NASA TECHNICAL NOTE

NASA TN D-7305

NASA TN D-7305

AN EXTENSION OF THE QZ ALGORITHM FOR SOLVING THE GENERALIZED MATRIX EIGENVALUE PROBLEM

by Robert C. Ward

Langley Research Center
Hampton, Va. 23665

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • JULY 1973

1. Report No.
NASA TN D-7305

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
AN EXTENSION OF THE QZ ALGORITHM FOR SOLVING THE GENERALIZED MATRIX EIGENVALUE PROBLEM

5. Report Date
July 1973

6. Performing Organization Code

7. Author(s)
Robert C. Ward

8. Performing Organization Report No.
L-8899

9. Performing Organization Name and Address
NASA Langley Research Center
Hampton, Va. 23665

10. Work Unit No.
501-06-01-11

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, D.C. 20546

13. Type of Report and Period Covered
Technical Note

14. Sponsoring Agency Code

15. Supplementary Notes
The basic information presented herein is a part of a thesis which will be offered in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Applied Mathematics, University of Virginia, Charlottesville, Virginia.

16. Abstract

An algorithm called the combination shift QZ algorithm is presented for solving the generalized matrix eigenvalue problem. This new algorithm is an extension of Moler and Stewart's QZ algorithm with some added features for saving time and operations. Also, some additional properties of the QR algorithm which were not practical to implement in the QZ algorithm can be generalized with the combination shift QZ algorithm. Numerous test cases are presented to give practical application tests for the algorithm. Based on the results presented in this paper, this algorithm should be preferred over existing algorithms which attempt to solve the class of generalized eigenproblems where both matrices are singular or nearly singular.

17. Key Words (Suggested by Author(s))
Generalized eigenvalue problem
Numerical analysis
Eigenvalues
Algorithms
QZ algorithms

18. Distribution Statement
Unclassified – Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	54	$3.00

*For sale by the National Technical Information Service, Springfield, Virginia 22151

CONTENTS

AN EXTENSION OF THE QZ ALGORITHM FOR SOLVING THE GENERALIZED MATRIX EIGENVALUE PROBLEM*

By Robert C. Ward
Langley Research Center

SUMMARY

An algorithm called the combination shift QZ algorithm is presented for solving the generalized matrix eigenvalue problem. This new algorithm is an extension of Moler and Stewart's QZ algorithm with some added features for saving time and operations. Also, some additional properties of the QR algorithm which were not practical to implement in the QZ algorithm can be generalized with the combination shift QZ algorithm. Numerous test cases are presented to give practical application tests for the algorithm. Based on the results presented in this paper, this algorithm should be preferred over existing algorithms which attempt to solve the class of generalized eigenproblems where both matrices are singular or nearly singular.

INTRODUCTION

There are numerous problems which occur frequently in the physical sciences that require solving the generalized eigenvalue problem

$$Ax = \lambda Bx \tag{1}$$

for λ and x where A and B are $n \times n$ real matrices, λ is a scalar, and x is a $n \times 1$ vector. To mention one example, it is well known (see Lancaster (ref. 1)) that the equations of motion for many mechanical and electrical systems may be written in the matrix form

$$A\ddot{p} + B\dot{p} + Cp = f \tag{2}$$

where A, B, and C are $n \times n$ real matrices, and p and f are time-dependent $n \times 1$ vectors. If a system with no damping $(B = 0)$, no forcing function $(f = 0)$, and solutions of the form $p(t) = e^{\lambda t}x$ (sinusoidal solutions) where x is independent of time is considered, then in terms of λ and x, equation (2) becomes

*The basic information presented herein is a part of a thesis which will be offered in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Applied Mathematics, University of Virginia, Charlottesville, Virginia.

$$Cx = -\lambda^2 Ax \tag{3}$$

which is the generalized eigenvalue problem in $-\lambda^2$. If the system is damped, the resulting equation can be transformed by the equations

$$\lambda x = y \tag{4}$$

$$\lambda Ay + By + Cx = 0 \tag{5}$$

into the block $2n \times 2n$ matrix generalized eigenvalue problem

$$\begin{bmatrix} 0 & I \\ -C & -B \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} I & 0 \\ 0 & A \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{6}$$

When solving the generalized eigenvalue problem, the roles of A and B can be reversed by solving for the reciprocals of the eigenvalues. That is, one could solve the problem

$$Bx = \mu Ax \tag{7}$$

where the eigenvalues λ of the original problem (eq. (1)) are given by

$$\lambda = \frac{1}{\mu} \tag{8}$$

An infinite eigenvalue of equation (1) is defined as a corresponding zero eigenvalue of equation (7).

In many application cases, A and B of equation (1) have some special properties which determine the type of eigenvalues present and influence the selection of an algorithm for solving the problem. First, consider the cases when B is well-conditioned with respect to inversion. When A is symmetric and B is positive definite as quite often occurs in physical applications, all the eigenvalues are real; Martin and Wilkinson (ref. 2) describe an algorithm which reduces this problem to the standard symmetric eigenvalue problem $Pz = \lambda z$. In addition, if A and B have band structure, Crawford (ref. 3) describes a modification to Martin and Wilkinson's algorithm to take advantage of the band matrices. Also, an algorithm by Golub, Underwood, and Wilkinson (ref. 4) using Lanczos method solves the band problem. If A and B do not have special properties, the generalized eigenvalue problem can be solved by solving the standard problem $B^{-1}Ax = \lambda x$. Normally, one would not want to form $B^{-1}A$ when A and B have special properties usable by algorithms. For example, in the preceding case where A was symmetric and B was positive definite, the problem was transformed into a standard symmetric eigenvalue problem which is faster and numerically more stable than the nonsymmetric problem $B^{-1}Ax = \lambda x$.

Now, consider the more complicated cases when B is ill-conditioned. When A is symmetric and B is positive semidefinite or positive definite but ill-conditioned, Fix and Heiberger (ref. 5) describe an algorithm for solving this problem which depends on determining the rank of several submatrices. This case is interesting in that the spectrum consists of both stable and unstable real eigenvalues and there exists the possibility of every scalar being an eigenvalue. Unstable eigenvalues are those which are sensitive to small changes in the matrix elements of A and B and thus cannot be computed accurately by a normal computational procedure. In general, rank determination is a difficult problem to solve on a computer; thus, Fix and Heiberger's algorithm runs into difficulty when there is not a clear separation between the stable and unstable eigenvalues. For a general ill-conditioned B matrix, Peters and Wilkinson (ref. 6) describe an algorithm which again depends on rank determination. Moler and Stewart (ref. 7) describe an algorithm which solves the generalized eigenvalue problem for arbitrary real matrices A and B by use of unitary transformations. This generalization of the double shift QR (ref. 8) is called the QZ algorithm and is particularly effective for the cases when B is singular or nearly singular. The algorithm presented in this paper is a combination of the QZ and a generalization of the single shift implicit QR and is referred to as the combination shift QZ algorithm. This algorithm is also effective on the singular or nearly singular B cases and is particularly effective on cases where a large number of real eigenvalues are expected, such as A symmetric and B positive semidefinite.

SYMBOLS

A,B,C	n by n matrices
A',B',C'	n by n matrices, next iterate of A, B, and C, respectively
$\overline{A},\overline{B}$	lower right 2 by 2 submatrices of A and B, respectively
a_{ij},b_{ij}	i,j elements in matrix A and B, respectively
$a_{10},a_{20},a_{30},$ S^2,u,r	scalars
$\tilde{C}$	n by n matrix related to C
$D_i,\overline{D}_i$	block diagonal matrices

p,x	n by 1 vectors
I	identity matrix
k,l,m,q	index arguments for derivation of test cases
n	size of matrices A and B
Q,Z	n by n orthogonal matrices
Q',Z',Z''	n by n orthogonal matrices
Q_i,Z_i	ith n by n orthogonal matrix in a sequence
x_i,y_i	n by 1 vectors associated with eigenvalue λ_i
$\alpha_i,\beta_i,\lambda,\lambda_i$	scalars
ϵ_0	basic machine roundoff error
$\left.\begin{matrix}\epsilon_1,\epsilon_2,\eta_1,\\ \eta_2,\tilde{\eta}_1,\tilde{\eta}_2\end{matrix}\right\}$	scalars, usually small numbers
σ	shift in single shift implicit QZ iteration
σ_1,σ_2	shifts in double shift QZ iteration
$\|[\ ]\|$	the norm of $[\ ]$
$[\ ]^H$	Hermitian of $[\ ]$
$[\ ]^T$	transpose of $[\ ]$

QZ ALGORITHM

Since a detailed description of the QZ algorithm is given in reference 7, only a brief summary will be given here for completeness. The algorithm is an iterative method for computing the decomposition guaranteed in the following theorem from reference 9:

Theorem: There are unitary matrices Q and Z so that QAZ and QBZ are both upper triangular.

If the decomposition can be accomplished, then the eigenvalues and eigenvectors are easily extracted by

$$\lambda_i = \frac{\alpha_i}{\beta_i} \tag{9}$$

$$x_i = Zy_i \tag{10}$$

where α_i and β_i are the diagonal elements of the QAZ and QBZ matrices, respectively, and y_i are the eigenvectors of the triangular system $QAZy_i = \lambda_i QBZy_i$.

Major Steps of QZ Algorithm

There are four major steps in the algorithm.

(1) Reduce A to upper Hessenberg form and at the same time reduce B to upper triangular form.

(2) Use a generalization of the double shift QR to put A in quasi-triangular form (upper block triangular form with 1 × 1 or 2 × 2 diagonal blocks) while keeping B in upper triangular form.

(3) Reduce A to upper triangular form and keep B in upper triangular form.

(4) Find the eigenvectors of the triangular system and back transform them to the original problem.

Step 1 is initiated by transforming B into upper triangular form by premultiplying by a unitary matrix, denoted Q', made up of a sequence of Householder reflections. Then Q'A is put in upper Hessenberg form by annihilating one element at a time in the order given below on a 5 × 5 example:

$$\begin{array}{c} Q'A \\ \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x^3 & x & x & x & x \\ x^2 & x^5 & x & x & x \\ x^1 & x^4 & x^6 & x & x \end{bmatrix} \end{array}$$

After each annihilation, for example by Q_i, the current B matrix is put back in triangular form by a postmultiplication with a Householder reflection, for example Z_i. This reflection does not affect the zeros introduced in the current A matrix. Thus, in the 5 × 5 matrix example, QAZ and QBZ would be upper Hessenberg and upper triangular, respectively, where $Q = Q_6Q_5Q_4Q_3Q_2Q_1Q'$ and $Z = Z_1Z_2Z_3Z_4Z_5Z_6$.

Step 2 is the generalization of the double shift QR algorithm applied to the standard eigenvalue problem $AB^{-1}x = \lambda x$ without forming B^{-1}. Since this step is the iterative step that the combination shift QZ algorithm alters, a more detailed description of this iteration is given later. For a complete description, the interested reader should consult reference 7.

Step 3 involves one QZ transformation, that is, one premultiplication by a unitary matrix Q and one postmultiplication by a unitary matrix Z on the quasi-triangular A and triangular B to put both matrices in triangular form. If a 2 × 2 diagonal block of A corresponds to a complex conjugate eigenvalue pair, complex arithmetic will be required in this step.

Step 4 is accomplished by solving the reduced triangular problem for its eigenvectors by a back-substitution process similar to the one used by Peters and Wilkinson (ref. 10) in the procedure "hqr2." The Z transformations (postmultiplication matrices) are accumulated and applied to the eigenvectors of the reduced system to obtain the eigenvectors of the original system. Recently, Kaufman (ref. 11) has pointed out that it is advantageous to solve the transposed problem for the left eigenvectors and thus accumulate the Q transformations for back transforming the triangular system vectors. This advantage can be seen in the discussion of the second step in the QZ algorithm.

Step 2 of QZ Algorithm

The iteration is motivated by assuming that B is nonsingular and by examining the double shift QR algorithm for $C = AB^{-1}$. Recall that A is upper Hessenberg and B is upper triangular as a result of step 1; thus, C is also upper Hessenberg.

Suppose one iteration of the double shift QR with shifts σ_1 and σ_2 is applied to C. Then a unitary matrix Q is found that makes the matrix QCQ^H upper Hessenberg, where Q^H denotes the Hermitian of Q, and the first row of Q is the first row of the orthogonal matrix which annihilates all the elements but the first in the first column of $\left(C - \sigma_1 I\right)\left(C - \sigma_2 I\right)$. The next iterate C' is then defined as

$$C' = QCQ^H \quad , \tag{11}$$

Consider what happens if a special form of the identity, that is, ZZ^H where Z is unitary is inserted in the matrix equation (1) to give

$$AZ\left(Z^H x\right) = \lambda BZ\left(Z^H x\right) \tag{12}$$

Then

$$\tilde{C} = AZ(BZ)^{-1} = AZZ^H B^{-1} = AB^{-1} = C \tag{13}$$

Since the unitary matrix Z does not change C or C', Z could be used effectively to keep A upper Hessenberg and B upper triangular during the iteration and not destroy the zeros introduced in step 1.

