{-1/2}$. Hence, for some perturbation matrices E_A and E_B, we have

$$0=\det[(A+E_A)-\lambda(B+E_B)], \qquad \lambda=0, \quad \varepsilon^{-1/2}.$$

By arguments similar to those given above, we can show that $\|E_A\|$ has order $\sqrt{\varepsilon}\|A\|$.

Returning to the $ACx = \lambda BDx$ problem, we can conclude from the preceding discussion that if we work with $(BD)^{-1}AC$, then our computed eigenvalues λ may not satisfy (2.1) for small perturbation matrices E_A, E_B, E_C and E_D.

3. A general unitary decomposition theorem. The results of the previous section place us in a difficult position in that we are seeking to solve (V) without forming AC, BD or $(BD)^{-1}$. The way out of this dilemma lies in the Decomposition 4, which we shall shortly prove and which the VZ algorithm computes. We motivate the discussion of this general result by reminding the reader that the QR, QZ and SVD algorithms, respectively, compute (in the limit) the following unitary decompositions.

DECOMPOSITION 1 (The Schur decomposition). *If $A \in R^{n\times n}$, then there exists an $n \times n$ unitary matrix Q such that $Q^*AQ = R$ is upper triangular. (The eigenvalues of A are given by the r_{ii}.)*

DECOMPOSITION 2 (The generalized Schur decomposition [6]). *If $A, B \in R^{n\times n}$, then there exist $n \times n$ unitary matrices Q and Z such that $QAZ = R$ and $QBZ = S$ are both upper triangular. (The generalized eigenvalues are then given by the ratios r_{ii}/s_{ii}, $s_{ii} \neq 0$.)*

DECOMPOSITION 3 (The singular value decomposition. See [1] for example). *If $A \in R^{m\times n}$, then there exist orthogonal matrices U and V of orders m and n, respectively, such that $U^TAV = D$ is a diagonal matrix with nonnegative entries. (The singular values of A are then given by the d_{ii}.)*

Notice (a) that Decomposition 2 indicates how problem (II) can be solved without trying to form $B^{-1}A$ and (b) how Decomposition 3 suggests how we can find the singular values of A without forming A^TA. The Decomposition 4 below has the Decompositions 2 and 3 as special cases and, not surprisingly, indicates how we can solve the $ACx = \lambda BDx$ problem without forming AC, BD or $(BD)^{-1}$. We shall need the following two lemmas.

LEMMA 1. *Any sequence of $n \times n$ unitary matrices $\{Q_K\}$ contains a converging subsequence $\{Q_{k_i}\}$ whose limit Q is itself unitary.*

Proof. The proof follows from the $\mathbb{C}^{n^2}$ version of the Bolzano–Weierstrass theorem. See [8, p. 105]. □

LEMMA 2. *Let $G \in \mathbb{C}^{n\times m}$ and $H \in \mathbb{C}^{m\times n}$ $(m \geqq n)$ have the property that* rank (G) = rank $(H) = n$. *If the product $GH = R$ is upper triangular, then there exists an $m \times m$ unitary matrix U such that GU and U^*H are both upper triangular.*

Proof. We can always find an $m \times n$ unitary matrix U such that U^*H is upper triangular [4]. Partition GU and U^*H as follows:

$$GU = [G_1 : G_2], \qquad U^*H = \begin{bmatrix} H_1 \\ 0 \end{bmatrix},$$

where G_1, $H_1 \in \mathbb{C}^{n\times n}$, $H_1 = \nabla$ and $G_2 \in \mathbb{C}^{n\times(m-n)}$. We must show that G_1 is upper triangular, but this is easy since

$$R = GH = (GU)(U^*H) = G_1H_1$$

implies

$$G_1 = RH_1^{-1} = \nabla\nabla^{-1} = \nabla.$$

(H_1^{-1} exists because $n = \text{rank}(H) = \text{rank}(U^*H) = \text{rank}(H_1)$.) □

We now come to the main result of this section. The decomposition which follows is computed (in the limit) by the *VZ* algorithm.

DECOMPOSITION 4. *Let A and B be in $R^{n\times m}$, and let C and D be in $R^{m\times n}$ with $m \geqq n$. There exist unitary matrices Q and U in $\mathbb{C}^{n\times n}$ and unitary matrices V and Z in $\mathbb{C}^{m\times m}$ such that*

$$\tilde{A} = QAZ = \triangledown, \qquad \tilde{B} = QBV = \triangledown,$$
$$\tilde{C} = Z^*CU = \triangledown, \qquad \tilde{D} = V^*DU = \triangledown.$$

Proof. First assume that A, B, C and D all have rank n and that the square $n\times n$ matrix BD is invertible. By Decomposition 1, we can find an $n\times n$ unitary matrix Q such that $Q(AC)(BD)^{-1}Q^* = \triangledown$. Using Lemma 2 (with $G = QAC$ and $H = (BD)^{-1}Q^*$), we can thus find an $n\times n$ unitary matrix U such that

$$Q(AC)U = \triangledown, \tag{3.1}$$

$$U^*(BD)^{-1}Q^* = \triangledown. \tag{3.2}$$

Inverting both sides of (3.2) gives

$$Q(BD)U = \triangledown. \tag{3.3}$$

Again we apply Lemma 2 to (3.1) and (3.3) to produce $m\times m$ unitary matrices Z and V such that, from (3.1), $QAZ = \triangledown$ and $Z^*CU = \triangledown$, and from (3.3) $QBV = \triangledown$ and $V^*DU = \triangledown$.

