ANL-7590

ANL-7590

# Argonne National Laboratory

## MATDIAG, A Program for Computing Multilevel S-matrix Resonance Parameters

by

Peter A. Moldauer, Richard N. Hwang, and Burton S. Garbow

PROPERTY OF
ARGONNE NATIONAL LAB
IDAHO LIBRARY

The facilities of Argonne National Laboratory are owned by the United States Government. Under the terms of a contract (W-31-109-Eng-38) between the U. S. Atomic Energy Commission, Argonne Universities Association and The University of Chicago, the University employs the staff and operates the Laboratory in accordance with policies and programs formulated, approved and reviewed by the Association.

MEMBERS OF ARGONNE UNIVERSITIES ASSOCIATION

The University of Arizona
Carnegie-Mellon University
Case Western Reserve University
The University of Chicago
University of Cincinnati
Illinois Institute of Technology
University of Illinois
Indiana University
Iowa State University
The University of Iowa
Kansas State University
The University of Kansas
Loyola University
Marquette University
Michigan State University
The University of Michigan
University of Minnesota
University of Missouri
Northwestern University
University of Notre Dame
The Ohio State University
Ohio University
The Pennsylvania State University
Purdue University
Saint Louis University
Southern Illinois University
University of Texas
Washington University
Wayne State University
The University of Wisconsin

## LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

Printed in the United States of America
Available from
Clearinghouse for Federal Scientific and Technical Information
National Bureau of Standards, U. S. Department of Commerce
Springfield, Virginia 22151
Price: Printed Copy $3.00; Microfiche $0.65

ANL-7590
Physics

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

# MATDIAG, A Program for Computing Multilevel S-matrix Resonance Parameters

by

Peter A. Moldauer, Richard N. Hwang,
and Burton S. Garbow

Reactor Physics Division

July 1969

# TABLE OF CONTENTS


| | Page |
|---|---|
| ABSTRACT | 5 |
| INTRODUCTION | 5 |
| THEORY | 6 |
| DESCRIPTION OF INPUT | 9 |
| DESCRIPTION OF OUTPUT | 10 |
| THE MATDIAG CODE | 10 |
| APPENDIX: Fortran Listing of the Program | 15 |
| REFERENCES | 28 |

# MATDIAG, A Program for Computing Multilevel S-matrix Resonance Parameters

by

Peter A. Moldauer, Richard N. Hwang,
and Burton S. Garbow

## ABSTRACT

The program MATDIAG computes cross-section resonance parameters from Wigner's R-matrix. The input R-matrix parameters may be specified individually, or they may be generated statistically. The generated resonance parameters are analyzed statistically and may be used for the calculation of Doppler-broadened multilevel cross sections in other programs.

## INTRODUCTION

The purpose of the computer program MATDIAG is to compute cross-section resonance parameters from Wigner's R-matrix.[1,2]

The R-matrix parameters (energy levels $\epsilon_\mu$ and channel amplitudes $\gamma_{\mu c}$) may either be specified individually in the input, or they may be generated statistically according to appropriate distribution laws.

The computed cross-section resonance parameters are the poles and residues of the collision matrix,[3] which are useful for a variety of purposes. They can be employed for the direct calculation of Doppler-broadened multilevel cross sections by means of the Adler-Adler cross-section code,[4-7] or for multilevel cross-section and neutron-flux calculations using the improved RABBLE program or other similar such programs.[8] Of particular interest to the fast-reactor research program is the Doppler effect of the fissile isotopes in the low keV region, which has been studied by Hwang with the help of the MATDIAG program.[8-10]

The statistical analysis of the S-matrix resonance parameters that is performed in the MATDIAG program is needed for the precise specification of average cross sections where resonances overlap and is also useful for the interpretation of cross-section fluctuations.[11] Moldauer[12,13] has used the program for the study of resonance-parameter statistics and for the clarification of the averaging process.

Two considerations necessitate the computation of S-matrix parameters in terms of R-matrix parameters in the region of overlapping resonances.

First, this step is required to ensure the unitarity of the collision matrix, which imposes severe and generally complicated restrictions on the resonance parameters, particularly when the number of competing open channels is small.[14] Second, there exists no satisfactory theory that gives the statistical properties of the S-matrix parameters directly, but there is a very extensive and well-developed statistical theory of R-matrix parameters.[14]

The program generates resonance parameters by inversion of the level matrix and therefore requires the diagonalization of a complex symmetric matrix whose dimension is equal to the number of resonances included in the calculation. The version of the program used in earlier applications[9-15] was written for the CDC-3600 computer and used in complex eigenvalue and eigenvector routines of Ehrlich[16] with a restriction to a maximum of 50 resonances. The program described in this report has been significantly improved in the following ways: (1) The program has been converted to run on the faster IBM system 360-75; (2) the complex eigenvalue and eigenvector routines* written by Garbow[17] were used in place of Ehrlich's routine (this change reduced considerably the computer time required); (3) the modified version is capable of treating a maximum of 120 poles; and (4) options are available to punch cards with Adler-Adler parameters[4-7] for the Doppler-broadened cross sections in the modified RABBLE[8] format for the calculation of self-shielded reaction cross sections.

## THEORY

The theory of multilevel and statistical cross-section calculations has been discussed in great detail in Ref. 11 and was further summarized in Ref. 18. We give here only an outline containing the definitions required for an understanding of the MATDIAG code.

The collision matrix elements can be written in the form

$$S_{cc'} = \exp[i(\phi_c + \phi_{c'})]\left[W_{cc'} - i\sum_{\mu}\frac{g_{\mu c}g_{\mu c'}}{E - \epsilon_{\mu} + \frac{1}{2}i\Gamma_{\mu}}\right], \tag{1}$$

where all parameters are assumed to be energy-independent constants and the multilevel cross section is given by

*The complex eigenvector routine is unpublished.

$$\sigma_{cc'} = \pi\lambda^2 |\delta_{cc'} - S_{cc'}|^2, \tag{2}$$

whose energy dependence is directly determined by the resonance energies $\epsilon_\mu$, the widths $\Gamma_\mu$, and the residue amplitudes $g_{\mu c}$, where the subscript c refers to a particular channel. The program calculates these parameters from the eigenvalues and eigenvectors of the matrix $B_{\mu\nu}$, which is specified in terms of R-matrix parameters $E_\mu$ and $\gamma_{\mu c}$ as follows:

$$B_{\mu\nu} = E_\mu \delta_{\mu\nu} - \frac{D}{4\pi} \sum_c (\sigma_c + i\tau_c)\, \hat{\gamma}_{\mu c} \hat{\gamma}_{\nu c}, \tag{3}$$

where, in the absence of a background R-matrix,*

$$\sigma_c + i\tau_c = 4\pi(S_c^0 + iP_c)\langle\gamma_{\mu c}^2\rangle / D \tag{4}$$

and

$$\hat{\gamma}_{\mu c} = \gamma_{\mu c} / \langle\gamma_{\mu c}^2\rangle^{1/2}, \tag{5}$$

where $S_c^0$ is the shift function with the R-matrix boundary condition subtracted. This parameter can often be selected to be zero. $P_c$ is the penetrability function, and D is the mean spacing of resonances; $\langle\gamma_{\mu c}^2\rangle$ is the value of $\gamma_{\mu c}^2$ averaged over all resonances $\mu$. These parameters are determined by the optical-model transmission coefficient $T_c$ in each channel c, which has the value

$$T_c = \frac{\tau_c}{(1 + \frac{1}{4}\tau_c)^2 + (\frac{1}{4}\sigma_c)^2}. \tag{6}$$

For a fission channel, $T_c$ should be associated with the penetration probability of Hill and Wheeler.[19]

The resonance energies $\xi_\mu$ and the widths $\Gamma_\mu$ are the real and imaginary parts of the eigenvalues of $B_{\mu\nu}$;

$$B_{\mu\nu} Z_\nu^{(\lambda)} = (\epsilon_\lambda - \tfrac{1}{2} i \Gamma_\lambda)\, Z_\mu^{(\lambda)}, \tag{7}$$

where $Z_\nu^{(\mu)}$ is the $\nu$th component of the $\mu$th eigenvector of $B_{\mu\nu}$.

