More on the QZ Algorithm for Solving The Generalized Eigenvalue Problem

Linda Kaufman

CU-CS-064-75

More on the QZ Algorithm for Solving the Generalized Eigenvalue Problem

by

Linda Kaufman
Department of Computer Science
University of Colorado
Boulder, Colorado 80302

Report #CU-CS-064-75 April 1975

Some Thoughts on the QZ Algorithm for Solving the Generalized Eigenvalue Problem

by

Linda Kaufman
Department of Computer Science
University of Colorado
Boulder, Colorado 80302

ABSTRACT

The QZ algorithm of Moler and Stewart solves the generalized eigenvalue problem of finding λ and $\underset{\sim}{x}$ such that $A\underset{\sim}{x} = \lambda B\underset{\sim}{x}$ for real square matrices A and B by first simultaneously reducing A to upper Hessenberg form and B to triangular form then iteratively reducing A to quasi triangular form (no two consecutive elements on the subdiagonal of A are non-negligible) while preserving the triangularity of B.
The algorithm is analogous to applying the QR algorithm to the standard eigenvalue problem $C\underset{\sim}{x} = \lambda\underset{\sim}{x}$ with $C = AB^{-1}$, but matrix inversion is never explicitly performed and the algorithm is not affected by the condition of the B matrix. In this paper a slight addition to the QZ algorithm is presented which permits the decoupling of the problem into two smaller subproblems when 2 consecutive elements on the subdiagonal of A are small but neither is negligible compared to the norm of A. We also discuss the possibility of sometimes using stabilized elementary transformations instead of Householder transformations.

1. Introduction

In the generalized eigenvalue problem one is given 2 square matrices A and B and asked to find the set of scalars λ and nonzero vectors $\underset{\sim}{x}$ such that

$$A\underset{\sim}{x} = \lambda B\underset{\sim}{x} \quad .$$

Most of the algorithms which purport to solve this problem are less than satisfactory when B is ill conditioned. An exception is the QZ algorithm for real A and B of Moler and Stewart [4] which does not require matrix inversion and is unaffected by the condition of B.

The QZ algorithm attempts to find orthogonal matrices Q and Z which simultaneously reduce A and B to upper triangular form. The eigenvalues of the original problem can be determined by dividing the diagonal elements of the triangularized A by the corresponding diagonal elements of the triangularized B matrix.

The QZ algorithm has 4 sections:

(1) Simultaneously reducing A to upper Hessenberg form (i.e., $a_{ij} = 0$ for $i > (j+1)$) and reducing B to upper triangular form i.e. ($b_{ij} = 0$ for $i > j$)

(2) Iteratively reducing A to quasi-triangular form while preserving the triangularity of B. A matrix is quasi-triangular if no two consecutive elements on the subdiagonal are nonzero. Thus A will look something like:

$$\begin{matrix} X & X & X & X & X & X \\ X & X & X & X & X & X \\ & & X & X & X & X \\ & & & X & X & X \\ & & & X & X & X \\ & & & & & X \end{matrix}$$

(3) The determination of the eigenvalues from the quasi-triangular and triangular matrices.

(4) The determination of the eigenvectors.

Step (2) is the heart of the algorithm and the most time consuming. It uses a sequence of Householder transformations (see Wilkinson [6]) to try to drive the subdiagonal elements of A to zero while preserving the triangularity of B. For n x n matrices each iteration in the step is essentially

(1) Construct the vector $\underset{\sim}{a}$ where (1.1)

$$a_1 = \left(\left(\frac{a_{n-1,n-1}}{b_{n-1,n-1}} - \frac{a_{11}}{b_{11}}\right) \times \left(\frac{a_{nn}}{b_{n,n}} - \frac{a_{11}}{b_{11}}\right) - \frac{a_{n-1,n}}{b_{n,n}} \cdot \frac{a_{n,n-1}}{b_{n-1,n-1}} + \frac{a_{n,n-1}}{b_{n-1,n-1}} \cdot \frac{b_{n-1,n}}{b_{n,n}} \cdot \frac{a_{11}}{b_{11}} \right) \times \frac{b_{11}}{a_{21}} + \frac{a_{12}}{b_{22}} - \frac{a_{11}}{b_{11}} \cdot \frac{b_{12}}{b_{22}} \quad .$$

$$a_2 = \frac{a_{22}}{b_{22}} + \frac{a_{11}}{b_{11}} - \frac{a_{21}}{b_{11}} \cdot \frac{b_{12}}{b_{22}} - \frac{a_{n,n}}{b_{n,n}} - \frac{a_{n-1,n-1}}{b_{n-1,n-1}} + \frac{a_{n,n-1}}{b_{n-1,n-1}} \frac{b_{n-1,n}}{b_{n,n}}$$

$$a_3 = \frac{a_{32}}{b_{22}}$$

$$a_i = 0 \quad \text{for } 4 \le i \le n$$

If any of the diagonal elements of B in the formula are 0, set them to $u||B||$, where u is the precision of the machine.