We have thus proved the decomposition in the case when our matrices obey some rather strict conditions on rank. We can nevertheless proceed without these restrictions. For general A, B, C and D let

$$D = Q_D\begin{bmatrix} R_D \\ 0 \end{bmatrix}, \qquad R_D = \triangledown \in \mathbb{C}^{n\times n},$$

be the Householder QR decomposition of D where R_D has nonnegative diagonal elements. Similarly, let

$$BQ_D = Q_B[R_B \vdots S_B], \qquad R_B = \triangledown \in \mathbb{C}^{n\times n}, \quad S_B \in \mathbb{C}^{n\times(m-n)}$$

be the QR decomposition of BQ_D. If

$$D_k = Q_D\begin{bmatrix} R_D + (1/k)I_n \\ 0 \end{bmatrix} \quad \text{and} \quad B_k = Q_B[R_B + (1/k)I_n \vdots S_B]Q_D^*,$$

then it can easily be shown that $\lim \|B_k - B\| = \lim \|D_k - D\| = 0$ and that, for every positive integer k, rank (B_K) = rank (D_K) = rank $(B_kD_k) = n$. We also have full rank matrices A_k and C_k which converge to A and C, respectively.

By the first part of the proof, we can find, for each k, unitary Q_k, U_k, V_k and Z_k such that $Q_kA_kZ_k$, $Q_kB_kV_k$, $Z_k^*C_kU_k$ and $V_k^*D_kU_k$ are all upper triangular. Using Lemma 2 repeatedly, we can find a subsequence $\{k_i\}$ of the positive integers such that $Q_{k_i} \to Q$, $U_{k_i} \to U$, $V_{k_i} \to V$ and $Z_{k_i} \to Z$. It is easy to verify that QAZ, QBV, Z^*CU and V^*DU are all upper triangular. □

Remark 1. Suppose that the diagonal elements of $\tilde{A}$, $\tilde{B}$, $\tilde{C}$ and $\tilde{D}$ are given by α_i, β_i, γ_i and δ_i, respectively. We then have $\prod_1^n (\alpha_i\gamma_i - \lambda\beta_i\delta_i) = \det(\tilde{A}\tilde{C}$

$-\lambda\tilde{B}\tilde{D}) = \det[(QAZ)(Z^*CU) - \lambda(QBV)(V^*DU)] = \det(QU)\det(AC - \lambda BD)$. Thus, the λ which make $AC - \lambda BD$ singular are given by the ratios

$$\frac{\alpha_i\gamma_i}{\beta_i\delta_i}, \qquad \beta_i\delta_i \neq 0,$$

which clearly demonstrates the relevance of this decomposition to the problem (V).

Remark 2. As an illustration of the generality of Decomposition 4, we use it to prove the singular value decomposition. Specifically, we can find unitary Q, U, V and Z such that

$$QA^*Z = \diagdown, \tag{3.4}$$

$$Z^*AU = \diagdown, \tag{3.5}$$

$$QI_{nm}V = \diagdown, \tag{3.6}$$

$$V^*I_{mn}U = \diagdown. \tag{3.7}$$

Equations (3.6) and (3.7) tell us that $QU = \diagdown$, which implies that $QU = D = \diagdown$ since a unitary triangular matrix must be diagonal. Substituting $Q = DU^*$ into (3.4), we find that

$$DU^*A^*Z = \diagdown \quad \text{or} \quad U^*A^*Z = \diagdown,$$

which implies

$$Z^*AU = \diagup. \tag{3.8}$$

By comparing (3.5) and (3.8), we can conclude that $Z^*AU = \diagdown$.