The residue amplitudes are given by

*The meaning of $\sigma_c$ and $\tau_c$ in more general circumstances is explained in Ref. 13.

$$g_{\mu c} = \sqrt{\frac{D\tau_c}{2\pi}} \frac{\tau_c - i\sigma_c}{(\tau_c^2 + \sigma_c^2)^{1/2}} \sum_\nu Z_\nu^{(\mu)} \hat{\gamma}_{\nu c}. \tag{8}$$

The eigenvectors are normalized so that

$$\sum_\nu Z_\nu^{(\lambda)^2} = 1, \tag{9}$$

and the normalization factor

$$\sum_\nu |Z_\nu^{(\lambda)}|^2 = N_\lambda \geq 1 \tag{10}$$

is computed that permits the definition of partial widths

$$\Gamma_{\mu c} = |g_{\mu c}|^2 / N_\mu, \tag{11}$$

which add up to the total width

$$\Gamma_\mu = \sum_c \Gamma_{\mu c}. \tag{12}$$

For statistical analysis, the program also calculates the real and imaginary parts of

$$\theta_{\mu c} = \sum_\nu Z_\nu^{(\mu)} \gamma_{\mu c} \tag{13}$$

as well as their absolute values and phase angles, and also the generalized transmission coefficients[11]

$$\theta_{\mu c} = 2\pi N_\mu^2 \Gamma_{\mu c}/D \tag{14}$$

and

$$\theta_\mu = \sum_c \theta_{\mu c} \tag{15}$$

as well as the statistical parameters[11,13]

$$B_c = \left| \langle g_{\mu c}^2 \rangle / \langle |g_{\mu c}|^2 \rangle \right|^2 \tag{16}$$

and

$$B = \langle B_c \rangle_c. \tag{17}$$

Option is also available to print and punch cards with the Adler-Adler amplitudes for the Doppler-broadened cross sections in the format of the modified RABBLE[8] so that the self-shielded cross sections can be readily calculated. The Adler-Adler amplitudes for the Doppler-broadened cross sections are related to the fundamental S-matrix parameters in the following way:[4-9]

$$G_{\mu}^{(x)} = \sum_{c'} \mathrm{Im}\, \zeta_{\mu}, \tag{18}$$

$$H_{\mu}^{(x)} = \sum_{c'} \mathrm{Re}\, \zeta_{\mu}, \tag{19}$$

$$G_{\mu}^{(T)} = \mathrm{Re}[g_{\mu c}^{2} \cdot \exp(-i2R/\lambda)] \tag{20}$$

and

$$H_{\mu}^{(T)} = \mathrm{Im}[g_{\mu c}^{2} \cdot \exp(-i2R/\lambda)], \tag{21}$$

where the complex reaction amplitude $\xi_{\mu}$ is defined as

$$\zeta_{\mu} = g_{\mu c}\, g_{\mu c'} \sum_{\mu'} \frac{g_{\mu' c}^{*} g_{\mu' c'}^{*}}{\epsilon_{\mu} - \epsilon_{\mu} - \frac{i}{2}(\Gamma_{\mu} + \Gamma_{\mu'})}. \tag{22}$$

Another option available in connection with the Doppler-effect studies is to punch out cards with the initially generated R-matrix parameters in the RABBLE format. Such information is useful in examining the validity of the Breit-Wigner equation, because these studies are often made on the basis of an <u>ad hoc</u> assumption that the single-level parameters and R-matrix parameters are identical.[8] This assumption is correct if resonances are widely spaced.

## DESCRIPTION OF INPUT

Input parameters include the number of resonances NN, the number of channels NR, and the R-matrix parameters $\sigma_c$, $\tau_c$, $E_{\mu}$, and $\hat{\gamma}_{\mu c}$.

The $E_{\mu}$ can be specified in three ways:

(1) They may be specified to have unit spacing with an average value $\langle E_{\mu} \rangle_{\mu}$ of zero (picket fence).

(2) They may be read in individually.

(3) They may be specified statistically so that the lowest $E_\mu$ and the mean spacing D are specified and the individual spacings $\Delta_\mu = E_{\mu+1} - E_\mu$ are generated randomly to follow the Wigner distribution

$$W(\Delta_\mu/D) = \frac{1}{2}\pi(\Delta_\mu/D)\exp[-\frac{\pi}{4}\Delta_\mu^2/D^2].$$

The $\hat{\gamma}_{\mu c}$ can be specified in two ways:

(1) They can be generated randomly to have a normal distribution with unit standard deviation. This means that the $\gamma^2_{\mu c}$ have a Porter-Thomas distribution.

(2) They can be read in individually.

## DESCRIPTION OF OUTPUT

The various collision-matrix resonance parameters are given in the output. In addition, certain statistical properties of these parameters are calculated, such as their averages, their dispersions and correlations, which are defined as follows:

$$\text{X AV} = \langle X_\mu \rangle \sum_\mu X_\mu/\text{NN},$$

$$\text{X MSQ} = (\langle X_\mu^2 \rangle - \langle X_\mu \rangle^2)/\langle X_\mu \rangle^2,$$

and

$$\text{X CORR} = (\langle X_{\mu+1} X_\mu \rangle - \langle X_\mu \rangle^2)/\langle X_\mu \rangle^2.$$

When channels with large transmission coefficients ($T_c > 0.5$) are present, the distributions of collision-matrix resonance parameters are apt to be distorted near the edges of the set of resonances.[13] For this reason, provision is made to exclude from the above averages a number NKK of resonances at each edge.

## THE MATDIAG CODE

The MATDIAG code consists of several subroutines, which are listed in the appendix. The function of the main program is to read input information, set up the matrix B, and then print and punch out the quantities of interest as obtained by other subroutines. To set up the matrix B, random sequences of $\gamma_{\mu c}$ and $E_\mu$ are generated by subroutines RANN and ESUB,

respectively, using the random numbers obtained in subroutine RANF.[20] Where the elements of $B_{\mu\nu}$ are exceedingly large, the matrix can be scaled down by a factor of SF specified by the users in order to avoid the possible overflow in the computation of the eigenvalues and the eigenvectors.

The most important routines required are clearly those that compute the complex eigenvalues and their corresponding complex eigenvectors. Because the dimensions of matrices of interest are generally very large, these routines must be fast and efficient to be of any practical interest. Two such programs available at Argonne National Laboratory are MATSUB[16] and FRANCC[17] with VCTR (unpublished). MATSUB, originally written for the CDC-3600 computer,[16] has been converted to the IBM 360/75. It was found, however, that FRANCC[17] is faster by as much as a factor of three, compared to MATSUB. Furthermore, it avoids the direct evaluation of the determinant, which, for large matrices of interest, causes overflow in MATSUB. Hence, FRANCC is recommended. Recently, another program that treats the complex symmetric matrices of interest became available. This program, written by Seaton,[21] is based on a modified Jacobi method. Preliminary tests have shown that Seaton's program is comparable to FRANCC in speed and is perhaps more efficient when the diagonal elements of the matrix B predominate or when only moderate accuracy is required. These programs will be compared further.

Once the complex eigenvalues and eigenvectors are obtained, the main program computes and prints out the S-matrix resonance parameters. Subroutines ORDER and ORDER1 order the quantities $|\epsilon_\mu - \epsilon_\nu| \Gamma_\mu/2$ and $N_\mu$ in increasing order so that the statistical behavior of these quantities can be examined more readily. Subroutine AVERAGE computes average quantities required in the determination of the averaged cross sections.