Suppose the matrix Q in equation (11) is known and the matrix Z which keeps the current A and B in the proper form is known. Define A' and B' by

$$A' = QAZ \tag{14}$$

$$B' = QBZ \tag{15}$$

and then

$$A'B'^{-1} = (QAZ)(QBZ)^{-1} = QAZZ^H B^{-1} Q^H = QAB^{-1} Q^H = C' \tag{16}$$

Thus, if Q and Z can be determined without forming B^{-1}, then the next iterates, A' and B', can be determined by equations (14) and (15).

The QZ iteration must then do two things. First, determine the correct first row of Q. Second, determine Q and Z so that Q retains the correct first row, QAZ is upper Hessenberg, and QBZ is upper triangular.

As mentioned earlier, the first row of Q is the first row of the orthogonal matrix which annihilates all the elements but the first in the first column of $\left(AB^{-1} - \sigma_1 I\right)\left(AB^{-1} - \sigma_2 I\right)$. Since A is upper Hessenberg and B is upper triangular, the first column of $\left(AB^{-1} - \sigma_1 I\right)\left(AB^{-1} - \sigma_2 I\right)$ is completely determined by σ_1, σ_2, and the first two columns of AB^{-1}. Only the nonsingularity of the upper 2 by 2 submatrix of B, that is, b_{11} and b_{22} nonzero, is required to find these columns. This is really no restriction since Moler and Stewart show how to handle the case of the singular or nearly singular submatrix.

The first column of $\left(AB^{-1} - \sigma_1 I\right)\left(AB^{-1} - \sigma_2 I\right)$ is called "the fictitious zeroth column" of AB^{-1} and is easily computed. The first column of AB^{-1} has two nonzero elements and the second column has three. The equations for these elements are

$$\left(AB^{-1}\right)_{11} = \frac{a_{11}}{b_{11}} \tag{17}$$

$$\left(AB^{-1}\right)_{21} = \frac{a_{21}}{b_{11}} \tag{18}$$

$$\left(AB^{-1}\right)_{12} = \frac{a_{12}}{b_{22}} - \frac{a_{11}b_{12}}{b_{11}b_{22}} \tag{19}$$

$$\left(AB^{-1}\right)_{22} = \frac{a_{22}}{b_{22}} - \frac{a_{21}b_{12}}{b_{11}b_{22}} \tag{20}$$

$$\left(AB^{-1}\right)_{32} = \frac{a_{32}}{b_{22}} \tag{21}$$

Since the rate of convergence for the QR algorithm is determined by the ratio of consecutive distinct eigenvalues, shifts are employed to make this ratio as small as possible. The shifts σ_1 and σ_2 are chosen to be the two zeros of the equation

$$\det\left(\overline{A} - \sigma\overline{B}\right) = 0 \tag{22}$$

where

$$\overline{A} = \begin{bmatrix} a_{n-1,n-1} & a_{n-1,n} \\ a_{n,n-1} & a_{n,n} \end{bmatrix}$$

$$\overline{B} = \begin{bmatrix} b_{n-1,n-1} & b_{n-1,n} \\ 0 & b_{n,n} \end{bmatrix}$$

and n is the order of the current A and B matrices. These shifts are not explicitly computed, but techniques similar to those used in "hqr2" (ref. 10) are applied instead. By denoting $n - 1$ by m, the three nonzero elements of the zeroth column of AB^{-1} are computed by the following formulas:

$$a_{10} = \left[\left(\frac{a_{mm}}{b_{mm}} - \frac{a_{11}}{b_{11}}\right)\left(\frac{a_{nn}}{b_{nn}} - \frac{a_{11}}{b_{11}}\right) - \left(\frac{a_{mn}}{b_{nn}}\right)\left(\frac{a_{nm}}{b_{mm}}\right) + \left(\frac{a_{nm}}{b_{mm}}\right)\left(\frac{b_{mn}}{b_{nn}}\right)\left(\frac{a_{11}}{b_{11}}\right)\right]\left(\frac{b_{11}}{a_{21}}\right) + \frac{a_{12}}{b_{22}} - \left(\frac{a_{11}}{b_{11}}\right)\left(\frac{b_{12}}{b_{22}}\right) \tag{23}$$

$$a_{20} = \left(\frac{a_{22}}{b_{22}} - \frac{a_{11}}{b_{11}}\right) - \left(\frac{a_{21}}{b_{11}}\right)\left(\frac{b_{12}}{b_{22}}\right) - \left(\frac{a_{mm}}{b_{mm}} - \frac{a_{11}}{b_{11}}\right) - \left(\frac{a_{nn}}{b_{nn}} - \frac{a_{11}}{b_{11}}\right) + \left(\frac{a_{nm}}{b_{mm}}\right)\left(\frac{b_{mn}}{b_{nn}}\right) \tag{24}$$

$$a_{30} = \frac{a_{32}}{b_{22}} \tag{25}$$

Note that a_{10}, a_{20}, and a_{30} are all real even if the roots (shifts) of equation (22) are complex. This is an important feature of the QZ algorithm.

After a_{10}, a_{20}, and a_{30} have been computed, the iteration involves only premultiplying and postmultiplying the A and B matrices by Householder transformations, the proper elements being annihilated each time. Since there are only three nonzero elements in the "zeroth" column, only the first three elements of the first row of Q are nonzero. Being illustrated on 6 by 6 matrices, A and B have the following form after applying the Householder transformation Q_1 for annihilating a_{20} and a_{30}:

$$\overset{Q_1A}{\begin{bmatrix} x & x & x & x & x & x \\ x & x & x & x & x & x \\ x & x & x & x & x & x \\ 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & x & x \end{bmatrix}} \qquad \overset{Q_1B}{\begin{bmatrix} x & x & x & x & x & x \\ x^2 & x & x & x & x & x \\ x^1 & x^1 & x & x & x & x \\ 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & 0 & x \end{bmatrix}}$$

The algorithm must now reduce Q_1A to upper Hessenberg and Q_1B to upper triangular without affecting the first row. This is done by postmultiplying by the Householder transformation Z_1' which annihilates the elements denoted by superscript 1 and then by the Householder transformation Z_1'' which annihilates the element in position denoted by superscript 2. Letting $Z_1 = Z_1'Z_1''$ yields the following forms:

$$\overset{Q_1AZ_1}{\begin{bmatrix} x & x & x & x & x & x \\ x & x & x & x & x & x \\ x^1 & x & x & x & x & x \\ x^1 & x & x & x & x & x \\ 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & x & x \end{bmatrix}} \qquad \overset{Q_1BZ_1}{\begin{bmatrix} x & x & x & x & x & x \\ 0 & x & x & x & x & x \\ 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & 0 & x \end{bmatrix}}$$

Now, annihilating the elements denoted by superscript 1 with a Householder transformation Q_2 yields

$$
\begin{array}{cc}
Q_2Q_1AZ_1 & Q_2Q_1BZ_1 \\
\begin{bmatrix}
x & x & x & x & x & x \\
x & x & x & x & x & x \\
0 & x & x & x & x & x \\
0 & x & x & x & x & x \\
0 & 0 & 0 & x & x & x \\
0 & 0 & 0 & 0 & x & x
\end{bmatrix}
&
\begin{bmatrix}
x & x & x & x & x & x \\
0 & x & x & x & x & x \\
0 & x^2 & x & x & x & x \\
0 & x^1 & x^1 & x & x & x \\
0 & 0 & 0 & 0 & x & x \\
0 & 0 & 0 & 0 & 0 & x
\end{bmatrix}
\end{array}
$$

Postmultiplying by Z_2, a product of Householder transformations annihilating the elements in the given order, reduces the current B matrix to triangular form. Then premultiplying by Q_3 annihilates the nonzero elements outside the Hessenberg form in the second column of $Q_2Q_1AZ_1Z_2$. This procedure continues until all the unwanted nonzero elements are pushed down to the lower right-hand corner and a Hessenberg matrix A' and an upper triangular matrix B' remain.

By letting the elements of the current transformed A and B matrices be denoted by a_{ij} and b_{ij}, respectively, the iteration can be summarized by the following outline:

```
(1) Compute $a_{10}$, $a_{20}$, and $a_{30}$ by equations (23), (24), and (25)
(2) For $k = 1, 2, . . ., n - 2$,
    (a) Determine $Q_k$ to annihilate $a_{k+1,k-1}$ and $a_{k+2,k-1}$
    (b) Determine $Z'_k$ to annihilate $b_{k+2,k+1}$ and $b_{k+2,k}$
    (c) Determine $Z''_k$ to annihilate $b_{k+1,k}$
(3) Determine $Q_{n-1}$ to annihilate $a_{n,n-2}$
(4) Determine $Z_{n-1}$ to annihilate $b_{n,n-1}$
```

The operation count (only operations of the highest order of n are counted) for one double shift iteration is $13n^2$ multiplications, $13n^2$ additions, and $3n$ square roots. If the eigenvectors are required, then the Z matrices must be accumulated, which adds $8n^2$ more multiplications and $8n^2$ more additions per iteration. If the transposed problem is solved, then the Q matrices are accumulated and add $5n^2$ more multiplications and $5n^2$ more additions per iteration instead of $8n^2$.

COMBINATION SHIFT QZ ALGORITHM

The combination shift QZ algorithm is basically the QZ algorithm with two improvements which take advantage of some opportunities for saving time and operations. Steps 1, 3, and 4 are not altered by the new algorithm. Thus, the iterative step which is the heart of the algorithm is the only step affected.

Step 2 was the generalization of the double shift QR. The double shift QR is used to solve the standard eigenvalue problem $AB^{-1}x = \lambda x$ because of the possibility of complex conjugate shifts σ_1 and σ_2. If the shifts are complex, the double shift version allows the continuation of the use of real arithmetic,.as pointed out earlier. However, if the shifts are real, this feature of the double shift version is no longer an advantage. Thus, a generalization of the single shift implicit QR algorithm might have some advantages when real shifts are encountered. After a discussion of this generalization and one of its properties that can be utilized, the second step of the new algorithm will be explained.

Single Shift Implicit QZ Iteration

Similar to the double shift generalization, the single shift implicit QZ iteration is motivated by assuming B is nonsingular and by examining the single shift implicit QR algorithm for $C = AB^{-1}$. Recall that C is an upper Hessenberg matrix because of step 1.

Suppose one iteration of the single shift implicit QR algorithm with shift σ is applied to C. Then a matrix Q is found so that the matrix QCQ^H is upper Hessenberg and the first row of Q is the first row of the orthogonal matrix which annihilates all the elements except the first in the first column of the matrix $(C - \sigma I)$. The next iterate C' is then defined as

$$C' = QCQ^H \tag{26}$$

As was the case in the double shift, A and B can be postmultiplied by a unitary matrix Z without altering C or C'. Thus again this technique can be used to keep A upper Hessenberg and B upper triangular during the iteration. Also by finding the proper matrices Q and Z, the next iterates A' and B' can be found without explicitly forming B^{-1} by using the equations

$$A' = QAZ \tag{27}$$

$$B' = QBZ \tag{28}$$

Thus, to perform one single shift implicit QZ iteration, the algorithm must do two things. First, determine the correct first row of Q. Second, determine Q and Z so that Q retains the correct first row, QAZ is upper Hessenberg, and QBZ is upper triangular.

The first row of Q is the first row of the orthogonal matrix which annihilates all the elements except the first in the first column of $\left(AB^{-1} - \sigma I\right)$. Since A is upper Hessenberg and B is upper triangular, the first column of $\left(AB^{-1} - \sigma I\right)$ is completely determined by σ, a_{11}, a_{21}, and b_{11}. In fact, the first column of $\left(AB^{-1} - \sigma I\right)$, called the fictitious zeroth column, has as its first two elements

$$a_{10} = \frac{a_{11}}{b_{11}} - \sigma \tag{29}$$

$$a_{20} = \frac{a_{21}}{b_{11}} \tag{30}$$

with the remaining elements all zero. Thus, a nonzero b_{11} is the only requirement on the nonsingularity of B. If b_{11} is zero, then a deflation can be carried out to reduce the order of the working matrices A and B. This procedure will be discussed later.

The second part is very similar to the second part of the double shift iteration. Premultiplication and postmultiplication by Householder transformations are alternatively used to annihilate the proper elements to reduce A to upper Hessenberg form and B to upper triangular form without affecting the first row. Let Q_1 be the unitary matrix which annihilates a_{20}; that is, the first row of Q_1 is the desired first row for Q. Then Q_1A and Q_1B have the following form on a 5 by 5 example:

$$\begin{array}{cc} Q_1A & Q_1B \\ \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \end{bmatrix} & \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \end{bmatrix} \end{array}$$

Now, Q_1B must be returned to upper triangular form while Q_1A is kept in upper Hessenberg form. Postmultiplying by Z_1 to annihilate the element in the b_{21} position yields

$$\begin{array}{cc} Q_1AZ_1 & Q_1BZ_1 \\ \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \end{bmatrix} & \begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \end{bmatrix} \end{array}$$

Premultiplying by Q_2 to annihilate the element in the a_{31} position yields

$$\begin{array}{cc} Q_2Q_1AZ_1 & Q_2Q_1BZ_1 \\ \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \end{bmatrix} & \begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \end{bmatrix} \end{array}$$

Postmultiplying by Z_2 to annihilate b_{32} yields

$$\begin{array}{cc} Q_2Q_1AZ_1Z_2 & Q_2Q_1BZ_1Z_2 \\ \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & 0 & x & x \end{bmatrix} & \begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \end{bmatrix} \end{array}$$

The process continues with Q_3 annihilating a_{42}, Z_3 annihilating b_{43}, Q_4 annihilating a_{53}, and finishing with Z_4 annihilating b_{54}. This procedure yields a unitary matrix $Q = Q_4Q_3Q_2Q_1$, a unitary matrix $Z = Z_1Z_2Z_3Z_4$, an upper Hessenberg matrix $A' = QAZ$, and an upper triangular matrix $B' = QBZ$ so that the first row of Q is the first row of Q_1 as required.