4. The underlying equivalence relation. Having established the importance of Decomposition 4 to the problem (V), we now turn our attention to its computation. Greater understanding is achieved if we introduce the idea of an "underlying equivalence relation". Behind every eigenvalue routine is an equivalence relation under which the sought-after quantities are preserved. The following table summarizes this for *QR*, *QZ* and *SVD*:

TABLE 1

Routine	Underlying Equivalence Relation	Relevance
QR	$A_2 \sim A_1$ iff $A_2 = Q^TA_1Q, \quad Q^TQ = I_n$	$\lambda(A_2) = \lambda(A_1)$
QZ	$(A_2, B_2) \sim (A_1, B_1)$ iff $A_2 = QA_1Z, \quad B_2 = QB_1Z, \quad Q^TQ = Z^TZ = I_n$	$\lambda(A_2, B_2) = \lambda(A_1, B_1)$
SVD	$A_2 \sim A_1$ iff $A_2 = U^TAV, U^TU = I_m, \quad V^TV = I_n$	$\lambda(A_2^TA_2) = \lambda(A_1^TA_1)$

(*These equivalence relations have obvious complex analogies.*)

As far as the VZ algorithm is concerned, it iterates upon 4-tuples of matrices and revolves around the following equivalence relation:

$$(A_2, B_2, C_2, D_2) \sim (A_1, B_1, C_1, D_1)$$

iff there exist orthogonal matrices Q, U, V and Z of appropriate orders such that

$$\begin{aligned} A_2 &= QA_1Z, & B_2 &= QB_1V, \\ C_2 &= Z^TC_1U, & D_2 &= V^TD_1U. \end{aligned} \tag{4.1}$$

It is easy to show that $\lambda(A_2C_2, B_2D_2) = \lambda(A_1C_1, B_1D_1)$. Thus, the sought-after eigenvalues are preserved whenever the four matrices are updated in the fashion indicated by (4.1).

5. The finite step initial reduction. It is typical in a QR process to begin by reducing the matrix (matrices) involved to some *equivalent* condensed form. We review this fact in the following table:

TABLE 2

Algorithm	Original Data	Equivalent Condensed Form
QR	A_0	$A_1 = Q^TA_0Q = \nwarrow$
QZ	A_0, B_0	$A_1 = QA_0Z = \nwarrow$ $B_1 = QB_0Z = \nwarrow$
SVD	A_0	$A_1 = U^TA_0V = \diagdown$

In each case the reduction to condensed form is accomplished in a finite number of operations.

The object of the initial reduction in VZ is to find orthogonal Q, U, V and Z such that

$$\begin{aligned} A_1 &= QA_0Z = \nwarrow, & B_1 &= QB_0V = \nwarrow, \\ C_1 &= Z^TC_0U = \nwarrow, & D_1 &= V^TD_0U = \nwarrow, \end{aligned} \tag{5.1}$$

where A_0, B_0, C_0 and D_0 are the four original matrices. For particular choices of these matrices, (5.1) reduces to the QR, QZ and SVD reductions summarized in Table 2. For example, if $C_0 = D_0 = I_n$, then $QA_0U = (QA_0Z)(Z^TI_nU) = \nwarrow\nwarrow = \nwarrow$ and, similarly, $QB_0U = (QB_0V)(V^TI_nU) = \nwarrow\nwarrow = \nwarrow$. This is precisely the condensed form of the QZ algorithm applied to A_0 and B_0.

We now adopt some conventions in order to describe efficiently how VZ computes the reduction (5.1). Let $H_r^d(k)$ denote the set of real $d \times d$ Householder matrices of the form $I_d + \rho uv^T$ where

(a) u and v are in R^d and are linearly dependent,

(b) $\rho = -2/u^Tv$,

(c) only components $k, k+1, \cdots, k+r-1$ of u are nonzero.

When $r = 2$ or 3, our Householder matrices will actually be of the modified type (see [5]), but we shall largely ignore this. It serves only to remark here that

premultiplication (postmultiplication) by a matrix in $H_r^d(k)$ affects only rows (columns) $k, k+1, \cdots, k+r-1$.

Another convention which we must understand is that the computed orthogonal matrices are denoted by Q, U, V and Z and are applied to the "current" arrays A, B, C and D in a fashion suggested by (4.1). For example, an orthogonal matrix applied on the right of the current B is called a "V", and its transpose V^T must be applied on the left of the current D. Similar comments can be made concerning the orthogonal matrices Q, U and Z, and together these comments make up the "rules of VZ" which must be obeyed if the eigenvalues of the original matrices are to be preserved.

We are now equipped to show how the reduction (5.1) is carried out. The process begins with three Householder triangularizations.

(i) Find an orthogonal Z so $Z^TC = \nabla$ and apply as follows:

$$C := Z^TC, \qquad A := AZ.$$

(ii) Find an orthogonal V so $V^TD = \nabla$ and apply as follows:

$$D := V^TD, \qquad B := BV.$$

(iii) Find an orthogonal Q so $QB = \nabla$ and apply as follows:

$$B := QB, \qquad A := QA.$$

The details of these triangularizations can be found in [4]. If, as an example, $m = 7$ and $n = 5$, then the updated matrices have the following form:

$$A = \begin{matrix} x & x & x & x & x & x & x \\ x & x & x & x & x & x & x \\ x & x & x & x & x & x & x \\ x & x & x & x & x & x & x \\ x & x & x & x & x & x & x \end{matrix}, \qquad B = \begin{matrix} x & x & x & x & x & x & x \\ 0 & x & x & x & x & x & x \\ 0 & 0 & x & x & x & x & x \\ 0 & 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & 0 & x & x & x \end{matrix},$$

$$C = \begin{matrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}, \qquad D = \begin{matrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}.$$

(Here "x" denotes an arbitrary nonzero element.)

