N70-20135
CR-102455

Final Report

Contract No. NAS8-21131

Covering period Nov. 4, 1968-Nov. 3, 1969

NATIONAL AERONAUTICS AND
SPACE ADMINISTRATION

USER'S GUIDE SOCDES

by John F. Cassidy, Jr.

M7FC

CASE FILE
COPY

Rensselaer Polytechnic Institute

Troy, New York

(ACCESSION NUMBER) (THRU)
88 1
(PAGES) (CODE)
(NASA CR OR TMX OR AD NUMBER) (CATEGORY)

Rensselaer Polytechnic Institute
Troy, New York 12181

Final Report

Contract No. NAS8-21131

Covering period Nov. 4, 1968-Nov. 3, 1969

NATIONAL AERONAUTICS AND
SPACE ADMINISTRATION

USER'S GUIDE SOCDES

by John F. Cassidy, Jr. MSFC

CASE FILE
COPY

Submitted on behalf of

Rob Roy
Professor of Systems Engineering

USER'S GUIDE: SOCDES

by

John F. Cassidy, Jr.

I. Introduction

A. Purpose: SOCDES is a digital computer program written in Fortran IV which solves a variety of SOC and SOC related time invariant, semi-infinite time interval optimal control problems. It will handle SOC and SOC sensitivity problems with Reverse problem and filter design options. Moreover it can solve the allstate problem.

B. Method: The necessary conditions defining the various SOC problems are solved via Newton-Raphson iteration.[1] The Lyapunov stability equations are solved by an eigensystem approach. The allstate problem is solved by Kleinman's method.[2]

C. Structure: The program consists of a main and 14 subroutines which are briefly described as follows.

MAIN - Initializes; controls various subroutines; inputs, outputs; calculates gain function, jacobian and stopping tolerances; and evaluates cost index.

DINP - Allows flexible input of matrix data.

LYPEIG - Solves general Lyapunov stability equation via eigensystem approach.

CALETA - Calculates ETA matrix used in LYPEIG.

HESSEN - Calculates Hessenberg form of matrix prior to allow efficient calculation of eigenvalues by QREIG.

QREIG - Calculates eigenvalues for a general non-symmetrix matrix by the QR method.

QRT - Used by QREIG.

HSQR - Squares a given matrix prior for use by EIGUEC.

EIGVEC - Finds eigenvectors for a general non-symmetric matrix by the Inverse Iteration method.

DLIN - Solves linear algebraic simultaneous equations or inverts matrices.

AQCAL - Allows flexible readin or calculation of the sensitivity tensor, AQ

RIMULT - Multiplies complex numbers.

PSOLV - Calculation Ricatti matrix for a given set of feedback gains.

RAYL - Calculates improved estimates of eigenvalues by Rayleigh coefficient method.

RICKA - Solves the allstate optimal control problem and Ricatti equation.

II Problem Definitions

SOCDES solves three basic problems which are defined below. In all cases the linear system under consideration is described by

$$\dot{\underline{x}} = A\underline{x} + B\underline{u} \; ; \; \underline{x}(0) = \underline{c} \qquad (1)$$

where $\underline{x}$ is an NS element state vector, $\underline{u}$ is an NC element control vector and A and B are the system and control coefficient matrices respectively.

A. Allstate Problem

The control, $\underline{u}$, is chosen to minimize, J_1, $J_1 = \frac{1}{2}\int_0^\infty (\underline{x}^T S\underline{x} + \underline{u}^T Q\underline{u})dt$ subject to Eq. 1. The optimal control $\underline{u}^o$ has a linear feedback structure and is given by

$$u = K^T\underline{x} = -Q^{-1}B^T P\underline{x} \tag{2}$$

where P is the solution to the Ricatti equation.

$$A^T P + PA + S - PBQ^{-1}B^T P = 0 \tag{3}$$

Given A, B, S, Q, and an initial stable guess for K, SOCDES solves Eq. 2 and 3 and prints P and K.

B. SOC Problem

The control, $\underline{u}$, is chosen to minimize J_2,

$$J_2 = \frac{1}{2}\int_0^\infty (\underline{x}^T S\underline{x} + \underline{x}^T\hat{S}\underline{x} + \underline{x}^T W\underline{u} + \underline{x}^T\hat{W}\underline{u} + \underline{u}^T Q\underline{u})\, dt$$

subject to Eq. (1) and the SOC control structure constraints. These control constraints are selected by the user and require that each control be linear feedback of only the available states. The program allows a more general control structure than is indicated in reference 1. Let the first NU states be available; then in the case of multiple controls ($NC \geq 2$) a different combination of from 1 to all of the available states may appear in each control. The ith and j^{th} optimal feedback gains are defined by

$$K_{ij} = \left(Q^{-1}\left(B^T P + \frac{W^T}{2}\right)\right)_{ji} \tag{4}$$

where P is the solution to

$$A_K^T P + P A_K + S + KQK^T = 0 \tag{5}$$

$$A_K = A - BK^T$$

and K has zero values corresponding to the states that will not appear in the controls. Let IK be a NU by NC matrix which defines the gain structure, i.e.,

$$IK = \begin{cases} 1 & \text{if } i^{th}\ j^{th} \text{ gain is non-zero} \\ 0 & \text{otherwise} \end{cases}$$

Given A, B, Q, S, W and a stable initial guess for K, SOCDES will calculate the optimal value of K by Newton-Raphson iteration of the gain function equation Eq. (6) obtained from the reduced necessary conditions, Eq. (4) and (5) while the Ricatti equation, Eq. (5) acts as a constant.

$$(GFM)_{ij} = (Q^{-1}(B^TP + \frac{W^T}{2}) - K^T)_{ij} = 0 \qquad (6)$$

The values of $\hat{W}$ and $\hat{S}$ are not required since their definitions were used to generate the reduced necessary conditions.

At the user's option, SOCDES will solve the Reverse SOC problem. Given a stable set of feedback gains, SOCDES will find a cost index defining a SOC problem for which the given gains are the optimal solutions. Given K and an arbitrary choice of a positive definite Q and positive semi-definite S, SOCDES will calculate W to complete the definition of the index, $\hat{S}$ and $\hat{W}$ are not calculated but may be found for their respective definitions.

C. Sensitivity Problems

Assume that A and B matrices describing the system are functions of a parameter vector, q, containing NPA parameters (NPA ≥ 3). The parameters have constant but unknown values which lie somewhere near the

known nominal values. The first order trajectory sensitivity variables, $\underline{z}_i$, are defined as follows.

$$\underline{z}_i = \frac{\partial x}{\partial q_i} \qquad i = 1, \text{NPA}$$

$$\underline{z}_i = A_{K_{q_i}} \underline{x} + A_K \underline{z}_i \; ; \; \underline{z}_i(0) = \underline{0}$$

where $A_K = A - BK^T$

$$A_{K_{q_i}} = \frac{\partial (A - BK^T)}{\partial q_i}$$

An augmented state vector, $\hat{x}$, and augmented control $\hat{u}$ are formal.

$$\hat{x} = \begin{bmatrix} \underline{x} \\ \underline{z}_i \\ \vdots \\ \underline{z}_{npa} \end{bmatrix} \quad ; \quad \hat{u} = \begin{bmatrix} u \\ m_1 \\ \vdots \\ m_{npa} \end{bmatrix}$$

and u and m are required to have the identical specified SOC structure in which neither the unavailable states nor the sensitivity variables are fed back.

$$\underline{u} = - K^T \underline{x} \tag{7}$$

$$m_i = - K^T \underline{z}_i \tag{8}$$

Then $\hat{\underline{u}}$ is chosen to minimize, J_3,

$$J_3 = \frac{1}{2} \int_0^\infty (\hat{\underline{x}}^T S \hat{\underline{x}} + \hat{\underline{x}}^T \hat{S} \hat{\underline{x}} + \hat{\underline{x}}^T W \hat{\underline{u}} + \hat{\underline{x}}^T \hat{W} \hat{\underline{u}} + \hat{\underline{u}} Q \hat{\underline{u}}) \, dt$$

where

$$S = \begin{bmatrix} S^1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0_{npa+1} \\ 0 & \cdots & 0 & S \end{bmatrix} \quad ; \quad S^i \text{ is a NS by NS partition block}$$

and

$$Q = \begin{bmatrix} Q^1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & Q^{npa+1} \end{bmatrix} \quad ; \quad Q^i \text{ is a NC by NC partition block}$$

subject to the dynamics of Eq. (1) and the SOC constraints of Eq. (7) and (8).