By letting the elements of the current transformed A and B matrices be denoted by a_{ij} and b_{ij}, respectively, the single shift implicit QZ iteration can be summarized by the following outline:

(1) Compute a_{10} and a_{20} by equations (29) and (30)

(2) For $k = 1, 2, \ldots, n - 1$,

(a) Determine Q_k to annihilate $a_{k+1,k-1}$

(b) Determine Z_k to annihilate $b_{k+1,k}$

The operation count for one iteration is $6n^2$ multiplications, $6n^2$ additions, and $2n$ square roots. If the eigenvectors are required and the Z matrices are accumulated, this procedure adds $3n^2$ more multiplications and $3n^2$ more additions per iteration. There is no advantage in solving the transposed problem and accumulating the Q matrices for this iteration since it would require the same number of operations. This condition is due to only one Z transformation being required to annihilate the B matrix elements as they become nonzero in this iteration whereas two Z transformations are required in the double shift QZ iteration.

As previously noted, the iteration runs into trouble if b_{11} is zero or negligible. (A matrix element is defined as negligible if the element in absolute value is less than ϵ_0 times the norm of the matrix. The infinity matrix norm is used in both the QZ and the combination shift QZ algorithms. The error ϵ_0 is the basic machine roundoff error. (See Wilkinson (ref. 12).)) The solution to this problem is to deflate from the top as is done in the double shift QZ. If b_{11} is negligible, it may be set equal to zero without difficulty since unitary transformations are being used. For a 4 by 4 example, A and B then have the form

$$\begin{array}{c} A \\ \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \end{bmatrix} \end{array} \qquad \begin{array}{c} B \\ \begin{bmatrix} 0 & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{bmatrix} \end{array}$$

Thus a unitary Q can be used to annihilate a_{21} without affecting the form of B. This procedure gives a zero subdiagonal element of A and reduces the problem. If b_{11} is not quite small enough to be set equal to zero, it will cause the shift to be felt only weakly since the division by b_{11} will overshadow σ in the equation for a_{10} and a_{20}. However, the iteration can still be used profitably in converging to a large eigenvalue at the top.

Both the double shift QZ and the single shift QZ have the property of reducing the problem when a negligible subdiagonal element of A is encountered. On a 6 by 6 example, if a_{32} is negligible, it can be set equal to zero and the eigenvalues of the full matrix problem can be found by solving for the eigenvalues of the following two diagonal systems:

$$\begin{array}{cc} A & B \\ \left[\begin{array}{cc|cccc} x & x & x & x & x & x \\ x & x & x & x & x & x \\ \hline 0 & 0 & x & x & x & x \\ 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & x & x \end{array}\right] & \left[\begin{array}{cc|cccc} x & x & x & x & x & x \\ 0 & x & x & x & x & x \\ \hline 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & 0 & x \end{array}\right] \end{array}$$

But because of the simplicity of the single shift iteration, one more generalization of the basic QR algorithm can be applied. The property of reducing the number of transformations by detecting two consecutive small subdiagonal elements of A can now be generalized.

Consecutive Small Subdiagonals

Suppose $a_{r,r-1}$ and $a_{r+1,r}$ are both "small," but not negligible. One would like to develop a test similar to the QR algorithm that would allow the iteration to start at column r, that is, the Q_k matrices would affect only rows r and below. Let the form of A and B on a 6 by 6 example for $r = 3$ be

$$\begin{array}{cc} A & B \\ \left[\begin{array}{cccccc} x & x & x & x & x & x \\ x & x & x & x & x & x \\ 0 & \epsilon_1 & x & x & x & x \\ 0 & 0 & \epsilon_2 & x & x & x \\ 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & x & x \end{array}\right] & \left[\begin{array}{cccccc} x & x & x & x & x & x \\ 0 & x & x & x & x & x \\ 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & 0 & x \end{array}\right] \end{array}$$

where $\epsilon_1(a_{32})$ and $\epsilon_2(a_{43})$ are "small" in some sense.

Suppose the iteration starts at column 3 instead of column 1. Then by making the assumption that b_{33} is nonzero or nonnegligible, one has

$$a_{10} = \frac{a_{33}}{b_{33}} - \sigma \tag{31}$$

$$a_{20} = \frac{\epsilon_2}{b_{33}} \tag{32}$$

A discussion will be given later on what can be done if this assumption is false. Let the transformation Q_3 which annihilates a_{20} be denoted as

$$Q_3 = I - p\left(\alpha p^T\right) \tag{33}$$

where the vector p and scalar α are found by the following set of equations:

$$p^T = (0, 0, 1, u, 0, 0) \tag{34}$$

$$S^2 = a_{10}^2 + a_{20}^2 \tag{35}$$

$$u = \frac{a_{20}}{a_{10} \pm S} \tag{36}$$

$$\alpha = \frac{2}{1 + u^2} \tag{37}$$

The sign of S in equation (36) is chosen so that S and a_{10} have the same sign. Thus, Q_3A and Q_3B would have the forms

$$\begin{array}{cc} Q_3A & Q_3B \\ \begin{bmatrix} x & x & x & x & x & x \\ x & x & x & x & x & x \\ 0 & \eta_1 & x & x & x & x \\ 0 & \eta_2 & x & x & x & x \\ 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & x & x \end{bmatrix} & \begin{bmatrix} x & x & x & x & x & x \\ 0 & x & x & x & x & x \\ 0 & 0 & x & x & x & x \\ 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & 0 & x \end{bmatrix} \end{array}$$

where

$$\eta_1 = \epsilon_1 - \frac{2\epsilon_1}{1 + u^2} \tag{38}$$

$$\eta_2 = - \frac{2\epsilon_1 u}{1 + u^2} \tag{39}$$

If η_2 is negligible, then one is justified in starting the iteration with the third column. One would set η_2 equal to zero, apply Z_3 to annihilate b_{43} and introduce a nonzero in a_{53}, apply Q_4 to annihilate a_{53} and introduce a nonzero in b_{54}, and so forth. Hence, an easily computable bound on $|\eta_2|$ must be obtained. From equations (36), (35), and the sign selection, one finds

$$0 \leqq |u| = \left| \frac{a_{20}}{a_{10} \pm S} \right| \leqq \left| \frac{a_{20}}{a_{10} \pm \sqrt{a_{10}^2}} \right| = \left| \frac{a_{20}}{2a_{10}} \right| \tag{40}$$

By using equations (39) and (40),

$$|\eta_2| = \left| \frac{2\epsilon_1 u}{1 + u^2} \right| \leqq |2\epsilon_1 u| \leqq \left| \frac{\epsilon_1 a_{20}}{a_{10}} \right| \tag{41}$$

From equations (31), (32), and (41),

$$|\eta_2| \leqq \left| \frac{\epsilon_1 \epsilon_2}{a_{33} - \sigma b_{33}} \right| = \left| \frac{a_{32} a_{43}}{a_{33} - \sigma b_{33}} \right| \tag{42}$$

Equations (40), (41), and (42) require a_{10} to be nonzero. It is conceivable for a_{10} to be zero. If this occurs, η_2 would be equal to $\pm\epsilon_1$ and would be nonnegligible. Therefore, the following test inserting the general index r, is used for determining the negligibility of η_2 instead of equation (42):

$$\left| a_{r,r-1} a_{r+1,r} \right| \leqq \left| a_{rr} - \sigma b_{rr} \right| \epsilon_0 \left| |A| \right| \tag{43}$$

If this inequality holds, the iteration can start in column r instead of column 1. In this specific example, the iteration can start in column 3 instead of column 1.

Now, consider equation (38) and try to derive an expression for η_1 when η_2 is negligible

$$\eta_1 = \epsilon_1 - \frac{2\epsilon_1}{1 + u^2} = \frac{-\epsilon_1 - \epsilon_1 u^2 + 2\epsilon_1 u^2}{1 + u^2} \tag{44}$$

Thus, one has

$$\eta_1 = -\epsilon_1 + \left[\frac{2\epsilon_1 u}{1 + u^2}\right] u = -\epsilon_1 - \eta_2 u \tag{45}$$

From equation (35),

$$\left|S\right| \geqq \left|a_{20}\right| \tag{46}$$

which yields from equation (36) and the sign selection

$$\left|u\right| = \left|\frac{a_{20}}{a_{10} \pm S}\right| \leqq \left|\frac{a_{20}}{\left|a_{10}\right| + \left|a_{20}\right|}\right| \leqq 1 \tag{47}$$

Since η_2 is negligible, one now has an expression for η_1

$$\eta_1 = -\epsilon_1 = -a_{32} \tag{48}$$

which just involves changing signs.

There is only one difficulty to clear up. The fact that b_{33} is not negligible was used even though the final result that is tested (eq. (43)) does not require this. If b_{33} is negligible, then one may set b_{33} equal to zero and try to perform a reduction. After setting b_{33} equal to zero, A and B have the forms

$$\begin{array}{cc}
A & B \\
\begin{bmatrix}
x & x & x & x & x & x \\
x & x & x & x & x & x \\
0 & \epsilon_1 & x & x & x & x \\
0 & 0 & \epsilon_2 & x & x & x \\
0 & 0 & 0 & x & x & x \\
0 & 0 & 0 & 0 & x & x
\end{bmatrix}
&
\begin{bmatrix}
x & x & x & x & x & x \\
0 & x & x & x & x & x \\
0 & 0 & 0 & x & x & x \\
0 & 0 & 0 & x & x & x \\
0 & 0 & 0 & 0 & x & x \\
0 & 0 & 0 & 0 & 0 & x
\end{bmatrix}
\end{array}$$

It is desirable to annihilate ϵ_2 by a Householder transformation to reduce the problem. By setting

$$a_{10} = a_{33} \tag{49}$$

$$a_{20} = \epsilon_2 \tag{50}$$

and going through the analysis of equations (33) to (41), $\tilde{\eta}_2$ in this case obeys the following inequality:

$$\left|\tilde{\eta}_2\right| \leq \left|\frac{\epsilon_1\epsilon_2}{a_{33}}\right| \tag{51}$$

If $\tilde{\eta}_2$ is negligible, $\tilde{\eta}_1$ can be found through the same analysis as equations (44) to (48) with the same result

$$\tilde{\eta}_1 = -\epsilon_1 \tag{52}$$

Thus, if $\left|\frac{\epsilon_1\epsilon_2}{a_{33}}\right|$ is negligible, the sign on ϵ_1 can be changed, ϵ_2 annihilated, and the problem reduced.

In summary, the algorithm tests the subdiagonal elements $a_{r,r-1}$ and $a_{r+1,r}$ $(r = n - 1, n-2, \ldots, 2)$ each iteration for the validity of equation (43) if b_{rr} is nonnegligible, and equation (43) with $\left|a_{rr} - \sigma b_{rr}\right|$ replaced by $\left|a_{rr}\right|$ if b_{rr} is negligible. Suppose equation (43) is valid for subdiagonal elements $a_{i,i-1}$ and $a_{i+1,i}$. The algorithm then proceeds according to the negligibility of b_{ii}. If b_{ii} is not negligible, the iteration starts a column i. If b_{ii} is negligible, the matrix problem is reduced into two smaller matrix problems.

Step 2 of Combination Shift QZ Algorithm

The second step of the combination shift QZ algorithm can now be stated; that is,

> reduce A to quasi-triangular form while keeping B in upper triangular form by using a combination of the double shift QZ and the single shift implicit QZ.

The type of iteration used to determine the next iterates depends on the type of shifts computed. By using equation (22), the algorithm determines whether the shifts are real or complex. If they are complex, a double shift QZ iteration is performed as explained earlier. If they are real, a single shift implicit QZ iteration is performed by using the shift which is closest to the value $\frac{a_{nn}}{b_{nn}}$, where a_{nn} and b_{nn} are the n,n elements of the current transformed A and B matrices, respectively.

The reasoning behind the selection of this step is obvious. If the shifts are complex, then the calculation remains in real arithmetic by the double shift and is probably converging to complex eigenvalues. If the shifts are real, then the iteration is probably converging to at least one real eigenvalue which will emerge as $\frac{a_{nn}}{b_{nn}}$.