Now the trick is to reduce A to upper Hessenberg while preserving the triangularity of the other three matrices. Our technique is patterned after the QZ initial reduction, and we present a few steps of it with our 7×5 example so as to convey the general idea.

Find $Q \in H_2^5(4)$ to zero a_{51}.

$$(5.2)\quad A := QA = \begin{matrix} x & x & x & x & x & x & x \\ x & x & x & x & x & x & x \\ x & x & x & x & x & x & x \\ x & x & x & x & x & x & x \\ 0 & x & x & x & x & x & x \end{matrix}, \qquad B := QB = \begin{matrix} x & x & x & x & x & x & x \\ 0 & x & x & x & x & x & x \\ 0 & 0 & x & x & x & x & x \\ 0 & 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & + & x & x & x \end{matrix}.$$

(Here "+" indicates the presence of an unwanted nonzero entry.) The C and D arrays are unchanged, and observe that in the process of zeroing a_{51}, a nonzero element was generated in the (5, 4) position of B.

Find $V \in H_2^7(4)$ to zero b_{54}.

$$(5.3)\quad B := BV = \begin{matrix} x & x & x & x & x & x & x \\ 0 & x & x & x & x & x & x \\ 0 & 0 & x & x & x & x & x \\ 0 & 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & 0 & x & x & x \end{matrix}, \qquad D := V^T D = \begin{matrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & + & x \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}.$$

The A and C arrays remain the same while in the process of zeroing b_{54}, a nonzero element was generated in the (5,4) position of D.

Find $U \in H_2^5(4)$ to zero d_{54}.

$$(5.4)\quad D := DU = \begin{matrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}, \qquad C := CU = \begin{matrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & + & x \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}.$$

The result of this step is that a nonzero element was generated in the (5,4) position of C. The A and B arrays are left unchanged.

Find $Z \in H_2^7 4)$ to zero c_{54}.

$$C := ZC = \begin{matrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x, \\ 0 & 0 & 0 & 0 & x \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \qquad A := AZ^T = \begin{matrix} x & x & x & x & x & x & x \\ x & x & x & x & x & x & x \\ x & x & x & x & x & x & x. \\ x & x & x & x & x & x & x \\ 0 & x & x & x & x & x & x \end{matrix}$$

With this application of Z, no unwanted nonzero elements were generated. This "zero chasing" technique, evident in (5.2)–(5.5), is repeated as a_{41}, a_{31}, a_{52}, a_{42} and a_{53} are zeroed in turn. The process terminates with A, B, C and D in the condensed form specified by (5.1). For general m and n, the VZ initial reduction goes as follows:

1. For $k = 1, 2, \cdots, n$, find $Z^T \in H_{m-k+1}^m(k)$ to zero $c_{k+1,k}, \cdots, c_{mk}$. Apply Z as follows: $C := Z^T C$, $A := AZ$.

2. For $k = 1, 2, \cdots, n$, find $V^T \in H_{m-k+1}^m(k)$ to zero $d_{k+1,k}, \cdots, d_{mk}$. Apply V as follows: $D := V^T D$, $B := BV$.

3. For $k = 1, 2, \cdots, n-1$, find $Q \in H_{n-k+1}^n(k)$ to zero $b_{k+1,k}, \cdots, b_{nk}$. Apply Q as follows: $A := QA$, $B := QB$.

4. For $l = 1, 2, \cdots, n-2$ and $k = n-1, n-2, \cdots, l+1$, do steps (i)–(iv) (assume $n > 2$):
   (i) Find $Q \in H_2^n(k)$ to zero $a_{k+1,l}$. Apply: $A := QA$, $B := QB$.
   (ii) Find $V \in H_2^m(k)$ to zero $b_{k+1,k}$. Apply: $B := BV$, $D := V^T D$.
   (iii) Find $U \in H_2^n(k)$ to zero $d_{k+1,k}$. Apply: $D := DU$, $C := CU$.
   (iv) Find $Z \in H_2^m(k)$ to zero $c_{k+1,k}$. Apply: $C := ZC$, $A := AZ^T$.

Assuming $m = n$, this reduction requires about $14\,n^3$ multiplies, $14\,n^3$ adds and $6\,n^2$ square roots. This is about $2\frac{1}{2}$ times the work involved in the QZ initial reduction and about 9 times the work involved in the comparable reduction of a single matrix as in QR.