Given A, B, Q^i, S^i, W, $A_{K_{q_i}}$, and a stable guess for K, SOCDES will calculate the optimal value of K by Newton Raphson iteration of the gain function equation. The SOC sensitivity problem enjoys all of the features of the regular SOC problem including the Reverse problem and expanded multiple control structure capabilities. The augmented Ricatti matrix can be partitioned into NS by NS blocks. Since some of the blocks are identically zero, only the non-zero blocks are printed with the following convention.

$$\hat{P} = \begin{bmatrix} P^1 & P^{npa+1^T} & \cdots & P^{2npa+1^T} \\ P^{npa+2} & P^2 & 0 \cdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ P^{2npa+1} & 0 \cdots & 0 & P^{npa+1} \end{bmatrix}$$

D. Filter Design Procedure

SOCDES may be used to synthesize optimal compensating filters.[3] The basic dynamics of the filter are chosen by the user and augmented to the plant dynamics. With the aid of the multiple control structure feature, SOCDES solves either SOC problems and produces the poles and zeros of the filter as well as feedback gains.

The general features of the SOCDES program are shown in Block Diagram 1. The options are controlled by the user via data read in at the start of execution. After each problem is solved, the optimal index is evaluated and the average, maximum, and minimum values for any unit length initial conditions are printed. If the regular SOC problem is solved, the user may elect to solve the allstate problem and compare the solutions. The average, maximum and minimum values of the ratio of the SOC to allstate indices is calculated for all initial conditions such that the optimal value of the allstate index is one.[4] The user may elect to create and solve a new problem by introducing new weightings. There weightings may be read in or obtained by adding previously read in weighting perturbations to the present weightings. A basic multiple run option allows the user to place one data set after another to solve different problems with the same run. Note that for a regular SOC problem, W may be read in or calculated by a Reverse problem solution.

III Use of the Program

In this section the input and output variables are defined, the user written subroutines are explained and a sample problem with output is presented.

There are two basic types of input data, program control data and problem data. An alphabetical listing of program control variables appears in Table 1 while the problem data is defined in Table 2. The flexible input scheme of SOCDES allows the user to modify two subroutines, AQCAL and DINP. Subroutine AQCAL zeros AQ and then branches on the data set

number, IC1. Since in general AQ is a sparse matrix the user may specify the non-zero elements of AQ at the proper branch points or elect to read in AQ in the format of his choice. Subroutine DINP reads in by columns A, B, K, S, and Q in a standard 5E15.5 format. However if long problems are solved, NS $\geq$15, then a more compact format such as the A format may be used. This format may be changed and this small subroutine can be compiled without the recompilation of the rest of SOCDES. The standard format for input data is described in Table 3. After read in most of the data is printed out for checking purposes and future reference. Also, the user may label each run with up to 160 characters.

Example Problem

Consider a second order linear system with two inaccurately known plant parameters, a and b. Assume that the first state is available and that a and b have nominal values of zero.

$$\dot{\underline{x}} = A\underline{x} + Bu \quad ; \quad u = -k\,x_1$$

$$A = \begin{bmatrix} a & 1 \\ -b & -2 \end{bmatrix} \quad ; \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Let k = 2 be the initial guess for the feedback gain. Let us solve the Reverse problem for the following weightings and update the sensitivity weighting twice by the identity matrix and solve the corresponding SOC sensitivity problems.

$$S^1 = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} \quad ; \quad S^2 = S^3 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

$$DS^1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad ; \qquad DS^2 = DS^3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

We will allow a maximum of 15 iterations, print as much as possible after each iteration, and stop when the gain tolerance is below .0001. To run this problem the data cards to be used are described in Table 4 and standard versions of AQCAL and DINP are shown in Tables 5 and 6. The letter b indicates a blank in a numerical data field. The actual data cards are shown in Fig. 1 and the actual output is given in Fig. 2.

IV Restrictions

A. The original feedback gain guess for each problem must correspond to a stable closed loop system. The guess may be zero if the open loop system is stable. If ICASE $\geq$ 1, and no numerical problems are encountered, the stable gains from the previous problem will be used to initiate the new problem. This restriction is necessary for convergence of the Newton Raphson algorithm.

B. The state and control weightings must be diagonal. This is a programming restriction made to conserve input-output effort. For most problems diagonal weightings are sufficient.

C. The present dimensions of the program impose the following limits.

$$NS \leq 25, \quad NC \leq 25, \quad NPA \leq 3, \quad NJ \leq 25$$

D. If the closed loop system, $A_K = A - BK^T$, has repeated roots, the program may fail, since a full set of independent eigenvectors may not exist. If EIGVEC encounters repeated eigenvectors, it will print a warning message and attempt to find a full set of eigenvectors.

E. In order to insure the existence of a solution to a SOC problem, the control weighting, Q^i, must be positive definite, and the state weighting, S^i, must be positive semi-definite and observable. Since S^i is positive semi-definite, it can be factored.

$$S^i = H^T H$$

S^i is said to be observable if (H,A) are an observable pair as defined by Kalman.[4]

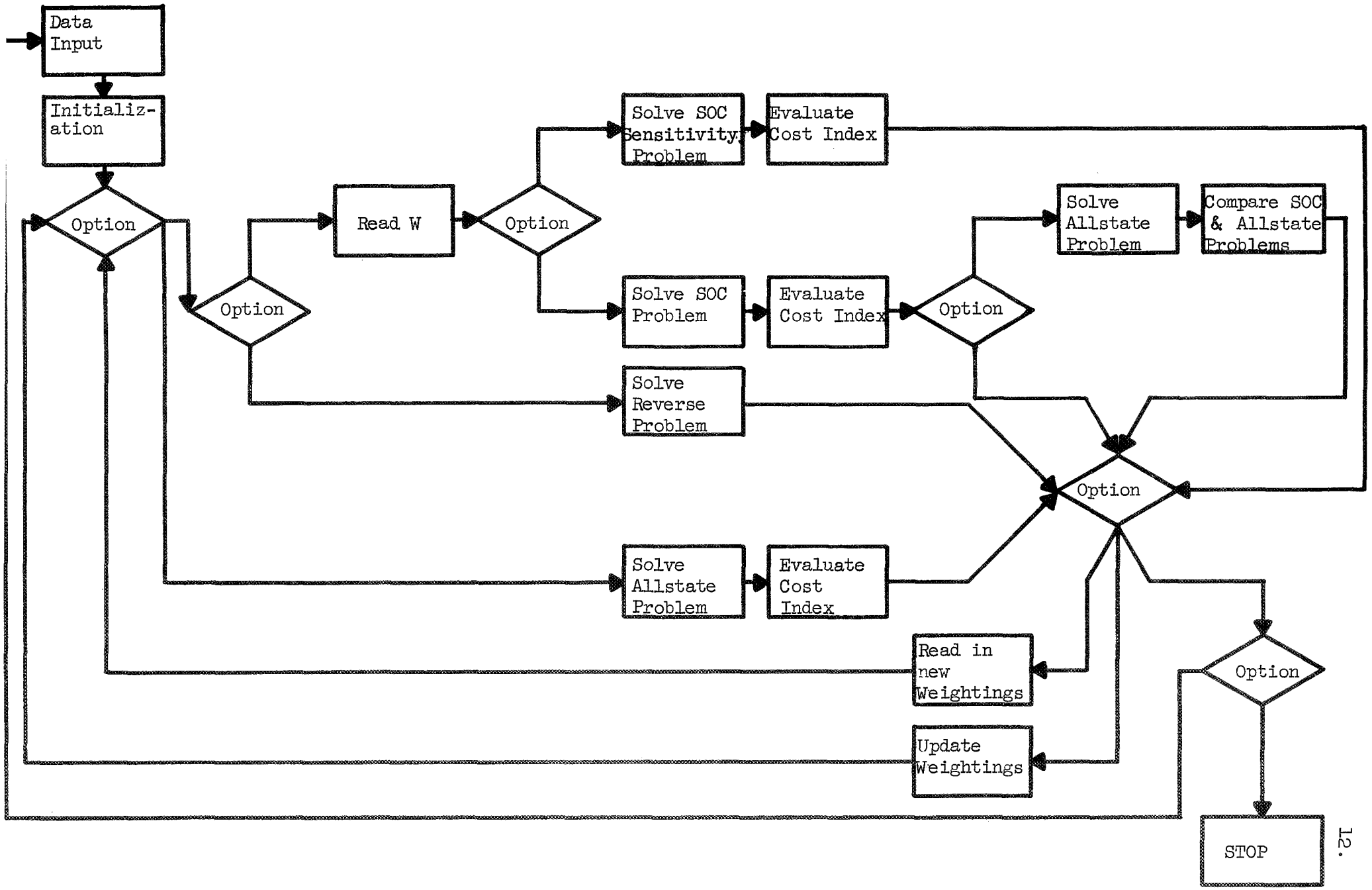
V. User Hints

A. The choice of the gain stopping tolerance may require some care. For some problems on some computers with short word lengths, round-off and truncation errors may prevent the Newton Raphson algorithm from reducing the gain function below a certain value. SOCDES uses a percentage gain change stopping tolerance rather than a gain function norm, since the former is less sensitive to problems of this type. It is usually more efficient to pick a large tolerance .05 or .01 and then reduce it to 1.10^{-3} or 1.10^{-5} on later runs if no numerical problems develop.

