SOLUTION OF A COMPLEX QUADRATIC EIGENVALUE PROBLEM
RELATED TO PIPE FLOW

Donald A. Gignac

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED.

COMPUTATION AND MATHEMATICS DEPARTMENT
RESEARCH AND DEVELOPMENT REPORT

AUGUST 1974

Report 4503
The Naval Ship Research and Development Center is a U. S. Navy center for laboratory effort directed at achieving improved sea and air vehicles. It was formed in March 1967 by merging the David Taylor Model Basin at Carderock, Maryland with the Marine Engineering Laboratory at Annapolis, Maryland.

Naval Ship Research and Development Center
Bethesda, Md. 20034

MAJOR NSRDC ORGANIZATIONAL COMPONENTS

*REPORT ORIGINATOR

NSRDC
COMMANDER 00
TECHNICAL DIRECTOR 01

OFFICER-IN-CHARGE
CARDEROCK 05

OFFICER-IN-CHARGE
ANAPOLIS 04

SYSTEMS
DEVELOPMENT
DEPARTMENT 11

SHIP PERFORMANCE
DEPARTMENT 15

STRUCTURES
DEPARTMENT 17

AVIATION AND
SURFACE EFFECTS
DEPARTMENT 16

COMPUTATION
AND MATHEMATICS
DEPARTMENT 18

SHIP ACOUSTICS
DEPARTMENT 19

PROPULSION AND
AUXILIARY SYSTEMS
DEPARTMENT 27

MATERIALS
DEPARTMENT 28

CENTRAL
INSTRUMENTATION
DEPARTMENT 29
**Title:** Solution of a Complex Quadratic Eigenvalue Problem Related to Pipe Flow

**Author:** Donald A. Gignac

**Performing Organization:** Naval Ship Research & Development Center, Bethesda, Maryland 20034

**Report Date:** August 1974

**Number of Pages:** 45

**Security Classification:** UNCLASSIFIED

**Distribution Statement:** APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED.

**Abstract:**
A complex quadratic eigenvalue problem of order n (n = 20, 30, 40, ...) encountered in an investigation of pipe flow was reduced to that of computing the eigenvalues of either a real or a complex matrix of order 2n, depending on the linearization technique used. To get satisfactory results using the CDC 6000 series computer, double precision versions of the appropriate EISPACK subroutines were required. Although the modified subroutines used for the real matrix computations required less core than those used for the...
20. continued:

complex matrix, the calculation time was not significantly faster than that for the complex matrix. The use of "balancing" subroutines merits further investigation.
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>THE PROBLEM</td>
<td>2</td>
</tr>
<tr>
<td>THE SUBROUTINES</td>
<td>4</td>
</tr>
<tr>
<td>THE RESULTS</td>
<td>5</td>
</tr>
<tr>
<td>OBSERVATIONS AND CONCLUSIONS</td>
<td>11</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>12</td>
</tr>
<tr>
<td>APPENDIX A - PROGRAM LISTINGS</td>
<td>13</td>
</tr>
</tbody>
</table>

LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE 1</td>
<td>Second- and Third-Quadrant Eigenvalues Obtained from Complex Matrix Approach.</td>
<td>7</td>
</tr>
<tr>
<td>TABLE 2</td>
<td>Second- and Third-Quadrant Eigenvalues Obtained from Real Matrix Approach.</td>
<td>9</td>
</tr>
</tbody>
</table>
INTRODUCTION

The work described in this report was undertaken in response to a request by Dr. R.J. Hansen of the Naval Research Laboratory who was investigating a pipe flow problem in fluid mechanics\(^1\) using a computer program containing an eigenvalue subroutine for complex matrices. The unsatisfactory results previously obtained with this computer program were thought to be due to the complex eigenvalue subroutine then being used, and it was thought that substituting an appropriate sequence of subroutines from the EISPACK subroutine library\(^2\) might solve the problem. However, for certain matrices generated by the program, the EISPACK subroutines did not produce satisfactory results either.

At first it was felt that the matrices had not been generated with sufficient accuracy, but when further effort in that direction proved unproductive, it was conjectured that the source of the difficulty might lie in the fact that the matrix was generated in double precision while its eigenvalues were computed in single precision. At this point the question arises: If it is desirable or necessary to generate these matrices in double precision,\(^*\) then should not their eigenvalues be computed in double precision also?

This question should be considered in light of the following observations.


\(^*\) To accumulate dot products in double precision on the CDC 6400 so as to minimize roundoff error, the whole dot product procedure must be done in double precision, costly though that may be.
First, under certain circumstances even the 48-bit mantissa of the single-precision word of the CDC 6400 may not be sufficient to withstand the accumulation of round-off error in the course of computing the eigenvalues of a complex matrix of rather low order.

Second, if A is the complex matrix generated in double precision and B is the complex matrix obtained by truncating A to single precision, the respective elements of A and B differ by increments of order 10^{-16}. Clearly either matrix can be regarded as having been obtained by perturbing the elements of the other. The crucial question is whether this perturbation significantly changes the eigenvalues, or even alters their very nature?

This report documents an investigation of these difficulties for Hansen's complex eigenvalue problem as well as for an associated real eigenvalue problem.

THE PROBLEM

If A, B, and C are complex matrices of order n, we may define the matrix equation

$$(\lambda^2 A + \lambda B + C)X = 0$$  \hspace{1cm} (1)

as the quadratic eigenvalue problem of order n. If A is invertible, the solution of this problem is equivalent to computing the eigenvalues and eigenvectors of a complex matrix of order 2n.

Two methods can be used to reduce this quadratic eigenvalue problem to a matrix eigenvalue problem. First the substitution of $Y = XX$ into the above equation results in the linear eigenvalue problem of order 2n.

$$\begin{bmatrix} -\lambda I - A & -B \\ -A^{-1}C & -A^{-1}B \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} = \lambda \begin{bmatrix} X \\ Y \end{bmatrix} .$$  \hspace{1cm} (2)

The second method involves multiplying the quadratic problem by $A^{-1}$ and completing the square to obtain

$$[(\lambda I + \frac{1}{2} A^{-1}B)^2 - (\frac{1}{2} A^{-1}B)^2 + A^{-1}C]X = 0 .$$  \hspace{1cm} (3)
Next, \( Y \) is implicitly defined by the relationship

\[
(\lambda I + \frac{1}{2} A^{-1}B)X = A^{-1}Y.
\]  

(4)

Then this relationship is substituted in Equation (3) and multiplication by \( A \) and simplification gives the linear eigenvalue problem

\[
\begin{pmatrix}
\frac{1}{2} A^{-1}B & A^{-1} \\
\frac{1}{4} BA^{-1}B - C & -\frac{1}{2} BA^{-1}
\end{pmatrix}
\begin{pmatrix}
X \\
Y
\end{pmatrix}
= \lambda
\begin{pmatrix}
X \\
Y
\end{pmatrix}.
\]

The problem at hand is of the form

\[
[\lambda^2 P + (2\theta E + \frac{1}{\theta} Pi) + (-J + Ei)] X = 0
\]

where \( P, E, \) and \( J \) are real matrices of order 2\( \theta \) or more, \( \theta \) is a real parameter, and \( i \) is the square root of \(-1\). The first linearization procedure results in the complex eigenvalue problem

\[
\begin{pmatrix}
0 & I \\
p^{-1}J - p^{-1}Ei & 2\theta p^{-1}E + \frac{1}{\theta} Ii
\end{pmatrix}
\begin{pmatrix}
X \\
Y
\end{pmatrix}
= \lambda
\begin{pmatrix}
X \\
Y
\end{pmatrix}.
\]

However, the second linearization procedure produces the complex eigenvalue problem

\[
\begin{pmatrix}
-\theta p^{-1}E & -\frac{1}{2\theta} Ii \\
\theta^2 EP^{-1}E - \frac{1}{4\theta^2} P + J & -\theta EP^{-1} - \frac{1}{2\theta} Ii
\end{pmatrix}
\begin{pmatrix}
X \\
Y
\end{pmatrix}
= \lambda
\begin{pmatrix}
X \\
Y
\end{pmatrix}.
\]

In this instance the coefficient matrix is the difference of a real non-symmetric matrix and a complex scalar multiple of the identity matrix. This coefficient matrix \( C \) is similar to a matrix of simpler form. If we let \( D \) be the matrix
\[
\begin{pmatrix}
- \mathbf{I} & 0 \\
- \delta \mathbf{E} & \mathbf{I}
\end{pmatrix}
\]

then the inverse of \( \mathbf{D} \) is

\[
\begin{pmatrix}
- \mathbf{I} & 0 \\
- \delta \mathbf{E} & \mathbf{I}
\end{pmatrix}
\]

and \( \mathbf{D}^{-1}\mathbf{C}\mathbf{D} \) is

\[
\begin{pmatrix}
-2\delta \mathbf{P}^{-1}\mathbf{E} & \mathbf{P}^{-1} \\
- \frac{1}{4\delta^2} \mathbf{P} + \mathbf{J} & 0
\end{pmatrix} - \frac{1}{2\delta} \mathbf{I}i
\]

Thus we may obtain the eigenvalues of the complex matrix \( \mathbf{C} \) by computing the eigenvalues of the real non-symmetric matrix

\[
\begin{pmatrix}
-2\delta \mathbf{P}^{-1}\mathbf{E} & \mathbf{P}^{-1} \\
- \frac{1}{4\delta^2} \mathbf{P} + \mathbf{J} & 0
\end{pmatrix}
\]

and subtracting the quantity \( \frac{1}{2\delta} \) from their imaginary parts.