(2) Find the Householder transformation H_1 which annihilates a_2 and a_3 and premultiply A and B by H_1 to obtain new A and B matrices. The new B matrix will not be upper triangular.

(3) Multiply A and B on the right and left by sequences of Householder transformations to chase unwanted nonzero elements down the second subdiagonal of A and the subdiagonal of B and out the bottom of the matrices.

It is hoped that within a few iterations some of the subdiagonal elements of A will be so small that numerically they may be considered as zero. The algorithm only works on a portion of the original problem in which all the subdiagonal elements of A are non-negiligible, and the value of n in our description is the dimension of that portion. Since each iteration requires $13n^2$ multiplications when eigenvectors are not requested and $21nn_0$ when they are, where n_0 is the dimension of the original problem, it is desirable to reduce the problem to as small a portion as possible. In section 2 we describe a modification of the QZ algorithm which sometimes permits the reduction of the size of the problem. It is based on an idea of Francis [1] for the QR algorithm for solving $A\underset{\sim}{x} = \lambda \underset{\sim}{x}$. The modification considers the case when two consecutive elements on the subdiagonal of A are non-negligible, but sufficiently small that it is numerically safe to decouple the problem into two smaller problems. This situation is not uncommon.

In section 3 we discuss an essential modification of the algorithm which will guarantee that if the second subdiagonal element of B is zero then within 1 iteration either a_{21} or a_{32} will be zero in exact arithmetic. One can construct examples with $b_{22} = 0$, on which the original algorithm will breakdown. In section 4 we propose substituting stabilized elementary transformations for the orthogonal column transformations.

Section 2: Two consecutive small elements

In this section we describe a modification to step 2 of the QZ algorithm which in some cases will save computational effort. Let us consider matrices A and B with the following form.

$$A \qquad\qquad\qquad\qquad\qquad\qquad B$$

$$\begin{array}{l cc|ccccc}
 & X & X & X & X & X & X & X \\
 & X & X & X & X & X & X & X \\
\cline{2-8}
m^{th}\ \text{row} & & X & X & X & X & X & X \\
 & & & X & X & X & X & X \\
 & & & & X & X & X & X \\
 & & & & & X & X & X \\
 & & & & & & X & X
\end{array}
\qquad\qquad
\begin{array}{ccccccc}
X & X & X & X & X & X & X \\
 & X & X & X & X & X & X \\
 & & X & X & X & X & X \\
 & & & X & X & X & X \\
 & & & & X & X & X \\
 & & & & & X & X \\
 & & & & & & X
\end{array}$$

An element a_{ij} of A will be considered negligible if $|a_{ij}| \leq \|A\|_{\infty} \cdot u$ where u is the precision of the machine. We assume that none of the subdiagonal elements of A are negligible so ordinarily the QZ algorithm would be applied to the full matrices.

Let us assume that instead of applying the QZ algorithm to the full matrices, we begin the iteration with their m th columns and use a_{mm}, b_{mm}, $a_{m+1,m+1}$, $b_{m+1,m+1}$ etc., for $a_{11}, b_{11}, a_{22}, b_{22}$, etc. in the formulae in (1.1). The transformation Q_1 affects the m th, m+1 st, and m+2 nd rows of the entire A matrix including its m-1 st column. If the elements introduced in the m-1 st column below the subdiagonal are non-negligible then our starting with the m th row and column has introduced a large perturbation into the problem. On the contrary, if they are negligible we have not introduced a perturbation larger than that which the other elements of the matrix probably already possess because of roundoff error, and we have managed to decrease the operation count.

Let us look at $a_{m+1,m-1}$, and $a_{m+2,m-1}$ after applying Q_1 to A. They are given by

$$a_{m+1,m-1} = x = -a_{m,m-1}a_2 \;/\; \sigma \qquad (2.1)$$
$$a_{m+2,m-1} = y = -a_{m,m-1}a_3 \;/\; \sigma$$
$$\text{where } \sigma = \text{sign } (a_1)\ (a_1^2 + a_2^2 + a_3^2)^{1/2} \quad .$$

Certainly we have not introduced a great perturbation if $|a_{m+1,m-1}|$ + $|a_{m+2,m-1}|$ are negligible i.e.

$$(|a_{m+1,m-1}| + |a_{m+2,m-1}|) \leq u||A||$$

which by (2.1) implies $(|a_{m,m-1}|| (|a_2| + |a_3|) \leq \sigma\ u||A||$.