B. If state, control, and cross term weightings are chosen arbitrarily, there is no guarantee that a solution to the SOC problem exists. A conservative procedure is to solve the Reverse problem and then probe the stable gain space by perturbing the weightings.

C. Care must be taken in the choice of the weightings. Aside from theoretical considerations, an improper choice of weightings matrices may lead to numerical problems. If the initial choice of S^i and Q^i are arbitrary such as in the Reverse problem, then the following suggestion may

reduce numerical problems. After the choice of an initial gain matrix is made, then the elements of the weighting matrices may be chosen so that the norms of the state weighting matrices and KQ^iK^T are of the same order of magnitude. For the example problem, the given weightings would be a poor choice with an initial guess of $k = 50$. In this case $Q^1 = Q^2 = Q^3 = .001$ would be a much better choice.



BLOCK DIAGRAM 1

TABLE 1 PROGRAM CONTROL DATA

Symbol	Type	Value	Definition
INCASE	Integer	-	The number of different weightings sets and linear problems to be solved for a particular data set. See REVPRB definition.
ICR	Integer	0 1	Weighting increments are read in and weightings are updated. New weightings are read in.
IC1	Integer	-	Data set number.
IC2	Integer	-	Number of month: current date.
IC3	Integer	-	Number of day.
IC4	Integer	-	Last two digits of year
IIRAY	Integer	-	Number of times Rayleigh coefficients to be used.
IK	Integer matrix	-	Gain indicator matrix ; defined in Section II.
ILAB	Integer vector	-	Data set label
IPRT	Integer		Print control, After each iteration, print :-
		2	Iteration number, current and stopping gain tolerances, current gains and gain change, and eigenvalues of closed loop system.
		1	Above plus gain function and Jacobian.
		0	Above plus lower triangular portion of symmetric Ricatti matrix.
			After problem is solved, print:-

TABLE 1 con't

		2	Iterations needed to solve problem, final and stopping gain tolerances, gains, and eigenvalues of closed loop systems.
		1	Above plus gain function matrix.
		0	Above plus lower triangular portion of symmetric Ricatti matrix.
IRIC	Integer		Allstate solution control.
		0	No solution
		1	Solve allstate for comparison with SOC solution.
		2	Solve allstate problem only.
KSTOL	Real	-	Gain stopping tolerances. Iterations cease when % change of each gains is less than KSTOL.
MAXIT	Integer	-	Maximum number of iterations if tolerance is not obtained after MAXIT iterations warning message is printed and program solves current values and goes on to next step.
REVPRB	Integer		Reverse problem control
		0	Reverse problem is not solved and W is read in. If only the allstate problem is to solved (IRIC=2), then REVPRB should be 0 and all zero values of W(NU by NC) should be read in.
		1	Reverse problem is solved. If ICASE > 0, new weightings are obtained under control of ICR.

TABLE 2 PROBLEM DATA

Symbol	Type	Definition
A	Real matrix	System matrix, $\dot{\underline{x}}=A\underline{x} + B\underline{u}$
AQ	Real matrix	$AQ(i,j,l) = \dfrac{\partial(A-BK^T)ij}{\partial q_l}$
B	Real matrix	Control coefficient matrix
K	Real matrix	Feedback gain matrix (NU by NC). The zero gains corresponding to the unavailable states are not read in. If a control does not depend on all of the available states, zeros are read in at the proper locations.
NC	Integer	Number of controls.
NJ	Integer	Total number of non-zero feedback gains.
NPA	Integer	Number of parameters, NPA $\leq$ 3. If NPA = 0, then SOC problem is solved.
NS	Integer	Number of state variables.
NU	Integer	Number of available states.
Q	Real matrix	Control weighting. To reduce input effort, diagonal weighting matrices are assumed. $Q(i,j) = Q_{ii}{}^{j}$
S	Real matrix	State weighting $S(i,j) = S_{ii}{}^{j}$
W	Real matrix	Cross term weighting. Since elements of W corresponding to the unavailable states are zero, only the upper NU by NC portion of W is considered.

TABLE 3 STANDARD INPUT FORMAT

Number of Cards	Variable	Format	Comments
1	IC1, IC2, IC3, IC4	4I2	Data set number and date
2	ILAB	20A4	Up to 160 characters
1	NS, NC, NU, NJ, NPA	16I5	See Table 2
1	REVPRB, ICASE, ICR, IRIC, IIRAY	16I5	See Table 1
$\frac{NC \cdot NU}{16}$	IK	16I5	NC·NU elements Table 1
1	MAXIT, IPRT, KSTOL	2I5, E20.0	See Table 1
$\frac{NS^2}{5}$	A*	5E15.5	5 elements per card Table 2
$\frac{NC \cdot NS}{5}$	B*	5E15.5	See Table 2
$\frac{NC \cdot NU}{5}$	K*	5E15.5	Upper NU by NC portion of feedback gain matrix
$\frac{NS(NPA+1)}{5}$	S*	5E15.5	Diagonal state weighting Table 2
$\frac{NC(NPA+1)}{5}$	Q*	5E15.5	Diagonal control weighting
-	AQ	-	If Subroutine AQCAL is modified to read in AQ, then data must be placed here.
$\frac{NC \cdot NU}{5}$	W	5E15.5	If Reverse Problem is not solved (REVPRB=0), then upper NU by NC portion of W is read in here.

*These variables are read in via Subroutine DINP.

$\frac{NS(NPA+1)}{5}$	DS	5E15.5	If ICR=0 diagonal elements of DS read in here. S=S+DS
$\frac{NC(NPA+1)}{5}$	DQ	5E15.5	If ICR=0 diagonal elements of DQ read in here. Q=Q+DQ
$\frac{NC \cdot NU}{5}$	DW	5E15.5	If ICR=0, upper NC·NU portion of DW read in here. W=W+DW
$\frac{NS(NPA+1)}{5}$	S**	5E15.5	If ICR=1, read new S
$\frac{NC(NPA+1)}{5}$	Q**	5E15.5	If ICR=1, read new Q
$\frac{NC \cdot NU}{5}$	W**	5E15.5	If ICR=1, read new W

**If the Reverse problem is not solved then these variables are not read in for the first problem. Consistent with ICASE additional values of S,Q, and W are placed after the first set.