Since the double shift QZ may be used, it is still an advantage to solve the transposed problem and accumulate the Q transformations. This method will insure only having to accumulate one Householder reflection per iteration.

THEORETICAL COMPARISON OF THE COMBINATION SHIFT QZ AND THE QZ ITERATIONS

One important and interesting form of comparison is operation count (only multiplications and divisions of the highest order of n are stated here). The operation count for one double shift QZ iteration is $13n^2$, whereas the operation count for one combination shift QZ iteration depends on the type of shifts encountered. If the shifts are complex, then the count is the same as that of the double shift. If the shifts are real, then one shift is performed and the iteration requires $6n^2$ operations. (One should note that there are a few more logical statements and multiplications in the combination shift iteration because of the shift-type determination, but these are of order unity per iteration.)

To emphasize what happens in the real case, suppose the two algorithms are converging to a real eigenvalue and the shifts in both the double shift QZ and the combination shift QZ are real. The combination shift QZ would iterate with $6n^2$ operations, obtain a new shift estimate, iterate again with $6n^2$ operations, obtain a new shift estimate, and so on until convergence. The double shift QZ would use both shifts and iterate with $13n^2$ operations, obtain two new shift estimates, iterate again with $13n^2$ operations, and so on until convergence. Thus, the combination shift QZ can perform two iterations with a better shift estimate for the second iteration with n^2 fewer operations than the double shift QZ. Consequently, when trying to converge to a real eigenvalue, it appears that the combination shift would save on the number of operations per two shifts as well as possibly on the number of iterations because of the improved shift estimates. When trying to converge to a complex eigenvalue pair, the shifts should be complex and the two algorithms should be roughly equivalent. One would then expect that the combination shift QZ iteration would be faster than the double shift QZ iteration when real eigenvalues are present, but would at worst (all complex eigenvalues) be only slightly slower. One would also expect that the savings due to the combination shift would be somewhat proportional to the number of real eigenvalues found. These expectations are confirmed by the numerical results presented later.

The numerical tests also indicate that on large matrices, for example of order 50, time and iterations are generally gained or lost in computing the first 50 percent of the eigenvalues. There are two reasons for this gain or loss. First, the size of the matrix is reduced as an eigenvalue or an eigenvalue pair is found. Thus, there are more operations associated with finding the earlier eigenvalues, and an iteration saved at the beginning is worth more than an iteration saved at the end. Second, the earlier iterations help the later iterations by orienting the eigenvalues in approximate order and giving better estimates for the shifts. (This is also the reason that extra early iterations are not completely wasted.) In large matrices, the average rate of determining eigenvalues is generally one eigenvalue per two shift iterations (one double shift or two single shifts) or better after the first 50 percent of the eigenvalues are found. It would then appear that the combination shift QZ algorithm would benefit from all the real eigenvalues being in a position to be found first. This advantage is somewhat offset by the property discussed in the next paragraph.

Orientation of the eigenvalues plays a further role in determining the relative merits of the combination shift iteration. If a real eigenvalue would normally be found between two complex pairs of eigenvalues, the combination shift would operate more efficiently than the QZ since it has the capability of finding just one real eigenvalue. The double shift algorithm would have to disorder the eigenvalues in order to make use of both shifts in the iteration, or would have to perform an iteration with one shift which does not give immediate help in extracting the eigenvalues. To illustrate the point, consider the following example:

$$\overset{A}{\begin{bmatrix} 0 & -3 & -3 \\ 1 & 1 & -2 \\ 0 & 1 & -2 \end{bmatrix}} \qquad \overset{B}{\begin{bmatrix} 1 & 0 & -3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}}$$

The eigenvalues of $Ax = \lambda Bx$ are $-3, \frac{1}{2} \pm \frac{\sqrt{11}}{2} i$. The two shifts σ_1 and σ_2 used in the double shift are -2 and 0. The shift -2 can be used to good advantage in finding the eigenvalue -3, but the shift 0 does not provide much help in converging to the eigenvalues. The combination shift would iterate by using a shift of -2 and would obtain a new shift for the next iteration which should be an even closer approximation to -3. This property of the algorithms indicates that the combination shift should operate more efficiently than the QZ when the real eigenvalues are not bunched together. But when the real eigenvalues are bunched, both shifts in the double shift could be used advantageously and the savings of the combination shift QZ is not as large.

As previously mentioned, the simplicity of the single shift iteration allows one to check for consecutive small subdiagonals. Of course, the time and operations saved by utilizing this property is a function of the number of consecutive small subdiagonals detected by the algorithm and the position of these small elements along the subdiagonal. In the 180 test cases presented in this paper, consecutive small subdiagonals were detected on the average of once every 8.7 single shift iterations. Since the double shift iteration does not have this capability without consuming considerably more time, this is a positive feature for the single shift iteration.

The single shift iteration had to deal with the problem of a possible negligible b_{11} element. This problem was solved by a deflation from the top; that is, finding an infinite eigenvalue and reducing the order of the matrix problem by one. If b_{11} was small but not negligible, then an iteration was performed with a shift essentially equal to zero. This shift was useful in finding the large eigenvalue and thus deflating from the top, but did not provide much help in converging to the stable eigenvalue and deflating at the bottom. The double shift iteration has the same problem with b_{11} and the additional problem of a negligible or almost negligible b_{22}. The double shift iteration cannot take advantage of a negligible b_{22} and is forced to perform an iteration with a shift essentially equal to zero when b_{22} is negligible as well as almost negligible. Hence, the combination shift QZ algorithm would not be as likely to perform an iteration which does not help in converging to a stable eigenvalue as the QZ algorithm.

Since both iterations involve only unitary transformations, they are both stable and well defined and, as expected, the accuracies of the two different iteration strategies are roughly equivalent.

NUMERICAL RESULTS

In order to determine the relative merits of the combination shift QZ algorithm, numerous test cases were run on the Control Data Corporation (CDC) 6600 computers at Langley Research Center. The results of these test cases were compared with the results of the same test cases by using Moler and Stewart's QZ algorithm. The QZ algorithm used was a FORTRAN computer code supplied by Dr. Cleve Moler of University of New Mexico. For the combination shift QZ algorithm, only the subroutine of this code involving the iteration (that is, step 2 of the algorithm) was modified. The test cases were divided into six categories depending on the percentage of real eigenvalues possessed by the test case. The appendix gives the details on the generation of the matrices for these tests. Iteration times for the different test cases are given in tables I to VI. Operation count and total algorithm time for category I are given in tables VII and VIII, respectively.

The first category is one which is often found in the physical sciences; that is, the generalized eigenproblem with all real eigenvalues. For all the test cases in this category, A is a symmetric matrix and B is a symmetric, positive definite, nearly singular matrix. For all the test cases in this category except I-6, A is also nearly singular. Cases I-1, I-2, I-3, and I-6 test the algorithms on problems which consist mainly of stable eigenvalues. The problem has two unstable eigenvalues, in test cases I-1, I-2, and I-3 and three unstable eigenvalues in test cases I-6. Cases I-4 and I-5 test the algorithms on problems consisting of an approximately equal number of stable and unstable eigenvalues. Most of the stable eigenvalues in these two cases are nearly zero. Table I reports the percentage of the QZ iteration time required by the combination shift QZ iteration. For the 30 test cases tried, the average time saved was over 35 percent and the deviation from this time was small. However, there was a definite trend toward larger savings on smaller matrices. To give an operation count comparison, a counter was inserted into both the combination shift QZ iteration and the QZ iteration to count all multiplications and divisions of order unity and above per iteration. Table VII reports the result of this comparison. As one would expect, results similar to table I are obtained with slightly larger percentages saved. Table VIII reports on the time comparisons of the complete algorithms. Since the algorithms are identical except for the iteration step, the results are also similar to those reported for the iteration time except with the percentages closer to 100, as expected.

The second category is the one with the test cases which have 80 percent real eigenvalues. Table II reports the iteration time comparison for this category. The test cases II-1 have complex eigenvalues with larger magnitudes than the real eigenvalues. The test cases II-2 are just the opposite with the complex eigenvalues having smaller magnitudes than the real eigenvalues. Test cases II-3 and II-4 are problems which have each complex conjugate pair of eigenvalues isolated in magnitude; that is, if the eigenvalues were ordered by magnitude, there would be at least one real eigenvalue between every complex conjugate pair. Test cases II-5 and II-6 are problems which have the complex eigenvalues grouped together by magnitude; that is, if the eigenvalues were ordered by magnitude, there would be more than one complex conjugate pair between the real eigenvalues. For the 30 test cases tried in this category, the average time saved in the iteration section of the algorithm is over 25 percent. Again, there is a trend for saving more on the smaller matrices. Also, the deviations are larger on the smaller matrices because of the larger effect of just one more or less iteration.

The third category contains the test cases with 60 percent real eigenvalues. Table III reports on the iteration time comparison between the QZ and the combination shift QZ. Again, test cases III-1 have complex eigenvalues with larger magnitudes, III-2 has complex eigenvalues with smaller magnitudes, III-3 and III-4 have complex eigenvalues isolated, and III-5 and III-6 have complex eigenvalues grouped. For the 30 test cases tried in this category, the average iteration time saved is almost 30 percent.

Table IV and table V report the results of the fourth and fifth category, respectively. The fourth category contains matrices with 40 percent of the eigenvalues real and the fifth category with 20 percent of the eigenvalues real. Again, test cases IV-1, V-1, IV-2, and V-2 have the same property as the respective cases in the two previous categories. Test cases IV-3, V-3, IV-4, V-4, IV-5, V-5, IV-6, and V-6 have their real eigenvalues ordered like the complex eigenvalues are ordered in the respective cases of categories II and III. In the fourth category, a savings of almost 20 percent is realized in the iteration time. The average iteration time saved in the fifth category is 13 percent.

The sixth and last category is that of the all complex eigenvalue cases. Table VI reports the results of this category. Cases VI-1 and VI-2 test the algorithms on problems which consist mainly of stable purely imaginary eigenvalues. Cases VI-3 and VI-4 test the algorithms on problems which have eigenvalues with small and large imaginary parts relative to the real part and eigenvalues with real and imaginary parts of the same order of magnitude. Cases VI-5 and VI-6 test the algorithms on problems consisting of an approximately equal number of stable and unstable purely imaginary eigenvalues. Most of the stable eigenvalues in these two cases are nearly zero. An explanation for the savings in this category is that some real shifts were encountered during the iteration and the combination shift QZ algorithm returned to the shift determination strategy quicker, and thus avoided as many real shifts as the QZ algorithm. Also, small real shifts on the nearly zero complex eigenvalues are used effectively. The average savings on test cases in this category is over 5 percent.

In all the test cases presented in this paper, the stable eigenvalues from the two algorithms are exact to approximately 12 significant figures. Neither algorithm has shown consistently more accuracy than the other. Also, the unstable eigenvalues have been determined as accurately as possible by both algorithms. To determine the accuracy of the unstable eigenvalues, one must check the accuracy of α_i and β_i, the diagonal elements of the resulting triangular A and B matrices, respectively. The α_i and β_i are accurate up to a perturbation of order $||A||\epsilon_0$ and $||B||\epsilon_0$, respectively. For example, if the norms of the original matrices were of order unity and α_i and β_i were of order 10^{-13}, then the eigenvalue λ_i would have approximately one accurate digit even though λ_i would be of order unity itself. To illustrate the point even further, test cases I-4 and I-5 for the 50 by 50 matrices are examples of eigenproblems which theoretically have all real eigenvalues, but both algorithms found some complex eigenvalues among the unstable eigenvalues. But, as expected, the imaginary parts of these eigenvalues had zero accuracy. Complex eigenvalues were the result of the nonpreserving-symmetry property which both algorithms possess.

By looking at all the test cases, one can identify some definite trends concerning individual problems. They are as follows:

(1) The more real eigenvalues a problem has, the more one can expect to save by using the combination shift QZ algorithm. Figure 1 gives a graphic view of this tendency. It shows a graph of the percentage of the QZ iteration time used by the combination shift QZ iteration plotted against the percentage of real eigenvalues. One standard deviation band about the average is also shown.