Since computing square roots is costly, and we know that $|a_1| \leq \sigma$, a practical criterion for negligibility would be

$$|a_{m,m-1}|(|a_2| + |a_3|) \leq |a_1|\ u||A|| \quad . \qquad (2.2)$$

Subsequent column transformations in the iteration will not affect the (m-1)st column of A. Subsequent row transformations will affect the column, but one is guaranteed that

$|a_{i,m-1}| \leq (x^2 + y^2)^{1/2}$ where x and y are given in (2.1) for $i > m$.

Thus criterion (2.2) is a sufficient test for a negligible perturbation for all subsequent transformations in the iteration.

This analysis leads to the idea that for m = n-2,n-3 until 1 one should form the corresponding a_1,a_2, and a_3 and if criterion (2.2) is satisfied,begin the iteration with the m^th row and column. Of course the formulae for the a's are complicated and the idea is open to the criticism that often as much work can be expended in looking for a decoupling as would be saved if the decoupling were not performed.

Therefore the computation of the a's should be done as efficiently as possible. In particular, all expressions in (1.1), which can be computed once and not repeated for each value of m, should certainly be done only once. Therefore it is suggested that the quantities

$$\begin{aligned} q &= a_{n-1,n-1}/b_{n-1,n-1} \\ t &= a_{n,n}/b_{n,n} \\ v &= a_{n,n-1}/b_{n-1,n-1} \\ w &= a_{n-1,n}/b_{n,n}\cdot v \\ x &= v\cdot b_{n-1,n}/b_{n,n} \\ z &= q + t - x \end{aligned} \qquad (2.3)$$

be computed before hand. Also one should have $u||A||$ handy. This can be done once for the whole program.

Now consider multiplying the formulae for a_1, a_2, a_3 in (1.1) by b_{22}. If b_{22} is nonzero, neither Q_1 nor criterion (2.2) is changed, and one has saved a few multiplications. If it is zero, one zero must appear on the subdiagonal of A by the end of the iteration, as the analysis given in Section 3 indicates.

If the formulae in (1.1) are also multiplied by b_{11}, the formulae become more complicated and no computational effort is saved. However, one does discover that if $b_{m,m}$ is zero then it is equivalent to setting

$$\begin{aligned} a_1 &= a_{m,m} \\ a_2 &= a_{m+1,m} \\ a_3 &= 0 \end{aligned}$$

In this case criterion (2.2) is simply

$$|a_{m,m-1} \times a_{m+1,m}| \leq |a_{m,m}| \; u||A|| .$$

If the criterion is passed and Q_1 is applied to A and B, one will discover that B will remain upper triangular and A will have the form

$$\begin{array}{l|cc|ccccc} & X & X & X & X & X & X & X \\ & X & X & X & X & X & X & X \\ \hline m^{th} \text{ row} & & X & X & X & X & X & X \\ & & & & X & X & X & X \\ & & & & X & X & X & X \\ & & & & & X & X & X \\ & & & & & & X & X \end{array} ,$$

so that the problem has been completely decoupled. One should now begin the iteration with the (m+1)st row and column by following Moler's and Stewart's [4] suggestion of deflation from the top if $b_{m+1,m+1}$ is zero. If $b_{m+1,m+1}$ is not zero, the last values of a_1, a_2, and a_3 calculated determine Q_1 for the iteration.

An Algorithm to determine if decoupling can be safely done:

1) Assume $a_{i+1,i}$, for i=k,k+1,...n are non neglible. Compute the quantities given in (2.3) with $b_{n,n}$ and $b_{n-1,n-1}$ replaced by $u||B||$ if they are zero.

2) Set m to n-2. (We assume $(n-k) > 2$.)

3) If $b_{m,m}$ is not 0, go to step (4). If $m > k$ and

$$|a_{m,m-1} \cdot a_{m+1,m}| > |a_{m,m}| \; u||A|| ,$$

go to step(6). Compute the Householder reflection which annihilates the second elements in the vector $(a_{m,m}, a_{m+1,m})^T$ and apply it to the mth and m+1st rows of A and B. Increment m and k by 1, and if m is greater than n-2, the whole iteration is complete. If $b_{m,m}$ is nonzero, go to step (6), else return to the beginning of step(3).

4) Compute the quantities

$c = a_{m+1,m}/b_{m,m}$

$d = a_{m,m}/b_{m,m}$

$r = q-d$

$s = t-d$

$a_1 = ((r\times s-w+x \times d)/c) \times b_{m+1,m+1} + a_{m,m+1} - d\times b_{m,m+1}$

$a_2 = a_{m+1,m+1} - c\times b_{m,m+1} - b_{m+1,m+1}\times(z-d)$

$a_3 = a_{m+2,m+1}$

If m is k go to step (6)

5) If $|a_{m,m-1}| (|a_2| + |a_3|) > |a_1| u ||A||$ then decrease m by 1 and return to step(3).

6) Start the main part of the second step of the QZ algorithm at row m. Remember that column transformations must be applied to rows k through m and that if $k\neq m$, Q_1 will change the sign of $a_{m,m-1}$.