TABLE 4 SAMPLE PROBLEM DATA

Card Number	Data	Format	Comment
1	b1b8b169	4I4	Run number and date
2	"2nd Order Sample Problem"	20A4	label card
3	-		blank label card
4	2,1,1,1,2	16I5	NS,NC,NU,NJ,NPA
5	1,2,0,0,0	16I5	REVPRB,ICASE,ICR,IRIC IIRAY
6	1	16I5	IK
7	15,0,.0001	2I5,E20.0	MAXIT,IPRT,KSTOL
8	0,0,1,-2	5E15.5	A
9	0,1	"	B
10	2	"	K
11	3,3,1,2,1	"	S
12	2	"	S
13	1,1,1	"	Q
14	0,0,1,1,1	"	DS
15	1	"	DS
16	0,0,0	"	DQ
17	0	"	DW

TABLE 5 AQCAL

```
      Subroutine AQCAL (AQ,NS,NPA,IC1)
      Dimension  AQ(10,10,3)
      DO 10 L = 1,NPA
      DO 10 I = 1, NS
      DO 10 J = 1,NS
10    AQ(I,J,L) = 0.0
      Go to (100,200,300,400,500), IC1
100   Continue
      AQ(1,1,2) = 1.0
      AQ(2,1,1) =-1.0
      Return
200   Continue
      Return
300   Continue
      Return
400   Continue
      Return
500   Continue
      Return
      End
```

TABLE 6 DINP

```
      Subroutine DINP (A,B,K,S,Q,NS,NC,NU,NOP)
      Real K(25,25)
      Dimension A(25,25), B(25,25), S(25,25), Q(25,25)
15    Format (5E15.5)
      IO1=1
      Read (IO1,15) ((A(I,J), I=1,NS),  J=1,NS)
      Read (IO1,15) ((B(I,J), I=1,NS),  J=1,NC)
      Read (IO1,15) ((K(I,J), I=1,NU),  J=1,NC)
      Read (IO1,15) ((S(I,J), I=1,NS),  J=1,NOP)
      Read (IO1,15) ((Q(I,J), I=1,NC),  J=1,NOP)
      Return
      End
```

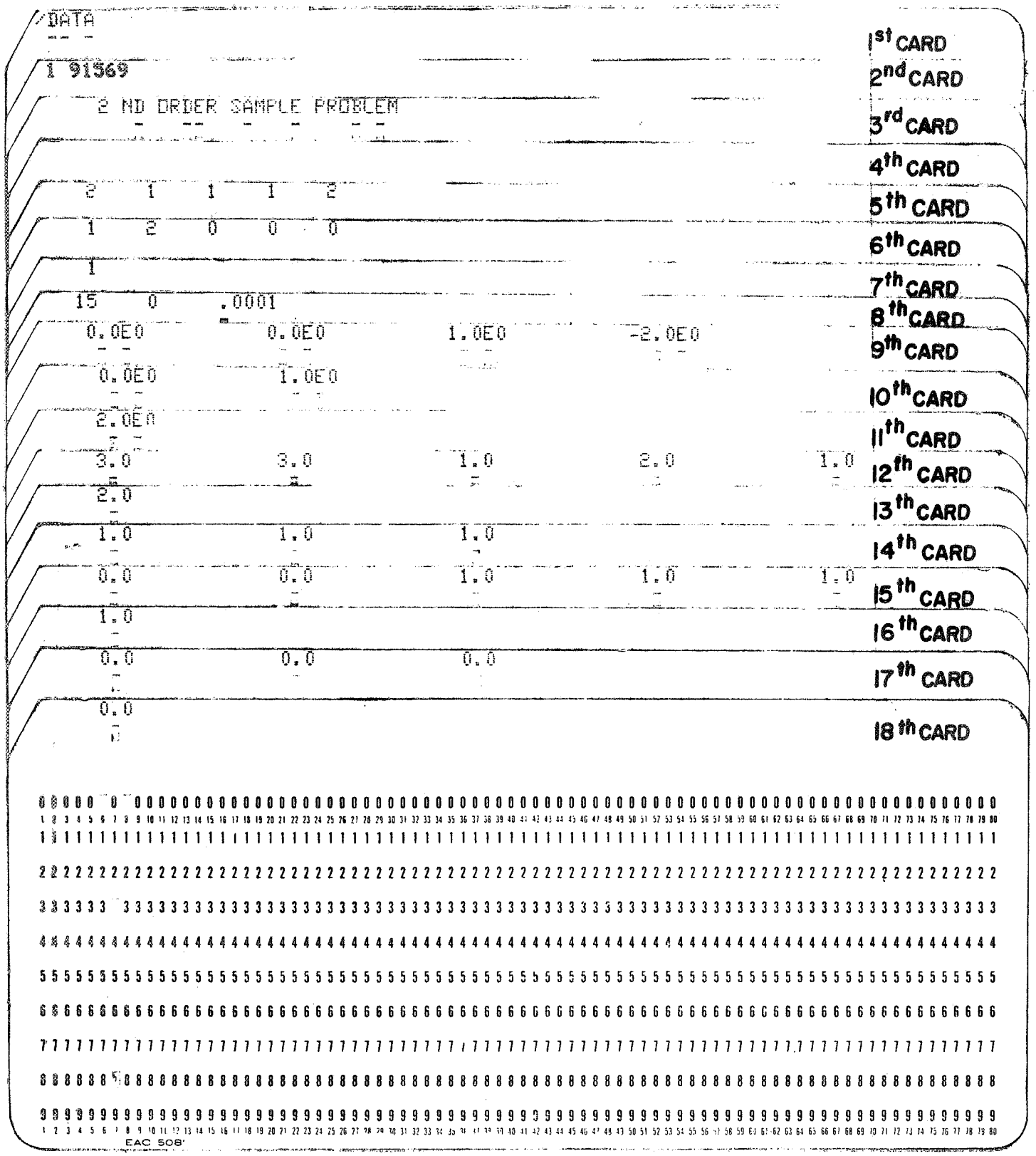


Figure 1 - Data Cards Used for Input to Sample Problem

```
                    SCCDES PRCGRAM     **     RENSSELAFR POLYTECHNIC INSTITUTE     **          J.F.CASSIDY

                EIGEN-SYSTEM VERSICN       ***        JULY 1969



****     RUN NUMBER  1                          9/15/69     ****

                    2 ND ORDER SAMPLE PRCBLEM


                ***     INPUT CATA    ***

      NUMBER CF STATES =   2
      NUMBER CF CCNTROLS =   1
      THE FIRST   1 STATES WILL BE FEC BACK



      THE RAYLEIGH CCEFFICIENTS WILL BE USEC   0 TIMES.



      THERE ARE   1 NON-ZERC FEEDBACK GAINS.
      MAXIMUM NUMBER CF ITERATICNS =  15
      PRINT CCNTRCL =   0
      FEEDBACK GAIN STOPPING TCLERANCE =    0.1000E-03

                                  *      GAIN INDICATCR MATRIX      *

 1

                                  *     A     *


 C.CCCCE-C1   1.CCCCE CC
```

Figure 2 Sample Problem Output

```
C.CCCCE-C1  -2.CCCOE OC

                                    *     B'     *

C.CCCCE-01   1.CCCCE CC

                                    *     K'     *

2.CCCCE CO

                                    *     DIAGCNAL ELEMENTS OF S     *

3.CCCCE CC   3.CCCCE CC   1.C000E 00   2.0000E 00   1.0000E 00   2.0000E 00

                                    *     DIAGONAL ELEMENTS OF Q     *

1.CCCOE CO   1.CCCCE CC   1.C000E 00

                                    *     AC - PARAMETER NUMBER   1     *

C.CCCCE-C1   C.CCCCE-C1

-1.CCCOE CC   C.CCCCE-C1

                                    *     AC - PARAMETER NUMBER   2     *
```

Figure 2 Sample Problem Output (continued)

```
1.CCCOE CC   C.CCCCE-01

C.CCCCE-C1   C.CCCCE-01


                                    ****     REVERSE PROBLEM SOLUTION     ****


                                    *     W*     *


-3.C312E CC


                          ***     CHARACTERISTIC RCOTS OF CLOSED LOOP SYSTEM.     ***
          REAL        IMAGINARY                    REAL        IMAGINARY

       -1.CCCC         1.CCCC                  -1.0000        -1.0000
```

Figure 2 Sample Problem Output (continued)

```
                    **     NEW WEIGHTINGS     **

                                    *     DIAGONAL ELEMENTS OF S     *

  3.0000E 00   3.0000E 00   2.0000E 00   3.0000E 00   2.0000E 00   3.0000E 00

                                    *     DIAGONAL ELEMENTS OF Q     *

  1.0000E 00   1.0000E 00   1.0000E 00

                                    *     W'     *

 -3.0312E 00

******************
                    ITERATION NUMBER   1

                                    GAIN TOLERANCE/STOPPING TOLERANCE =    0.1073E 00/   0.1000E-03

                                    *     K'     *

  2.2403E 00

                                    *   TRANSPOSED GAIN INCREMENT MATRIX   *
```