(2) One can expect a larger savings on smaller matrices by using the combination shift QZ than on the larger matrices. This tendency is noted in practically all the test cases presented. Figure 2 gives a graphic view of the average and one standard deviation band of this trend. Also, as noted earlier, the standard deviation is larger for small matrices because of the greater effect of one iteration and the "nonsettling" of the eigen-

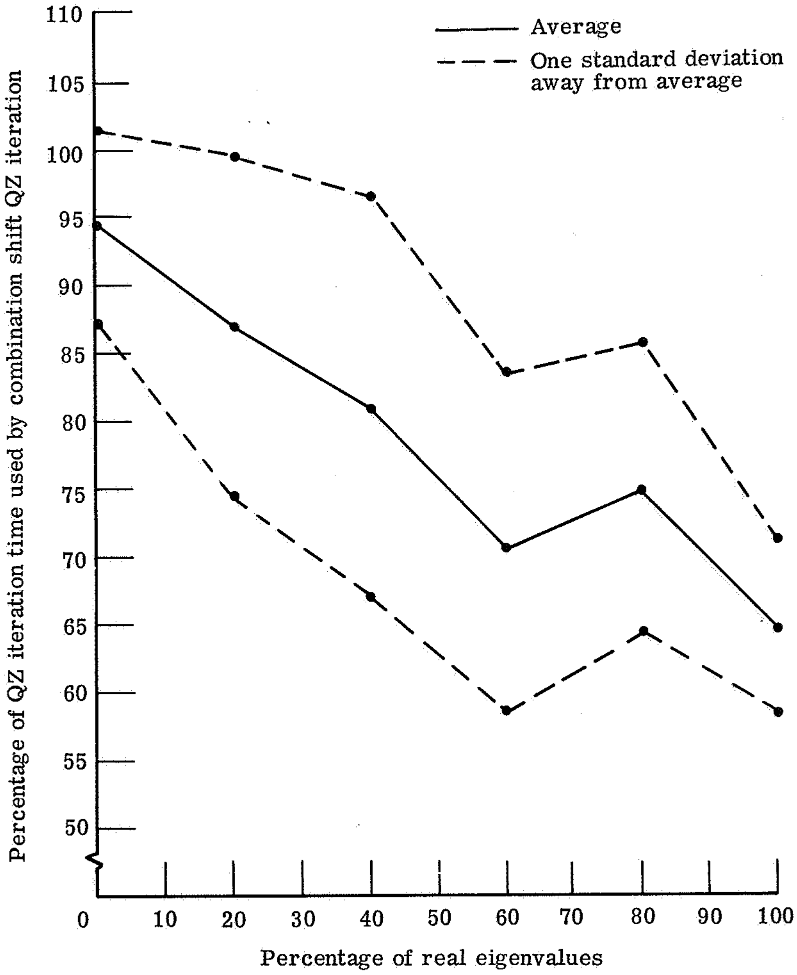

Figure 1.- Iteration time comparison by percentage of real eigenvalues.

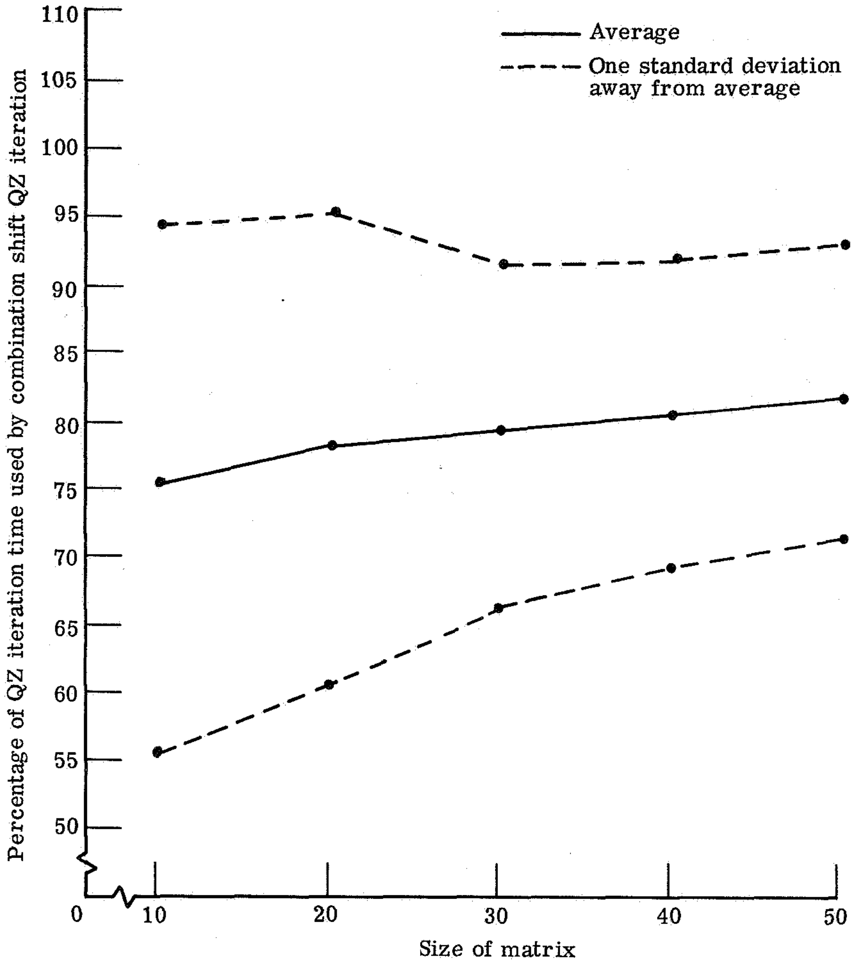

Figure 2.- Iteration time comparison by size of matrix.

values; that is, because of the lack of enough previous iterations, the eigenvalues have not settled down to a specified location or order.

(3) The standard deviation tends to be larger for the cases which have a more equal distribution of real and complex eigenvalues. This difference may be attributed to a higher rate of reordering real and complex eigenvalues so that both shifts in the QZ algorithm are used effectively.

(4) If the real eigenvalues have smaller magnitudes than the complex eigenvalues, one can expect a greater savings by using the combination shift QZ since the smaller eigenvalues tend to be found first. This tendency supports the comments presented in the previous section on the theoretical comparison.

The results presented in the tables correspond to eigenvalue computation only. In the test cases which have been tried, approximately the same percentages resulted when the eigenvector calculation was added. The eigenvectors were calculated as they were in Moler and Stewart's algorithm and not by the preferred method suggested by Kaufman.

As does the QZ algorithm, the combination shift QZ algorithm handles the "ill-disposed" problem when $\det(A - \lambda B)$ vanishes identically, that is, when any λ can be considered as an eigenvalue. This case appears with an "essentially" zero diagonal element on the final triangular matrices at the same relative location.

CONCLUDING REMARKS

The algorithm presented in this paper, called the combination shift QZ algorithm, solves the generalized eigenvalue problem. It should be used when both matrices are singular or nearly singular and tests indicate it is particularly effective on eigenproblems which have a large percentage of real eigenvalues. Based on the results presented in this paper, it should be preferred over existing algorithms which attempt to solve this class of eigenproblems.

Langley Research Center,
National Aeronautics and Space Administration,
Hampton, Va., May 7, 1973.

APPENDIX

TEST CASES

Test cases were generated so that they fell into six basic categories. These categories were pairs of matrices which in the generalized eigenproblem sense have:

Category I – 100 percent real eigenvalues

Category II – 80 percent real eigenvalues

Category III – 60 percent real eigenvalues

Category IV – 40 percent real eigenvalues

Category V – 20 percent real eigenvalues

Category VI – 0 percent real eigenvalues

Within each category, six matrices were generated as a function of the matrix size N.

To help present the details on the generation of the test cases, several matrices need to be defined. First, three orthogonal matrices and a tridiagonal matrix are as follows:

U symmetric matrix whose ij element is $\sin \frac{ij\pi}{N + 1}$

T matrix with diagonal equal to 10 and both superdiagonals and subdiagonals equal to 4

V orthogonal eigenvector matrix of T

P orthogonal eigenvector matrix of a symmetric random number matrix with random numbers uniformly distributed in the interval $[-5., 5.]$

Next, several diagonal and block diagonal matrices, which are functions of the variables indicated and which define the eigenvalues, are defined as follows:

$$D_1(N) = \text{diag}\{1, 3, 5, \ldots, N - 1; 10^{-10}, 10^{-11}, 10^{-12};$$
$$-N + 6, -N + 8, -N + 10, \ldots, -4, -2\}$$

$$D_2(N) = \text{diag}\{1, 2, 3, \ldots, N - 2; 10^{-11}, 10^{-12}\}$$

$$D_3(N) = \mathrm{diag}\left\{1, 2, 3; \left(\frac{N}{2} + 1\right)10^{-12}, \left(\frac{N}{2}\right)10^{-12}, \left(\frac{N}{2} - 1\right)10^{-12}, \ldots, (4)10^{-12}; \right.$$
$$\left. 1, 2, 3, \ldots, \frac{N}{2} - 1\right\}$$

$$D_4(N) = \mathrm{diag}\left\{3, 5, 7; 4, 5, 6, \ldots, \frac{N}{2} + 1; 10^{-12}, 10^{-12}, 10^{-12}, \ldots, 10^{-12}\right\}$$

$$D_5(N,k) = \mathrm{diag}\left\{\begin{bmatrix} 0 & N \\ -N & 0 \end{bmatrix}, \begin{bmatrix} 0 & N - 2 \\ -(N - 2) & 0 \end{bmatrix}, \begin{bmatrix} 0 & N - 4 \\ -(N - 4) & 0 \end{bmatrix}, \right.$$
$$\left. \ldots \begin{bmatrix} 0 & \frac{kN}{5} + 2 \\ -\left(\frac{kN}{5} + 2\right) & 0 \end{bmatrix}; \frac{kN}{5}, \frac{kN}{5} - 1, \frac{kN}{5} - 2, \ldots, 3; 10^{-10}, 10^{-11}\right\}.$$

$$D_6(N) = \mathrm{diag}\left\{10^{-11}, 10^{-11}; 1, 1, 1, \ldots, 1\right\}$$

$$D_7(N,k) = \mathrm{diag}\left\{N, N - 1, N - 2, \ldots, \frac{kN}{5} + 1; \begin{bmatrix} 0 & \frac{kN}{5} \\ -\frac{kN}{5} & 0 \end{bmatrix}, \begin{bmatrix} 0 & \frac{kN}{5} - 2 \\ -\left(\frac{kN}{5} - 2\right) & 0 \end{bmatrix}, \right.$$
$$\left. \begin{bmatrix} 0 & \frac{kN}{5} - 4 \\ -\left(\frac{kN}{5} - 4\right) & 0 \end{bmatrix}, \ldots \begin{bmatrix} 0 & 4 \\ -4 & 0 \end{bmatrix}; \begin{bmatrix} 0 & 10^{-11} \\ -10^{-11} & 0 \end{bmatrix}\right\}$$

$$D_8(N) = \mathrm{diag}\left\{10^{-11}, 10^{-10}; 1, 1, 1, \ldots, 1\right\}$$

$$D_9(N,k,l,m,q) = \mathrm{diag}\left\{10^{-11},\ 10^{-10};\ 3,\ 4,\ 5,\ .\ .\ .,\ k;\ \begin{bmatrix} 0 & k+1 \\ -(k+1) & 0 \end{bmatrix}, \begin{bmatrix} 0 & k+3 \\ -(k+3) & 0 \end{bmatrix},\right.$$

$$\begin{bmatrix} 0 & k+5 \\ -(k+5) & 0 \end{bmatrix},\ .\ .\ .,\ \begin{bmatrix} 0 & l-1 \\ -(l-1) & 0 \end{bmatrix};\ l+1,\ l+2,\ l+3,\ .\ .\ .,\ m;$$

$$\begin{bmatrix} 0 & m+1 \\ -(m+1) & 0 \end{bmatrix}, \begin{bmatrix} 0 & m+3 \\ -(m+3) & 0 \end{bmatrix}, \begin{bmatrix} 0 & m+5 \\ -(m+5) & 0 \end{bmatrix},$$

$$\left. .\ .\ .\ \begin{bmatrix} 0 & q-1 \\ -(q-1) & 0 \end{bmatrix};\ q+1,\ q+2,\ q+3,\ .\ .\ .,\ N\right\}$$

$$D_{10}(N) = \mathrm{diag}\left\{1,\ 1,\ 1,\ .\ .\ .,\ 1;\ 10^{-10},\ 10^{-11}\right\}$$

$$D_{11}(N,k,l,m,q) = \mathrm{diag}\left\{\begin{bmatrix} 0 & 10^{-11} \\ -10^{-10} & 0 \end{bmatrix}; \begin{bmatrix} 0 & 3 \\ -3 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 5 \\ -5 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 7 \\ -7 & 0 \end{bmatrix},\right.$$

$$.\ .\ .\ \begin{bmatrix} 0 & k-1 \\ -(k-1) & 0 \end{bmatrix};\ k+1,\ k+2,\ k+3,\ .\ .\ .,\ l;$$