This algorithm requires at most $11(n-3) + 9$ multiplications and $10(n-3) + 9$ additions and $n-3$ comparisons. Considering the fact that each iteration of step (2) of the QZ algorithm costs $13n^2 + 0(n)$ multiplications and the same number of additions when eigenvectors are not computed and significantly more when they are, the price of this modification to the algorithm is not as exorbitant as it might seem at first glance. In the long view it should certainly pay for itself by decreasing the values of n in the n^2 term.

It is interesting to consider the behavior of the algorithm on the following example.

$$\begin{array}{ccccccc} & & & A & & & \\ 1 & 3 & 5 & 7 & 9 & 2 & 4 \\ 2 & 4 & 6 & 8 & 1 & 3 & 5 \\ & 1 & 2 & 3 & 4 & 5 & 8 \\ & & 10^{-9} & 7 & 5 & 3 & 1 \\ & & & 10^{-9} & 5 & 8 & 1 \\ & & & & 1 & 1 & 2 \\ & & & & & 1 & 1 \end{array} \qquad \begin{array}{ccccccc} & & & B & & & \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ & 1 & 2 & 3 & 4 & 5 & 6 \\ & & 1 & 2 & 3 & 4 & 5 \\ & & & 1 & 2 & 3 & 4 \\ & & & & 1 & 2 & 3 \\ & & & & & 1 & 1 \\ & & & & & & 1 \end{array}$$

The algorithm was implemented on the CDC 6400 where $u = 2^{-47} \sim 7 \times 10^{-15}$. For the first 3 iterations when m was 4, (a_1, a_2, a_3) was approximately $(10^{10}, 10, 1)$ and thus criterion (2.2) was passed and the iterations began at row 4. At the end of iteration 3, the subdiagonal of A looked like

$$(2\ ,\ 1\ ,\ -10^{-9},\ -3.29\text{x}10^{-11},\ 1.9\text{x}10^{-4},\ 1.11\text{x}10^{-7})^T$$

and an unplanned, welcomed phenomenon occured: criterion (2.2) was passed when m was 5. Hence the fourth iteration started at row 5. By the end of this iteration, a_{76} was small enough that it could be neglected. The following iteration again started at row 4, and at its completion the subdiagonal of A looked like

$$(2\ ,\ 1\ ,\ 10^{-9}\ ,\ -2.9\text{x}10^{-22}\ ,\ 1.67\text{x}10^{-11}\ ,\ -2.3\text{x}10^{-15})^T.$$

Thus the last 3 rows of A were in quasi-triangular form and one could concentrate on the top 4x4 matrix, which was reduced to quasi-triangular form in 2 iterations.

With the original algorithm all iterations began at row 1. The algorithm also reduced the problem to the top 4x4 submatrices after 5 iterations, but a_{43} had grown considerably and the subdiagonal of A looked like

$$(-3\text{x}10^{-2}\ , -2.3\ ,\ 1.5\text{x}10^{-2}\ ,\ -2\text{x}10^{-23}\ ,\ 1\text{x}10^{-10}\ ,\ 2.3\text{x}10^{-15})^T.$$

In contrast to the modified algorithm, 3 iterations were required to reduce the problem to the top 3 rows and 2 more to completely reduce the A matrix to quasi-triangular form. Thus the original algorithm required 3 more small iterations. When eigenvectors were required about 50% more time was required for the original algorithm than for the modified one.

The eigenvalues computed by the two algorithms are given below:

Modified Algorithm	Original Algorithm
$-9.9999999999994\times10^{-1}$	$-9.9999999999983\times10^{-1}$
$-9.2218442367418\times10^{-15}$	$-1.4617199904744\times10^{-14}$
1.9999999999999	1.9999999999998
6.9999999943482	6.9999999943482
-1.7320508076675	-1.7320508076675
4.0000000019230	4.0000000019228
1.7320508073967	1.7320508073967

Obviously there is not much difference between the sets of eigenvalues, but since the problem has eigenvalues at -1,2, and 0, those given by the modified algorithm are preferable.

Although in the above example, searching for small elements increased the efficiency of the algorithm significantly and gave slightly more accurate eigenvalues, in general one should not expect this to be the case. On various random examples generated, the time for both algorithms were very similar.