Figure 2 Sample Problem Output (continued)

```
2.4C31E-C1

                    ***    CHARACTERISTIC RCCTS OF CLOSED LCOP SYSTEM.    ***
        REAL        IMAGINARY                  REAL        IMAGINARY

      -1.CCCC        1.1137                  -1.0000       -1.1137

                              *    TRANSPOSED GAIN FUNCTION MATRIX    *

5.2877E-C2

                              *   JACCBIAN MATRIX   *

-2.C156E CC

                              **    LCWER TRIANGULAR PORTION OF RICATTI MATRIX.    **

1.2315F C1

3.8C88E CC   2.6544E CC

                              *    LCWER TRIANGULAR PORTION OF RICATTI MATRIX.        BLOCK  2    *

6.568CE CC

1.5665E CC   1.5333E CC
```

Figure 2 Sample Problem Output (continued)

```
                                        *     LCWER TRIANGULAR PORTION OF RICATTI MATRIX.          BLOCK  3     *

 6.568CE CC

 1.5665E CC   1.5233E OC

                                        *     LCWER TRIANGULAR PORTION OF RICATTI MATRIX.          BLOCK  4     *

-3.2424E-C1

-7.3294E-C1  -1.7481E-C1

                                        *     LCWER TRIANGULAR PORTION OF RICATTI MATRIX.          BLOCK  5     *

 3.79C5E CC

 1.8575E CC   7.3294E-C1

******************

******************
                 ITERATICN NUMBER   2

                                        GAIN TCLERANCE/STCPPING TOLERANCE =    0.1448E-01/   0.1000E-03

                                        *     K'     *
```

Figure 2 Sample Problem Output (continued)

```
2.2732E CC

                    *   TRANSPOSED GAIN INCREMENT MATRIX   *

3.2925E-C2

                 ***    CHARACTERISTIC RCCTS OF CLOSED LOOP SYSTEM.    ***
      REAL        IMAGINARY                  REAL        IMAGINARY

    -1.CCCC        1.1284                  -1.0000        -1.1284

                    *    TRANSPOSED GAIN FUNCTION MATRIX     *

7.2C98E-C4

                    *   JACCBIAN MATRIX   *

-1.6C6CE CC

                    **    LOWER TRIANGULAR PORTION OF RICATTI MATRIX.    **

1.2349E 01

3.7096E CC   2.6448E CC

                    *    LCWER TRIANGULAR PORTION OF RICATTI MATRIX.        BLOCK  2    *
```

Figure 2 Sample Problem Output (continued)

```
 6.6499E 00

 1.5765E 00   1.5383E 00

                                    *     LOWER TRIANGULAR PORTION OF RICATTI MATRIX.          BLOCK  3     *


 6.6499E 00

 1.5765E 00   1.5383E 00

                                    *     LOWER TRIANGULAR PORTION OF RICATTI MATRIX.          BLOCK  4     *


-3.1851E-01

-7.3132E-01  -1.7338E-01

                                    *     LOWER TRIANGULAR PORTION OF RICATTI MATRIX.          BLOCK  5     *


 3.7995E 00

 1.8568E 00   7.3132E-01

******************

******************
                ITERATION NUMBER   3


                                    GAIN TOLERANCE/STOPPING TOLERANCE =    0.2029E-03/   0.1000E-03
```

Figure 2 Sample Problem Output (continued)

```
                                    *    K'    *

2.2737E CC

                                    *   TRANSPOSED GAIN INCREMENT MATRIX   *

4.6142E-C4

                              ***    CHARACTERISTIC ROOTS OF CLOSED LOOP SYSTEM.    ***
          REAL        IMAGINARY                    REAL        IMAGINARY

       -1.CCCC        1.1286                    -1.0000        -1.1286

                                    *    TRANSPOSED GAIN FUNCTION MATRIX    *

3.8147E-C6

                                    *   JACOBIAN MATRIX   *

-1.5625E CC

                                    **    LOWER TRIANGULAR PORTION OF RICATTI MATRIX.    **

1.2350E C1
```

Figure 2 Sample Problem Output (continued)

```
3.7893E 00   2.6447E 00

                                  *    LOWER TRIANGULAR PORTION OF RICATTI MATRIX.        BLOCK  2    *


6.6510E 00

1.5767E 00   1.5383E 00

                                  *    LOWER TRIANGULAR PORTION OF RICATTI MATRIX.        BLOCK  3    *


6.6510E 00

1.5767E 00   1.5383E 00

                                  *    LOWER TRIANGULAR PORTION OF RICATTI MATRIX.        BLOCK  4    *


-3.1844E-01

-7.3130E-01  -1.7336E-01

                                  *    LOWER TRIANGULAR PORTION OF RICATTI MATRIX.        BLOCK  5    *


3.7996E 00

1.8568E 00   7.3130E-01

******************
```

Figure 2 Sample Problem Output (continued)

```
******************

******************

                                    PROBLEM SOLUTION IS COMPLETE.

                    GAIN TOLERANCE WAS SATISFIED AFTER =  4 ITERATIONS.


                            GAIN TOLERANCE/STOPPING TOLERANCE =    0.1074E-05/   0.1000E-03

                            *     K'     *

2.2737E 00

                    ***    CHARACTERISTIC ROOTS OF CLOSED LOOP SYSTEM.    ***
        REAL        IMAGINARY                  REAL        IMAGINARY

     -1.0000        1.1286                  -1.0000        -1.1286

                            *    TRANSPOSED GAIN FUNCTION MATRIX    *

1.9073E-06

                            **    LOWER TRIANGULAR PORTION OF RICATTI MATRIX.    **
```

Figure 2 Sample Problem Output (continued)

```
1.2350E 01

3.7893E 00   2.6447E 00

                                  *    LOWER TRIANGULAR PORTION OF RICATTI MATRIX.        BLOCK  2    *


6.6510E 00

1.5767E 00   1.5383E 00

                                  *    LOWER TRIANGULAR PORTION OF RICATTI MATRIX.        BLOCK  3    *


6.6510E 00

1.5767E 00   1.5383E 00

                                  *    LOWER TRIANGULAR PORTION OF RICATTI MATRIX.        BLOCK  4    *


-3.1844E-01

-7.3130E-01  -1.7336E-01

                                  *    LOWER TRIANGULAR PORTION OF RICATTI MATRIX.        BLOCK  5    *


3.7996E 00

1.8568E 00   7.3130E-01
```

Figure 2 Sample Problem Output (continued)

```
******************

                    **    COST INDEX EVALUATION    **

          THE COST INDEX IS EVALUATED FOR UNIT LENGTH INITIAL CONDITION VECTORS.

     AVERAGE COST =    3.749
     MAXIMUM COST =    6.827
     MINIMUM COST =   0.6702

          **     NEW WEIGHTINGS     **

                                *     DIAGONAL ELEMENTS OF S     *

 3.0000E 00   3.0000E 00   3.0000E 00   4.0000E 00   3.0000E 00   4.0000E 00

                                *     DIAGONAL ELEMENTS OF Q     *

 1.0000E 00   1.0000E 00   1.0000E 00

                                *     W'     *

-3.0312E 00

******************
          ITERATION NUMBER   1

                                GAIN TOLERANCE/STOPPING TOLERANCE =    0.8663E-01/   0.1000E-03
```

Figure 2 Sample Problem Output (continued)

```
2.4894E 00

                    *   TRANSPOSED GAIN INCREMENT MATRIX   *

2.1566E-01

                  ***    CHARACTERISTIC ROOTS OF CLOSED LOOP SYSTEM.    ***
        REAL        IMAGINARY                  REAL        IMAGINARY

     -1.0000        1.2204                 -1.0000        -1.2204

                    *    TRANSPOSED GAIN FUNCTION MATRIX     *

3.4317E-02

                    *   JACOBIAN MATRIX   *

-1.8280E 00

                    **    LOWER TRIANGULAR PORTION OF RICATTI MATRIX.    **

1.3546E 01

4.0393E 00   2.7696E 00
```

Figure 2 Sample Problem Output (continued)

```
                                    *     LCWER TRIANGULAR PORTION OF RICATTI MATRIX.          BLOCK  2     *

  8.4831E CC

  1.8472E CC   1.9236E CC

                                    *     LCWER TRIANGULAR PORTION OF RICATTI MATRIX.          BLOCK  3     *

  8.4831E CC

  1.8472E CC   1.9236E CC

                                    *     LCWER TRIANGULAR PORTION OF RICATTI MATRIX.          BLOCK  4     *

 -2.42C6E-C1

 -8.5193E-C1  -1.8551E-C1

                                    *     LCWER TRIANGULAR PORTION OF RICATTI MATRIX.          BLOCK  5     *

  4.6C49E CC

  2.1657E CC   8.5193E-01

******************

******************
```