THE SUBROUTINES

All the single-precision subroutines (CBAL, COMHES, COMLR, BALANC, ELMHES, ORTHES, HQR) are the standard EISPACK subroutines. They contain instructions pertaining to their calling sequence. The double-precision subroutines (DBAL, DOMHES, DOMLR, DALANC, DLMHES, DRTHES, DQR) were obtained from the corresponding EISPACK subroutines by making the following changes:

- All single-precision variables were changed to double precision.
- The single-precision criterion for the CDC 6400, \( 2^{-47} \), was changed to the double-precision criterion, \( 2^{-95} \).
The various library subroutines (ABS, SQRT, etc.) were changed to their double-precision counterparts. In the case of DOMLR, it was necessary to supply double-precision subroutines for complex division and complex square-root, since these operations are not available on the CDC 6400. The subroutines provided (CLXDVDE and CLXSQRT) are merely FORTRAN translations of the ALGOL procedures CABS, CDIV, and CSQRT of Martin and Wilkinson.\(^3\)

- The argument lists for these double-precision subroutines differ from the calling sequences for their EISPACK counterparts only in that all non-integer arrays were double precision instead of single precision.

**THE RESULTS**

The eigenvalues of the complex matrix provided by the first linearization technique were obtained using the following procedure*:

- The complex matrix was balanced.
- The balanced matrix was reduced to Hessenberg form by stabilized elementary similarity transformations.
- The eigenvalues of the Hessenberg matrix were computed using the LR algorithm for complex matrices.

This procedure was carried out in three different ways:

1) The complex matrix generated in double precision was truncated to single precision and the single-precision EISPACK subroutines CBAL, COMHES, COMLR were used (Table 1a).

2) The complex matrix generated in double precision was truncated to single precision and the double-precision versions DBAL, DOMHES, DOMLR

---


of the EISPACK subroutines were used (Table 1b).

3) DBAL, DOMHES, DOMLR were used on the complex matrix generated in double precision (Table 1c).

The considerable simplification of the eigenvalue problem achieved by using the second linearization technique was not anticipated at the outset of this investigation. When a detailed examination of the eigenvalues computed from the matrix of the first linearization technique suggested that a different linearization technique might yield a matrix more amenable to computation, other linearization techniques were investigated.

The eigenvalues of the real matrix provided by the second linearization technique were obtained using the following procedure:

- The real non-symmetric matrix was balanced.
- The balanced matrix was reduced to Hessenberg form using either stabilized elementary similarity transformations or orthogonal transformations.
- The eigenvalues of the Hessenberg matrix were computed using the QR algorithm.

This procedure was carried out in four different ways.

1) The real matrix generated in double precision was truncated to single precision and the single precision EISPACK subroutines BALANC, ORTHES, and HQR were used (Table 2a).

2) The real matrix generated in double precision was truncated to single precision and the double precision versions DALANC, DRTHES, and DQR of the EISPACK subroutines were used (Table 2b).

3) DALANCE, DRTHES, and DQR were used on the real matrix generated in double precision (Table 2c).

4) DRTHES and DQR were used on the real matrix generated in double precision (Table 2d).

Of the 60 eigenvalues computed, only those that fall in the second and third quadrants are presented in the tables that follow. However, their behavior is representative of the eigenvalues as a whole. The computation times shown are those required to compute all 60 of the eigenvalues.
TABLE 1 - SECOND AND THIRD QUADRANT EIGENVALUES
OBTAINED FROM COMPLEX MATRIX APPROACH

Table 1a
CBAL, COMHES, COMLR Subroutines 21.77 Seconds

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1.66895348951915E+02</td>
<td>0.453962332056731E+02</td>
</tr>
<tr>
<td>2</td>
<td>-3.16749508523681E+02</td>
<td>0.248351140020612E+00</td>
</tr>
<tr>
<td>3</td>
<td>-3.4888408302488E+02</td>
<td>0.221199870220845E+02</td>
</tr>
<tr>
<td>1</td>
<td>-8.16293526043538E+00</td>
<td>-1.358111109722657E+03</td>
</tr>
<tr>
<td>2</td>
<td>-8.136399999160820E+00</td>
<td>-7.04598564725760E+01</td>
</tr>
<tr>
<td>3</td>
<td>-2.6251036877122E+01</td>
<td>-1.3447760343908E+03</td>
</tr>
<tr>
<td>4</td>
<td>-2.62511208539763E+01</td>
<td>-8.37954276808919E+01</td>
</tr>
<tr>
<td>5</td>
<td>-5.7054636332099E+01</td>
<td>-1.328076034536E+03</td>
</tr>
<tr>
<td>6</td>
<td>-5.7054980976807E+01</td>
<td>-1.0049417473950E+02</td>
</tr>
<tr>
<td>7</td>
<td>-1.1227054905432E+02</td>
<td>-1.3145972874669E+03</td>
</tr>
<tr>
<td>8</td>
<td>-1.1227092516163E+02</td>
<td>-1.14021920263506E+02</td>
</tr>
<tr>
<td>9</td>
<td>-1.6689417380842E+02</td>
<td>-1.88253246825858E+03</td>
</tr>
<tr>
<td>10</td>
<td>-2.0537737392610E+02</td>
<td>-1.3302794525257E+03</td>
</tr>
<tr>
<td>11</td>
<td>-2.0537768621726E+02</td>
<td>-0.98292388380672E+01</td>
</tr>
<tr>
<td>12</td>
<td>-3.1674879144651E+02</td>
<td>-1.43105511121843E+03</td>
</tr>
<tr>
<td>13</td>
<td>-3.4888304538186E+02</td>
<td>-0.164977092569118E+03</td>
</tr>
</tbody>
</table>

Table 1b
DBAL, DOMHES, DOMLR Subroutines 96.19 Seconds
(Matrix has been truncated to single precision significance)

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1.64163526301686D+02</td>
<td>0.444157924738040D+02</td>
</tr>
<tr>
<td>2</td>
<td>-3.09194743856593D+02</td>
<td>0.79511187632182D-01</td>
</tr>
<tr>
<td>3</td>
<td>-3.42104294907155D+02</td>
<td>0.214986849517767D+02</td>
</tr>
<tr>
<td>1</td>
<td>-4.91817227667278D+00</td>
<td>-1.36123802804776D+03</td>
</tr>
<tr>
<td>2</td>
<td>-4.91817235024852D+00</td>
<td>-6.73334031666884D+01</td>
</tr>
<tr>
<td>3</td>
<td>-2.36178882980058D+01</td>
<td>-8.17847614039992D+01</td>
</tr>
<tr>
<td>4</td>
<td>-2.36178886587554D+01</td>
<td>-1.3467866891774D+03</td>
</tr>
<tr>
<td>5</td>
<td>-5.44207538548246D+01</td>
<td>-0.983990980218899D+01</td>
</tr>
<tr>
<td>6</td>
<td>-5.44207544558938D+01</td>
<td>-1.33017233233870D+03</td>
</tr>
<tr>
<td>7</td>
<td>-1.08592821202510D+02</td>
<td>-1.11690142642918D+02</td>
</tr>
<tr>
<td>8</td>
<td>-1.08592821807477D+02</td>
<td>-1.31688128825279D+03</td>
</tr>
<tr>
<td>9</td>
<td>-1.64163521583046D+02</td>
<td>-0.187272935438159D+03</td>
</tr>
<tr>
<td>10</td>
<td>-1.99544704290707D+00</td>
<td>-1.33174702838537D+03</td>
</tr>
<tr>
<td>11</td>
<td>-1.99544704372173D+00</td>
<td>-0.968244033591238D+03</td>
</tr>
<tr>
<td>12</td>
<td>-3.09194742208070D+02</td>
<td>-0.142936654328604D+03</td>
</tr>
<tr>
<td>13</td>
<td>-3.42104291482651D+02</td>
<td>-0.164355828892842D+03</td>
</tr>
</tbody>
</table>
TABLE 1 (CONTINUED)

<table>
<thead>
<tr>
<th></th>
<th>DBAL, DOMHES, DOMLR Subroutines</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-.161635728329553D+02</td>
<td>.432950045889481D+02</td>
</tr>
<tr>
<td>2</td>
<td>-.335577057717484D+02</td>
<td>.207044804956883D+02</td>
</tr>
<tr>
<td>1</td>
<td>-.204921408502598D+00</td>
<td>-.685273791773558D+01</td>
</tr>
<tr>
<td>2</td>
<td>-.204921408502598D+00</td>
<td>-.136004404939408D+03</td>
</tr>
<tr>
<td>3</td>
<td>-.208873186667484D+01</td>
<td>-.134653761423907D+03</td>
</tr>
<tr>
<td>4</td>
<td>-.208873186667484D+01</td>
<td>-.820338143323628D+01</td>
</tr>
<tr>
<td>5</td>
<td>-.512410079791166D+01</td>
<td>-.133057379553590D+03</td>
</tr>
<tr>
<td>6</td>
<td>-.512410079791166D+01</td>
<td>-.979976330355343D+01</td>
</tr>
<tr>
<td>7</td>
<td>-.104158704992893D+02</td>
<td>-.13175172309115D+03</td>
</tr>
<tr>
<td>8</td>
<td>-.104158704992893D+02</td>
<td>-.111054185480280D+02</td>
</tr>
<tr>
<td>9</td>
<td>-.161635728329553D+02</td>
<td>-.186152147464091D+03</td>
</tr>
<tr>
<td>10</td>
<td>-.193106276285779D+02</td>
<td>-.133126188924393D+03</td>
</tr>
<tr>
<td>11</td>
<td>-.193106276285779D+02</td>
<td>-.9730953932749970+01</td>
</tr>
<tr>
<td>12</td>
<td>-.301520647479031D+02</td>
<td>-.14256568129680D+03</td>
</tr>
<tr>
<td>13</td>
<td>-.301520647479031D+02</td>
<td>-.291474727463061D+00</td>
</tr>
<tr>
<td>14</td>
<td>-.335577057717484D+02</td>
<td>-.163561623352832D+03</td>
</tr>
<tr>
<td></td>
<td>BALANC, ORTHES, HQR Subroutines</td>
<td>DALANC, DRTHES, DQR Subroutines</td>
</tr>
<tr>
<td>-------</td>
<td>-------------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>1</td>
<td>-160756150689513E+02</td>
<td>161637993977430D+02</td>
</tr>
<tr>
<td>2</td>
<td>-335103274930459E+02</td>
<td>335103274930459E+02</td>
</tr>
<tr>
<td>3</td>
<td>-174021213374999E+00</td>
<td>-174021213374999E+00</td>
</tr>
<tr>
<td>4</td>
<td>-205661137487414E+01</td>
<td>-205661137487414E+01</td>
</tr>
<tr>
<td>5</td>
<td>-508934051840720E+01</td>
<td>-508934051840720E+01</td>
</tr>
<tr>
<td>6</td>
<td>-103794737355472E+01</td>
<td>-103794737355472E+01</td>
</tr>
<tr>
<td>7</td>
<td>-160756150689513E+01</td>
<td>-160756150689513E+01</td>
</tr>
<tr>
<td>8</td>
<td>-192775922492228E+01</td>
<td>-192775922492228E+01</td>
</tr>
<tr>
<td>9</td>
<td>-301234424853915E+02</td>
<td>-301234424853915E+02</td>
</tr>
<tr>
<td>10</td>
<td>-335103274930459E+02</td>
<td>-335103274930459E+02</td>
</tr>
</tbody>
</table>