Other characteristics of this code will be given as follows:

1. Input

| Card | Format | Variables | Description |
|---|---|---|---|
| 1 | (16A5) | | Title |
| 2 | (12I5, E12.6) | NN | Number of poles $\leq 120$. |
| | | NR | Number of channels $\leq 300$. |
| | | NKK | Number of edge poles on each side of the sample excluded from statistical analysis. |
| | | NRTEST | = 0 Same $\sigma, \tau$ for all channels.<br>= 1 Different $\sigma, \tau$ for each channel. |

| Card | Format | Variables | Description |
|---|---|---|---|
| 2 (Contd.) | | NK | Number of channels in printout ≤300. |
| | | NNRGEN | Number of random number generator or number of sequential problems. |
| | | NOPT1 | = 1 Picket fence $E_\mu$.<br>= 2 Read in $E_\mu$.<br>= 3 $E_\mu$ generated from the Wigner distribution. |
| | | NOPT2 | = 0 $\gamma$'s generated from the normal distribution.<br>= 1 $\gamma$'s are read in. |
| | | NOPT3 | = 0 Real and imaginary parts of the complex pole are not punched out.<br>= 1 Real and imaginary parts of the complex poles are punched with cards in (8E10.3) format. |
| | | NU | Number of incident neutron channels. |
| | | NF | Number of fission channels (NU + NF ≤ NR). |
| | | KADLER | = 1 Adler-Adler parameters for the Doppler-broadened cross sections are printed and punched out in the modified RABBLE format[8] (6E12.5/3E12.5).<br>= 0 Adler-Adler parameters are not computed. |
| | | SF | Scaling factor for the input matrix. |

NOTE: If KADLER = 0, skip card 3.

| | | | |
|---|---|---|---|
| 3 | (E12.6, I6) | GFACT | Statistical spin g-factor. |
| | | MWIDTH | = 1 Total R-matrix partial widths $\sum_c (2P_c)\gamma^2_{\mu c}$ for neutron, fission, and capture are printed and punched in RABBLE format (5E12.6).<br>= 0 R-matrix partial widths are not given. |

| Card | Format | Variables | Description |
|---|---|---|---|
| 4* | (8F10.6) or (2F10.6) | ELR(I), ERI(I) | ELR(I) = $\sigma_c$ ELI(I) = $\tau_c$ for NRTEST = 1. for NRTEST = 0. |
| 5 | (8I10) | NRGEN (M) (M=1,NNRGEN) | Random number generators for $\gamma$'s if NOPT2 = 0. |
| | (8F10.0) | GAMMA(I,K) (K=1,NR) | Read in $\gamma$'s if NOPT2 = 1. |

NOTE: For NOPT1 = 1, skip cards 6 and 7. $E_\mu$'s are generated in ESUB using $\langle D \rangle = 1$.

| | | | |
|---|---|---|---|
| 6 | (8F10.0) | E(I) (I=1,NN) | If NOPT1 = 2. |
| | (2F10.4, I10) | DBAR | If NOPT1 = 3. Average spacing (eV) (usually taken to be unity in the studies of the average cross section).[11-15] |
| | | ENUT | Arbitrary initial energy of the interval (eV). |
| | | NR | Random number generator for $E_\mu$. (Here, NR is local variable used in ESUB only.) |
| 7 | (E12.6) | DBAR | If NOPT1 = 2. Otherwise, skip this card. (Average spacing eV.) |

2. Output

The output of MATDIAG is listed in order as follows:

(1) Input information.

(2) Generated sequences of $\gamma_{\mu c}/\langle\gamma^2_{\mu c}\rangle^{1/2}$ for each channel.

(3) Generated $E_\mu$.

(4) A MATRIX (the Matrix B).

(5) EIGENVALUE (real and imaginary parts $\epsilon_\lambda$, $\frac{1}{2}\Gamma_\lambda$).
N (the normalization factor $N_\lambda$).
EIGENVECTOR (real and imaginary parts of the components of $Z_\mu^{(\lambda)}$).

*If KADLER > 0, $\sigma_c$ and $\tau_c$ for neutron channel are read in first and are followed by the fission and capture channels.

(6) Eigenvalues ($\lambda_\mu$), associated eigenvectors (Z), and estimated errors. Define a vector v such that $v = (B - \lambda_\mu) Z$. The estimated error is defined as the element of vector v that possesses the largest magnitude.

(7) Listing of H (real parts $\epsilon_\lambda$ of eigenvalues).
ΔH (real eigenvalue spacings $\epsilon_{\lambda+1} - \epsilon_\lambda$).
G (half widths $\frac{1}{2}\Gamma_\lambda$).
N (normalization factors $N_\lambda$).
Statistics of H, G, N.

(8) ΔH, G, and N in increasing order.

(9) RE THETA (Re $\theta_{\lambda c}$), IM THETA (Im $\theta_{\lambda c}$), MAG THETA**2 ($|\theta|_{\lambda c}$), ARG THETA, (phase angle of $\theta_{\lambda c}$), BG($\Gamma_{\lambda c}$), and BT($\theta_{\lambda c}$) listed per one channel c and all λ, also B($B_c$) and statistics of channel parameters.
Repeated for as many channels as called for.

(10) Adler-Adler parameters: $G_\mu^{(F)}$, $G_\mu^{(R)}$, $G_\mu^{(T)}$, $H_\mu^{(F)}$, $H_\mu^{(R)}$, $H_\mu^{(T)}$, $\epsilon_\mu$, and $\Gamma_\mu/2$.

## 3. Time Estimation

The computer time required depends strongly on the number of poles considered and also, to some extent, depends on the total number of channels assumed. The computer times required for test problems using 105 channels are tabulated as follows:

| Number of Poles | Computer Time, min |
|---|---|
| 50 | 7 |
| 80 | 35 |
| 120 | 127 |