We conclude this section with the remark that as far as subsequent computations are concerned, we are left with an equivalent *square* problem. To see this, partition our reduced matrices $\hat{A}$, $\hat{B}$, $\hat{C}$ and $\hat{D}$ as follows:

$$\hat{A} = [A_1 \vdots A_2], \quad \hat{B} = [B_1 \vdots B_2], \quad \hat{C} = \begin{bmatrix} C_1 \\ 0 \end{bmatrix}, \quad \hat{D} = \begin{bmatrix} D_1 \\ 0 \end{bmatrix},$$

where A_1, B_1, C_1 and D_1 are all in $R^{n \times n}$. Clearly, $\hat{A}\hat{C} = A_1 C_1$ and $\hat{B}\hat{D} = B_1 D_1$.

6. The VZ iteration. We now turn our attention to the iterative portion of the algorithm having reduced the four matrices to the condensed form of the previous section. Let A, B, C and D denote these reduced matrices. The iteration we are about to describe has a simple task—find orthogonal matrices Q, U, V and Z so that

(6.1) $$QAZ = \nabla, \quad QBV = \nabla, \quad Z^T C U = \nabla, \quad V^T D U = \nabla$$

and

(6.2) QAZ is somehow "more upper triangular" than A.

The objective (6.2) is purposely imprecise, but together with (6.1), it does inform us that the general idea is to drive the subdiagonal elements in A to zero while preserving the triangularity in the other three matrices.

Before outlining how this is accomplished, we must mention how the $AC-\lambda BD$ problem deflates. Prior to a VZ iteration, the subdiagonal of A and the diagonals of B, C and D are checked for small elements. If any of these elements are negligible, then the problem can be decoupled into two smaller subproblems. This is obvious if any of the subdiagonal elements in A are small. Less obvious is the fact that if any of the b_{ii}, c_{ii} or d_{ii} are small, then we can still effect a decoupling. This is accomplished by some straight forward "zero chasing", the details of which we shall omit.

In checking matrix entries for smallness, we have four separate criteria: $\varepsilon_A = \varepsilon\|A\|$, $\varepsilon_B = \varepsilon\|B\|$, $\varepsilon_C = \varepsilon\|C\|$ and $\varepsilon_D = \varepsilon\|D\|$. An element x_{ij} of a matrix X is considered negligible if $|x_{ij}| \leqq \varepsilon_X$. Thus, when looking for small elements, candidates are judged relative to the matrix from which they come.

We now describe the details of a VZ iteration, and henceforth assume that $|a_{i,i-1}| > \varepsilon_A$, $|b_{ii}| > \varepsilon_B$, $|c_{ii}| > \varepsilon_C$ and $|d_{ii}| > \varepsilon_D$. This will enable us to form the matrix $F = AC(BD)^{-1}$ in the theoretical discussion later. For clarity, we shall work a 6×6 example.

Let Q_1 be an arbitrary orthogonal matrix in $H_3^6(1)$. We shall show later how Q_1 is picked so that the goal (6.2) is realized. Applying Q_1 to A and B gives

$$A := Q_1A = \begin{matrix} x & x & x & x & x & x \\ x & x & x & x & x & x \\ + & x & x & x & x & x \\ 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & x & x \end{matrix}, \qquad B := Q_1B = \begin{matrix} x & x & x & x & x & x \\ + & x & x & x & x & x \\ + & + & x & x & x & x \\ 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & 0 & x \end{matrix}$$

while the arrays C and D remain unchanged. With an eye to restoring the system to condensed form, we

(a) find $V_1 \in H_3^6(1)$ to zero b_{31} and b_{32}; apply: $B := BV_1$, $D := V_1^TD$;

(b) find $V_1' \in H_2^6(1)$ to zero b_{21}; apply: $B := BV_1'$, $D := (V_1')^TD$.

The arrays A and C are left alone by these operations while B and D assume the following forms:

$$B = \begin{matrix} x & x & x & x & x & x \\ 0 & x & x & x & x & x \\ 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & 0 & x \end{matrix}, \qquad D = \begin{matrix} x & x & x & x & x & x \\ + & x & x & x & x & x \\ + & + & x & x & x & x \\ 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & 0 & x \end{matrix}.$$

Next, we

(a) find $U_1 \in H_3^6(3)$ to zero d_{31} and d_{32}; Apply: $D := DU_1$, $C := CU_1$;

(b) find $U_1' \in H_2^6(3)$ to zero d_{21}; Apply: $D := DU_1'$, $C := CU_1'$.

The arrays A and B are untouched, while C and D take the following shapes:

$$VD = \begin{matrix} x & x & x & x & x & x \\ 0 & x & x & x & x & x \\ 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & 0 & x \end{matrix}, \qquad C = \begin{matrix} x & x & x & x & x & x \\ + & x & x & x & x & x \\ + & + & x & x & x & x \\ 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & 0 & x \end{matrix}$$

Now we chase the unwanted nonzero elements in C to the array A:

(a) Find $Z_1 \in H_3^6(1)$ to zero c_{21} and c_{31}. Apply: $C := Z_1 C$, $A := AZ_1^T$.