Table 2a

Table 2b

(Matrix has been truncated to single precision significance)
### Table 2 (continued)

#### Table 2c

<table>
<thead>
<tr>
<th></th>
<th>DALANCE, DRTTHES, DQR Subroutines</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>81.25 Seconds</td>
</tr>
<tr>
<td>1</td>
<td>-.161635728329553D+02</td>
<td>.432950045889481D+02</td>
</tr>
<tr>
<td>2</td>
<td>-.3355770577717484D+02</td>
<td>.207044804956883D+02</td>
</tr>
<tr>
<td>2</td>
<td>-.204921408502594D+00</td>
<td>-.685273791773559D+01</td>
</tr>
<tr>
<td>3</td>
<td>-.204921408502594D+00</td>
<td>-.136004404939408D+03</td>
</tr>
<tr>
<td>4</td>
<td>-.208873186667484D+01</td>
<td>-.820338143323628D+01</td>
</tr>
<tr>
<td>5</td>
<td>-.208873186667484D+01</td>
<td>-.134653761423907D+03</td>
</tr>
<tr>
<td>6</td>
<td>-.512410079791165D+01</td>
<td>-.979976330355343D+01</td>
</tr>
<tr>
<td>7</td>
<td>-.512410079791165D+01</td>
<td>-.13305737955590D+03</td>
</tr>
<tr>
<td>8</td>
<td>-.104158704992893D+02</td>
<td>-.111054185480280D+02</td>
</tr>
<tr>
<td>9</td>
<td>-.104158704992893D+02</td>
<td>-.131751724309115D+03</td>
</tr>
<tr>
<td>10</td>
<td>-.161635728329553D+02</td>
<td>-.186152147446091D+03</td>
</tr>
<tr>
<td>11</td>
<td>-.161635728329553D+02</td>
<td>-.133126188924393D+03</td>
</tr>
<tr>
<td>12</td>
<td>-.193106276285779D+02</td>
<td>-.291474727463069D+00</td>
</tr>
<tr>
<td>13</td>
<td>-.193106276285779D+02</td>
<td>-.142565668129680D+03</td>
</tr>
<tr>
<td>14</td>
<td>-.335577057717484D+02</td>
<td>-.163561623352832D+03</td>
</tr>
</tbody>
</table>

#### Table 2d

|        | DRTTHES, DQR Subroutines          | Time          |
|        |                                   | 76.27 Seconds |
|        | (No balancing)                    |               |
| 1      | -.161635728329026D+02             | .432950045889481D+02 |
| 2      | -.335577057717281D+02             | .207044804956883D+02 |
| 1      | -.204921408518517D+00             | -.685273791773917D+01 |
| 2      | -.204921408518517D+00             | -.136004404939404D+03 |
| 3      | -.208873186669188D+01             | -.820338143324120D+01 |
| 4      | -.208873186669188D+01             | -.134653761423902D+03 |
| 5      | -.512410079793201D+01             | -.979976330356033D+01 |
| 6      | -.512410079793201D+01             | -.133057379555833D+03 |
| 7      | -.104158704993175D+02             | -.111054185480356D+02 |
| 8      | -.104158704993175D+02             | -.131751724309108D+03 |
| 9      | -.161635728329026D+02             | -.186152147446035D+03 |
| 10     | -.193106276286219D+02             | -.973095393274980D+01 |
| 11     | -.193106276286219D+02             | -.133126188924393D+03 |
| 12     | -.301520647479031D+02             | -.291474727463069D+00 |
| 13     | -.301520647479031D+02             | -.142565668129680D+03 |
| 14     | -.335577057717281D+02             | -.163561623352906D+03 |
OBSERVATIONS AND CONCLUSIONS

Study of the preceding tables led to the following observations:

1) The situation as regards the complex matrix requires double-precision eigenvalue computation on the CDC 6400 computer, as indicated by a comparison of the eigenvalues in Tables 1a and 1c, realizing that
   i) the real parts of the eigenvalues must be equal in pairs, and
   ii) the sequence of double-precision subroutines for the complex LR algorithm obtains eigenvalues satisfying i) by iterating directly for each root and not by some deflation procedure which involves solving a quadratic equation, a procedure which could provide equal real parts.

   The situation for the real matrix is similar. The eigenvalues in Tables 2a and 2c agree to only a few decimal places at best, but the eigenvalues in Tables 1c and 2c are virtually identical. However, the eigenvalues in Table 2c, unlike those in Table 1c, appear to be the result of a series of quadratic deflations. Unfortunately, the double-precision computations take more than four times the time required for the single-precision computations.

2) Since the double-precision subroutines are assumed to be more accurate than the corresponding single-precision subroutines, Tables 1b and 1c and Tables 2b and 2c indicate that the original matrix should be generated and input to the eigenvalue subroutines in double precision. Since the 'b' and 'c' eigenvalues are computed using the same double-precision subroutines, the substantial discrepancies between the 'b' and 'c' results can only be explained by the sensitivity of the eigenvalues. Since the 'c' eigenvalues meet the pairwise equality condition on their real parts, the double-precision matrix is the preferred input.

3) The situation with respect to the use of balancing subroutines is not clear. Comparison of Tables 2c and 2d indicates that the use of DALANC, the double-precision balancing subroutine for real matrices, is desirable. However, in one instance the use of DBAL, the double-precision balancing matrix for complex matrices, made no appreciable difference. The use of
balancing subroutines is therefore recommended as a matter of policy, although the value of DBAL appears open to question and to merit further investigation.

4) ELMHES and ORTHES, the Hessenberg reduction subroutines for real matrices, and their double precision counterparts DLMHES and DRTHES give comparable accuracy and require similar computation times respectively.

ACKNOWLEDGMENTS

The author wishes to thank Dr. Elizabeth H. Cuthill (1805) for her invaluable assistance and Dr. R.J. Hansen of the Naval Research Laboratory and Professor Matthew Kelleher of the Naval Postgraduate School for their interest and encouragement.
NOTE: It is to be emphasized that the subroutines listed herein are solely the present author's responsibility and were obtained by the modifications described in this report. The comment cards are those of the original single-precision EISPACK subroutines.
SUBROUTINE DBAL(NM,N,AR,AL,LOW,IGH,SCALE)

INTEGER I,J,K,L,M,N,JJ,NM,IGH,LOW,IEEXC
REAL AR(NM,N),AL(NM,N),SCALE(N)
REAL C,F,G,R,S,82,RADIX
DOUBLE PRECISION AR(NM,N)
REAL ABS
LOGICAL NOCONV

THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE
CBALANCE, WHICH IS A COMPLEX VERSION OF BALANCE,
NUM. MATH. 13, 293-304(1969) BY PARLETT AND REINSCH.

THIS SUBROUTINE BALANCES A COMPLEX MATRIX AND ISOLATES
EIGENVALUES WHENEVER POSSIBLE.

ON INPUT-
NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL
ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM
DIMENSION STATEMENT,
N IS THE ORDER OF THE MATRIX,
AR AND AL CONTAIN THE REAL AND IMAGINARY PARTS,
RESPECTIVELY, OF THE COMPLEX MATRIX TO BE BALANCED.

ON OUTPUT-
AR AND AL CONTAIN THE REAL AND IMAGINARY PARTS,
RESPECTIVELY, OF THE BALANCED MATRIX,
LOW AND IGH ARE TWO INTEGERS SUCH THAT AR(I,J) AND AL(I,J)
ARE EQUAL TO ZERO IF
(1) I IS GREATER THAN J AND
(2) J=1,...,LOW-1 OR I=IGH+1,...,N,
SCALE CONTAINS INFORMATION DETERMINING THE
PERMUTATIONS AND SCALING FACTORS USED.

SUPPOSE THAT THE PRINCIPAL SUBMATRIX IN ROWS LOW THROUGH IGH
HAS BEEN BALANCED, THAT P(J) DENOTES THE INDEX INTERCHANGED
WITH J DURING THE PERMUTATION STEP, AND THAT THE ELEMENTS
OF THE DIAGONAL MATRIX USED ARE DENOTED BY D(I,J). THEN
SCALE(J) = P(J), FOR J = 1,...,LOW-1
= D(J,J), J = LOW,...,IGH
= P(J), J = IGH+1,...,N.

THE ORDER IN WHICH THE INTERCHANGES ARE MADE IS N TO IGH+1,
THEN 1 TO LOW-1.

NOTE THAT 1 IS RETURNED FOR IGH IF IGH IS ZERO FORMALLY.
THE ALGOL PROCEDURE EXC CONTAINED IN CBALANCE APPEARS IN CBAL IN LINE. (NOTE THAT THE ALGOL ROLES OF IDENTIFIERS K, L HAVE BEEN REVERSED.)

ARITHMETIC IS REAL THROUGHOUT.

QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW, APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY

THE ALGOL PROCEDURE EXC CONTAINED IN CBALANCE APPEARS IN CBAL IN LINE. (NOTE THAT THE ALGOL ROLES OF IDENTIFIERS K, L HAVE BEEN REVERSED.)

ARITHMETIC IS REAL THROUGHOUT.

QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW, APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY

******* RADIX IS A MACHINE DEPENDENT PARAMETER SPECIFYING THE BASE OF THE MACHINE FLOATING POINT REPRESENTATION. 

RADIX = 2.

RADIX = 2.

******** FOR J=L STEP -1 UNTIL 1 DO -- *********

CONTINUE

GO TO (80,130), IEXC

******** SEARCH FOR ROWS ISOLATING AN EIGENVALUE AND PUSH THEM DOWN *********

80 IF (L .EQ. 1) GO TO 280

L = L - 1

80 IF (L .EQ. 1) GO TO 280

L = L - 1

******** FOR J=L STEP -1 UNTIL 1 DO -- *********

100 DO 120 JJ = I, L

J = L + 1 - JJ

DO 110 I = 1, L
IF (I .EQ. J) GO TO 110
   IF (AR(I,J) .NE. 0.0 .OR. AI(I,J) .NE. 0.0) GO TO 120
110  CONTINUE
C
   M = L
   IEXC = 1
   GO TO 20
120  CONTINUE
C
   GO TO 140
C
   ********** SEARCH FOR COLUMNS ISOLATING AN EIGENVALUE
   AND PUSH THEM LEFT **********
C
130  K = K + 1
C
140  DO 170 J = K, L
C
   DO 150 I = K, L
      IF (I .EQ. J) GO TO 150
      IF (AR(I,J) .NE. 0.0 .OR. AI(I,J) .NE. 0.0) GO TO 170
150  CONTINUE
C
   M = K
   IEXC = 2
   GO TO 20
170  CONTINUE
C
   ********** NOW BALANCE THE SUBMATRIX IN ROWS K TO L **********
C
180  SCLAE(I) = 1.0
C
   ********** ITERATIVE LOOP FOR NORM REDUCTION **********
190  NOCONV = .FALSE.
C
   DO 270 I = K, L
      C = 0.0
      R = 0.0
C
   DO 200 J = K, L
C
   IF (J .EQ. I) GO TO 200
C
   * C = C + ABS(AR(J,I)) + ABS(AI(J,I))
   * R = R + ABS(AR(I,J)) + ABS(AI(I,J))
   C = C + DABS(AR(J,I)) + DABS(AI(J,I))
   R = R + DABS(AR(I,J)) + DABS(AI(I,J))
C
200  CONTINUE
C
   G = R / RADIX
   F = 1.0
   S = C + R
210  IF (C .GE. G) GO TO 220
C
   F = F * RADIX
   C = C * B2
   GO TO 210
220  G = R * RADIX
230  IF (C .LT. G) GO TO 240
C
   F = F / RADIX
   C = C / B2
   GO TO 230
C  ********** NOW BALANCE **********
240  IF ((C + R) / F .GE. 0.95 * S) GO TO 270
     G = 1.0 / F
     SCALE(I) = SCALE(I) * F
     NOCONV = *TRUE.*
C
     DO 250 J = K, N
          AR(I,J) = AR(I,J) * G
          AI(I,J) = AI(I,J) * G
     250    CONTINUE
C
     DO 260 J = 1, L
          AR(J,I) = AR(J,I) * F
          AI(J,I) = AI(J,I) * F
     260    CONTINUE
C
     270    CONTINUE
C
     IF (NOCONV) GO TO 190
C
     280    LOW = K
            IGH = L
            RETURN
C  ********** LAST CARD OF CBAL **********
END
SUBROUTINE DOMHES(NM,N,LOW,IGH,AR,AI,INT)

INTEGER I,J,N,LA,N,IGH,KP1,LOW,M1,MP1
REAL AR(NM,N),AI(NM,N)
REAL XR,XI,YR,YI
DOUBLE PRECISION AR(NM,N),AI(NM,N),XR,XI,YR,YI
REAL ABS
INTEGER INT(IGH)
COMPLEX XY
REAL TI(2),T2(2)

EQUIVALENCE (X,TI(I),XR),(TI(2),XI),(YT2(1),YR),(T2(2),YI)

THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE COMHES,
NUM. MATH. 12, 349-368(1968) BY MARTIN AND WILKINSON.

GIVEN A COMPLEX GENERAL MATRIX, THIS SUBROUTINE
REDUCES A SUBMATRIX SITUATED IN ROWS AND COLUMNS
LOW THROUGH IGH TO UPPER HESSENBERG FORM BY
STABILIZED ELEMENTARY SIMILARITY TRANSFORMATIONS.

ON INPUT-
NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL
ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM
DIMENSION STATEMENT,
N IS THE ORDER OF THE MATRIX,
LOW AND IGH ARE INTEGERS DETERMINED BY THE BALANCING
SUBROUTINE CBAL. IF CBAL HAS NOT BEEN USED,
SET LOW=1, IGH=N,

AR AND AI CONTAIN THE REAL AND IMAGINARY PARTS,
RESPECTIVELY, OF THE COMPLEX INPUT MATRIX.

ON OUTPUT-
AR AND AI CONTAIN THE REAL AND IMAGINARY PARTS,
RESPECTIVELY, OF THE HESSENBERG MATRIX, THE
MULTIPLIERS WHICH WERE USED IN THE REDUCTION
ARE STORED IN THE REMAINING TRIANGLES UNDER THE
HESSENBERG MATRIX,

INT CONTAINS INFORMATION ON THE ROWS AND COLUMNS
INTERCHANGED IN THE REDUCTION.
ONLY ELEMENTS LOW THROUGH IGH ARE USED.

ARITHMETIC IS REAL EXCEPT FOR THE REPLACEMENT OF THE ALGOL
PROCEDURE CDIV BY COMPLEX DIVISION.
QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,
C

APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY

---------- ----------------------

LA = IGH - 1
KPi = LOW + 1
IF (LA .LT. KPi) GO TO 200

DO 180 M = KPi, LA
    MM1 = M - 1
    XR = 0.0
    XI = 0.0
    I = M

    DO 100 J = M, MM1
        X = (ABS(AR(J,MM1)) + ABS(AI(J,MM1))) .LE. ABS(XR) + ABS(XI)
        IF (X .LE. 1.D0) GO TO 100
        IF (ABS(AR(J,MM1)) + DABS(AI(J,MM1)) .LE. DABS(XR) + DABS(XI)) GO TO 100
        XR = AR(J,MM1)
        XI = AI(J,MM1)
        I = J
    100 CONTINUE

INT(M) = I
IF (I .EQ. M) GO TO 130

*********** INTERCHANGE ROWS AND COLUMNS OF AR AND AI ***********

DO 110 J = MM1, N
    YR = AR(I, J)
    AR(I, J) = AR(M, J)
    AR(M, J) = YR
    YI = AI(I, J)
    AI(I, J) = AI(M, J)
    AI(M, J) = YI

110 CONTINUE

DO 120 J = 1, IGH
    YR = AR(J, I)
    AR(J, I) = AR(J, M)
    AR(J, M) = YR
    YI = AI(J, I)
    AI(J, I) = AI(J, M)
    AI(J, M) = YI

120 CONTINUE

*********** END INTERCHANGE ***********

130 IF (XR .EQ. 0.0 .AND. XI .EQ. 0.0) GO TO 100

MP1 = M + 1

DO 160 I = MP1, IGH
    YR = AR(I, MM1)
    YI = AI(I, MM1)
    IF (YR .EQ. 0.0 .AND. YI .EQ. 0.0) GO TO 160
    Y = Y / X
    CALL CLXDVDE(YR, YI, YR, YI, XR, XI)
    AR(I, MM1) = YR

160 CONTINUE
AI(I,MM1) = YI

DO 140 J = M, N
   AR(I,J) = AR(I,J) - YR * AR(M,J) + YI * AI(M,J)
   AI(I,J) = AI(I,J) - YR * AI(M,J) - YI * AR(M,J)
140 CONTINUE

DO 150 J = 1, IGH
   AR(J,M) = AR(J,M) + YR * AR(J,I) - YI * AI(J,I)
   AI(J,M) = AI(J,M) + YR * AI(J,I) + YI * AR(J,I)
150 CONTINUE

CONTINUE

CONTINUE

CONTINUE

RETURN

*********** LAST CARD OF COMHES ***********

END
SUBROUTINE DOHLR(NMNLOW,IGH,HR,HI,WR,WI,IERR)

INTEGER I,J,L,M,N,ENLLMMNMIGHIMI,ITS,LOW,MP1,ENM1,IERR
REAL HR(NMN),HI(NMN),WR(NMN),WIN(N)
REAL SI,SR,ITI,TR,XI,XR,YI,YR,ZZI,ZZR,MACHEP
DOUBLE PRECISION HR(NMN),HI(NMN),WR(NMN),WIN(N),SI,SR,ITI,TR,XI,XR,
             YI,YR,ZZI,ZZR,MACHEP,ZZ,ZZI

EQUIVALENCE (ZR,ZZR),(ZI,ZZI)
REAL ABS
COMPLEX X,Y,Z
COMPLEX CSQRT,CSMPLX
REAL Ti(2),T2(2),T3(2)

EQUIVALENCE (X,T1(1),XR),(T1(2),XI),(Y,T2(1),YW),(T2(2),YI),
             (Z,T3(1),ZZR),(T3(2),ZZI)

C THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE COMLR,
C NUM. MATH. 12, 369-376(1968) BY MARTIN AND WILKINSON.
C HANDBOOK FOR AUTO. COMP., VOL. II-LINEAR ALGEBRA, 396-403 (1971).
C THIS SUBROUTINE FINDS THE EIGENVALUES OF A COMPLEX
C UPPER HESSENBERG MATRIX BY THE MODIFIED LR METHOD.