Figure 2 Sample Problem Output (continued)

```
ITERATION NUMBER   2

                    GAIN TOLERANCE/STOPPING TOLERANCE =    0.8943E-02/   0.1000E-03

                    *     K*     *

2.5118E 00

                    *   TRANSPOSED GAIN INCREMENT MATRIX   *

2.2463E-02

                  ***    CHARACTERISTIC ROOTS OF CLOSED LOOP SYSTEM.     ***
      REAL        IMAGINARY                  REAL        IMAGINARY

   -1.0000         1.2296                 -1.0000        -1.2296

                    *     TRANSPOSED GAIN FUNCTION MATRIX     *

2.9182E-04

                    *   JACOBIAN MATRIX   *

-1.5277E 00

                    **     LOWER TRIANGULAR PORTION OF RICATTI MATRIX.     **
```

Figure 2 Sample Problem Output (continued)

```
1.3576E 01

4.0277E 00   2.7639E 00

                                  *    LOWER TRIANGULAR PORTION OF RICATTI MATRIX.        BLOCK  2    *


8.5453E 00

1.8531E 00   1.9265E 00

                                  *    LOWER TRIANGULAR PORTION OF RICATTI MATRIX.        BLOCK  3    *


8.5453E 00

1.8531E 00   1.9265E 00

                                  *    LOWER TRIANGULAR PORTION OF RICATTI MATRIX.        BLOCK  4    *


-2.3775E-01

-8.5051E-01  -1.8444E-01

                                  *    LOWER TRIANGULAR PORTION OF RICATTI MATRIX.        BLOCK  5    *


4.6118E 00
```

Figure 2 Sample Problem Output (continued)

)

2.1643E 00 8.5C51E-01

****

Figure 2 Sample Problem Output
(continued)

```
******************

******************

                                    PROBLEM SOLUTION IS COMPLETE.

                    GAIN TOLERANCE WAS SATISFIED AFTER =  3 ITERATIONS.


                          GAIN TOLERANCE/STOPPING TOLERANCE =    0.7734E-04/   0.1000E-03

                          *     K'     *

2.512CE CC

                    ***    CHARACTERISTIC ROOTS OF CLOSED LOOP SYSTEM.     ***
        REAL        IMAGINARY                  REAL        IMAGINARY

      -1.CCCC        1.2296                  -1.0000       -1.2296

                          *     TRANSPOSED GAIN FUNCTION MATRIX     *

C.CCCOE-C1

                          **     LOWER TRIANGULAR PORTION OF RICATTI MATRIX.     **
```

Figure 2 Sample Problem Output (continued)

```
1.3576E 01

4.0276E 00   2.7638E 00

                                  *    LOWER TRIANGULAR PORTION OF RICATTI MATRIX.        BLOCK  2    *


8.5458E 00

1.8531E 00   1.9266E 00

                                  *    LOWER TRIANGULAR PORTION OF RICATTI MATRIX.        BLOCK  3    *


8.5458E 00

1.8531E 00   1.9266E 00

                                  *    LOWER TRIANGULAR PORTION OF RICATTI MATRIX.        BLOCK  4    *


-2.3771E-01

-8.5050E-01  -1.8443E-01

                                  *    LOWER TRIANGULAR PORTION OF RICATTI MATRIX.        BLOCK  5    *


4.6119E 00

2.1643E 00   8.5050E-01
```

Figure 2 Sample Problem Output (continued)

```
******************

                    **    COST INDEX EVALUATION    **

              THE COST INDEX IS EVALUATED FOR UNIT LENGTH INITIAL CONDITION VECTORS.

      AVERAGE COST =    4.085
      MAXIMUM COST =    7.456
      MINIMUM COST =   0.7142

******************

*****     END OF RUN NUMBER  1                    9/15/69     *****

******************
***ERROR***  CONTROL CARD ENCOUNTERED ON UNIT 1 DURING EXECUTION.

     PROGRAMME WAS EXECUTING LINE    62 IN ROUTINE M/PROG WHEN TERMINATION OCCURRED

COMPILE TIME=    73.40 SEC,EXECUTION TIME=    12.65 SEC,OBJECT CODE=   93824 BYTES,ARRAY AREA=   39640 BYTES,UNUSED=   63144 BYTES
        /STOP
```

Figure 2 Sample Problem Output (continued)

References

1. Cassidy, J. "Optimal Control with Unavailable States", NASA Contract NAS8-21131, Final Report Vol. III.

2. Kleinman, D. "On An Iterative Technique For Ricatti Equation Computation", Transactions On Automatic Control, vol. AC-13, pp. 114-115, February 1968.

3. _____ Monthly Progress Report, RPI-NASA contract NAS8-21131 March 4 - April 4, 1969.