# APPENDIX

## Fortran Listing of the Program

```
C     PROGRAM MATDIAG
C
C      THIS PROGRAM HAS BEEN TRANSLATED FOR THE     360/50
C      WITH RELEASE 1-A OF THE MOD-50 TRANSDECK
C                                                        JDB
C
      DIMENSION EVECR(120,120),EVECI(120,120)
      DIMENSION E(120),ELR(120),ELI(120),GAMMA(120,300),TITLE(16)
      DIMENSION  H(120),G(120),DH(120),THETR(120),THETI(120),TMAG(120),
     1NSH(120),NRGEN(120),EN(120),ENN(120),DEL(120),BGAMMA(120),
     2BTHETA(120),SAVEH(120),SAVEG(120),SGREAL(120),SGIMAG(120),
     3SAVEN(120)
      DIMENSION A(120,120),VALU(120,1),B(120,120)
      DIMENSION  NPR(2),SUMAT(2),SUMABG(2),SUMABT(2),SUMTMSQ(2),
     1SUMBGMSQ(2),SUMBTMSQ(2),SUMTCOR(2),SUMRGCOR(2),SUMRTCOR(2),SUMB(2)
      DIMENSION GF(120),GR(120),GT(120),HF(120),HR(120),
     1HT(120),HALGAM(120)
      COMMON DRAR
      COMMON /QR/ ITER(240), V(120), DUMMY(360)
      COMPLEX*16 TEMP1
      REAL*8 SGREAL,SGIMAG
      COMPLEX*16 A,VALU,B,V,TEMP
      REAL*8 E,SUMR,SUMI
      COMPLEX*16 DUMMY1,DUMMY2,DUMMY3,PRODF,PRODG
      REAL*8 DUMMY4,DUMMY5,DUMMY6,DUMMY7,DUMMY8,DUMMY9
      REAL*8 HALGAM
      REAL*8 G,H,DH,EN
      REAL*8      ANORM,VNORM,RESID,DUMMY
      REAL*8 SF
      COMPLEX DUM,CNORM,DAMN1,DAMN2,DUMDUM,SCRW
C
   10 READ(60,20)TITLE
   20 FORMAT(16A5)
      WRITE(61,30)TITLE
   30 FORMAT(1H116A5)
      READ(60,40)NN,NR,NKK,NRTEST,NK,NNRGEN,NOPT1,NOPT2,NOPT3,NU,NF,
     1KADLER,SF
   40 FORMAT(12I5,E12.6)
      WRITE(61,1)SF
    1 FORMAT(16H SCALING FACTOR=E15.6)
      WRITE(61,45)NN,NR,NKK
   45 FORMAT(1H08X1HN9X1HR6X4HEDGE/(3I10))
      IF(KADLER.LE.0)GO TO 4
      READ(60,123)GFACT,MWIDTH
  123 FORMAT(E12.6,I6)
    4 IF(NRTEST)80,50,80
   50 CONTINUE
      READ(60,60)ELR(1),ELI(1)
   60 FORMAT(8F10.0)
      WRITE(61,65)ELR(1),ELI(1)
   65 FORMAT(1H010X4HRE L11X4HIM L/(2E15.7))
      DO70K=2,NR
      ELR(K)=ELR(1)
      ELI(K)=ELI(1)
   70 CONTINUE
      GOTO90
   80 CONTINUE
      READ(60,60)((ELR(K),ELI(K)),K=1,NR)
      WRITE(61,85)
   85 FORMAT(1H010X4HRE L11X4HIM L)
      WRITE(61,160)((ELR(K),ELI(K)),K=1,NR)
   90 CONTINUE
      DO 91 I=1,NR
```

```
      ELR(I)=ELR(I)/(4.*3.14159265)
      ELI(I)=ELI(I)/(4.*3.14159265)
   91 CONTINUE
      IF(NOPT2 .GT.0)GO TO 95
      READ(60,42)(NRGEN(M),M=1,NNRGEN)
   42 FORMAT(8I10)
      WRITE(61,96)
   96 FORMAT(32H0RANDOM NUMBER GENERATOR NUMBERS)
      WRITE(61,42)(NRGEN(M),M=1,NNRGEN)
   95 CONTINUE
      DO400M=1,NNRGEN
      IF(NOPT2.GT.0)GO TO 101
      WRITE(61,97)NRGEN(M)
   97 FORMAT(16H1RANDOM NO. GEN=I10)
      DO100I=1,NN
      DO100K=1,NR
      GAMMA(I,K)=RANN(NRGEN(M))
  100 CONTINUE
      GO TO 105
  101 CONTINUE
      DO102 K=1,NR
      READ(60,60)(GAMMA(I,K),I=1,NN)
  102 CONTINUE
  105 CONTINUE
      WRITE(61,106)
  106 FORMAT(6H0GAMMA)
      DO 107 I=1,NN
      WRITE(61,160)(GAMMA(I,K),K=1,NR)
  107 CONTINUE
      CALLESUB(NN,E,NOPT1)
      WRITE(61,108)
  108 FORMAT(2H0E)
      WRITE(61,160)(E(J),J=1,NN)
      IF(MWIDTH.LE.0)GO TO 3219
      NFF=NU+NF
      NUP=NU+1
      NGA=NU+NF+1
      DO 1108 I=1,NN
      DO 7709 K=1,NU
      GANN=DSQRT(E(I))*2.*ELI(1)*GAMMA(I,K)**2*DBAR
      GAFF=0.0
      GARR=0.0
      DO 1109 L=NUP,NFF
      GAFF=GAFF+2.*ELI(L)*GAMMA(I,L)**2*DBAR
 1109 CONTINUE
      DO 7709 J=NGA,NR
      GARR=GARR+2.*ELI(J)*GAMMA(I,J)**2*DBAR
 7709 CONTINUE
      PRINT 2209,GANN,GARR,GAFF
 2209 FORMAT(4H GN=E12.6,4H GR=E12.6,4H GF=E12.6)
      PUNCH 3309,E(I),GANN,GARR,GAFF,GFACT
 3309 FORMAT(5F12.6)
 1108 CONTINUE
 3219 DO 140 I=1,NN
      DO140J=1,NN
      SUMR=0.
      SUMI=0.
      DO110K=1,NR
      SUMR=SUMR+ELR(K)*GAMMA(I,K)*GAMMA(J,K)*DBAR
      SUMI=SUMI+ELI(K)*GAMMA(I,K)*GAMMA(J,K)*DBAR
  110 CONTINUE
      IF(I-J)130,120,130
  120 CONTINUE
      A(I,J)=((1.0D0,0.0D0)*(E(J)-SUMR)+(0.0D0,1.0D0)*(-SUMI))/SF
      GOTO140
  130 CONTINUE
```

```
      A(I,J)=((1.0D0,0.0D0)*(-SUMR)+(0.0D0,1.0D0)*(-SUMI))/SF
  140 CONTINUE
      DO 3 J=1,NN
      DO 3 K=1,NN
    3 B(J,K)=A(J,K)
      WRITE(61,150)
  150 FORMAT(9H1A MATRIX)
      DO 170 I=1,NN
      WRITE(61,160)((A(I,J)),J=1,NN)
  160 FORMAT(8E15.7)
  170 CONTINUE
      CALL CLOCK(T1)
      CALL FRANCC(A,VALU,NN,ANORM,120)
      CALL CLOCK(T2)
      TT=T2-T1
      WRITE(61,5000)ANORM,TT
      WRITE(61,8000)
      TT=0.0
      DO 190 I=1,NN
C**** NEXT        STATEMENT  ADDED OR REWRITTEN TO BYPASS BUG IN COMPILER
      TEMP1=-VALU(I,1)
      DO 88 J=1,NN
      DO 88 K=1,NN
   88 A(J,K)=B(J,K)
      CALL CLOCK(T1)
      CALL VCTR(A,V,NN,VALU(I,1))
      CALL CLOCK(T2)
      TT=TT+(T2-T1)
      DAMN1=VALU(I,1)
      H(I)=REAL(DAMN1) *SF
      G(I)=AIMAG(DAMN1)*SF
      DUM=0.
      DO 999 J=1,NN
  999 DUM=DUM+V(J)**2
      CNORM=CSQRT(DUM)
      DO 1111 J=1,NN
      V(J)=V(J)/CNORM
      DAMN2=V(J)
      EVECR(I,J)=REAL(DAMN2)
      EVECI(I,J)=AIMAG(DAMN2)
 1111 CONTINUE
      RESID=0.0
      DO 1195 J=1,NN
C**** NEXT        STATEMENT  ADDED OR REWRITTEN TO BYPASS BUG IN COMPILER
      TEMP=TEMP1*V(J)
      DO 92 K=1,NN
   92 TEMP=TEMP+B(J,K)*V(K)
      IF(CDABS(TEMP)-RESID)1195,1195,93
   93 RESID=CDABS(TEMP)
 1195 CONTINUE
      WRITE(61,8500)(VALU(I,1),RESID,(V(J),J=1,NN))
  190 CONTINUE
      WRITE(61,9000)TT
 5000 FORMAT (13H0MATRIX NORM=1PD24.15,  12H   TIME(MS)=1PF8.0/
     162H0          REAL                    IMAGINARY          ITERATIONS)
 6000 FORMAT (1H0 1PD24.15,D24.15,I9)
 7000 FORMAT (53H1THE QR ALGORITHM FOR EIGENVALUES OF COMPLEX MATRICES)
 8000 FORMAT (↑0      EIGENVALUE, ASSOCIATED EIGENVECTOR, AND RESIDUAL↑)
 8500 FORMAT (1H0 1PD24.15,D24.15,D20.2/(1H0 0P4(F18.10,F15.10)))
      DO 273 I=1,NN
 9000 FORMAT (↑0 TIME(MS) FOR EIGENVECTORS=↑ 1PF12.0)
      EN(I)=0.
      DO 272 J=1,NN
      EN(I)=EN(I)+EVECR(I,J)**2+EVECI(I,J)**2
  272 CONTINUE
      ENN(I)=EN(I)
```

```
273 CONTINUE
    CALLORDER1(H,NSH,NN)
    DO 200 I=1,NN
    NS=NSH(I)
    SAVEH(I)=H(I)
    SAVEG(I)=G(NS)
    HALGAM(I)=-G(NS)
    SAVEN(I)=EN(NS)
200 CONTINUE
    IF(NOPT3.EQ.0)GO TO 220
    PRINT 280,NN,NR
    PUNCH 280,NN,NR
280 FORMAT(2I10)
    PRINT 210,(SAVEH(I),SAVEG(I),I=1,NN)
    PUNCH 210,(SAVEH(I),SAVEG(I),I=1,NN)
210 FORMAT(8E10.3)
220 CONTINUE
    NNM1=NN-1
    DO230I=1,NNM1
    DH(I)=H(I+1)-H(I)
230 CONTINUE
    DH(NN)=0.
    WRITE(61,241)
241 FORMAT(1H113X1HH10X5HH DIF14X1HG14X1HN)
    WRITE(61,251)((H(I),DH(I),SAVEG(I),SAVEN(I)),I=1,NN)
251 FORMAT(4E15.7)
    NPQ=0
    CALLAVERAGE(SAVEN,NN,NPQ,AEN,AENSQ)
    CALLAVERAGE(SAVEG,NN,NPQ,AG,AGSQ)
    CALLAVERAGE(DH,NNM1,NPQ,ADH,ADHSQ)
    DHMSQ=(ADHSQ/ADH**2)-1.
    GMSQ=(AGSQ/AG**2)-1.
    ENMSQ=(AENSQ/AEN**2)-1.
    WRITE(61,265)ADH,DHMSQ,AG,GMSQ,AEN,ENMSQ
265 FORMAT(1H09X5HDH AV9X6HDH MSQ11X4HG AV10X5HG MSQ11X4HN AV
   110X5HN MSQ/(6E15.7))
    NPQ=NKK
    CALLAVERAGE(SAVEN,NN,NPQ,AEN,AENSQ)
    CALLAVERAGE(SAVEG,NN,NPQ,AG,AGSQ)
    CALLAVERAGE(DH,NNM1,NPQ,ADH,ADHSQ)
    DHMSQ=(ADHSQ/ADH**2)-1.
    GMSQ=(AGSQ/AG**2)-1.
    ENMSQ=(AENSQ/AEN**2)-1.
    WRITE(61,266)ADH,DHMSQ,AG,GMSQ,AEN,ENMSQ