ON INPUT-
NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL
ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM
DIMENSION STATEMENT,
N IS THE ORDER OF THE MATRIX,
LOW AND IGH ARE INTEGERS DETERMINED BY THE BALANCING
SUBROUTINE CBAL. IF CBAL HAS NOT BEEN USED,
SET LOW=1, IGH=N,
HR AND HI CONTAIN THE REAL AND IMAGINARY PARTS,
RESPECTIVELY, OF THE COMPLEX UPPER HESSENBERG MATRIX.
THEIR LOWER TRIANGLES BELOW THE SUBDIAGONAL CONTAIN THE
MULTIPLIERS WHICH WERE USED IN THE REDUCTION BY COMHES,
IF PER-formed.

ON OUTPUT-
THE UPPER HESSENBERG PORTIONS OF HR AND HI HAVE BEEN
DESTROYED. THEREFORE, THEY MUST BE SAVED BEFORE
CALLING COMLR IF SUBSEQUENT CALCULATION OF
EIGENVECTORS IS TO BE PER-formed,
WR AND WI CONTAIN THE REAL AND IMAGINARY PARTS,
RESPECTIVELY, OF THE EIGENVALUES. IF AN ERROR
EXIT IS MADE, THE EIGENVALUES SHOULD BE CORRECT
FOR INDICES IERR+1,...,N,
IERR IS SET TO
ZERO FOR NORMAL RETURN,
J IF THE J-TH EIGENVALUE HAS NOT BEEN
DETERMINED AFTER 30 ITERATIONS.

ARITHMETIC IS REAL EXCEPT FOR THE REPLACEMENT OF THE ALGOL
PROCEDURE DIV BY COMPLEX DIVISION AND USE OF THE SUBROUTINES
CSQRT AND CMPLX IN COMPUTING COMPLEX SQUARE ROOTS.

QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,
APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY

******** MACHEP IS A MACHINE DEPENDENT PARAMETER SPECIFYING
THE RELATIVE PRECISION OF FLOATING POINT ARITHMETIC.

********

MACHEP = 2.**(-95)

IERR = 0

********** STORE ROOTS ISOLATED BY GBAL **********

180 DO 200 I = 1, N
    IF (I .GE. LOW .AND. I .LE. IGH) GO TO 200
    WR(I) = HR(I,I)
    WI(I) = HI(I,I)
200 CONTINUE

EN = IGH
TR = 0.0
TI = 0.0

********** SEARCH FOR NEXT EIGENVALUE **********

220 IF (EN .LT. LOW) GO TO 1001
    ITS = 0
    ENMi = EN - 1
    ********** LOOK FOR SINGLE SMALL SUB-DIAGONAL ELEMENT
    FOR L=EN STEP -1 UNTIL LOW -- **********

240 DO 260 LL = LOW, EN
    L = EN + LOW - LL
    IF (L .EQ. LOW) GO TO 300
    * IF (ABS(HR(L,L-1)) + ABS(HI(L,L-1)) .LE.
    * X MACHEP * (ABS(HR(L-1,L-1)) + ABS(HI(L-1,L-1))
    * X + ABS(HR(LL)) + ABS(HI(LL))) GO TO 300
    IF (DABS(HR(L,L-1)) + DABS(HI(L,L-1)) .LE.
    X MACHEP * (DABS(HR(L-1,L-1)) + DABS(HI(L-1,L-1))
    X + DABS(HR(LL)) + DABS(HI(LL))) GO TO 300

260 CONTINUE

********** FORMSHIFT **********

300 IF (L .EQ. EN) GO TO 660
    IF (ITS .EQ. 30) GO TO 1000
    IF (ITS .EQ. 10 .OR. ITS .EQ. 20) GO TO 320
    SR = HR(EN,EN)
    SI = HI(EN,EN)
    XR = HR(ENM1,EN) * HR(EN,EN1) - HI(ENM1,EN) * HI(EN,EN1)
    XI = HR(ENM1,EN) * HI(EN,EN1) + HI(ENM1,EN) * HR(EN,EN1)
IF (XR .EQ. 0.0 .AND. XI .EQ. 0.0) GO TO 340

YR = (HR(ENM1,EN1) - SR) / 2.0

YI = (HI(EN1,ENM1) - SI) / 2.0

* Z = CSQRT(CMPLX(YR**2+XI**2+XR**2,0.0*YR*YI+XI))

CALL CLSQRT(ZR,ZI,YR*YI+XR,2.0*YR*YI+XI)

* IF (YR*ZZR + XI*ZZI .LT. 0.0) Z = -Z

IF (YR*ZZR + XI*ZZI .GE. 0.0) GO TO 301

ZR = ZR
ZI = ZI

* X = X / (Y + Z)

301 CALL CLXVODE(XR,XI,XR,XI,SR+ZR,YI+ZI)

SR = SR - XR
SI = SI - XI

GO TO 340

************ FORM EXCEPTIONAL SHIFT ************

* 320 SR = ABS(HR(EN1,EN1)) + ABS(HR(EN1,EN1))

320 SR = DABS(HR(EN1,EN1)) + DABS(HR(EN1,EN1))

* SI = ABS(HI(EN1,EN1)) + ABS(HI(EN1,EN1))

SI = DABS(HI(EN1,EN1)) + DABS(HI(EN1,EN1))

C 340 DO 360 I = LOW, EN

HR(I,I) = HR(I,I) - SR
HI(I,I) = HI(I,I) - SI

360 CONTINUE

C TR = TR + SR
TI = TI + SI

ITS = ITS + 1

************ LOOK FOR TWO CONSECUTIVE SMALL

SUB-DIAGONAL ELEMENTS ************

* XR = ABS(HR(M-1,EN1)) + ABS(HI(EN1,EN1))

XR = DABS(HR(M-1,EN1)) + DABS(HI(EN1,EN1))

* YR = ABS(HI(M-1,EN1)) + ABS(HI(EN1,EN1))

YR = DABS(HI(M-1,EN1)) + DABS(HI(EN1,EN1))

ZRR = DABS(HR(M-1,EN1)) + DABS(HI(EN1,EN1))

************ FOR M=EN-1 STEP -1 UNTIL L DO -- ************

DO 380 MM = L, EN

M = EN + L - MM

IF (M .EQ. L) GO TO 420

YR = YR

* YR = ABS(HR(M,M-1)) + ABS(HI(M,M-1))

XR = DABS(HR(M,M-1)) + DABS(HI(M,M-1))

XI = ZRR

XR = ABS(HR(M-1,M-1)) + ABS(HI(M-1,M-1))

XR = DABS(HR(M-1,M-1)) + DABS(HI(M-1,M-1))

IF (YR .LE. MACHEP * ZR / YI + (ZRR + XR + XI)) GO TO 420

380 CONTINUE

************ TRIANGULAR DECOMPOSITION H=L*R ************

420 MP1 = M + 1

C 420 MP1 = M + 1

DO 520 I = MP1, EN

IM1 = I - 1

XR = HR(IM1,IM1)

XI = HI(IM1,IM1)
YR = HR(I, I1)
YI = HI(I, I1)
* IF (ABS(XR) + ABS(XI) .GE. ABS(YR) + ABS(YI)) GO TO 460
IF (DABS(XR) + DABS(XI) .GE. DABS(YR) + DABS(YI)) GO TO 460
C ********** INTERCHANGE ROWS OF HR AND HI **********
DO 440 J = 1, N
  ZZR = HR(I1, J)
  HR(I1, J) = HR(I, J)
  HR(I, J) = ZZR
  ZZI = HI(I1, J)
  HI(I1, J) = HI(I, J)
  HI(I, J) = ZZI
440 CONTINUE
C
Z = X / Y
CALL CLXDVDE(ZR, ZI, XR, XI, YR, YI)
WR(I) = 1.0
GO TO 480
* 460 Z = Y / X
C
460 CALL CLXDVDE(ZR, ZI, YR, YI, XR, XI)
480 HR(I, I1) = ZZR
HI(I, I1) = ZZI
C
DO 500 J = I, EN
  HR(I, J) = HR(I, J) - ZZR * HR(I1, J) + ZZI * HI(I1, J)
  HI(I, J) = HI(I, J) - ZZR * HI(I1, J) + ZZI * HR(I1, J)
500 CONTINUE
C
520 CONTINUE
C
********** COMPOSITION R*L=H **********
DO 640 J = M, EN
  XR = HR(J, J-1)
  XI = HI(J, J-1)
  HR(J, J-1) = 0.0
  HI(J, J-1) = 0.0
C
640 CONTINUE
C
********** INTERCHANGE COLUMNS OF HR AND HI, IF NECESSARY **********
C
IF (WR(J) .LE. 0.0) GO TO 580
580 DO 600 I = L, J
  ZZR = HR(I, J-1)
  HR(I, J-1) = HR(I, J)
  HR(I, J) = ZZR
  ZZI = HI(I, J-1)
  HI(I, J-1) = HI(I, J)
  HI(I, J) = ZZI
600 CONTINUE
C
640 CONTINUE
GO TO 240

****** A ROOT FOUND ********

660 WR(EN) = HR(EN,EN) + TR
  WI(EN) = HI(EN,EN) + TI
  EN = ENM1
  GO TO 220

****** SET ERROR -- NO CONVERGENCE TO AN EIGENVALUE AFTER 30 ITERATIONS ********

1000 IERR = EN
1001 RETURN

****** LAST CARD OF COMLR ********
END
SUBROUTINE CLXDVDE(ZR,ZI,XR,XI,YR,YI)

DOUBLE PRECISION ZR,ZI,XR,XI,YR,YI,H,QR,QI,HH

IF (DABS(YR)+DABS(YI)) 3,1,3

1 WRITE(6,2)
2 FORMAT(1H1/14(1H0/),30X,*CLXDVDE MESSAGE - YOU HAVE JUST ATTEMPTED
1 TO DIVIDE BY ZERO.*)
STOP

3 IF (DABS(YR)-DABS(YI)) 5,5,4

4 H=YI/YR
   HH=H*YI+YR
   QR=(XR+H*XI)/HH
   QI=(XI-H*XR)/HH
   ZR=QR
   ZI=QI
   RETURN

5 H=YR/YI
   HH=H*YR+YI
   QR=(H*XR+XI)/HH
   QI=(H*XI-XR)/HH
   ZR=QR
   ZI=QI
   RETURN.
END
SUBROUTINE CLXSQRT(YR,YI,XR,XI)
DOUBLE PRECISION YR,YI,XR,XI,XR1,XI1,H,DPCABS

FIRST COMPUTE DPCABS, THE ABSOLUTE VALUE OF THE RADICAND.