(Equations continued on next page)

$$\begin{bmatrix} 0 & l + 1 \\ -(l + 1) & 0 \end{bmatrix}, \begin{bmatrix} 0 & l + 3 \\ -(l + 3) & 0 \end{bmatrix}, \begin{bmatrix} 0 & l + 5 \\ -(l + 5) & 0 \end{bmatrix},$$

$$\ldots \begin{bmatrix} 0 & m - 1 \\ -(m - 1) & 0 \end{bmatrix}; m + 1, m + 2, m + 3, \ldots, q;$$

$$\begin{bmatrix} 0 & q + 1 \\ -(q + 1) & 0 \end{bmatrix}, \begin{bmatrix} 0 & q + 3 \\ -(q + 3) & 0 \end{bmatrix}, \begin{bmatrix} 0 & q + 5 \\ -(q + 5) & 0 \end{bmatrix},$$

$$\left. \ldots \begin{bmatrix} 0 & N - 1 \\ -(N - 1) & 0 \end{bmatrix} \right\}$$

$$D_{12}(N) = \operatorname{diag}\left\{ \begin{bmatrix} 0 & N \\ -N & 0 \end{bmatrix}, \begin{bmatrix} 0 & N - 2 \\ -(N - 2) & 0 \end{bmatrix}, \begin{bmatrix} 0 & N - 4 \\ -(N - 4) & 0 \end{bmatrix}, \right.$$

$$\left. \ldots \begin{bmatrix} 0 & 4 \\ -4 & 0 \end{bmatrix}; \begin{bmatrix} 0 & 10^{-11} \\ -10^{-11} & 0 \end{bmatrix} \right\}$$

$$D_{13}(N) = \operatorname{diag}\left\{ \begin{bmatrix} 0 & 10^{-10} \\ -10^{-10} & 0 \end{bmatrix}, \begin{bmatrix} 2. & 1.000000001 \\ -1.000000001 & 0 \end{bmatrix}, \begin{bmatrix} 2. & 1.0001 \\ -1.0001 & 0 \end{bmatrix}; \right.$$

(Equations continued on next page)

$$\left.\begin{bmatrix} 2. & 7. \\ -7. & 0 \end{bmatrix}, \begin{bmatrix} 2. & 9. \\ -9. & 0. \end{bmatrix}, \begin{bmatrix} 2. & 11. \\ -11. & 0 \end{bmatrix}, \ldots \begin{bmatrix} 2. & N - 1 \\ -(N - 1) & 0 \end{bmatrix}\right\}$$

$$D_{14}(N,k) = \text{diag}\left\{\begin{bmatrix} 0 & 10^{-10} \\ -10^{-10} & 0 \end{bmatrix}, \begin{bmatrix} 0 & 10^{-10} \\ -10^{-10} & 0 \end{bmatrix}, \begin{bmatrix} 0 & 10^{-10} \\ -10^{-10} & 0 \end{bmatrix},\right.$$

$$\ldots \begin{bmatrix} 0 & 10^{-10} \\ -10^{-10} & 0 \end{bmatrix}; \begin{bmatrix} 0 & k + 1 \\ -(k + 1) & 0 \end{bmatrix}; \begin{bmatrix} 0 & k + 3 \\ -(k + 3) & 0 \end{bmatrix}$$

$$\left.\begin{bmatrix} 0 & k + 5 \\ -(k + 5) & 0 \end{bmatrix}, \begin{bmatrix} 0 & k + 7 \\ -(k + 7) & 0 \end{bmatrix}, \ldots \begin{bmatrix} 0 & N - 1 \\ -(N - 1) & 0 \end{bmatrix}\right\}$$

$$D_{15}(N,k) = \text{diag}\{\underbrace{1, 1, 1, \ldots, 1}_{k + 2 \text{ components}}; 10^{-10}, 10^{-10}, 10^{-10}, \ldots, 10^{-10}\}$$

Some of the block diagonal matrices are constructed from block diagonal submatrices of dimension 10. They are as follows:

$$\overline{D}_1(N) = \text{diag}\left\{10^{-11}, 10^{-10}, N - 2, \begin{bmatrix} 0 & N - 3 \\ -(N - 3) & 0 \end{bmatrix}; N - 5, N - 6, N - 7, N - 8, N - 9\right\}$$

$$\overline{D}_2(N,k) = \mathrm{diag}\left\langle N - 10k, N - 10k - 1, N - 10k - 2, \begin{bmatrix} 0 & N - 10k - 3 \\ -(N - 10k - 3) & 0 \end{bmatrix}, \right.$$

$$\left. N - 10k - 5, N - 10k - 6, N - 10k - 7, N - 10k - 8, N - 10k - 9 \right\rangle$$

$$\overline{D}_3 = \mathrm{diag}\left\langle 10, 9, 8, \begin{bmatrix} 0 & 7 \\ -7 & 0 \end{bmatrix}, 5, 4, 3, 10^{-10}, 10^{-11} \right\rangle$$

$$\overline{D}_4(N) = \mathrm{diag}\left\langle 10^{-11}, 10^{-10}, \begin{bmatrix} 0 & N - 2 \\ -(N - 2) & 0 \end{bmatrix}, N - 4, N - 5, \right.$$

$$\left. \begin{bmatrix} 0 & N - 6 \\ -(N - 6) & 0 \end{bmatrix}, N - 8, N - 9 \right\rangle$$

$$\overline{D}_5(N,k) = \mathrm{diag}\left\langle N - 10k, N - 10k - 1, \begin{bmatrix} 0 & N - 10k - 2 \\ -(N - 10k - 2) & 0 \end{bmatrix}, \right.$$

$$N - 10k - 4, N - 10k - 5, \begin{bmatrix} 0 & N - 10k - 6 \\ -(N - 10k - 6) & 0 \end{bmatrix},$$

$$\left. N - 10k - 8, N - 10k - 9 \right\rangle$$

$$\overline{D}_6 = \mathrm{diag}\left\{10,\ 9,\ \begin{bmatrix} 0 & 8 \\ -8 & 0 \end{bmatrix},\ 6,\ 5,\ \begin{bmatrix} 0 & 4 \\ -4 & 0 \end{bmatrix},\ 10^{-10},\ 10^{-11}\right\}$$

$$\overline{D}_7(N) = \mathrm{diag}\left\{\begin{bmatrix} 10^{-11} & N \\ -N & 10^{-10} \end{bmatrix},\ N - 2,\ N - 3,\ \begin{bmatrix} 0 & N - 4 \\ -(N - 4) & 0 \end{bmatrix},\right.$$

$$\left. N - 6,\ N - 7,\ \begin{bmatrix} 0 & N - 8 \\ -(N - 8) & 0 \end{bmatrix}\right\}$$

$$\overline{D}_8(N,k) = \mathrm{diag}\left\{\begin{bmatrix} 0 & N - 10k \\ -(N - 10k) & 0 \end{bmatrix},\ N - 10k - 2,\ N - 10k - 3,\right.$$

$$\begin{bmatrix} 0 & N - 10k - 4 \\ -(N - 10k - 4) & 0 \end{bmatrix},\ N - 10k - 6,\ N - 10k - 7,$$

$$\left.\begin{bmatrix} 0 & N - 10k - 8 \\ -(N - 10k - 8) & 0 \end{bmatrix}\right\}$$

$$\overline{D}_9 = \mathrm{diag}\left\{\begin{bmatrix} 0 & 10 \\ -10 & 0 \end{bmatrix},\ 8,\ 7,\ \begin{bmatrix} 0 & 6 \\ -6 & 0 \end{bmatrix},\ 4,\ 3,\ \begin{bmatrix} 10^{-10} & 2 \\ -2 & 10^{-11} \end{bmatrix}\right\}$$

$$\overline{D}_{10}(N) = \text{diag}\left\{\begin{bmatrix} 10^{-11} & N \\ -N & 10^{-10} \end{bmatrix}, N - 2, \begin{bmatrix} 0 & N - 3 \\ -(N - 3) & 0 \end{bmatrix}, \begin{bmatrix} 0 & N - 5 \\ -(N - 5) & 0 \end{bmatrix},\right.$$

$$\left. N - 7, \begin{bmatrix} 0 & N - 8 \\ -(N - 8) & 0 \end{bmatrix}\right\}$$

$$\overline{D}_{11}(N,k) = \text{diag}\left\{\begin{bmatrix} 0 & N - 10k \\ -(N - 10k) & 0 \end{bmatrix}, N - 10k - 2, \begin{bmatrix} 0 & N - 10k - 3 \\ -(N - 10k - 3) & 0 \end{bmatrix},\right.$$

$$\left. \begin{bmatrix} 0 & N - 10k - 5 \\ -(N - 10k - 5) & 0 \end{bmatrix}, N - 10k - 7, \begin{bmatrix} 0 & N - 10k - 8 \\ -(N - 10k - 8) & 0 \end{bmatrix}\right\}$$

$$\overline{D}_{12} = \text{diag}\left\{\begin{bmatrix} 0 & 10 \\ -10 & 0 \end{bmatrix}, 8, \begin{bmatrix} 0 & 7 \\ -7 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 5 \\ -5 & 0 \end{bmatrix}, 3, \begin{bmatrix} 10^{-10} & 2 \\ -2 & 10^{-11} \end{bmatrix}\right\}$$

The test cases can now be defined as

I-1: $\left.\begin{aligned} A &= U^T D_1(N)\, U \\ B &= U^T D_2(N)\, U \end{aligned}\right\}$ $N = 10, 20, 30, 40, 50$

I-2: $\left.\begin{aligned} A &= P^T D_1(N)\, P \\ B &= P^T D_2(N)\, P \end{aligned}\right\}$ $N = 10, 20, 30, 40, 50$

I-3: $\left.\begin{aligned} A &= V^T D_1(N)\, V \\ B &= V^T D_2(N)\, V \end{aligned}\right\}$ $N = 10, 20, 30, 40, 50$

I-4:

$$\left.\begin{aligned} A &= U^T D_3(N)\, U \\ B &= U^T D_4(N)\, U \end{aligned}\right\} \qquad N = 10, 20, 30, 40, 50$$

I-5:

$$\left.\begin{aligned} A &= V^T D_3(N)\, V \\ B &= V^T D_4(N)\, V \end{aligned}\right\} \qquad N = 10, 20, 30, 40, 50$$

I-6:

A = Symmetric random number matrix with random numbers uniformly distributed in the interval $[-10., 10.]$

$B = C^T C$ where the first $N - 3$ rows of C are random numbers uniformly distributed in the interval $[-5., 5.]$ and the last three rows are linear combinations of the preceding rows with a perturbation on the order of 10^{-11} added to each element in these rows.

II-1:

$$\left.\begin{aligned} A &= T\, D_5(N,4)\, U \\ B &= T\, D_6(N)\, U \end{aligned}\right\} \qquad N = 10, 20, 30, 40, 50$$

II-2:

$$\left.\begin{aligned} A &= T\, D_7(N,1)\, U \\ B &= T\, D_8(N)\, U \end{aligned}\right\} \qquad N = 10, 20, 30, 40, 50$$

II-3:

$$N = 10 \qquad \left\{\begin{aligned} A &= T\, \bar{D}_1(10)\, U \\ B &= T\, D_{10}(10)\, U \end{aligned}\right.$$

$$N = 20 \qquad \left\{\begin{aligned} A &= T\, \mathrm{diag}\left\{\bar{D}_1(20),\ \bar{D}_2(20,\ 1)\right\}\, U \\ B &= T\, D_{10}(20)\, U \end{aligned}\right.$$

$$N = 30 \qquad \left\{\begin{aligned} A &= T\, \mathrm{diag}\left\{\bar{D}_1(30),\ \bar{D}_2(30,\ 1),\ \bar{D}_2(30,\ 2)\right\}\, U \\ B &= T\, D_{10}(30)\, U \end{aligned}\right.$$

$$N = 40 \qquad \left\{\begin{aligned} A &= T\, \mathrm{diag}\Big\{\bar{D}_1(40),\ \bar{D}_2(40,\ 1),\ \bar{D}_2(40,\ 2), \\ &\qquad \bar{D}_2(40,\ 3)\Big\}\, U \\ B &= T\, D_{10}(40)\, U \end{aligned}\right.$$

$$N = 50 \qquad \left\{\begin{aligned} A &= T\, \mathrm{diag}\Big\{\bar{D}_1(50),\ \bar{D}_2(50,\ 1),\ \bar{D}_2(50,\ 2), \\ &\qquad \bar{D}_2(50,\ 3),\ \bar{D}_2(50,\ 4)\Big\}\, U \\ B &= T\, D_{10}(50)\, U \end{aligned}\right.$$

II-4:

N = 10 $$\begin{cases} A = T\,\overline{D}_3\,U \\ B = T\,D_8(10)\,U \end{cases}$$

N = 20 $$\begin{cases} A = T\,\text{diag}\left\{\overline{D}_2(20, 0), \overline{D}_3\right\}U \\ B = T\,D_8(20)\,U \end{cases}$$

N = 30 $$\begin{cases} A = T\,\text{diag}\left\{\overline{D}_2(30, 0), \overline{D}_2(30, 1), \overline{D}_3\right\}U \\ B = T\,D_8(30)\,U \end{cases}$$

N = 40 $$\begin{cases} A = T\,\text{diag}\left\{\overline{D}_2(40, 0), \overline{D}_2(40, 1), \overline{D}_2(40, 2), \overline{D}_3\right\}U \\ B = T\,D_8(40)\,U \end{cases}$$

N = 50 $$\begin{cases} A = T\,\text{diag}\left\{\overline{D}_2(50, 0), \overline{D}_2(50, 1), \overline{D}_2(50, 2),\right. \\ \qquad \left.\overline{D}_2(50, 3), \overline{D}_3\right\}U \\ B = T\,D_8(50)\,U \end{cases}$$

II-5:

N = 10 $$\begin{cases} A = T\,D_9(10, 4, 6, 10, -)\,U \\ B = T\,D_{10}(10)\,U \end{cases}$$