266 FORMAT(1H06HCENTER3X5HDH AV9X6HDH MSQ11X4HG AV10X5HG MSQ11X4HN AV
   110X5HN MSQ/(6E15.7))
    CALLORDER(DH,NNM1)
    CALLORDER(G,NN)
    CALLORDER(EN,NN)
    WRITE(61,261)
261 FORMAT(1H17HORDERED2X5HH DIF14X1HG14X1HN)
    WRITE(61,264)((DH(I),G(I),EN(I)),I=1,NN)
264 FORMAT(3E15.7)
    DO 215 L=1,2
    SUMAT(L)=0.
    SUMABG(L)=0.
    SUMABT(L)=0.
    SUMTMSQ(L)=0.
    SUMBGMSQ(L)=0.
    SUMBTMSQ(L)=0.
    SUMTCOR(L)=0.
    SUMBGCOR(L)=0.
    SUMBTCOR(L)=0.
    SUMB(L)=0.
215 CONTINUE
    DO370K=1,NR
```

```
      DO 295 I=1,NN
      NS=NSH(I)
      THETR(I)=0.
      THETI(I)=0.
      DO290J=1,NN
      THETR(I)=THETR(I)+EVECR(NS,J)*GAMMA(J,K)
      THETI(I)=THETI(I)+EVECI(NS,J)*GAMMA(J,K)
  290 CONTINUE
      DUMARG=DRAR*ELI(K)*2.
      DUMLSQ=ELI(K)**2+ELR(K)**2
      SCRW=CMPLX(ELI(K),-ELR(K))
      DUMDUM=SQRT(DUMARG)*SCRW/SQRT(DUMLSQ)
      A(I,K)=DUMDUM*CMPLX(THETR(I),THETI(I))
      SGREAL(I)=A(I,K)
      SGIMAG(I)=(0.0D0,-1.0D0)*A(I,K)
      TMAG(I)=THETR(I)**2+THETI(I)**2
      BGAMMA(I)=2.*ELI(K)*TMAG(I)*DBAR/ENN(NS)
      BTHETA(I)=4.*3.14159265*ELI(K)*TMAG(I)*ENN(NS)
      DEL(I)=ATAN2 (THETI(I),THETR(I))
      DEL(I)=DEL(I)/(2.*3.14159265)
  295 CONTINUE
      IF(NOPT3.EQ.0)GO TO 310
      PRINT 300,(SGREAL(I),SGIMAG(I),I=1,NN)
      PUNCH 300,(SGREAL(I),SGIMAG(I),I=1,NN)
  300 FORMAT(8F10.3)
  310 CONTINUE
      NPR(1)=0.
      NPR(2)=NKK
      DO420L=1,2
      NPQ=NPR(L)
      CALL AVERAGE(TMAG,NN,NPQ,ATMAG,ATMAGSQ)
      CALLAVERAGE(BGAMMA,NN,NPQ,ABG,ABGSQ)
      CALLAVERAGE(BTHETA,NN,NPQ,ABT,ABTSQ)
      TMAGCORR=0
      BGCORR=0
      BTCORR=0
      NMN1=1+NPQ
      NMN2=NN-NPQ-1
      NMN3=NN-NPQ
      DO315I=NMN1,NMN2
      TMAGCORR=TMAGCORR+TMAG(I)*TMAG (I+1)
      BGCORR=BGCORR+BGAMMA(I)*BGAMMA (I+1)
      BTCORR=BTCORR+BTHETA(I)*BTHETA (I+1)
  315 CONTINUE
      TMAGMSQ=(ATMAGSQ/ATMAG**2)-1.
      BGMSQ=(ABGSQ/ABG**2)-1.
      BTMSQ=(ABTSQ/ABT**2)-1.
      TMAGCORR=(TMAGCORR/((NN-1-2*NPQ)*ATMAG*ATMAG))-1
      BGCORR=(BGCORR/((NN-1-2*NPQ)*ABG*ABG))-1
      BTCORR=(BTCORR/((NN-1-2*NPQ)*ABT*ABT))-1
      CALL AVERAGE(THETR,NN,NPQ,ATHETR,ATHETRSQ)
      CALL AVERAGE(THETI,NN,NPQ,ATHETI,ATHETISQ)
      BR=ATHETRSQ-ATHETISQ
      BI=0.
      DO316I=NMN1,NMN3
      BI=BI+THETR(I)*THETI(I)
  316 CONTINUE
      BI=2.*BI/(NN-2*NPQ)
      BB=(BR**2+BI**2)/ATMAG**2
      SUMAT(L)=SUMAT(L)+ATMAG
      SUMABG(L)=SUMABG(L)+ABG
      SUMABT(L)=SUMABT(L)+ABT
      SUMTMSQ(L)=SUMTMSQ(L)+TMAGMSQ
      SUMBGMSQ(L)=SUMBGMSQ(L)+BGMSQ
      SUMBTMSQ(L)=SUMBTMSQ(L)+BTMSQ
      SUMTCOR(L)=SUMTCOR(L)+TMAGCORR
```

```
      SUMBGCOR(L)=SUMBGCOR(L)+BGCORR
      SUMBTCOR(L)=SUMBTCOR(L)+BTCORR
      SUMB(L)=SUMB(L)+BB
      IF(L-1)318,318,380
  318 CONTINUE
      IF(K-NK)317,317,360
  317 CONTINUE
      WRITE(61,320)K
  320 FORMAT(1H16X8HRE THETA7X8HIM THETA3X12HMAG THETA**2
     16X 9HARG THETA13X2HBG13X2HBT5X2HK=I3)
      WRITE(61,325)((THETR(I),THETI(I),TMAG(I),DEL(I),BGAMMA(I),
     1BTHETA(I)),I=1,NN)
  325 FORMAT(6E15.7)
  360 CONTINUE
      WRITE(61,329)K
  329 FORMAT(1H04X5HBG AV4X6HBG MSQ3X7HBG CORR5X5HBT AV4X6HBT MSQ
     13X7HBT CORR5X5HTH AV4X6HTH MSQ3X7HTH CORR5X1HB5X2HK=I3)
  380 CONTINUE
      WRITE(61,330)ABG,BGMSQ,BGCORR,ABT,BTMSQ,BTCORR,ATMAG,TMAGMSQ,
     1TMAGCORR,BB
  330 FORMAT(10E10.2)
  420 CONTINUE
  370 CONTINUE
      DO430L=1,2
      TUMAT    =SUMAT(L)/NR
      TUMARG    =SUMABG(L)/NR
      TUMABT    =SUMABT(L)/NR
      TUMTMSQ    =SUMTMSQ(L)/NR
      TUMBGMSQ    =SUMBGMSQ(L)/NR
      TUMBTMSQ    =SUMBTMSQ(L)/NR
      TUMTCOR    =SUMTCOR(L)/NR
      TUMBGCOR    =SUMBGCOR(L)/NR
      TUMBTCOR    =SUMBTCOR(L)/NR
      TUMB    =SUMB(L)/NR
      IF(L-1)417,417,470
  417 CONTINUE
      WRITE(61,375)
  375 FORMAT(45H1AVERAGES OVER R, TOTAL POLES / CENTRAL POLES)
      WRITE(61,329)
  470 CONTINUE
      WRITE(61,330)TUMABG,TUMBGMSQ,TUMBGCOR,TUMABT,TUMBTMSQ,TUMBTCOR
     1,TUMAT,TUMTMSQ,TUMTCOR,TUMB
  430 CONTINUE
      NOPT1=0
  400 CONTINUE
      IF(KADLER.LE.0)GO TO10
      NGA=NU+NF+1
      NUP=NU+1
      NFF=NU+NF
      DO 9113 I=1,NN
      DUMMY1=0.0
      DUMMY2=0.0
      DUMMY3=0.0
      DO 9112 K=1,NU
      ARGP=DSQRT(H(I))*3.80933E-03
      DUMMY3=DUMMY3+A(I,K)**2*CMPLX(COS(ARGP),-SIN(ARGP))
      DO 9111 J=1,NN
      PRODF=0.0
      PRODG=0.0
      DO 9114 L=NUP,NFF
      PRODF=PRODF+DCONJG(A(J,K))*DCONJG(A(J,L))*A(I,K)*A(I,L)
 9114 CONTINUE
      IF(NGA.GT.NR)GO TO 6666
      DO 9117 M=NGA,NR
      PRODG=PRODG+DCONJG(A(J,K))*DCONJG(A(J,M))*A(I,K)*A(I,M)
 9117 CONTINUE
```

```
      DUMMY2=DUMMY2+PRODG/DCMPLX((H(I)-H(J)),-(HALGAM(I)+HALGAM(J)))
 6666 DUMMY1=DUMMY1+PRODF/DCMPLX((H(I)-H(J)),-(HALGAM(I)+HALGAM(J)))
 9111 CONTINUE
 9112 CONTINUE
      DUMMY4=DUMMY1
      DUMMY5=(0.0,-1.0)*DUMMY1
      DUMMY6=DUMMY2
      DUMMY7=(0.0,-1.0)*DUMMY2
      DUMMY8=DUMMY3
      DUMMY9=(0.0,-1.0)*DUMMY3
      GF(I)=DUMMY5
      GR(I)=DUMMY7
      GT(I)=DUMMY8
      HF(I)=-DUMMY4
      HR(I)=-DUMMY6
      HT(I)=DUMMY9
      PRINT 9119,GF(I),GR(I),GT(I),HF(I),HR(I),HT(I)
 9119 FORMAT(4H GF=E15.6,4H GR=E15.6,4H GT=E15.6,4H HF=E15.6,4H HR=
     1E15.6,4H HT=E15.6)
      PRINT 9229,HALGAM(I),H(I)
 9229 FORMAT(8H HALGAM=E15.6,7H SENUT=E15.6)
      PUNCH 3311,H(I),GT(I),GR(I),GF(I),HT(I),HR(I),HF(I),HALGAM(I),
     1GFACT
 3311 FORMAT(1P6E12.5/1P3E12.5)
 9113 CONTINUE
      GOTO10                                                            269
                                                                        271
      END
      SUBROUTINE VCTR   (A,V,N,ALPHA)                                 VCTR0001
      DIMENSION A(120,120),V(120)
C**** REMOVE OR MODIFY NEXT       STATEMENT  IN SINGLE PRECISION VERSION
      COMPLEX*16 A,V,ALPHA,R,C
                                                                      VCTR0003
      A(1,1)=A(1,1)-ALPHA                                             VCTR0004
 6    DO 15 I=2,N                                                     VCTR0005
      A(I,I)=A(I,I)-ALPHA
C**** NEXT        STATEMENT  ADDED OR REWRITTEN TO BYPASS BUG IN COMPILER
      V(I)=(1.0D0,0.0D0)                                              VCTR0006
 70   II=I-1                                                          VCTR0007
 7    DO 15 J=1,II
C**** CHANGE FUNCTION NAMES IN NEXT  TWO  STATEMENTS IN SINGLE PRECISION
    8 IF(CDABS(A(I,J)))9,15,9
 9    IF(CDABS(A(J,J))-CDABS(A(I,J)))11,10,10
                                                                      VCTR0010
 10   R=A(I,J)/A(J,J)                                                 VCTR0011
      GO TO 130                                                       VCTR0012
 11   R=A(J,J)/A(I,J)                                                 VCTR0013
      DO 12 K=1,N                                                     VCTR0014
      C=A(J,K)                                                        VCTR0015
      A(J,K)=A(I,K)                                                   VCTR0016