Section 3: An analysis when b_{22} is zero

In [5] Ward analyzes of the convergence of the QZ algorithm when the formula for $\underset{\sim}{a}$ is given by (1.1). His theorems specifically exclude the case when b_{22} is 0, for in this case the algorithm can fail to converge as witnessed by the following example:

$$\begin{array}{cccccc} \multicolumn{6}{c}{A} \\ 1 & -1 & 0 & 1 & 0 & 1 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ & -1 & -1 & 1 & 0 & 1 \\ & & 1 & 0 & 1 & 0 \\ & & & 1 & 1 & 0 \\ & & & & 1 & -1 \end{array} \qquad \begin{array}{cccccc} \multicolumn{6}{c}{B} \\ 1 & 0 & 0 & 0 & 0 & 1 \\ & 0 & 0 & 1 & 1 & 0 \\ & & -1 & 0 & 0 & 1 \\ & & & 1 & 1 & 0 \\ & & & & 1 & -1 \\ & & & & & 0 \end{array}$$

After normalization, the vector $\underset{\sim}{a}$ will always be $(0,0,1,0,0,0)^T$ and both matrices will alternate between the original ones given above and the matrices A' and B' given below.

$$\begin{array}{cccccc} \multicolumn{6}{c}{A'} \\ -1 & -1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ & -1 & 1 & 1 & 0 & 1 \\ & & -1 & 0 & 1 & 0 \\ & & & 1 & 1 & 0 \\ & & & & 1 & -1 \end{array} \qquad \begin{array}{cccccc} \multicolumn{6}{c}{B'} \\ -1 & 0 & 0 & 0 & 0 & 1 \\ & 0 & 0 & 1 & 1 & 0 \\ & & 1 & 0 & 0 & 1 \\ & & & 1 & 1 & 0 \\ & & & & 1 & -1 \\ & & & & & 0 \end{array}$$

Actually, all known implementations of the QZ algorithm use an ad-hoc vector for $\underset{\sim}{a}$ if convergence is not attained after 10 iterations.

If, however, the formula for $\underset{\sim}{a}$ in (1.1) had originally been multiplied by b_{22}, then $\underset{\sim}{a}$ would have been (-1,0,-1,0,0,0) and after 1 iteration a_{32} would be zero.

This is not an isolated example and we shall show that a zero will always appear on the subdiagonal of A if b_{22} is 0 and if the algorithm is carried out in exact arithmetic with the formula for $\underset{\sim}{a}$ multiplied by b_{22}.

Let us assume we are applying the second step of the QZ algorithm to an upper Hessenberg matrix A with no negligible subdiagonal elements and an upper triangular matrix B with $b_{22} = 0$. Since the problem can be deflated from the top when $b_{11} = 0$, we'll assume b_{11} is nonzero. The algorithm begins with the creation of the vector $\underset{\sim}{a}$ using the formulae in (1.1) multiplied by b_{22} as in section 2. Since b_{22} is assumed to be 0, we have

$$\begin{aligned} a_1 &= a_{12} - a_{11}b_{12}/b_{11} \\ a_2 &= a_{22} - a_{21}b_{12}/b_{11} \\ a_3 &= a_{3,2} \\ a_i &= 0 \qquad i \geq 3 \end{aligned} \tag{3.1}$$

Thus $\underset{\sim}{a} = \underset{\sim}{a}_2 - b_{12}/b_{11}\underset{\sim}{a}_1$ where $\underset{\sim}{a}_i$ denotes the i^{th} column of the A matrix.

Then the algorithm constructs the Householder transformation Q_1 such that $Q_1\underset{\sim}{a} = -\text{sign}\ (a_1)||\underset{\sim}{a}||_2\underset{\sim}{e}_1$

where $\underset{\sim}{e}_1$ is the first column of the identity matrix and applies Q_1 to A and B.

Because of the formula for $\underset{\sim}{a}$ we have

$$Q_1(\underset{\sim}{a}_2 - b_{12}/b_{11}\underset{\sim}{a}_1) = -\text{sign}\ (a_1)||\underset{\sim}{a}||_2\underset{\sim}{e}_1. \tag{3.2}$$

This fact is the key to the following theorem:

Theorem: After one iteration, the double shift QZ algorithm applied to an upper Hessenberg matrix A with no zero subdiagonal elements and an upper triangular matrix B with b_{11} nonzero and b_{22} zero, produces an A matrix in which either a_{21} is zero or a_{32} is zero if exact arithmetic is used.

Proof: The matrix $B' = Q_1B$ has the form

$$\begin{matrix} X & X & X & X & X & X \\ X & X & X & X & X & X \\ X & X & X & X & X & X \\ & & & X & X & X \\ & & & & X & X \\ & & & & & X \end{matrix}$$

and the third part of the iteration (see section 1) begins with the construction of V_1 such that $B'V_1$ is upper triangular. The matrix V_1 is the product of two matrices $V_1'V_1''$ where V_1' is the Householder transformation designed to zero b_{31}' and b_{32}' and V_1'' is the Householder transformation designed to zero the (2,1) element of $B'' = B'V_1'$.