XR1=DBABS(XR)
XI1=DBABS(XI)

1 IF (XI1-XR1) 2,2,1
   XR1=XI1
   XI1=H
2 IF (XI1) 4,3,4
3 DPCABS=XR1
   GO TO 5
4 DPCABS=XR1*DSQRT(1.0+(XI1/XR1)*(XI1/XR1))
5 H=DSQRT((DBABS(XR)+DPCABS)/2.0)
   IF (XI) 6,7,6
6 XI=XI/(2.0*H)
7 IF (XR) 9,8,8
8 YR=H
    YI=XI
    RETURN
9 IF (XI) 11,10,10
10 YR=XI
    YI=1
    RETURN
11 YR=-XI
    YI=-H
    RETURN
END
SUBROUTINE BALANC(NY4,NA,LOW,IGH,SCALE)

INTEGER I,J,K,L,M,NJJ,NM,IGHLOW,IEXC

REAL A(NM,N),SCALE(N)

DOUBLE PRECISION A(NM,N),SCALE(N),C,F,G,R,S,B2,RADIX

REAL ABS

LOGICAL NOCONV

C THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE
C BALANCE, NUM. MATH. 13, 293-304 (1969) BY PARLETT AND
C REINSCH. HANDBOOK FOR AUTO. COMP., VOL. II-LINEAR ALGEBRA,

C THIS SUBROUTINE BALANCES A REAL MATRIX AND ISOLATES
C EIGENVALUES WHENEVER POSSIBLE.

ON INPUT-

NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL
ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM
DIMENSION STATEMENT,

N IS THE ORDER OF THE MATRIX,

A CONTAINS THE INPUT MATRIX TO BE BALANCED.

ON OUTPUT-

A CONTAINS THE BALANCED MATRIX,

LOW AND IGH ARE TWO INTEGERS SUCH THAT A(I,J)
IS EQUAL TO ZERO IF
(1) I IS GREATER THAN J AND
(2) J=1,...,LOW-1 OR I=IGH+1,...,N,

SCALE CONTAINS INFORMATION DETERMINING THE
PERMUTATIONS AND SCALING FACTORS USED.

SUPPOSE THAT THE PRINCIPAL SUBMATRIX IN ROWS LOW THROUGH IGH
HAS BEEN BALANCED, THAT PIJ DENOTES THE INDEX INTERCHANGED
WITH J DURING THE PERMUTATION STEP, AND THAT THE ELEMENTS
OF THE DIAGONAL MATRIX USED ARE DENOTED BY D(I,J). THEN
SCALE(J) = P(J), FOR J = 1,...,LOW-1
= D(J,J), J = LOW,...,IGH
= P(J), J = IGH+1,...,N.

THE ORDER IN WHICH THE INTERCHANGES ARE MADE IS N TO IGH+1,
THEN 1 TO LOW-1.

NOTE THAT 1 IS RETURNED FOR IGH IF IGH IS ZERO FORMALLY.

THE ALGOL PROCEDURE EXC CONTAINED IN BALANCE APPEARS IN

C
BALANCE IN LINE. (NOTE THAT THE ALGOL ROLES OF IDENTIFIERS HAVE BEEN REVERSED.)

QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO R. S. GARBOW,
APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY

NOTE THAT THE ALGOL ROLES OF IDENTIFIERS HAVE BEEN REVERSED.

QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO R. S. GARBOW,
APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY

************ RADIX IS A MACHINE DEPENDENT PARAMETER SPECIFYING
THE BASE OF THE MACHINE FLOATING POINT REPRESENTATION.

RADIX = 2.

82 = RADIX * RADIX
K = 1
L = N
GO TO 100

*********** IN-LINE PROCEDURE FOR ROW AND COLUMN EXCHANGE ***********

20 SCALE (M) = J
IF (J .EQ. M) GO TO 50

DO 50 I = 1, L
F = A(I,J)
A(I,J) = A(I,M)
A(I,M) = F
50 CONTINUE

DO 40 I = K, N
F = A(J,I)
A(J,I) = A(M,I)
A(M,I) = F
40 CONTINUE

GO TO (30,130), IEK

*********** SEARCH FOR ROWS ISOLATING AN EIGENVALUE ***********

80 IF (L .EQ. 1) GO TO 280
L = L - 1

*********** FOR J = L STEP -1 UNTIL 1 DO -- ***********

100 DO 120 JJ = 1, L
J = L + 1 - JJ

DO 110 I = 1, L
IF (I .EQ. J) GO TO 110
IF (A(J,I) .NE. 0.0) GO TO 120
110 CONTINUE

M = L
IEK = 1
GO TO 20

120 CONTINUE

GO TO 140
************ SEARCH FOR COLUMNS ISOLATING AN EIGENVALUE

** AND PUSH THEM LEFT **********

130 K = K + 1

C

140 DO 170 J = K, L

C

150 DO 150 I = K, L

C  IF (I, = EQ. J) GO TO 150

C  IF (A(I,J) .NE. 0.0) GO TO 170

C  CONTINUE

C

M = <

IEXC = 2

GO TO 20

C

170 CONTINUE

C  ********** NOW BALANCE THE SUBMATRIX IN ROWS K TO L **********

C  DO 180 I = K, L

C

180 SCALE(I) = 1.0

C  ********** ITERATIVE LOOP FOR NORM REDUCTION ***********

C  NOCONV = .FALSE.

C

190 DO 270 I = K, L

C  J = J + 0

C  R = 0.0

C

200 DO 200 J = K, L

C  IF (J, .EQ. I) GO TO 200

C  C = C + ABS(A(I,J))

C  C = C +ABS(A(I,J))

C  R = R + ABS(A(I,J))

C  R = R + ABS(A(I,J))

C

210 IF (C .GE. G) GO TO 220

C  F = = / RADIUS

C  S = S + R

C

220 G = R * RADIUS

C

230 IF (C .LT. G) GO TO 240

C  F = S / RADIUS

C  S = S + R

C

240 IF ((C + R) .GE. 0.95 * S) GO TO 270

C  G = 1.0 / F

C  SCALE(I) = SCALE(I) * F

C  NOCONV = .TRUE.

C

250 DO 250 J = K, N

C  A(I,J) = A(I,J) * G

C

250 DO 250 J = 1, L

C

69215109

69215110

69215111

69215112

69215113

69215114

69215115

69215116

69215117

69215118

69215119

69215120

69215121

69215122

69215123

69215124

69215125

69215126

69215127

69215128

69215129

69215130

69215131

69215132

69215133

69215134

69215135

69215136

69215137

69215138

69215139

69215140

69215141

69215142

69215143

69215144

69215145

69215146

69215147

69215148

69215149

69215150

69215151

69215152

69215153

69215154

69215155

69215156

69215157

69215158

69215159

69215160

69215161

30
260  \[ A(J,I) = A(J,I) \times F \]

C  

270 CONTINUE

C  

IF (NOCONV) GO TO 190

C  

280 LOW = K
    IGH = L
    RETJFN

C  

********** LAST CARD OF BALANC **********
END
SUBROUTINE ELMHES(NW,N,LOW,IGH,A,INT)

INTEGER I,J,M,N,LA,MM,IGH,KP1,LOW,MM1,MP1
* REAL A(N,N)
* REAL X,Y
DOUBLE PRECISION A(IM,N),X,Y

THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE ELMHES,
NUM. MATH. 12, 349-368(1968) BY MARTIN AND WILKINSON. 
GIVEN A REAL GENERAL MATRIX, THIS SUBROUTINE 
REDUCES A SUBMATRIX SITUATED IN ROWS AND COLUMNS 
LOW THROUGH IGH TO JPPER HESSENBERG FORM BY 
STABILIZED ELEMENTARY SIMILARITY TRANSFORMATIONS.

ON INPUT-
NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL 
ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM 
DIMENSION STATEMENT,
N IS THE ORDER OF THE MATRIX,
LOW AND IGH ARE INTEGERS DETERMINED BY THE BALANCING 
SUBROUTINE BALANC, IF BALANC HAS NOT BEEN USED,
SET LOW=1, IGH=N,
A CONTAINS THE INPUT MATRIX.

ON OUTPUT-
A CONTAINS THE HESSENBERG MATRIX, THE MULTIPLIERS 
WHICH WERE USED IN THE REDUCTION ARE STORED IN THE 
REMAIING TRIANGLE UNDER THE HESSENBERG MATRIX,
INT CONTAINS INFORMATION ON THE ROWS AND COLUMNS 
INTERCHANGED IN THE REDUCTION,
ONLY ELEMENTS LOW THROUGH IGH ARE USED.

QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARROW,
APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY

LA = IGH - 1
KP1 = LW + 1
IF (LA.LT. KP1) GO TO 200
DO 180 M = KP1, LA
   M1 = M - 1
   X = 0.0
   I = M
C
   DO 100 J = M1, IGH
*   IF (ABS(A(J,M1)) .LE. ABS(X)) GO TO 100
    IF (DABS(A(J,M1)) .LE. DABS(X)) GO TO 100
    X = A(J,M1)
    I = J
  100 CONTINUE
C
   INT(M) = I
   IF (I .EQ. M) GO TO 133
C
   ********** INTERCHANGE ROWS AND COLUMNS OF A **********
   DO 110 J = M1, IGH
      Y = A(I,J)
      A(I,J) = A(M,J)
      A(M,J) = Y
  110 CONTINUE
C
   DO 120 J = 1, IGH
      Y = A(J,I)
      A(J,I) = A(J,M1)
      A(J,M1) = Y
  120 CONTINUE
C
   ********** END INTERCHANGE **********
   IF (X .EQ. C.O) GO TO 180
   MP1 = M + 1
C
   DO 150 I = MP1, IGH
      Y = A(I,M1)
      IF (Y .EQ. G.0) GO TO 160
      Y = Y / X
      A(I,M1) = Y
C
   DO 140 J = M1, N
      A(I,J) = A(I,J) - Y * A(M,J)
  140 CONTINUE
C
   DO 150 J = 1, IGH
      A(J,M1) = A(J,M1) + Y * A(J,I)
  150 CONTINUE
C
  160 CONTINUE
C
  180 CONTINUE
C
  200 RETURN
C
   ********** LAST CARD OF ELMHES **********
   END
SUBROUTINE ORTHES(N,LOW,IGH,A,ORT)
SUBROUTINE DRTHE S(N1, LOWI GHA,ORT)

INTEGER I,J,M,N,II,J,JLAM,NIGH,LP,ORT
REAL A(4M,N),ORT(IGH)
REAL F,G,H,SCALE

DOUL F :RECISION A(4M,N),ORT(IGH)
F,G,HSCALE

REAL SQTABSSIGN

C THIS SUBROUTINE IS TRANSLATION
C OF THE ALGOL PROCEDURE ORTHES,
C NUM. MATH. 12, 349-368(1968)
C BY MARTIN AND WILKINSON.
C HAN)BOO< FOR AUO. 30OMP.,
C VOL.II-LINEAR ALG

C GIVEN A REAL GENERAL
C REDUCES
C ORTHOGONAL SIMI.
C ORTHOGONAL SIMI.
C ON INPUT-
C ON OUTPJT-

A CONTAINS THE INPUT MATRIX.
A CONTAINS THE HESSENBERG MATRIX. INFORMATION ABOUT
THE ORTHOGONAL TRANSFORMATIONS USED IN THE REDUCTION
IS STORED IN THE REMAINING TRIANGLE UNDER THE
HESSENBERG MATRIX.
ORT CONTAINS FURTHER INFORMATION ABOUT THE TRANSFORMATIONS.
ONLY ELEMENTS LOW THROUGH IGH ARE JSED.

QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,
APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY

LA = IGH - 1
KPI = LOW + 1
IF (LA .LT. KPI) GO TO 200
DO 180 M = KPI, LA
M = 3.0
ORT(M) = G,0
SCALE = 0.0

C********** SCALE COLUMN (ALGOL TOL THEN NOT NEEDED) **********
DO 93 I = M, IGH
   SCALE = SCALE + ABS(A(I, M-1))
90 SCALE = SCALE + DABS(A(I, M-1))
C
IF (SCALE .EQ. 0.0) GO TO 180
MP = M + IGH

C********** FOR I=IGH STEP -1 UNTIL M DO -- **********
DO 110 II = M, IGH
   I = MP - II
   ORT(I) = A(I, M-1) / SCALE
   H = H + ORT(I) * ORT(I)
100 CONTINUE
C
G = -SIGN(SQRT(M), ORT(M))
S = -DSIGN(DSQRTH), ORT(M))
H = 4 - ORT(M) * G
ORT(M) = ORT(M) - G

C********** FORM (I-U*UT)/H * A **********
DO 130 J = M, N
   F = 0.0
   F = F / H
   DO 120 II = M, IGH
      I = MP - II
      F = F + ORT(I) * A(I, J)
110 CONTINUE
C
   F = F / H
   DO 120 J = M, IGH
      I = MP - JJ
      F = F + ORT(J) * A(I, J)
140 CONTINUE
C
   F = F / H
   DO 150 J = M, IGH
      I = MP - JJ
      A(I, J) = A(I, J) - F * ORT(J)
150 CONTINUE
C
   ORT(M) = SCALE * ORT(M)
   A(M, M-1) = SCALE * G
C
180 CONTINUE
200 RETURN

********** LAST CARD OF ORTHES **********

END
C

* SUBROUTINE HOR(INM,N,LOW,IGH,H,MR,WI,IERR)
* SUBROUTINE DOP(INM,N,LOW,IGH,H,MR,WI,IERR)
C
INTEGER I,J,K,L,M,N,EN,LL,MM,NA,NH,IGH,ITS,LOW,MP2,ENM2,IERR
* REAL H(INM,N),MR(N),WI(N)
* REAL P,T,S,T,N,X,Y,ZZ,MACHEP
DOUBLE PRECISION H(INM,N),MR(N),WI(N),P,T,S,T,N,X,Y,ZZ,MACHEP
C
REAL SORT,AS,SIGN
C
INTEGER MINO
LOGICAL NOTLAS
C
THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE HQR,
NUM. MATH. 14, 219-231(1970) BY MARTIN, PETERS, AND WILKINSON.
C
THIS SUBROUTINE FINDS THE EIGENVALUES OF A REAL
UPPER HESSENBERG MATRIX BY THE QR METHOD.
C
ON INPUT-

4M MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL
ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM
DIMENSION STATEMENT,
C
4 IS THE ORDER OF THE MATRIX,
C
LOW AND IGH ARE INTEGERS DETERMINED BY THE BALANCING
SUBROUTINE BALANC. IF BALANC HAS NOT BEEN USFD,
SET LOW=I, IGH=N,
C
H CONTAINS THE UPPER HESSENBERG MATRIX. INFORMATION ABOUT
THE TRANSFORMATIONS USED IN THE REDUCTION TO HESSENBERG
FORM BY ELMHFS OR ORTHES, IF PERFORMED, IS STORED
IN THE REMAINING TRIANGLE UNDER THE HESSENBERG MATRIX.
C
ON OUTPUT-

I HAS BEEN DESTROYED. THEREFORE, IT MUST BE SAVED
BEFORE CALLING HOR IF SUBSEQUENT CALCULATION AND
BACK TRANSFORMATION OF EIGENVECTORS IS TO BE PERFORMED,
C
MR AND WI CONTAIN THE REAL AND IMAGINARY PARTS,
RESPECTIVELY, OF THE EIGENVALUES. THE EIGENVALUES
ARE UNORDERED EXCEPT THAT COMPLEX CONJUGATE PAIRS
OF VALUES APPEAR CONSECUTIVELY WITH THE EIGENVALUE
HAVING THE POSITIVE IMAGINARY PART FIRST. IF AN
ERROR EXIT IS MADE, THE EIGENVALUES SHOULD BE CORRECT
FOR INDICES IERR+1,...,N,
C
IERR IS SET TO
ZERO FOR NORMAL RETURN,
J IF THE J-TH EIGENVALUE HAS NOT BEEN

C

06215001
06215002
06215003
06215004
06215005
06215006
06215007
06215008
06215009
06215010
06215011
06215012
06215013
06215014
06215015
06215016
06215017
06215018
06215019
06215020
06215021
06215022
06215023
06215024
06215025
06215026
06215027
06215028
06215029
06215030
06215031
06215032
06215033
06215034
06215035
06215036
06215037
06215038
06215039
06215040
06215041
06215042
06215043
06215044
06215045
06215046
06215047
06215048
06215049
06215050
06215051
06215052
06215053
DETERMINED AFTER 3 ITERATIONS.

QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBO9,
APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY

------------------------------------------------------------------

*********** MACHEP IS A MACHINE DEPENDENT PARAMETER SPECIFYING
THE RELATIVE PRECISION OF FLOATING POINT ARITHMETIC.

***********

* MACHEP = 2.**(-47)
MACHEP = 2. DO**(-95)

------------------------------------------------------------------

IERR = J

************ STOP POINTS ISOLATED BY BALANC ************

DO 50 I = 1, N
   IF (I .GE. LOW .AND. I .LE. IGH) GO TO 50
   WR(I) = H(I,I)
   WI(I) = 0.0
50 CONTINUE

EN = IGH
T = 0.0

************ SEARCH FOR NEXT EIGENVALUES ************

60 IF (EN .LT. LOW) GO TO 1001
   ITS = J
   NA = EN - 1
   ENM2 = NA - 1
   FOR = -EN STEP -1 UNTIL LOW DO -- ***********

70 DO 80 LL = LCW, EN
   L = EN + LOW - L
   IF (L .EQ. LOW) GO TO 100
   IF (ARS(H(LL-1)) .LE. MACHEP * (ABS(H(LL-1)) + ABS(H(LL)))) GO TO 100
   IF (ARS(H(LL-1)) .LE. MACHEP * (DABS(H(LL-1)) + ABS(H(LL)))) GO TO 100
   CONTINUE

80 CONTINUE

************ FORM SHIFT ************

100 X = H(EN,EN)
   IF (L .EQ. EN) GO TO 270
   Y = H(NA,NA)
   W = H(EN,NA) * H(NA,EN)
   IF (L .EQ. NA) GO TO 280
   IF (ITS .EQ. 3D) GO TO 100C
   IF (ITS .NE. 1D .AND. ITS .NE. 2D) GO TO 130

************ FORM EXCEPTIONAL SHIFT ************

T = T + X

D0 120 I = LOW, EN
   H(I,I) = HII,I) - X
120 CONTINUE

S = AES(H(EN,NA)) + ABS(H(NA,ENM2))
S = JABS(H(EN,NA)) + JABS(H(NA,ENM2))

38
\[ X = 0.73 \times S \]
\[ Y = X \]
\[ W = -0.4375 \times S \times S \]

130 \text{ ITS = ITS} + 1

*********** LOOK FOR TWO CONSECUTIVE SMALL

SUB-DIAGONAL ELEMENTS.