4. Jameson, A. "Optimization of Linear System of Constrained Configuration", Grumman Co. Aerodynamics Report 393-68-1.

APPENDIX I

Program Listing

```
/JOB          4188      MCBRINN,LINES=54,PAGES=200
      REAL KTOL,KSTOL
      INTEGER REVPRB
      DOUBLE PRECISION EE(100),EF(100)
      DOUBLE PRECISION CT(10,10),V(10,10),RR(10),RI(10)
      REAL K(10,10),JAC(10,10)
      DIMENSION A(10,10),B(10,10),Q(10,10),S(10,10),QI(10,10),AK(10,10),
     1RIC(10,10),RICI(10,10),DW(10,10),GFM(10,10),PK(10,10),DUMA(10,10),
     1DQ(10,10),SAVK(10,10),BQ(10,10),DUMB(10,10),PKIJ(10,10),BQI(10,10)
     1,DUME(10,10),DUMC(10,10,7),P(10,10,7),DUMY(10,10,3),AQ(10,10,3),W(
     110,10),W2(10,10),DUM(10,10),DS(10,10),IK(10,10)
      DIMENSION ILAB(40)
      COMMON K
      COMMON DUM,RR,RI,V,AQ,B,DUMA,Q,DUMC,TTT
      COMMON IK,NOP,NS,NC,ISYM,INT,NS2,NU,NPA
C     PROGRAM WILL ONLY HANDLE DIAGONAL WEIGHTING MATRICES.
    1 FORMAT(1H1)
   10 FORMAT(20A4)
   15 FORMAT(5E15.5)
 18   FORMAT(20A4)
   20 FORMAT(4E20.4)
  25  FORMAT(16I5)
 30   FORMAT(2I5,E20.4)
   40 FORMAT(1X,40I3)
   50 FORMAT(/,5(/1X,1P10E13.4))
 55   FORMAT(1X,10G13.5)
   60 FORMAT(/,5(/1X,1P10D13.4))
  100 FORMAT(T25,'SOCDES PROGRAM      **      RENSSELAER POLYTECHNIC'
     A' INSTITUTE      **           J.F.CASSIDY')
 101  FORMAT(//T20,'EIGEN-SYSTEM VERSION        ***         JULY 1969'////
     1)
  102 FORMAT(T20,'***      INPUT DATA     ***',//T10,'NUMBER OF STATES = '
     C,I3,/T10,'NUMBER OF CONTROLS = ',I3,/T10,'THE FIRST ',I3,' STATES
     1WILL BE FED BACK'///)
  103 FORMAT(//T40,'*      A      *'//)
  104 FORMAT(//T40,'*      B',1H',5X,'*'//)
  105 FORMAT(//T40,'*      K',1H',5X,'*'//)
  106 FORMAT(//T40,'*      DIAGONAL ELEMENTS OF S      *'//)
  107 FORMAT(//T40,'*      DIAGONAL ELEMENTS OF Q      *'//)
  108 FORMAT(//T40,'*      W',1H',5X,'*'//)
  109 FORMAT(/////,T40,'****      REVERSE PROBLEM SOLUTION      ****'//)
  110 FORMAT(///T20,'**      NEW WEIGHTINGS      **'//)
 112  FORMAT(//T50,'PROBLEM SOLUTION IS COMPLETE.'///T30,'GAIN TOLERANCE
     1 WAS SATISFIED AFTER =',I3,' ITERATIONS.'///)
 114  FORMAT(//T40,'GAIN TOLERANCE/STOPPING TOLERANCE = ',E13.4,'/',F13.
     14)
 115  FORMAT(T10,'MAXIMUM NUMBER OF ITERATIONS = ',I3,/T10,'PRINT CONTRO
     1L = ',I3,/T10,'FEEDBACK GAIN STOPPING TOLERANCE = ',E13.4)
 116  FORMAT(///T40,'*      TRANSPOSED GAIN FUNCTION MATRIX      *'//)
 117  FORMAT(//T40,'**      LOWER TRIANGULAR PORTION OF RICATTI MATRIX.
     1   **'///)
 119  FORMAT(T20,'ITERATION NUMBER ',I3//)
 120  FORMAT(//2X,'*****************')
 121  FORMAT(//T40,'*   TRANSPOSED GAIN INCREMENT MATRIX   *'//)
```

```
 122  FORMAT(//T40,'*   JACOBIAN MATRIX   *'//)
  124 FORMAT(//T40,'*      GAIN INDICATOR MATRIX      *',//)
 126  FORMAT(//T10,'THERE ARE ',I3,' NON-ZERO FEEDBACK GAINS.')
  130 FORMAT(////T30,'***     CHARACTERISTIC ROOTS OF CLOSED LOOP SYSTEM
     1.     ***'//T15,'REAL',T27,'IMAGINARY',T55,'REAL',T67,'IMAGINARY'/
     2/)
 132  FORMAT(T10,G13.5,T25,G13.5,T50,G13.5,T65,G13.5)
  135 FORMAT(//,T40,'*     LOWER TRIANGULAR PORTION OF RICATTI MATRIX.
     F        BLOCK ',I2,'     *',///)
 138  FORMAT(4I2)
 140  FORMAT(///T3,'*****     END OF RUN NUMBER ',I2,T50,I2,'/',I2,'/',
     1I2,5X,'*****')
 142  FORMAT(///T40,'*     AQ - PARAMETER NUMBER ',I3,'     *'//)
 145  FORMAT(////T30,'**    COST INDEX EVALUATION    **'///T20,'THE COST
     1 INDEX IS EVALUATED FOR UNIT LENGTH INITIAL CONDITION VECTORS.'///
     2T10,'AVERAGE COST = ',G12.4/T10,'MAXIMUM COST = ',G12.4/T10,'MINIM
     3UM COST = ',G12.4)
 147  FORMAT(////T3,'****     RUN NUMBER ',I2,T50,I2,'/',I2,'/',I2,5X,
     1'****'//T20,20A4,/T20,20A4///)
 148  FORMAT(///T22,'WITH RESPECT TO ALL INITIAL CONDITION VECTORS SUCH
     1THAT THE ALLSTATE COST IS UNITY,'/T20,'A PERFORMANCE INDICATOR IS FORMED
     1FORMED BY CALCULATING THE RATIO OF THE SOC COST TO THE ALLSTATE CO
     1ST.'///T10,'AVERAGE INDICATOR = ',G12.4/T10,'MAXIMUM INDICATOR = '
     1,G12.4/T10,'MINIMUM INDICATOR = ',G12.4//)
 152  FORMAT(//T10,'THE RAYLEIGH COEFFICIENTS WILL BE USED ',I3,' TIMES.
     1'///)
C     DATA INPUT AND INITIALIZATION
 7000 CONTINUE
      IO1=1
      IO2=2
      IO3=3
      ICK=1
      ICK=0
      ISS=0
      EPS=1.0E-15
C     READ AND WRITE DATA.
      READ(IO1,138) IC1,IC2,IC3,IC4
      READ(IO1,18) ILAB
      READ (IO1,25) NS,NC,NU,NJ,NPA
      READ(IO1,25) REVPRB,ICASE,ICR,IRIC,IIRAY
C     SKIP TO NEXT PAGE AND WRITE HEADING.
      WRITE(IO3,1)
      WRITE(IO3,100)
      WRITE(IO3,101)
      WRITE(IO3,147) IC1,IC2,IC3,IC4,ILAB
      NOP=NPA+1
      IAQ=1
      INT=-1
      NS2=NS*NS
      TTT=0.0
      NJ2=NJ*NJ
      NP=(NS*NS+NS)/2
      NP2=NP*NP
      WRITE(IO3,102) NS,NC,NU
```

```
      IRAY=IIRAY+1
      WRITE(IO3,152) IIRAY
      WRITE(IO3,126) NJ
      READ(IO1,25) ((IK(I,J),I=1,NU),J=1,NC)
      READ(IO1,30) MAXIT,IPRT,KSTOL
      WRITE(IO3,115) MAXIT,IPRT,KSTOL
      WRITE(IO3,124)
      DO 542 I=1,NC
  542 WRITE(IO3,40) (IK(J,I),J=1,NU)
      MAXIT=MAXIT+1
      DO 480 I=1,NS
      DO 480 L=1,NC
      GFM(I,L)=0.0E0
      SAVK(I,L)=0.0E0
      K(I,L)=0.0E0
  480 W(I,L)=0.0E0
      DO 500 I=1,NS
      DO 500 L=1,NOP
 500  S(I,J)=0.0E0
      DO 510 I=1,NC
      DO 510 L=1,NOP
      Q(I,L)=0.0E0
  510 QI(I,L)=0.0E0
C     MATRICES BY COLUMNS.
      CALL DINP(A,B,K,S,Q,NS,NC,NU,NOP)
C     TO REDUCE INPUT, S AND Q ARE ASSUMED TO BE DIAGONAL.
      WRITE (IO3,103)
      DO 520 I=1,NS
  520 WRITE (IO3,50) (A(I,J),J=1,NS)
      WRITE (IO3,104)
      DO 530 I=1,NC
  530 WRITE (IO3,50) (B(J,I),J=1,NS)
      WRITE(IO3,105)
      DO 540 I=1,NC
  540 WRITE(IO3,50) (K(J,I),J=1,NU)
      WRITE(IO3,106)
      WRITE (IO3,50) ((S(I,J),I=1,NS),J=1,NOP)
      WRITE(IO3,107)
      WRITE (IO3,50) ((Q(I,J),I=1,NC),J=1,NOP)
C     INVERT Q ** QI
      DO 550 J=1,NOP
      DO 550 I=1,NC
  550 QI(I,J)=1.0/Q(I,J)
      IF(NPA) 420,425,420
 420  CALL AQCAL(AQ,NS,NPA,IC1)
      IAQ=IAQ+1