(b) Find $Z_1' \in H_2^6(2)$ to zero c_{32}. Apply: $C := Z_1' C$, $A := (AZ_1')^T$.

C and A now have the form

$$C = \begin{matrix} x & x & x & x & x & x \\ 0 & x & x & x & x & x \\ 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & 0 & x \end{matrix}, \qquad A = \begin{matrix} x & x & x & x & x & x \\ x & x & x & x & x & x \\ + & x & x & x & x & x \\ + & + & x & x & x & x \\ 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & x & x \end{matrix},$$

while the arrays B and D are unchanged. As a final step in the cycle, we find a $Q_2 \in H_3^6(2)$ to zero a_{31} and a_{41}. Applying Q_2 to A and B gives

$$A = \begin{matrix} x & x & x & x & x & x \\ x & x & x & x & x & x \\ 0 & x & x & x & x & x \\ 0 & + & x & x & x & x \\ 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & x & x \end{matrix}, \qquad B = \begin{matrix} x & x & x & x & x & x \\ 0 & x & x & x & x & x \\ 0 & + & x & x & x & x \\ 0 & + & + & x & x & x \\ 0 & 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & 0 & x \end{matrix}$$

This fully illustrates one cycle within a VZ iteration. Just as the first cycle (depicted above) involved rows and columns 1, 2 and 3, so will the next cycle involve rows and columns 2, 3 and 4. In this way, the nonzero "+" quantities are chased from A to B to D to C to A, etc. The process terminates with the four matrices restored to condensed form, and we summarize this for general n:

1. For $k = 1, 2, \cdots, n-2$ do steps (a)–(g):

(a) if $k > 1$, find $Q_k \in H_3^n(k)$ to zero $a_{k+1,k-1}$ and $a_{k+2,k-1}$.

If $k = 1$, find $Q_1 \in H_n^3(k)$ in accordance with shift calculations (see below). In either case, apply: $A := Q_k A$, $B := Q_k B$.

(b) Find $V_k \in H_3^n(k)$ to zero $b_{k+2,k}$ and $b_{k+2,k+1}$. Apply: $B := BV_k$, $D := V_k^T D$.

(c) Find $V_k' \in H_2^n(k)$ to zero $b_{k+1,k}$. Apply: $B := BV_k'$, $D := (V_k')^T D$.

(d) Find $U_k \in H_3^n(k)$ to zero $d_{k+2,k}$ and $d_{k+2,k+1}$. Apply: $D := DU_k$, $C := CU_k$.

(e) Find $U_k' \in H_2^n(k)$ to zero $d_{k+1,k}$. Apply: $D := DU_k'$, $C := CU_k'$.

(f) Find $Z_k \in H_3^n(k)$ to zero $c_{k+1,k}$ and $c_{k+2,k}$. Apply: $C := Z_k C$, $A := AZ_k^T$.

(g) Find $Z_k' \in H_2^n(k+1)$ to zero $c_{k+2,k+1}$. Apply: $C := Z_k' C$, $A := A(Z_k')^T$.

(We are now only four plane rotations away from condensed form.)

2. Do steps (h)–(k):

(h) Find $Q_{n-1}' \in H_2^n(n-1)$ to zero $a_{n,n-2}$. Apply: $A := Q_{n-1}' A$, $B := Q_{n-1}' B$.

(i) Find $V_{n-1}' \in H_2^n(n-1)$ to zero $b_{n,n-1}$. Apply: $B := BV_{n-1}'$, $D := (V_{n-1}')^T$.

(j) Find $U_{n-1}' \in H_2^n(n-1)$ to zero $d_{n,n-1}$. Apply: $D := DU_{n-1}'$, $C := CU_{n-1}'$.

(k) Find $Z_{n-1}' \in H_2^n(n-1)$ to zero $c_{n,n-1}$. Apply: $C := Z_{n-1}' C$, $A := A(Z_{n-1}')^T$.

It can be verified that the A, B, C and D which emerge from this process are in condensed form (5.1). The volume of work entailed in a VZ iteration is approximately $29n^2$ multiplies, $29n^2$ adds and $2n$ square roots. This is roughly $2\frac{1}{4}$ times the work involved in a QZ iteration and about 6 times the work present in a QR (double implicit shift) iteration.