It is easy to verify that

$$b_{21}' = b_{11}a_2/\sigma$$
$$b_{31}' = b_{11}a_3/\sigma$$
$$b_{22}' = b_{12}a_2/\sigma$$
and $$b_{32}' = b_{12}a_3/\sigma \qquad \text{where } \sigma = \text{sign}(a_1)\sqrt{a_1^2 + a_2^2 + a_3^2}$$

where a_1, a_2, and a_3 are given in (3.1),

and that the top 3x3 submatrix of V_1' is given by

$$\begin{pmatrix} 1 - \dfrac{b_{11}^2d^2}{e(e+b_{33}')} & -\dfrac{b_{12}b_{11}d^2}{e(e+b_{33}')} & -\dfrac{b_{11}d}{e} \\ \dfrac{-b_{12}b_{11}d^2}{e(e+b_{33}')} & 1 - \dfrac{b_{12}^2d^2}{e(e+b_{33}')} & -\dfrac{b_{12}d}{e} \\ -\dfrac{b_{11}d}{e} & -\dfrac{b_{12}d}{e} & 1 - \dfrac{(b_{33}' + e)}{e} \end{pmatrix}$$

where $d = a_3/\sigma$ and $e = \left(\sqrt{b_{31}'^2 + b_{32}'^2 + b_{33}'^2}\right) \operatorname{sign}(b_{33}')$.

The elements b_{21}'' and b_{22}'' are thus

$$b_{21}'' = b_{11}\left[\frac{a_2}{\sigma}\left(1 - \frac{b_{11}^2 d^2}{e(e+b_{33}')} - \frac{b_{12}^2 d^2}{e(e+b_{33}')}\right) - \frac{d b_{23}'}{e}\right] \qquad (3.3)$$

and $$b_{2,2}'' = b_{12}\left[\frac{a_2}{\sigma}\left(\frac{-b_{11}^2 d^2}{e(e+b_{33}')} + 1 - \frac{b_{12}^2 d^2}{e(e+b_{33}')}\right) - \frac{d b_{23}'}{e}\right] .$$

We can now distinguish 2 cases:

Case 1: $b_{2,1}''$ is nonzero so that V_1'' is not the identity matrix.

Case 2: $b_{2,1}''$ is zero which by (3.3) implies $b_{2,2}''$ is also zero since b_{11} is nonzero.

For Case 1, one can show that the first column of V_1 is

$$\frac{1}{\sqrt{b_{12}^2 + b_{11}^2}}\,(b_{12}, -b_{11}, 0 \ldots 0)^T \quad .$$

Hence $AV_1\underset{\sim}{e}_1$ is

$$\frac{1}{\sqrt{b_{12}^2 + b_{11}^2}}\,(b_{12}\underset{\sim}{a}_1 - b_{11}\underset{\sim}{a}_2)$$

and by (3.2) we have

$Q_1AV_1\underset{\sim}{e}_1 = \beta\underset{\sim}{e}_1$ for some complicated constant β.

Future column transformations in the iteration do not touch the first column and future row transformations only touch the zero elements of the first column of Q_1AV_1. Thus the zero element in the (2,1) position will never be affected and so at the beginning of the next iteration a_{21} will be zero.

For Case 2, V_1'' is the identity matrix. The vectors $AV_1\underset{\sim}{e}_1$ and $AV_1\underset{\sim}{e}_2$ are

$$AV_1\underset{\sim}{e}_1 = \left(1 - \frac{b_{11}^2 d^2}{e(e+b_{33}')}\right)\underset{\sim}{a}_1 - \frac{b_{12}b_{11}d^2}{e(e+b_{33}')}\underset{\sim}{a}_2 - b_{11}\frac{d}{e}\underset{\sim}{a}_3$$

$$= -b_{11}\left(\frac{d^2}{e(e+b_{33}')}(b_{11}\underset{\sim}{a}_1 + b_{12}\underset{\sim}{a}_2) + \frac{d}{e}\underset{\sim}{a}_3\right) + \underset{\sim}{a}_1$$

$$\equiv b_{11}\underset{\sim}{g} + \underset{\sim}{a}_1 = b_{11}\left(\underset{\sim}{g} + \frac{\underset{\sim}{a}_1}{b_{11}}\right) \qquad (3.4)$$

and

$$AV_1\underset{\sim}{e}_2 = -b_{12}\left(\frac{d^2}{e(e+b_{33}')}(b_{11}\underset{\sim}{a}_1 + b_{12}\underset{\sim}{a}_2) + \frac{d}{e}\underset{\sim}{a}_3\right) + \underset{\sim}{a}_2$$

$$= b_{12}\underset{\sim}{g} + \underset{\sim}{a}_2 . \qquad (3.5)$$

Let A'' designate Q_1AV_1.