 12   A(I,K)=C                                                        VCTR0017
 130  JJ=J+1                                                          VCTR0018
 13   DO 14 K=JJ,N                                                    VCTR0019
 14   A(I,K)=A(I,K)-R*A(J,K)                                          VCTR0020
 15   CONTINUE                                                        VCTR0021
      C=A(N,N)                                                        VCTR0023
      DO 29 I=2,N                                                     VCTR0024
      JJ=N-I+1                                                        VCTR0025
      R=0.                                                            VCTR0026
      II=N-I+2                                                        VCTR0027
      DO 25 K=II,N                                                    VCTR0028
 25   R=R+A(JJ,K)*V(K)
C**** CHANGE FUNCTION NAME  IN NEXT       STATEMENT  IN SINGLE PRECISION
      IF(CDABS(A(JJ,JJ))-1.0E-10)27,27,28
                                                                      VCTR0030
 27   V(JJ)=1.                                                        VCTR0031
      C=0.                                                            VCTR0032
      DO 26 J=II,N                                                    VCTR0033
 26   V(J)=0.                                                         VCTR0034
      GO TO 29                                                        VCTR0035
 28   V(JJ)=(C-R)/A(JJ,JJ)
```

```
 29   CONTINUE                                                          VCTR0036
      RETURN                                                            VCTR0037
      END
      SUBROUTINE FRANCC (A,VALU,NSUB,ANORM,NMAX)
C
      DIMENSION A(NMAX,NSUB), VALU(NMAX)
C**** REMOVE OR MODIFY NEXT FOUR  STATEMENTS IN SINGLE PRECISION VERSION
      COMMON /QR/ ITER(240),DUMMY(600)
      COMPLEX*16 A,VALU,ANN,DIF,DISCSQ,DISC,E,G
      REAL*8     ANORM2,ANORM,EPS,DUMMY,DEL
      DATA EPS/Z3380000000000000/
C
      N=NSUB
C
C     REDUCE MATRIX TO UPPER HESSENBERG FORM (WITH REAL SUBDIAGONAL)
C
      CALL SUBDIC(A,N,NMAX)
C
C     COMPUTE MATRIX NORM
C
      ANORM2=0.0
      DO 40 I=1,N
      I1=I-1+1/I
      DO 40 J=I1,N
C**** CHANGE FUNCTION NAME  IN NEXT       STATEMENT  IN SINGLE PRECISION
      ANORM2=ANORM2+A(I,J)*DCONJG(A(I,J))
   40 CONTINUE
C**** CHANGE FUNCTION NAME  IN NEXT       STATEMENT  IN SINGLE PRECISION
      ANORM=DSQRT(ANORM2)
C
      DEL=ANORM*EPS
C
C     BEGINNING OF LOOP FOR ITERATIVE DETERMINATION OF EIGENVALUES
C     (ARRAY ITER HAS EFFECTIVELY BEEN CLEARED TO ZERO BY SUBDIA)
C
   50 K=NSUB+1-N
C
C     FIND ROOTS OF LOWER 2X2 MINOR
C
   60 ANN=A(N,N)
      IF (N-1) 250, 220, 70
   70 IF (CDABS   (A(N,N-1)) .LE.DEL) GO TO 220
C**** REMOVE OR MODIFY NEXT       STATEMENT  IN SINGLE PRECISION VERSION
      DIF=(A(N-1,N-1)-ANN)*(0.5D0,0.0D0)
      DISCSQ=DIF**2+A(N-1,N)*A(N,N-1)
C**** CHANGE FUNCTION NAME  IN NEXT       STATEMENT  IN SINGLE PRECISION
      DISC=CDSQRT(DISCSQ)
      E=DISC+DIF+ANN
      G=E-DISC-DISC
      IF (N.EQ.2) GO TO 230
      IF (CDABS   (A(N-1,N-2)) .LE.DEL) GO TO 230
C
C     CHOOSE SHIFT TOWARD ACCELERATING CONVERGENCE
C     (ROOT OF LOWER 2X2 MINOR CLOSEST TO LAST DIAGONAL ELEMENT)
C
C**** CHANGE FUNCTION NAMES IN NEXT  TWO  STATEMENTS IN SINGLE PRECISION
      Z1=(E-ANN)*DCONJG(E-ANN)
      Z2=(G-ANN)*DCONJG(G-ANN)
      IF (Z2.LT.Z1) E=G
C
C     PERFORM QR ITERATION
C
      CALL QR1(A,N,DEL,E,NMAX)
C
      ITER(K)=ITER(K)+1
      GO TO 60
```

```
C
C     SINGLE EIGENVALUE CONVERGED
C
  220 VALU(K)=ANN
      N=N-1
      GO TO 50
C
C     PAIR OF EIGENVALUES CONVERGED
C
  230 VALU(K)=E
      VALU(K+1)=G
      N=N-2
      GO TO 50
C
  250 RETURN
      END
      SUBROUTINE SUBDIC(A,N,NMAX)
C
C     HOUSEHOLDER REDUCTION OF COMPLEX MATRIX TO UPPER HESSENBERG FORM
C
      DIMENSION A(NMAX,N)
      DIMENSION WVEC(120),PVEC(120),QVEC(120),CWVEC(120)
      COMMON /QR/ WVEC, PVEC, CWVEC
      EQUIVALENCE (PVEC,QVEC)
C**** REMOVE OR MODIFY NEXT  TWO  STATEMENTS IN SINGLE PRECISION VERSION
      COMPLEX*16 A,WVEC,PVEC,QVEC,DIV,SCALAR,CWVEC
      REAL*8     TEMP,SUM,TEMP1
C
      DO 200 I=1,N
C
C     REDUCE COLUMN OF MATRIX
C
      WVEC(I)=0.0
      IF (I.EQ.N) GO TO 200
      I1=I+1
      I2=I1+1
C
C     SIMILARITY TRANSFORMATION TO PRODUCE REAL SUBDIAGONAL ELEMENT
C
      J=I1
C**** CHANGE FUNCTION NAME  IN NEXT       STATEMENT  IN SINGLE PRECISION
      TEMP1=CDABS(A(J,I))
      IF (TEMP1.EQ.0.0) GO TO 60
      DIV=A(J,I)/TEMP1
      DO 30 K=1,N
   30 A(K,J)=DIV*A(K,J)
      DO 40 K=I,N
   40 A(J,K)=A(J,K)/DIV
C
   60 IF (I2.GT.N) GO TO 200
      SUM=0.0
      DO 70 J=I2,N
C**** CHANGE FUNCTION NAME  IN NEXT       STATEMENT  IN SINGLE PRECISION
   70 SUM=SUM+A(J,I)*DCONJG(A(J,I))
      IF (SUM.EQ.0.0) GO TO 200
      J=I1
      TEMP=A(J,I)
C**** CHANGE FUNCTION NAMES IN NEXT THREE STATEMENTS IN SINGLE PRECISION
      SUM=DSQRT(SUM+ TEMP  **2)
      A(J,I)=DCMPLX(-DSIGN(SUM,TEMP),0.0D0)
      TEMP1=DSQRT(1.0+DABS(TEMP)/SUM)
      WVEC(J)=TEMP1
      CWVEC(J)=WVEC(J)
C**** CHANGE FUNCTION NAME  IN NEXT       STATEMENT  IN SINGLE PRECISION
      DIV=DSIGN(TEMP1*SUM,TEMP)
      DO 85 J=I2,N
```

```
      WVEC(J)=A(J,I)/DIV
C**** CHANGE FUNCTION NAME  IN NEXT       STATEMENT  IN SINGLE PRECISION
      CWVEC(J)=DCONJG(WVEC(J))
   85 CONTINUE
      SCALAR=0.0
      DO 95 J=I1,N
      PVEC(J)=0.0
      DO 90 K=I1,N
   90 PVEC(J)=PVEC(J)+A(K,J)*CWVEC(K)
      SCALAR=SCALAR+PVEC(J)*WVEC(J)
   95 CONTINUE
C**** REMOVE OR MODIFY NEXT       STATEMENT  IN SINGLE PRECISION VERSION
      SCALAR=SCALAR/(2.0D0,0.0D0)
      DO 120 J=I1,N
      QVEC(J)=PVEC(J)-SCALAR*CWVEC(J)
      DO 120 K=I1,N
      A(K,J)=A(K,J)-WVEC(K)*QVEC(J)
  120 CONTINUE
      DO 180 K=1,N
      QVEC(K)=-SCALAR*WVEC(K)
      DO 170 J=I1,N
  170 QVEC(K)=QVEC(K)+A(K,J)*WVEC(J)
      DO 180 J=I1,N
      A(K,J)=A(K,J)-QVEC(K)*CWVEC(J)
  180 CONTINUE
C
  200 CONTINUE
C
      RETURN
      END
      SUBROUTINE QR1(A,N,DEL,ZET,NMAX)
C
C     SINGLE COMPLEX QR ITERATION
C
      DIMENSION A(NMAX,N)
      COMMON /QR/ DUMMY(120),NU(120),MU(120),CMU(120)
C**** REMOVE OR MODIFY NEXT  TWO  STATEMENTS IN SINGLE PRECISION VERSION
      COMPLEX*16 A,ZET,MU,CMU,DIAG,W,Y,Z
      REAL*8     KAP,NU,SUPERD,TEMP1,DUMMY,DEL
      INTEGER    Q
C
      N1=N-1
      N2=N1-1
C
C     FIND Q
C
      DO 50 I1=1,N2
      I=N1-I1
      IF (CDABS   (A(I+1,I)) .LE.DEL) GO TO 60
   50 Q=I
C
C     SHIFT ORIGIN
C
   60 DO 100 I=Q,N
  100 A(I,I)=A(I,I)-ZET
C
C     REDUCE TO TRIANGLE (ROWS)
C
      DO 200 I=Q,N
      DIAG=A(I,I)
      SUPERD=0.0
      IF (I.EQ.N) GO TO 150
      SUPERD=A(I+1,I)
      A(I+1,I)=0.0
C**** CHANGE FUNCTION NAMES IN NEXT  TWO  STATEMENTS IN SINGLE PRECISION
  150 TEMP1=DIAG*DCONJG(DIAG)
```

```
      KAP=DSQRT(TEMP1+SUPERD**2)
      MU(I)=DIAG/KAP
C**** CHANGE FUNCTION NAME  IN NEXT       STATEMENT  IN SINGLE PRECISION
      CMU(I)=DCONJG(MU(I))
      NU(I)=SUPERD/KAP