It remains to show how Q_1 can be selected such that the goal (6.2) is realized. To this end, let k_1 and k_2 be the roots of the quadratic

$$\det\left[\begin{pmatrix} f_{rr} & f_{rn} \\ f_{nr} & f_{nn} \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right] = 0,$$

where $r = n-1$ and $F = (f_{ij})$ is the upper Hessenberg matrix $AC(BD)^{-1}$. Let $v \in R^n$ be the first column of the *real* matrix $(F - k_1 I_n)(F - k_2 I_n)$. It can be shown that v has the form

$$v^T = (x, y, z, 0, \cdots, 0),$$

and that, most importantly, we need only selected portions of $(BD)^{-1}$ to compute this "zeroth column". We now choose $Q_1 \in H_3^n(1)$ such that $Q_1 v$ is a scalar multiple of e_1, the first column of I_n. Define the orthogonal matrix Q by $Q = Q_{n-1}' Q_{n-2}, \cdots, Q_2 Q_1$. We can conclude from the construction of the Q's in (a) and (h) above that

$$e_1^T Q = e_1^T Q_1, \text{ (the first rows of } Q \text{ and } Q_1 \text{ are equal).} \tag{6.3}$$

Moreover, if we let U, V and Z to be the accumulation of the other three orthogonal transformations, then

$$\begin{aligned} QFQ^T &= QAC(BD)^{-1}Q^T \\ &= (QAZ)(Z^T CU)[(QBV)(V^T DU)]^{-1} \\ &= \blacksquare\blacksquare[\blacksquare\blacksquare]^{-1} = \blacksquare. \end{aligned} \tag{6.4}$$

Since k_1 and k_2 are the eigenvalues of the lower 2×2 portion of F, we see from (6.3), (6.4) and the Francis theorem [8, p. 352] that Q is precisely the matrix one would obtain by doing a double implicit QR iteration on the single matrix F. The implication of this is straightforward. Let A_k, B_k, C_k and D_k denote the matrices A, B, C and D after the kth VZ iteration. If we define F_k by $F_k = A_kC_k(B_kD_k)^{-1}$, then the above tells us that VZ acting upon the A_k, B_k, C_k and D_k is tantamount to QR acting upon the F_k. By what we know about the QR algorithm's convergence (see [8]), we see that the matrices F_k are tending to quasi-triangular form. (A quasi-triangular matrix is an upper Hessenberg matrix with no two adjacent subdiagonal elements nonzero.) But the constant, full rank triangularity of $C_k(B_kD_k)^{-1}$ leads us to conclude from the definition of F_k that the A_k are themselves tending to quasi-triangular form. We thus find that after enough iterations the original problem deflates into a collection of 1×1 and 2×2 subproblems by virtue of our setting small elements to zero as mentioned earlier in this section. We normally find that A is reduced to quasi-triangular form in about $1.3n$ iterations. This is consistent with the rate of convergence in the double implicit shift QR algorithm.

A word is in order about the errors incurred in the algorithm thus far. Let $\tilde{A}$, $\tilde{B}$, $\tilde{C}$ and $\tilde{D}$ denote the original matrices A, B, C and D after they have been transformed into quasi-triangular/triangular form. Because of the sole reliance upon orthogonal transformations, we know from Wilkinson [8] that $\tilde{A} = \tilde{Q}(A+E_A)\tilde{Z}$, $\tilde{B} = \tilde{Q}(B+E_B)\tilde{V}$, $\tilde{C} = \tilde{Z}^T(C+E_C)\tilde{U}$, and $\tilde{D} = \tilde{V}^T(D+E_D)\tilde{U}$ where Q, U, V and Z are exactly orthogonal and E_A, E_B, E_C and E_D are matrices whose norms are of order $\varepsilon\|A\|$, $\varepsilon\|B\|$, $\varepsilon\|C\|$ and $\varepsilon\|D\|$, respectively. In solving the 2×2 subproblems, great care must be exercised in order to preserve this kind of desirable perturbation result.

Finally, we remark that the generalized eigenvectors of the reduced problem can be obtained by a back substitution process. By applying the accumulated U's we can then get the eigenvectors to the initial problem.

7. An application of the *VZ* algorithm to the generalized singular value problem. We tested the VZ algorithm on the generalized singular value problem $A^TAx = \mu^2B^TBx$, where

$$A = \begin{bmatrix} 1 & 2 & 1 & 2 & -3 & -3 \\ 7 & 1 & -1 & 1 & -4 & -4 \\ -4 & 4 & 2 & 5 & -4 & -3 \\ -6 & -5 & -4 & 8 & 3 & 4 \\ 1 & -2 & -6 & 2 & -2 & 7 \\ -9 & 6 & 5 & -4 & -3 & 5 \\ 6 & -9 & 1 & -1 & 2 & 1 \\ 5 & 8 & -4 & 3 & -6 & -6 \\ -2 & 1 & 3 & -2 & 3 & -3 \\ -9 & -7 & 2 & 5 & 1 & 8 \end{bmatrix},$$

$$B = \begin{bmatrix} 4 & 1 & -3 & -6 & 5 & -1 \\ 3 & -4 & -6 & 1 & 4 & 2 \\ -5 & 7 & 8 & 3 & -9 & -4 \\ 6 & -6 & 4 & -5 & 2 & -1 \\ 1 & 3 & -9 & -7 & 5 & 7 \\ 2 & -4 & 4 & 2 & -6 & 2 \\ -7 & -5 & 1 & 4 & 2 & 5 \\ -8 & 2 & -7 & 5 & 7 & 1 \\ -9 & -5 & -3 & 1 & 8 & 8 \\ 6 & 4 & 5 & -6 & -1 & -8 \end{bmatrix}$$

Notice that the transpose of the vector (1, 1, 1, 1, 1, 1) is annihilated by both matrices. Thus, $A^TA - \mu^2 B^TB$ is singular for all scalars μ. To see how the VZ algorithm detected this, we tabulate the quantities α_i, β_i, γ_i and δ_i which are, respectively, the diagonal elements of the upper triangular QA^TZ, Z^TAU, QB^TV, and V^TBU.