From (3.2),(3.4), and (3.5) we have

$$\underset{\sim}{a}_2'' = b_{12}\left(Q_1\underset{\sim}{g} - \text{sign}(a_1)||a||_2\underset{\sim}{e}_1 + \frac{Q_1\underset{\sim}{a}_1}{b_{11}}\right) ,$$

$$\underset{\sim}{a}_1'' = b_{11}\left(Q_1\underset{\sim}{g} + \frac{Q_1\underset{\sim}{a}_1}{b_{11}}\right)$$

which implies that

$$a_{22}'' = \frac{b_{12}}{b_{11}} a_{21}'' , \qquad a_{32}'' = \frac{b_{12}a_{31}''}{b_{11}}$$

and $$a_{42}'' = \frac{b_{12}}{b_{11}} a_{41}'' \qquad (3.6)$$

The next segment of step (3) of the iteration involves finding the transformation Q_2 in the (2,3,4) plane which zeroes a_{31}'' and a_{41}''. By (3.6) this transformation will also zero a_{32}'' and a_{42}''. Since we have assumed b_{21}'' is zero which implies b_{22}'' is zero, the matrices Q_2A''

and Q_2B'' will look like

$$\begin{array}{cccccc}
X & X & X & X & X & X \\
X & X & X & X & X & X \\
 & & X & X & X & X \\
 & & X & X & X & X \\
 & & & X & X & X \\
 & & & & X & X
\end{array}
\qquad
\begin{array}{cccccc}
X & X & X & X & X & X \\
 & & X & X & X & X \\
 & & X & X & X & X \\
 & & X & X & X & X \\
 & & & & X & X \\
 & & & & & X
\end{array}$$

Future transformations in the iteration will not affect the zero element in the (3,2) position of the A matrix.

Section 4: Using Elementary Transformations

Each iteration of the second part of the QZ algorithm might be summarized as follows:

1) Compute shift parameters λ_1 and λ_2 , the eigenvalues of the problem

$$\begin{pmatrix} a_{n-1,n-1} - \lambda b_{n-1,n-1} & a_{n-1,n} - \lambda b_{n-1,n} \\ a_{n,n-1} & a_{n,n} - \lambda b_{n,n} \end{pmatrix} \underset{\sim}{x} = 0$$

2) Find an orthogonal matrix Q such that $Q(AB^{-1} - \lambda_1 I)(AB^{-1} - \lambda_2 I)$ is upper triangular.

3) Find an orthogonal matrix Z such that QBZ is upper triangular.

4) Replace A by QAZ and B by QBZ.

The basis of the QZ algorithm is that each iteration is equivalent to an iteration of the QR algorithm (see [1]) applied to $C = AB^{-1}$ with shifts λ_1 and λ_2 as defined above. The proof of this fact, given in [4], is independent of the form of Z and hence Z need not be orthogonal. In fact Z could be a product of stabilized elementary transformations such as in Gaussian elimination with partial pivoting. The new matrices $A' = QAZ$ and $B' = QBZ$ would not necessarily be those produced by the original algorithms, but, in the absence of roundoff error, the product AB^{-1} would be the same.

The advantage of using elementary transformations is efficiency. The operation counts given in the table 4.1 certainly indicate that per iteration using stabilized elementary transformations is faster. It is possible, however, for the total execution for the new algorithm to require more or fewer iterations. Because the shift strategy is based on A and B and not on AB^{-1}, one would not expect the two algorithms to follow the same steps. If the shift were changed to the eigenvalues of the lower 2 x 2 submatrix of AB^{-1}, then both algorithms would produce the same Q each iteration, in the absence of roundoff error, and the product AB^{-1} would always be the same. In practice the time for the second phase of the new algorithm is about 75% that of the original one when eigenvectors are not computed and 70% that of the original one when they are.

Table 4.1: Operation count per iteration(1)

	Original Algorithm		Elementary Transformations I		Elementary Transformations II	
	Eigenvectors		Eigenvectors		Eigenvectors	
	Without	With	Without	With	Without	With
Square roots	$3n$	$3n$	n	n	$2n$	$2n$
Multiplications(2)	$13n^2$	$21nn_o$	$8n^2$	$11nn_o$	$8n^2$	$10nn_o$
Additions(2)	$13n^2$	$21nn_o$	$8n^2$	$11nn_o$	$8n^2$	$10nn_o$
Memory Accesses(2),(3)	$46n^2$	$78nn_o$	$\geq 32n^2$ $\leq 44n^2$	$\geq 44nn_o$ $\leq 56nn_o$	$\geq 32n^2$ $\leq 44n^2$	$\geq 40nn_o$ $\leq 52nn_o$

(1) n refers to the size of the current subproblem and n_o to the size of the original problem.

(2) terms of order n are ignored.

(3) the exact number of memory accesses when using linear transformations depends on whether pivoting is performed.