FOR \( M = M - 2 \) STEP -1 UNTIL \( L \) DO -- ***********

DO 140 \( M = L \) \( \equiv \) \( NM2 \)
\[ 4 = \pm NM2 + L - M \]
\[ ZZ = H(M, M) \]
\[ \tau = X - ZZ \]
\[ S = Y - ZZ \]
\[ \delta = (R \times S - W) / H(M+1, M) + H(M, M+1) \]
\[ J = 4(M+1, M+1) - ZZ - \tau = S \]
\[ \tau = H(M+2, M+1) \]

\[ S = \text{ABS}(P) + \text{ABS}(Q) + \text{ABS}(R) \]
\[ S = \text{ABS}(P) + \text{ABS}(Q) + \text{ABS}(R) \]
\[ P = \delta / S \]
\[ Q = J / S \]
\[ R = S / S \]

IF \( \delta \cdot \text{EQ.} \) \( L \) \( \text{GO TO} \) 150

IF \( H(M, M-1) \times \text{ABS}(Q) \times \text{ABS}(R) \times \text{LE.} \text{MACHEP} \times \text{ABS}(P)\)

\[ X = (\text{ABS}(H(M-1, M-1)) \times \text{ABS}(ZZ) + \text{ABS}(H(M+1, M+1))) \text{GO TO} \) 150

IF \( \text{ABS}(H(M, M-1)) \times \text{ABS}(Q) \times \text{ABS}(R) \times \text{LE.} \text{MACHEP} \times \text{ABS}(P) \)

\[ X = (\text{ABS}(H(M-1, M-1)) \times \text{ABS}(ZZ) + \text{ABS}(H(M+1, M+1))) \text{GO TO} \) 150

140 CONTINUE

150 \( MP2 = M + 2 \)

DO 160 \( I = MP2, EN \)
\[ H(I, I-2) = 0.0 \]
IF \( I \cdot \text{NE.} \) \( MP2 \) \( \text{GO TO} \) 160
\[ H(I, I-3) = 0.0 \]

160 CONTINUE

*********** DOUBLE OR STEP INVOLVING ROWS \( L \) TO \( EN \) AND

COLUMNS \( M \) TO \( EN \) ***********

DO 260 \( K = M, NA \)
\[ \text{NTLS} = K \cdot \text{NE.} \ NA \]
IF \( K \cdot \text{NE.} \) \( M \) \( \text{GO TO} \) 173
\[ P = -H(K, K-1) \]
\[ Q = +H(K+1, K-1) \]
\[ R = 0.0 \]
IF \( \text{NTLS} \) \( R = H(K+2, K-1) \)

\[ X = \text{ABS}(P) + \text{ABS}(Q) + \text{ABS}(R) \]
\[ X = \text{ABS}(P) + \text{ABS}(Q) + \text{ABS}(R) \]
IF \( X \cdot \text{NE.} \) \( G.0 \) \( \text{GO TO} \) 260
\[ P = \tau / X \]
\[ Q = J / X \]
\[ R = S / X \]

IF \( P \cdot \text{NE.} \) \( M \) \( \text{GO TO} \) 180

170 \( S = \text{SIGN} \cdot \text{SORT} \cdot (P \times P + Q \times Q + R \times R), P \)
170 \( S = \text{SIGN} \cdot \text{SORT} \cdot (P \times P + Q \times Q + R \times R), P \)
IF \( K \cdot \text{NE.} \) \( M \) \( \text{GO TO} \) 180

\[ - (K, K-1) = -S \times X \]

GO TO 190

180 IF \( L \cdot \text{NE.} \) \( M \) \( H(K, K-1) = -H(K, K-1) \)

39
190  P = P + S
X = P / S
Y = Z / S
ZZ = R / S
Q = Z / P
R = R / P

C ********** Row Modification **********
30 210  J = K, EN
P = H(K, J) + J * H(K+1, J)
IF (.NOT. NOT AS) GO TO 300
P = P + R * H(K+2, J)
H(K+2, J) = H(K+2, J) - P * ZZ
260  H(K+1, J) = H(K+1, J) - P * Y
H(K, J) = H(K, J) - P * X
210  CONTINUE

C

J = IMOD(LN, K+3)

C ********** Column Modification **********
JO 230  I = L, J
P = X * H(I, K) + Y * H(I, K+1)
IF (.NOT. NOT AS) GO TO 220
P = P + ZZ * H(I, K+2)
H(I, K+2) = H(I, K+2) - P * R
H(I, K+1) = H(I, K+1) - P * Q
H(I, K) = H(I, K) - P
230  CONTINUE

C

260  CONTINUE

C

GO TO 70

C ********** One Root Found **********
270  WR(N) = X + T
WI(N) = 0. J
EN = NA
GO TO 60

C ********** Two Roots Found **********
280  P = (Y - X) / 2.0
Q = P * P + W
* ZZ = SQRT(AABS(Q))
ZZ = DSQRT(DABS(Q))
X = X + T
IF (Q .LT. 0. J) GO TO 320

C ********** Real Pair **********
C
Z2 = P + SIGN(ZZ, F)
ZZ = P + DSIGN4(ZZ, F)
WR(NA) = X + ZZ
WR(EN) = WR(NA)
IF (ZZ .NE. 0. J) WR(EN) = X - W / ZZ
WI(NA) = 0. J
WI(EN) = 3. J
GO TO 330
C

C ********** Complex Pair **********
320  WR(NA) = X + P
WR(EN) = X + P
WI(NA) = ZZ

40
WI(EN) = -ZZ

33C EN = ENM2
GO TO 6J

C ********** SET ERROR -- NO CONVERGENCE TO AN
C EIGENVALUE AFTER 30 ITERATIONS **********

1000 IERR = EN
1001 RETURN
C ********** LAST CARC OF HQR **********
END
### INITIAL DISTRIBUTION

<table>
<thead>
<tr>
<th>Copies</th>
<th>Copies:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ONR 430/R. LUNDEGARD</td>
<td>1 NOL 331/M. VANDER VORST</td>
</tr>
<tr>
<td>1 ONR 432/L. BRAM</td>
<td>1 NAVSHIPYD BREM/LIB</td>
</tr>
<tr>
<td>1 NRL/8441/J. HANSEN</td>
<td>1 NAVSHIPYD BSN/LIB</td>
</tr>
<tr>
<td>1 USNA DEPT MATH</td>
<td>1 NAVSHIPYD CHASN/LIB</td>
</tr>
<tr>
<td>1 USNA LIB</td>
<td>1 NAVSHIPYD MARE/LIB</td>
</tr>
<tr>
<td>1 NAVPGSCOL/59C1/G. CANTIN</td>
<td>1 NAVSHIPYD NORVA/LIB</td>
</tr>
<tr>
<td>1 NAVPGSCOL/M. KELLEHER</td>
<td>1 NAVSHIPYD PEARL/LIB</td>
</tr>
<tr>
<td>1 NAVPGSCOL/MATH DEPT</td>
<td>1 NAVSHIPYD PHILA/LIB</td>
</tr>
<tr>
<td>1 NAVPGSCOL/LIB</td>
<td>1 NAVSHIPYD PTSMH/LIB</td>
</tr>
<tr>
<td>1 NAVWARCOL</td>
<td>1 AIR FORCE AERO RES LABS/P. NIKOLAI</td>
</tr>
<tr>
<td>1 NROTC &amp; NAVADMINU, MIT</td>
<td></td>
</tr>
<tr>
<td>1 NOL 331/E. COHEN</td>
<td></td>
</tr>
</tbody>
</table>

### CENTER DISTRIBUTION

<table>
<thead>
<tr>
<th>Copies</th>
<th>Copies:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0000 NELSON PERRY W</td>
<td>1 1843 SCHOT JOANNA WOOD</td>
</tr>
<tr>
<td>1 1725 GIFFORD LEROY N JR</td>
<td>1 1844 DHIR SURENDRA K</td>
</tr>
<tr>
<td>1 1725 JONES REMBERT F JR</td>
<td>1 1844 EVERSTINE GORDON C</td>
</tr>
<tr>
<td>1 1725 RODERICK JOAN E</td>
<td>10 1844 GIGNAC DONALD A</td>
</tr>
<tr>
<td>1 1725 ROTH PETER N</td>
<td>1 1844 HENDerson FRANCIS M</td>
</tr>
<tr>
<td>1 1745 NG CHRISTOPHER</td>
<td>1 1844 MATULA PETRO</td>
</tr>
<tr>
<td>1 1800 GLEISSNER GENE H</td>
<td>1 1850 CORIN THOMAS</td>
</tr>
<tr>
<td>1 1802 SHANKS DANIEL</td>
<td>1 1860 SULIT ROBERT A</td>
</tr>
<tr>
<td>1 1802 FRENKIEL FRANCOIS N</td>
<td>1 1880 CAMARA ABEL W</td>
</tr>
<tr>
<td>1 1802 LUGT HANS J</td>
<td>1 1890 GRAY GILBERT R</td>
</tr>
<tr>
<td>1 1802 THEILHEIMER FEODOR</td>
<td>1 1890 TAYLOR NORA M</td>
</tr>
<tr>
<td>1 1805 CUTHILL ELIZABETH H</td>
<td>1 1892 GOOD SHARON E</td>
</tr>
<tr>
<td>1 1830 ERNST HERBERT M</td>
<td>1 1892 RUMSEY JUDITH J</td>
</tr>
<tr>
<td>1 1830 CULPEPPER LINWOOD M</td>
<td>1 1966 CASPAR JOHN R</td>
</tr>
<tr>
<td>1 1830 WALTON THOMAS S</td>
<td>1 1966 LIU YUAN-NING</td>
</tr>
<tr>
<td>1 1840</td>
<td>30 5614 REPORTS DISTRIBUTION</td>
</tr>
<tr>
<td>1 1842 MEALS L KENTON</td>
<td>1 5641 LIBRARY</td>
</tr>
<tr>
<td>1 1842 EDDY ROBERT P</td>
<td>1 5642 LIBRARY</td>
</tr>
</tbody>
</table>