N = 20 $$\begin{cases} A = T\,D_9(20, 5, 9, 20, -)\,U \\ B = T\,D_{10}(20)\,U \end{cases}$$

N = 30 $$\begin{cases} A = T\,D_9(30, 6, 12, 30, -)\,U \\ B = T\,D_{10}(30)\,U \end{cases}$$

N = 40 $$\begin{cases} A = T\,D_9(40, 4, 10, 18, 20)\,U \\ B = T\,D_{10}(40)\,U \end{cases}$$

N = 50 $$\begin{cases} A = T\,D_9(50, 5, 11, 21, 23)\,U \\ B = T\,D_{10}(50)\,U \end{cases}$$

II-6: Let $A' = A$ of test case II-5

$B' = B$ of test case II-5

$$\left.\begin{aligned} A &= U\,T^{-1}\,A'\,U^{-1}\,T \\ B &= U\,T^{-1}\,B'\,U^{-1}\,T \end{aligned}\right\} \qquad N = 10, 20, 30, 40, 50$$

III-1:

$$\left.\begin{aligned} A &= T\,D_5(N,3)\,U \\ B &= T\,D_6(N)\,U \end{aligned}\right\} \qquad N = 10,\ 20,\ 30,\ 40,\ 50$$

III-2:

$$\left.\begin{aligned} A &= T\,D_7(N,2)\,U \\ B &= T\,D_8(N)\,U \end{aligned}\right\} \qquad N = 10,\ 20,\ 30,\ 40,\ 50$$

III-3:

$$N = 10 \qquad \left\{\begin{aligned} A &= T\,\overline{D}_4(10)\,U \\ B &= T\,D_{10}(10)\,U \end{aligned}\right.$$

$$N = 20 \qquad \left\{\begin{aligned} A &= T\,\mathrm{diag}\left\{\overline{D}_4(20),\ \overline{D}_5(20,\ 1)\right\}\,U \\ B &= T\,D_{10}(20)\,U \end{aligned}\right.$$

$$N = 30 \qquad \left\{\begin{aligned} A &= T\,\mathrm{diag}\left\{\overline{D}_4(30),\ \overline{D}_5(30,\ 1),\ \overline{D}_5(30,\ 2)\right\}\,U \\ B &= T\,D_{10}(30)\,U \end{aligned}\right.$$

$$N = 40 \qquad \left\{\begin{aligned} A &= T\,\mathrm{diag}\left\{\overline{D}_4(40),\ \overline{D}_5(40,\ 1),\ \overline{D}_5(40,2),\right. \\ &\qquad \left.\overline{D}_5(40,\ 3)\right\}\,U \\ B &= T\,D_{10}(40)\,U \end{aligned}\right.$$

$$N = 50 \qquad \left\{\begin{aligned} A &= T\,\mathrm{diag}\left\{\overline{D}_4(50),\ \overline{D}_5(50,\ 1),\ \overline{D}_5(50,\ 2),\right. \\ &\qquad \left.\overline{D}_5(50,\ 3),\ \overline{D}_5(50,\ 4)\right\}\,U \\ B &= T\,D_{10}(50)\,U \end{aligned}\right.$$

III-4:

$$N = 10 \qquad \left\{\begin{aligned} A &= T\,\overline{D}_6\,U \\ B &= T\,D_8(10)\,U \end{aligned}\right.$$

$$N = 20 \qquad \left\{\begin{aligned} A &= T\,\mathrm{diag}\left\{\overline{D}_5(20,\ 0),\ \overline{D}_6\right\}\,U \\ B &= T\,D_8(20)\,U \end{aligned}\right.$$

$$N = 30 \qquad \left\{\begin{aligned} A &= T\,\mathrm{diag}\left\{\overline{D}_5(30,\ 0),\ \overline{D}_5(30,\ 1),\ \overline{D}_6\right\}\,U \\ B &= T\,D_8(30)\,U \end{aligned}\right.$$

$$N = 40 \qquad \begin{cases} A = T\ \mathrm{diag}\left\{\bar{D}_5(40,\ 0),\ \bar{D}_5(40,\ 1),\ \bar{D}_5(40,\ 2),\ \bar{D}_6\right\}\ U \\ B = T\ D_8(40)\ U \end{cases}$$

$$N = 50 \qquad \begin{cases} A = T\ \mathrm{diag}\left\{\bar{D}_5(50,\ 0),\ \bar{D}_5(50,\ 1),\ \bar{D}_5(50,\ 2),\right. \\ \qquad\qquad \left.\bar{D}_5(50,\ 3),\ \bar{D}_6\right\}\ U \\ B = T\ D_8(50)\ U \end{cases}$$

III-5:

$$N = 10 \qquad \begin{cases} A = T\ D_9(10,\ 5,\ 9,\ 10,\ -)\ U \\ B = T\ D_{10}(10)\ U \end{cases}$$

$$N = 20 \qquad \begin{cases} A = T\ D_9(20,\ 7,\ 15,\ 20,\ -)\ U \\ B = T\ D_{10}(20)\ U \end{cases}$$

$$N = 30 \qquad \begin{cases} A = T\ D_9(30,\ 5,\ 13,\ 21,\ 25)\ U \\ B = T\ D_{10}(30)\ U \end{cases}$$

$$N = 40 \qquad \begin{cases} A = T\ D_9(40,\ 8,\ 16,\ 24,\ 32)\ U \\ B = T\ D_{10}(40)\ U \end{cases}$$

$$N = 50 \qquad \begin{cases} A = T\ D_9(50,\ 7,\ 15,\ 27,\ 39)\ U \\ B = T\ D_{10}(50)\ U \end{cases}$$

III-6: Let $A' = A$ of test case III-5

$B' = B$ of test case III-5

$$\left.\begin{aligned} A &= U\ T^{-1}\ A'\ U^{-1}\ T \\ B &= U\ T^{-1}\ B'\ U^{-1}\ T \end{aligned}\right\} \qquad N = 10,\ 20,\ 30,\ 40,\ 50$$

IV-1:

$$\left.\begin{aligned} A &= T\ D_5(N,2)\ U \\ B &= T\ D_6(N)\ U \end{aligned}\right\} \qquad N = 10,\ 20,\ 30,\ 40,\ 50$$

IV-2:

$$\left.\begin{aligned} A &= T\ D_7(N,3)\ U \\ B &= T\ D_8(N)\ U \end{aligned}\right\} \qquad N = 10,\ 20,\ 30,\ 40,\ 50$$

IV-3:

$$N = 10 \qquad \begin{cases} A = T\ \bar{D}_7(10)\ U \\ B = T\ D_{10}(10)\ U \end{cases}$$

$$N = 20 \qquad \begin{cases} A = T\ \mathrm{diag}\left\{\overline{D}_7(20),\ \overline{D}_8(20,\ 1)\right\}\ U \\ B = T\ D_{10}(20)\ U \end{cases}$$

$$N = 30 \qquad \begin{cases} A = T\ \mathrm{diag}\left\{\overline{D}_7(30),\ \overline{D}_8(30,\ 1),\ \overline{D}_8(30,\ 2)\right\}\ U \\ B = T\ D_{10}(30)\ U \end{cases}$$

$$N = 40 \qquad \begin{cases} A = T\ \mathrm{diag}\left\{\overline{D}_7(40),\ \overline{D}_8(40,\ 1),\ \overline{D}_8(40,\ 2),\ \overline{D}_8(40,\ 3)\right\}\ U \\ B = T\ D_{10}(40)\ U \end{cases}$$

$$N = 50 \qquad \begin{cases} A = T\ \mathrm{diag}\left\{\overline{D}_7(50),\ \overline{D}_8(50,\ 1),\ \overline{D}_8(50,\ 2),\ \overline{D}_8(50,\ 3),\right. \\ \qquad \left.\overline{D}_8(50,\ 4)\right\}\ U \\ B = T\ D_{10}(50)\ U \end{cases}$$

IV-4:

$$N = 10 \qquad \begin{cases} A = T\ \overline{D}_9\ U \\ B = T\ D_8(10)\ U \end{cases}$$

$$N = 20 \qquad \begin{cases} A = T\ \mathrm{diag}\left\{\overline{D}_8(20,\ 0),\ \overline{D}_9\right\}\ U \\ B = T\ D_8(20)\ U \end{cases}$$

$$N = 30 \qquad \begin{cases} A = T\ \mathrm{diag}\left\{\overline{D}_8(30,\ 0),\ \overline{D}_8(30,\ 1),\ \overline{D}_9\right\}\ U \\ B = T\ D_8(30)\ U \end{cases}$$

$$N = 40 \qquad \begin{cases} A = T\ \mathrm{diag}\left\{\overline{D}_8(40,\ 0),\ \overline{D}_8(40,\ 1),\ \overline{D}_8(40,\ 2),\ \overline{D}_9\right\}\ U \\ B = T\ D_8(40)\ U \end{cases}$$

$$N = 50 \qquad \begin{cases} A = T\ \mathrm{diag}\left\{\overline{D}_8(50,\ 0),\ \overline{D}_8(50,\ 1),\ \overline{D}_8(50,\ 2),\right. \\ \qquad \left.\overline{D}_8(50,\ 3),\ \overline{D}_9\right\}\ U \\ B = T\ D_8(50)\ U \end{cases}$$

IV-5:

$$N = 10 \qquad \begin{cases} A = T\ D_9(10,\ 3,\ 9,\ 10,\ -)\ U \\ B = T\ D_{10}(10)\ U \end{cases}$$

$$N = 20 \qquad \begin{cases} A = T\,D_9(20, 3, 9, 13, 19)\,U \\ B = T\,D_{10}(20)\,U \end{cases}$$

$$N = 30 \qquad \begin{cases} A = T\,D_9(30, 6, 16, 20, 28)\,U \\ B = T\,D_{10}(30)\,U \end{cases}$$

$$N = 40 \qquad \begin{cases} A = T\,D_9(40, 6, 18, 22, 34)\,U \\ B = T\,D_{10}(40)\,U \end{cases}$$

$$N = 50 \qquad \begin{cases} A = T\,D_9(50, 10, 24, 30, 46)\,U \\ B = T\,D_{10}(50)\,U \end{cases}$$

IV-6: Let $A' = A$ of test case IV-5

$B' = B$ of test case IV-5

$$\left.\begin{array}{l} A = U\,T^{-1}\,A'\,U^{-1}\,T \\ B = U\,T^{-1}\,B'\,U^{-1}\,T \end{array}\right\} \qquad N = 10, 20, 30, 40, 50$$

V-1:

$$\left.\begin{array}{l} A = T\,D_5(N,1)\,U \\ B = T\,D_6(N)\,U \end{array}\right\} \qquad N = 10, 20, 30, 40, 50$$

V-2:

$$\left.\begin{array}{l} A = T\,D_7(N,4)\,U \\ B = T\,D_8(N)\,U \end{array}\right\} \qquad N = 10, 20, 30, 40, 50$$

V-3:

$$N = 10 \qquad \begin{cases} A = T\,\bar{D}_{10}(10)\,U \\ B = T\,D_{10}(10)\,U \end{cases}$$

$$N = 20 \qquad \begin{cases} A = T\,\mathrm{diag}\left\{\bar{D}_{10}(20), \bar{D}_{11}(20, 1)\right\}\,U \\ B = T\,D_{10}(20)\,U \end{cases}$$

$$N = 30 \qquad \begin{cases} A = T\,\mathrm{diag}\left\{\bar{D}_{10}(30), \bar{D}_{11}(30, 1), \bar{D}_{11}(30, 2)\right\}\,U \\ B = T\,D_{10}(30)\,U \end{cases}$$

$$N = 40 \qquad \begin{cases} A = T\,\mathrm{diag}\left\{\bar{D}_{10}(40), \bar{D}_{11}(40, 1), \bar{D}_{11}(40, 2), \bar{D}_{11}(40, 3)\right\}\,U \\ B = T\,D_{10}(40)\,U \end{cases}$$

$N = 50$ $$\begin{cases} A = T\ \mathrm{diag}\left\{\overline{D}_{10}(50),\ \overline{D}_{11}(50,\ 1),\ \overline{D}_{11}(50,\ 2),\ \overline{D}_{11}(50,\ 3),\right. \\ \qquad \left.\overline{D}_{11}(50,\ 4)\right\}\ U \\ B = T\ D_{10}(50)\ U \end{cases}$$

V-4:

$N = 10$ $$\begin{cases} A = T\ \overline{D}_{12}\ U \\ B = T\ D_{8}(10)\ U \end{cases}$$

$N = 20$ $$\begin{cases} A = T\ \mathrm{diag}\left\{\overline{D}_{11}(20,\ 0),\ \overline{D}_{12}\right\}\ U \\ B = T\ D_{8}(20)\ U \end{cases}$$

$N = 30$ $$\begin{cases} A = T\ \mathrm{diag}\left\{\overline{D}_{11}(30,\ 0),\ \overline{D}_{11}(30,\ 1),\ \overline{D}_{12}\right\}\ U \\ B = T\ D_{8}(30)\ U \end{cases}$$