The disadvantage of the new algorithm is stability. For both algorithms one can show that there exist matrices E, F, Q and Z such that the computed matrices $\overline{A}$ and $\overline{B}$ satisfy

$$\overline{A} = QAZ + E \text{ and } \overline{B} = QBZ + F$$

where $||E||$ and $||F||$ are bounded. However, the apriori bounds are much less for the original algorithm. If A_i and B_i represent the i^{th} computed A and B matrices and $\overline{A}$ and $\overline{B}$ the s^{th} computed matrices, then

$$||E|| \leq k_1 u \sum_{i=1}^{s} ||A_i||_2 ||N_i||_2 \text{ and } ||F|| \leq k_2 u \sum_{i=1}^{s} ||B_i||_2 ||N_i||_2$$

where k_1 and k_2 are constants, u is the precision of the machine and N_i is the product of the column transformations subsequently applied to A_i to form $\overline{A}$. When using orthogonal transformations $||A_i||_2$ is approximately $||A||_2$, $||B_i||_2$ is approximately $||B||_2$ and $||N_i||_2$ is approximately 1 for $i = 1,2 \ldots s$. When using elementary transformations one finds that the product of 2j column transformations can have numbers larger than the j^{th} Fibonacci number. Thus the bounds on $||N_i||$, $||B_i||$ and $||A||_i$ are huge.

In practice it is extremely rare that the proposed algorithm will cause trouble. However, from experience with similar algorithms with similar claims, I have no doubt that users will eventually find examples for which significant element growth occurs. A user has finally determined a complex hermetian matrix with well separated eigenvalues, [2] for which the LR algorithm ([6]), which uses stabilized elementary transformations exclusively, computes very poor eigenvectors and eigenvalues slightly in error (i.e. 9 accurate decimal places when one would expect 14). Several experiments were performed with this matrix. When the matrix was changed to a real one of twice the size and B was set to the identity matrix, the eigenvalue and eigenvectors computed by the proposed QZ algorithm yielded relative residuals of 10^{-12} rather than 10^{-14}. The results obtained when using a modification of the LZ algorithm [3] were a bit more disturbing. The LZ algorithm is designed for complex $A\underset{\sim}{x} = \lambda B\underset{\sim}{x}$. It uses stabilized elementary transformations and has the same general outline as the QZ algorithm, except that the vector $\underset{\sim}{a}$ in (1.1) is defined differently. In the modification of the LZ algorithm orthogonal row transformations were substituted for stabilized elementary row transformations, so that the situation was similar to the proposed QZ algorithm. Like those obtained using the LR algorithm, the computed eigenvectors had at times only 3 accurate decimal places,but the computed eigenvalues were correct to about 13 decimal places. Experience with this example suggests that using elementary transformations may sometimes cause trouble.

The huge numbers in the bounds for $||E||$ and $||F||$ can be significantly decreased if the arithmetic is rearranged when using elementary transformations.

Thus far we've assumed that the elementary transformations are replacing the column transformations of the Moler and Stewart algorithm. Thus a typical sequence of transformations would be a Householder transformation in the (k , k+1 , and k+2) planes to zero $a_{k+2,k-1}$ and $a_{k+1,k-1}$ followed by two column elementary transformations: one in the (k , k+1, k+2) planes to zero $b_{k+2,k}$ and $b_{k+2,k}$; the other in the (k,k+1) planes to zero $b_{k+1,k}$.

One could also use the following sequence of transformations

1) Zero $a_{k+2,k-1}$ using a Householder transformation in the (k+2,k+1) planes

2) Zero $b_{k+2,k+1}$ using an elementary column transformation in the (k+2,k+1) planes

3) Zero $a_{k+1,k-1}$ using a Householder transformation in the (k+1,k) planes

4) Zero $b_{k+1,k}$ using an elementary column transformation in the (k+1,k) planes.

For this algorithm the bound on $||\bar{A}||$ does not grow as fast; after t iterations, the computed matrix A_t satisfies $||A_t||_1 \leq 4^t \, ||A||_1$. The operation count for this algorithm is given in columns 5 and 6 of table 4.1.

Note one could also use elementary transformations for the first part of the QZ algorithm and face the same tradeoff between operation count and apriori stability bounds.

Acknowledgement: The author wishes to thank Robert Ward for his comments on the original manuscript.

REFERENCES

[1] J. G. Francis, "The QR Transformation - A Unitary Analogue of the LR Transformation", Computer Journal 4, 265-271, 332-345, 1961-1962.

[2] B. Gabow, Argonne National Labs, private communication, February 1976.

[3] L. Kaufman, "The LZ Algorithm to Solve the Generalized Eigenvalue Problem", SIAM J. Numer. Anal. 11, 997-1023, 1974.

[4] C. B. Moler and G. W. Stewart, "An Algorithm for Generalized Matrix Eigenvalue Problems", SIAM J. Numer. Anal. 10, 241-256, 1973.

[5] R. C. Ward, " The Combination Shift QZ Algorithm", SIAM J. Numer. Anal. 12, 835-853, 1975.

[6] J. H. Wilkinson, The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965.