      A(I,I)=KAP
      Y=MU(I)
      Z=CMU(I)
      IF (I.EQ.N) GO TO 250
      I1=I+1
      DO 200 J=I1,N
      W=A(I,J)
      A(I,J)=W*Z      +NU(I)*A(I+1,J)
      A(I+1,J)=Y     *A(I+1,J)-NU(I)*W
  200 CONTINUE
C
C     INVERSE OPERATION (COLUMNS)
C
  250 DO 300 J=Q,N
      Y=MU(J)
      Z=CMU(J)
      IF (J.EQ.N) GO TO 350
      J1=J+1
      DO 300 I=Q,J1
      W=A(I,J)
      A(I,J)=W*Y     +NU(J)*A(I,J+1)
      A(I,J+1)=Z      *A(I,J+1)-NU(J)*W
  300 CONTINUE
  350 DO 400 I=Q,N
      A(I,J)=Y     *A(I,J)
  400 A(I,I)=A(I,I)+ZET
      RETURN
      END
      FUNCTION RANN(NR)                                                 2
                                                                        2
C     ROUTINE TO FORM A SEQUENCE OF NORMALLY DISTRIBUTED PSUEDO-        2
C     RANDOM NUMBERS WITH ZERO MEAN AND UNIT STANDARD DEVIATION         2
C                                                                       2
      CALL RANSET(NR)                                                   3
      SUM=0.                                                            4
      DO 10 I=1,27                                                      5
      SUM=SUM+RANF(-1)                                                  6
   10 CONTINUE                                                          7
      SUM=SUM/27.                                                       8
      RANN=18.*(SUM-.5)                                                 9
      CALL RANGET(NR)                                                  10
      RETURN                                                           11
      END                                                              12
      SUBROUTINEAVERAGE(A,N,NKK,AV,AVSQ)
C                                                                       2
C      THIS PROGRAM HAS BEEN TRANSLATED FOR THE     360/50             2
C      WITH RELEASE 1-A OF THE MOD-50 TRANSDECK                        2
C                                                   JDB                2
C                                                                       3
      DIMENSIONA(1)                                                     4
      AV=0.                                                             5
      AVSQ=0.
      NKK1=NKK+1
      NKK2=N-NKK
      DO 10 I=NKK1,NKK2                                                 7
      AV=AV+A(I)                                                        8
      AVSQ=AVSQ+A(I)**2                                                 9
   10 CONTINUE
      AV=AV/(N-2*NKK)
      AVSQ=AVSQ/(N-2*NKK)                                              12
      RETURN                                                           13
      END
```

```
      SUBROUTINEORDER(A,N)
C
C      THIS PROGRAM HAS BEEN TRANSLATED FOR THE      360/50
C      WITH RELEASE 1-A OF THE MOD-50 TRANSDECK
C                                                        JDB
      DIMENSIONA(1)
      REAL*8 A,DUM
      NM1=N-1
      DO20I=1,NM1
      DO20J=I,N
      IF(A(I)-A(J))20,20,10
   10 CONTINUE
      DUM=A(I)
      A(I)=A(J)
      A(J)=DUM
   20 CONTINUE
      RETURN
      END
      SUBROUTINE ORDER1(A,NSA,N)
C
C      THIS PROGRAM HAS BEEN TRANSLATED FOR THE      360/50
C      WITH RELEASE 1-A OF THE MOD-50 TRANSDECK
C                                                        JDB
      DIMENSION A(1),NSA(1),SAV(120)
      REAL*8 A,SAV
      DO 10 I=1,N
      SAV(I)=A(I)
   10 CONTINUE
      CALL ORDER(A,N)
      DO 50 I=1,N
      DO 20 J=1,N
      IF(A(I)-SAV(J))20,40,20
   20 CONTINUE
      PRINT 30
   30 FORMAT(16H1ERROR IN ORDER1)
      STOP
   40 CONTINUE
      NSA(I)=J
   50 CONTINUE
      RETURN
      END
      SUBROUTINE HISTGRAM(A,NAP,NAM,DX,NMAX)
C
C      THIS PROGRAM HAS BEEN TRANSLATED FOR THE      360/50
C      WITH RELEASE 1-A OF THE MOD-50 TRANSDECK
C                                                        JDB
      DIMENSION NAP(1),NAM(1)
      N=ABS(A)/DX
      N=N+1
      IF(N-NMAX)10,10,40
   10 CONTINUE
      IF(A)20,30,30
   20 CONTINUE
      NAM(N)=NAM(N)+1
      GO TO 40
   30 CONTINUE
      NAP(N)=NAP(N)+1
   40 CONTINUE
      RETURN
      END
      FUNCTION RANF(J)
C     RANDOM NUMBER GENERATOR OF FORM X(I+1)=X(I)*(2**16+11) MOD 2**31
      EQUIVALENCE (X,IX)
      DATA IX/3125/
    2 IX=IX*65547
      IF(IX)5,6,6
```

```
    5 IX=IX+2147483647+1
    6 IF (J.GE.0) GOTO 8
      IF (IX.LT.8388608) GOTO 7
      YFL=IX/8
      RANF=YFL*.3725291E-8
      RETURN
    7 YFL=IX
       RANF=YFL*.4656613E-9
      RETURN
    8 RANF=X
      RETURN
      ENTRY IRANF(J)
   12 IX=IX*65547
      IF(IX)15,16,16
   15 IX=IX+2147483647+1
   16 IRANF=IX
      RETURN
      ENTRY RANSET(J)
      IX=J
      RANSET=0.
      RETURN
      ENTRY RANGET(J)
      J=IX
      RANGET=0.
      RETURN
      END
      SUBROUTINEESUB(NN,E,NOPT)                                          2
C                                                                        2
C       THIS PROGRAM HAS BEEN TRANSLATED FOR THE     360/50              2
C       WITH RELEASE 1-A OF THE MOD-50 TRANSDECK                         2
C                                                   JDB                  2
C  IF NOPT=3 , E(I) CHOSEN FROM WIGNER DISTRIBUTION
      DIMENSIONE(120)
      COMMON DBAR
      REAL*8 E
      SQRTF(X)=SQRT(X)                                                   4
      IF(NOPT-1)30,10,1                                                  5
    1 IF (NOPT -3) 23,24,23                                              6
   10 CONTINUE                                                           7
      TERM=NN+1                                                          8
      TERM=TERM/2.                                                       9
      DO20I=1,NN                                                        10
      E(I)=I-TERM                                                       11
   20 CONTINUE
      DBAR=1.0                                                          12
      GO TO 30                                                          13
   23 CONTINUE                                                          14
      READ(60,25)(E(I),I=1,NN)                                          15
   25 FORMAT(8F10.0)
      READ(60,1999)DBAR
 1999 FORMAT(E12.6)                                                      16
      GO TO 30                                                          17
   24 READ(60,26) DBAR,ENUT,NR
   26 FORMAT(2F10.4,I10)                                                19
      CALL RANSET(NR)
      E(1)=ENUT+DBAR * SQRTF(-ALOG(RANF(-1)))*1.12837916/2.
      DO 33 I=2,NN                                                      21
   33 E(I)=E(I-1)+ DBAR * SQRTF(-ALOG(RANF(-1)))*1.12837916             22
   30 CONTINUE                                                          23
      RETURN                                                            24
      END                                                               25
```