$N = 40$ $$\begin{cases} A = T\ \mathrm{diag}\left\{\overline{D}_{11}(40,\ 0),\ \overline{D}_{11}(40,\ 1),\ \overline{D}_{11}(40,\ 2),\ \overline{D}_{12}\right\}\ U \\ B = T\ D_{8}(40)\ U \end{cases}$$

$N = 50$ $$\begin{cases} A = T\ \mathrm{diag}\left\{\overline{D}_{11}(50,\ 0),\ \overline{D}_{11}(50,\ 1),\ \overline{D}_{11}(50,\ 2),\right. \\ \qquad \left.\overline{D}_{11}(50,\ 3),\ \overline{D}_{12}\right\}\ U \\ B = T\ D_{8}(50)\ U \end{cases}$$

V-5:

$N = 10$ $$\begin{cases} A = T\ D_{11}(10,\ 6,\ 8,\ 10,\ -)\ U \\ B = T\ D_{10}(10)\ U \end{cases}$$

$N = 20$ $$\begin{cases} A = T\ D_{11}(20,\ 6,\ 10,\ 20,\ -)\ U \\ B = T\ D_{10}(20)\ U \end{cases}$$

$N = 30$ $$\begin{cases} A = T\ D_{11}(30,\ 8,\ 14,\ 30,\ -)\ U \\ B = T\ D_{10}(30)\ U \end{cases}$$

$N = 40$ $$\begin{cases} A = T\ D_{11}(40,\ 6,\ 10,\ 20,\ 24)\ U \\ B = T\ D_{10}(40)\ U \end{cases}$$

$$N = 50 \qquad \begin{cases} A = T\,D_{11}(50, 10, 16, 24, 28)\,U \\ B = T\,D_{10}(50)\,U \end{cases}$$

V-6: Let $A' = A$ of test case V-5

$B' = B$ of test case V-5

$$\left.\begin{aligned} A &= U\,T^{-1}\,A'\,U^{-1}\,T \\ B &= U\,T^{-1}\,B'\,U^{-1}\,T \end{aligned}\right\} \qquad N = 10, 20, 30, 40, 50$$

VI-1:

$$\left.\begin{aligned} A &= T\,D_{12}(N)\,U \\ B &= T\,D_{6}(N)\,U \end{aligned}\right\} \qquad N = 10, 20, 30, 40, 50$$

VI-2:

$$\left.\begin{aligned} A &= U\,D_{12}(N)\,T \\ B &= U\,D_{6}(N)\,T \end{aligned}\right\} \qquad N = 10, 20, 30, 40, 50$$

VI-3:

$$\left.\begin{aligned} A &= T\,D_{13}(N)\,U \\ B &= T\,D_{10}(N)\,U \end{aligned}\right\} \qquad N = 10, 20, 30, 40, 50$$

VI-4:

$$\left.\begin{aligned} A &= U\,D_{13}(N)\,T \\ B &= U\,D_{10}(N)\,T \end{aligned}\right\} \qquad N = 10, 20, 30, 40, 50$$

VI-5: Let $m = 2\left[\frac{N}{4}\right]$ where $[r]$ means the greatest integer not greater than r

$A = T\,D_{14}(N,m)\,U$

$B = T\,D_{15}(N,m)\,U$

VI-6: Let $m = m$ of test case VI-5

$A = U\,D_{14}(N,m)\,T$

$B = U\,D_{15}(N,m)\,T$

REFERENCES

1. Lancaster, Peter: Lambda-Matrices and Vibrating Systems. Pergamon Press, Inc., c.1966.

2. Martin, R. S.; and Wilkinson, J. H.: Reduction of the Symmetric Eigenproblem $Ax = \lambda Bx$ and Related Problems to Standard Form. Numer. Math., Bd. 11, Heft 2, Springer-Verlag, Feb. 1968, pp. 99-110.

3. Crawford, Charles Raymond: The Numerical Solution of the Generalized Eigenvalue Problem. Ph. D. Diss., Univ. of Michigan, 1970.

4. Golub, G. H.; Underwood, R.; and Wilkinson, J. H.: The Lanczos Algorithm for the Symmetric $Ax = \lambda Bx$ Problem. STAN-CS-72-270, Stanford Univ., Mar. 1972.

5. Fix, G.; and Heiberger, R.: An Algorithm for the Ill-Conditioned Generalized Eigenvalue Problem. SIAM J. Numer. Anal., vol. 9, no. 1, Mar. 1972, pp. 78-88.

6. Peters, G.; and Wilkinson, J. H.: $Ax = \lambda Bx$ and the Generalized Eigenproblem. SIAM J. Numer. Anal., vol. 7, no. 4, Dec. 1970, pp. 479-492.

7. Moler, C. B.; and Stewart, G. W.: An Algorithm for Generalized Matrix Eigenvalue Problems. Tech. Rep. No. 3 (Contract N00014-67-A-0181-0023), Univ. of Michigan, Feb. 1972.

8. Wilkinson, J. H.: The Algebraic Eigenvalue Problem. Clarendon Press (Oxford), 1965.

9. Stewart, G. W.: On the Sensitivity of the Eigenvalue Problem $Ax = \lambda Bx$. CNA-13, Univ. of Texas at Austin, Mar. 1971.

10. Peters, G.; and Wilkinson, J. H.: Eigenvectors of Real and Complex Matrices by LR and QR Triangularizations. Numer. Math., Bd. 16, Heft 3, Springer-Verlag, c.1970, pp. 181-204.

11. Kaufman, Linda: A Generalization of the LR Algorithm To Solve $Ax = \lambda Bx$. STAN-CS-72-276, Stanford Univ., Apr. 1972.

12. Wilkinson, J. H.: Rounding Errors in Algebraic Processes. Prentice-Hall, Inc., c.1963.

TABLE I.- ITERATION TIME COMPARISON FOR CASES WITH ALL REAL EIGENVALUES

Test case	Percent of QZ used by combination shift QZ for matrix size of –					Row average	Row standard deviation
	N = 10	N = 20	N = 30	N = 40	N = 50		
I-1	70.5	61.9	58.8	68.0	65.2	64.9	3.9
I-2	44.9	52.1	67.5	65.3	65.6	59.1	8.8
I-3	55.8	55.4	66.2	68.0	65.0	62.1	5.1
I-4	62.8	64.7	62.8	67.2	68.3	65.2	2.3
I-5	62.8	66.1	61.9	66.5	70.1	65.5	2.4
I-6	69.0	72.1	73.5	69.8	73.8	71.6	3.1
Column average . . .	61.0	62.1	65.1	67.5	68.0		
Column standard deviation	8.4	6.2	4.9	1.6	3.2		
Average percentage for these cases							64.7
Standard deviation for these cases							6.4

TABLE II.- ITERATION TIME COMPARISON FOR CASES WITH 80 PERCENT REAL EIGENVALUES

Test case	Percent of QZ used by combination shift QZ for matrix size of –					Row average	Row standard deviation
	N = 10	N = 20	N = 30	N = 40	N = 50		
II-1	58.0	70.7	78.6	77.6	77.9	72.6	7.4
II-2	104.7	97.7	80.9	88.1	74.6	89.2	10.9
II-3	60.5	63.6	69.8	70.3	76.5	68.1	6.1
II-4	60.5	69.3	71.9	73.8	76.8	70.5	5.0
II-5	70.7	62.0	70.0	78.0	69.0	69.9	5.6
II-6	65.7	85.9	85.6	74.2	85.2	79.3	8.3
Column average . . .	70.0	74.9	76.1	77.0	76.7		
Column standard deviation	12.7	12.6	6.4	5.6	4.2		
Average percentage for these cases							74.9
Standard deviation for these cases							10.7

TABLE III.- ITERATION TIME COMPARISON FOR CASES WITH 60 PERCENT REAL EIGENVALUES

Test case	Percent of QZ used by combination shift QZ for matrix size of –					Row average	Row standard deviation
	N = 10	N = 20	N = 30	N = 40	N = 50		
III-1	60.0	71.2	71.3	82.0	84.9	73.9	8.7
III-2	104.4	94.2	92.4	77.7	77.8	89.3	10.3
III-3	58.3	65.9	72.4	66.5	71.6	66.9	5.6
III-4	58.3	65.9	66.8	61.8	71.1	64.8	4.1
III-5	46.2	70.5	70.3	73.3	78.9	67.8	11.5
III-6	63.2	49.5	52.1	65.0	77.9	61.5	10.4
Column average . . .	65.1	69.5	70.9	71.1	77.0		
Column standard deviation	18.2	13.3	11.7	6.7	5.2		
Average percentage for these cases							70.7
Standard deviation for these cases							12.7

TABLE IV.- ITERATION TIME COMPARISON FOR CASES WITH 40 PERCENT REAL EIGENVALUES

Test case	Percent of QZ used by combination shift QZ for matrix size of –					Row average	Row standard deviation
	N = 10	N = 20	N = 30	N = 40	N = 50		
IV-1	60.0	60.1	70.4	84.4	81.7	71.3	10.5
IV-2	112.5	118.8	94.8	94.7	90.4	102.2	11.6
IV-3	66.7	59.8	73.2	73.1	80.4	70.6	7.3
IV-4	61.3	73.5	78.8	75.8	79.9	73.9	6.2
IV-5	78.0	84.7	76.9	82.8	77.0	79.9	2.7
IV-6	101.6	97.0	89.0	89.9	90.7	93.6	5.6
Column average . . .	80.0	82.3	80.5	83.5	83.4		
Column standard deviation	20.3	21.0	8.8	6.9	4.4		
Average percentage for these cases							81.9
Standard deviation for these cases							14.5

TABLE V.- ITERATION TIME COMPARISON FOR CASES WITH 20 PERCENT REAL EIGENVALUES

Test case	Percent of QZ used by combination shift QZ for matrix size of –					Row average	Row standard deviation
	N = 10	N = 20	N = 30	N = 40	N = 50		
V-1	81.4	91.4	79.2	83.7	87.4	84.6	4.7
V-2	100.0	107.9	104.2	100.5	100.4	102.6	3.1
V-3	96.8	85.3	82.4	86.1	84.5	87.0	5.4
V-4	91.8	87.4	86.0	87.0	81.5	86.7	4.2
V-5	88.9	66.4	84.3	83.6	94.9	83.6	9.7
V-6	41.7	64.8	83.8	97.0	98.3	77.1	21.5
Column average . . .	83.4	83.9	86.7	89.6	91.2		
Column standard deviation	19.7	14.6	7.6	7.3	6.7		
Average percentage for these cases							87.0
Standard deviation for these cases							12.4

TABLE VI.- ITERATION TIME COMPARISON FOR CASES WITH ALL COMPLEX EIGENVALUES

Test case	Percent of QZ used by combination shift QZ for matrix size of –					Row average	Row standard deviation
	N = 10	N = 20	N = 30	N = 40	N = 50		
VI-1	102.4	101.1	101.1	100.1	99.5	100.8	3.0
VI-2	82.9	88.3	92.5	95.0	88.3	89.4	4.1
VI-3	85.0	100.8	94.8	95.3	96.2	94.4	5.5
VI-4	102.4	89.5	99.1	95.4	98.8	97.0	5.2
VI-5	100.0	101.1	101.0	100.9	100.6	100.7	2.0
VI-6	75.6	91.8	75.2	87.1	95.7	85.1	8.2
Column average . . .	91.4	95.4	94.0	95.6	96.5		
Column standard deviation	10.5	6.2	8.4	5.2	4.4		
Average percentage for these cases							94.6
Standard deviation for these cases							7.2

TABLE VII.- OPERATION COUNT COMPARISON FOR CASES WITH ALL REAL EIGENVALUES

Test case	Percent of QZ used by combination shift QZ for matrix size of –					Row average
	N = 10	N = 20	N = 30	N = 40	N = 50	
I-1	63.0	58.9	56.7	67.0	64.7	62.1
I-2	39.6	48.9	65.3	64.3	65.3	56.7
I-3	49.5	52.2	64.3	67.0	64.5	59.5
I-4	56.1	60.6	59.8	64.1	66.4	61.4
I-5	56.1	62.1	58.7	64.4	68.2	61.9
I-6	60.2	68.5	71.5	68.4	73.2	68.4
Column average	54.1	58.5	62.7	65.9	67.1	
Average percentage for these cases						61.7

TABLE VIII.- ALGORITHM TIME COMPARISON FOR CASES WITH ALL REAL EIGENVALUES

Test case	Percent of QZ used by combination shift QZ for matrix size of –					Row average
	N = 10	N = 20	N = 30	N = 40	N = 50	
I-1	82.7	77.6	76.3	82.5	81.2	80.1
I-2	65.0	71.4	81.8	80.7	81.5	76.1
I-3	70.7	73.9	81.3	82.5	81.2	77.9
I-4	79.5	83.1	84.7	88.6	89.5	85.1
I-5	77.0	84.0	83.9	88.2	90.5	84.7
I-6	79.5	83.8	84.9	83.2	85.3	83.3
Column average	75.7	78.9	82.2	84.3	84.8	
Average percentage for these cases						81.2

NASA-Langley, 1973 —— 19

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons. Also includes conference proceedings with either limited or unlimited distribution.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include final reports of major projects, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546