	α_i					γ_i			
18.	4408	9224	2101	4200	13.	8919	7585	5565	7000
14.	2621	7822	4083	0700	10.	9609	5294	3934	9100
15.	4007	5100	8590	0000	8.	9356	6755	8123	6070
6.	9798	0352	5926	1540	7.	0220	2359	9356	9530
14.	0941	9284	0413	0300	17.	0261	3202	3272	9700
0.	0000	0000	0000	0009	0.	0000	0000	0000	0012
	β_i					δ_i			
9.	4162	8385	9280	6950	7.	0935	1678	2628	8390
18.	8916	4327	8751	4100	14.	5188	4906	7692	8600
14.	0796	5698	4272	1200	8.	1691	5361	0971	4360
22.	3549	3610	3468	4500	22.	4901	5878	4526	1100
11.	3857	9924	2506	4700	13.	7543	2586	2278	0800
0.	0000	0000	0000	0007	0.	0000	0000	0000	0015

Our computed generalized singular values are then given by

$$\mu_i = \left(\frac{\alpha_i \gamma_i}{\beta_i \delta_i}\right)^{1/2},$$

1.9584	0445	3145	9270
0.7549	4640	7448	0300
1.0938	3027	7119	8620
0.3122	2650	1672	7363
1.2378	7470	1654	2610
1.0638	8743	0396	0920

(These results were obtained using an IBM 360/67 with REAL*8 arithmetic.) Observe that μ_6 is determined as the quotient of rounding errors, but that this is not at all obvious if we examine the set of μ_i alone. Hence, we see the importance of the "QZ philosophy" which in our case advises that we return the α_i, β_i, γ_i and δ_i to the user so that *he* can compute and interpret the quotients $\alpha_i\gamma_i/\beta_i\delta_i$.

As we remarked, the matrices A and B above have the property that $A^TA - \mu^2B^TB$ is singular for all scalars μ. Hence, we expect that regardless of μ,

$$\begin{aligned}\prod_1^n (\alpha_i\gamma_i - \mu^2\beta_i\delta_i) &= \det[(QA^TZ)(Z^TAU) - \mu^2(QB^TV)(V^TBU)] \\ &= \det(QU)\det[A^TA - \mu^2B^TB] \\ &= 0,\end{aligned}$$

which implies $\alpha_k\gamma_k = \beta_k\delta_k = 0$ for some k. Modulo rounding errors, we find this to be the case of $k = 6$.

While the VZ algorithm can successfully solve the generalized singular value problem, even in pathological situations as above, it does have the drawback of being very inefficient. This is because it takes no advantage of the symmetry inherent in the generalized singular value problem. Other algorithms exist which do take advantage of symmetry but, as usual, at some expense [7]. Some of these routines break down when B is rank deficient, while others experience difficulty only when A and B have nontrivially intersecting nullspaces. Despite this, their economy over VZ makes their use more attractive than that of the general algorithm presented in this paper.

Acknowledgments. This work was part of the author's Ph.D. thesis written at the University of Michigan under Prof. Cleve B. Moler. The author wishes to thank Professor Moler for his assistance.

REFERENCES

[1] G. FORSYTHE AND C. B. MOLER, *Computer Solution of Linear Algebraic Systems*, Prentice-Hall, Englewood Cliffs, N.J., 1967.

[2] J. G. F. FRANCIS, *The QR transformation—A unitary analogue to the LR transformation*, Comput. J., 4 (1961–62), pp. 265–271, 332–345.

[3] G. GOLUB AND C. REINSCH, *Singular value decomposition and least squares solution*, Numer. Math., 14 (1970), pp. 403–420.

[4] A. S. HOUSEHOLDER, *Unitary triangularization of a nonsymettric matrix*, J. Assoc. Comput. Mach., 5 (1958), pp. 339–42.

[5] C. MOLER AND G. W. STEWART, *An algorithm for generalized matrix eigenvalue problems*, this Journal, 10 (1973), pp. 241–256.

[6] G. W. STEWART, *On the sensitivity of the eigenvalue problem $Ax = \lambda Bx$*, this Journal, 9 (1972), pp. 669–686.

[7] C. VAN LOAN, *Generalized singular value problems*, submitted.

[8] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Oxford University Press, Oxford, 1965.