## REFERENCES


1. E. P. Wigner and L. Eisenbud, Phys. Rev. *72*, 29 (1947).

2. H. B. Willard, L. C. Biedenharn, P. Huber, and E. Baumgartner, *Fast Neutron Physics, Part II*, p. 1217, J. B. Marion and J. Fowler, editors, Interscience Publishers, Inc. (1963).

3. J. Humblet and L. Rosenfeld, Nucl. Phys. *26*, 529 (1961).

4. D. B. Adler and F. T. Adler, "Neutron Cross Sections in Fissile Elements," *Proceedings of the Conference on Breeding, Economics, and Safety in Large Fast Power Reactors*, October 7-10, 1963, pp. 695-709, ANL-6792 (Dec 1963).

5. F. T. Adler and D. B. Adler, *Proc. National Topical Meeting on Reactor Physics in the Resonance and Thermal Regions*, p. 47, MIT Press, Cambridge (1966).

6. F. T. Adler and D. B. Adler, *Proc. Neutron Cross Sections Technol. Conf.* p. 873, CONF-660303, Book 2 (1966).

7. D. B. Adler and F. T. Adler, Trans. Am. Nucl. Soc. *5*(1), 53 (1962).

8. R. N. Hwang, Nucl. Sci. Engr. (to be published).

9. R. N. Hwang, Nucl. Sci. Engr. *36*, 67 (1969).

10. R. N. Hwang, Nucl. Sci. Engr. *36*, 82 (1969).

11. P. A. Moldauer, Phys. Rev. *135*, B642 (1964).

12. P. A. Moldauer, Phys. Rev. *136*, B947 (1964).

13. P. A. Moldauer, Phys. Rev. *171*, 1164 (1968).

14. P. A. Moldauer, *Interfering Resonances*, ANL-7467 (July 1968).

15. P. A. Moldauer, Phys. Rev. Letters *18*, 249 (1967).

16. L. W. Ehrlich, *MATSUB, Eigenvalues and Eigenvectors of Complex Non-hermitian Matrices*, University of Texas Computation Center (1961).

17. B. S. Garbow, *FRANCC, Eigenvalues of Complex Matrices*, ANL-F251S (1968).

18. P. A. Moldauer, Rev. Mod. Phys. *36*, 1079 (1964).

19. D. L. Hill and J. A. Wheeler, Phys. Rev. *89*, 1102 (1953).

20. N. W. Clark, *RANF, OS/360 FORTRAN IV Routine to Produce a Series of Random Numbers*, ANL-G552S (1967).

21. M. J. Seaton, Computer Journal *12*, No. 2, 156 (1969).

ARGONNE NATIONAL LAB WEST
3 4444 00007860 0

t

B