      DO 422 II=1,NPA
      WRITE(IO3,142) II
      DO 422 I=1,NS
 422  WRITE(IO3,50) (AQ(I,J,II),J=1,NS)
  425 CONTINUE
C     IF REVPRB=0, READ AND WRITE W.
      IF(REVPRB) 579,575,579
  575 READ(IO1,15)   ((W(I,J),I=1,NU),J=1,NC)
```

```
      WRITE(IO3,108)
      DO 578 I=1,NC
      WRITE(IO3,50) (W(J,I),J=1,NU)
      DO 578 L=1,NU
  578 W2(L,I)=0.5*W(L,I)
      GO TO 6C0
  579 CONTINUE
C     SOLVE LYAPUNOV EQ. FOR P.
C     AK'*P+P*AK=-S-KQK'
      DO 585 I=1,NS
      DO 585 J=1,NS
      AK(I,J)=A(I,J)
      DO 585 L=1,NC
      IF(NU-J) 585,581,581
 581  KZ=IK(J,L)
      IF(KZ) 582,585,582
  582 AK(I,J)=AK(I,J)-B(I,L)*K(J,L)
  585 CONTINUE
      CALL PSOLV(AK,P,S,IRAY)
C     W2=0.5*W
C     W2=(KQ-PB)*I1
      DO 590 I=1,NU
      DO 590 J=1,NC
      W2(I,J)=0.0
      KZ=IK(I,J)
      IF(KZ) 587,590,587
  587 DO 588 L=1,NS
 588  W2(I,J)=W2(I,J)-P(I,L,1)*B(L,J)
      W2(I,J)=W2(I,J)+K(I,J)*Q(J,1)
  589 CONTINUE
      W(I,J)=2.0*W2(I,J)
  590 CONTINUE
      WRITE(IO3,109)
      WRITE(IO3,108)
      DO 595 I=1,NC
  595 WRITE(IO3,50) (W(J,I),J=1,NU)
      WRITE(IO3,130)
      WRITE(IO3,132) (RR(I),RI(I),I=1,NS)
  600 CONTINUE
C     CHECK MULTIPLE CASE DATA.
C     IF ICASE=0, STOP.
      IF(ICASE) 620,610,620
  610 CONTINUE
      WRITE(IO3,120)
      WRITE(IO3,140) IC1,IC2,IC3,IC4
      WRITE(IO3,120)
      GO TO 7C00
C     IF ICR=0, READ WEIGHTING INCREMENTS AND UPDATE.
C
C     IF ICR=1, READ NEW WEIGHTINGS.
  620 WRITE(IO3,1)
  635 DO 6000 IJA=1,ICASE
      IAE=REVPRB+ICASE-1
      IF(IAE) 5890,690,5890
```

```
 5890 NT=IJA+ICR
      IF(NT-1) 680,680,675
 680  CONTINUE
      READ(IO1,15) ((DS(I,L),I=1,NS),L=1,NOP)
      READ(IO1,15) ((DQ(I,L),I=1,NC),L=1,NOP)
      READ(IO1,15) ((DW(I,J),I=1,NU),J=1,NC)
 675  NT=IJA+REVPRB
      IF(NT-1) 690,690,638
  638 IF(ICR-1) 645,640,645
  640 READ(IO1,15) (( S(I,L),I=1,NS),L=1,NOP)
      READ(IO1,15) (( Q(I,L),I=1,NC),L=1,NOP)
      READ(IO1,15) ((W(I,J),I=1,NU),J=1,NC)
      GO TO 650
  645 DO 648 I=1,NS
      DO 648 J=1,NOP
  648 S(I,J)=S(I,J)+DS(I,J)
      DO 649 J=1,NC
      DO 651 L=1,NOP
  651 Q(J,L)=Q(J,L)+DQ(J,L)
      DO 649 L=1,NU
      W(L,J)=W(L,J)+DW(L,J)
  649 W2(L,J)=0.5*W(L,J)
  650 WRITE(IO3,110)
      WRITE(IO3,106)
      WRITE (IO3,50) ((S(I,J),I=1,NS),J=1,NOP)
      WRITE(IO3,107)
      WRITE (IO3,50) ((Q(I,J),I=1,NC),J=1,NOP)
      WRITE(IO3,108)
      DO 652 I=1,NC
  652 WRITE(IO3,50) (W(J,I),J=1,NU)
  660 DO 670 L=1,NOP
      DO 670 I=1,NC
  670 QI(I,L)=1.0/Q(I,L)
 690  CONTINUE
      IF(IRIC) 692,697,692
C     IF IRIC=0   NOTHING
C     IRIC=1  AND NPA=0    SOLVE ALL STATE FOR COMPARISON
C     IRIC=2  SOLVE ALLSTATE INSTEAD OF SOC.
 692  IF(NPA) 697,693,697
 693  DO 1694 I=1,NS
      DO 1694 J=1,NC
 1694 DUM(I,J)=K(I,J)
      CALL RICKA(A,B,S,P,GFM,Q,QI,W,DUM,NS,NC,KSTOL,FNORM,MAXIT,NUMIT,
     1IPRT,IRAY)
      INT=-1
      IF(IRIC-1) 697,1693,697
 1693 DO 1696 I=1,NS
      DO 1696 J=1,NS
 1696 RIC(I,J)=P(I,J,1)
C
C     ITERATION LOOP
C     SOLVE SOC NECESSARY CONDITION EQUATIONS BY
C     NEWTON-RAPHSEN ITERATION.
C
```

```
C     MAXIT = MAXIMUM NUMBER OF ITERATIONS.
  697 CONTINUE
      DO 1000 IBA=1,MAXIT
      IF(IRIC-2) 699,5000,699
 699  ITA=IBA-1
C     ITERATIONS WILL STOP WHEN PERCENTAGE GAIN CHANGE IS BELOW
C     SPECIFIED VALUE, KSTOL.
C     IPRT - CONTROLS PRINT OPTION.
C
C     SOLVE FOR P.
C     AK'P+PAK=-S-KQK'
      DO 700 I=1,NS
      DO 700 J=1,NS
      AK(I,J)=A(I,J)
      DO 700 L=1,NC
      IF(NU-J) 700,694,694
 694  KZ=IK(J,L)
      IF(KZ) 696,700,696
  696 AK(I,J)=AK(I,J)-B(I,L)*K(J,L)
  700 CONTINUE
      CALL PSOLV(AK,P,S,IRAY)
C     CALCULATE GAIN FUNTION MATRIX GFM.
C     GFM=(W2+PB)* Q(INVERS) - K
      DO 710 I=1,NU
      DO 710 J=1,NC
      KZ=IK(I,J)
      IF(KZ) 709,710,709
  709 DUM(I,J)=W2(I,J)
      DO 711 L=1,NS
  711 DUM(I,J)=DUM(I,J)+P(I,L,1)*B(L,J)
  710 CONTINUE
      DO 721 I=1,NU
      DO 721 J=1,NC
      KZ=IK(I,J)
      IF(KZ) 719,721,719
  719 GFM(I,J)=-K(I,J)+DUM(I,J)*QI(J,1)
  721 CONTINUE
C     PRINT OPTION.
      KTOL=0.0E0
      IF(ITA) 738,850,738
C     CHECK GAIN TOLERANCE.
C     KTOL = MAX(/K(L,J)SUBI+1 - K(L,J)SUBI/ / /K(L,J)SUBI)
C
  738 DO 740 I=1,NU
      DO 740 J=1,NC
      KZ=IK(I,J)
      IF(KZ) 741,740,741
  741 CC=SAVK(I,J)/K(I,J)
      DC=ABS(CC)
      IF(DC-KTOL) 740,740,739
  739 KTOL=DC
  740 CONTINUE
C
C     PRINT AREA.
```

```
      IF(KTOL-KSTOL) 800,840,840
  800 CONTINUE
C     WRITE GFM,P,KTOL,KSTOL,K
      WRITE(IO3,1)
      WRITE(IO3,120)
      WRITE(IO3,120)
      WRITE(IO3,112) ITA
      WRITE(IO3,114) KTOL,KSTOL
      WRITE(IO3,105)
      DO 810 I=1,NC
 810  WRITE(IO3,50) (K(J,I),J=1,NU)
      WRITE(IO3,130)
      WRITE(IO3,132) (RR(I),RI(I),I=1,NS)
      IF(IPRT-2) 812,6000,812
 812  WRITE(IO3,116)
      DO 815 I=1,NC
 815  WRITE(IO3,50) (GFM(J,I),J=1,NU)
      IF(IPRT-1) 816,6000,816
 816  WRITE(IO3,117)
      DO 818 I=1,NS
  818 WRITE(IO3,50) (P(I,J,1),J=1,I)
      IF(NPA) 822,830,822
  822 NN=NOP+NPA
      DO 820 II=2,NN
      WRITE (IO3,135) II
      DO 820 I=1,NS
  820 WRITE(IO3,50) (P(I,J,II),J=1,I)
 5000 CONTINUE
  830 WRITE (IO3,120)
      COST1=0.0
      DO 1850 I=1,NS
      COST1=COST1+P(I,I,1)
      DO 1850 J=1,NS
 1850 DUM(I,J)=P(I,J,1)
      COST1=COST1/NS
      L3=10
      CALL HESSEN(DUM,CT,NS,L3)
      IPRNT=0
      CALL QREIG(CT,NS,RR,RI,IPRNT,L3)
      COST2=0.0
      COST3=1.0E20
      DO 1860 I=1,NS
      COST=RR(I)
      IF(COST-COST2) 1864,1862,1862
 1862 COST2=COST
 1864 IF(COST-COST3) 1866,1866,1860
 1866 COST3=COST
 1860 CONTINUE
      COST1=0.5*COST1
      COST2=0.5*COST2
      COST3=0.5*COST3
      WRITE(IO3,145) COST1,COST2,COST3
      IF(IRIC-1) 6000,1863,6000
 1863 IF(NPA) 6000,1867,6000
```

```
 1867 DO 1870 I=1,NS
      DO 1870 J=1,NS
 1870 RICI(I,J)=0.0
      DO 1872 I=1,NS
 1872 RICI(I,I)=1.0
      L=0
      DO 1875 I=1,NS
      DO 1875 J=1,NS
      L=L+1
      EF(L)=RICI(J,I)
 1875 EE(L)=RIC(J,I)
      EPS=1.0E-15
      IER=0
      ICODE=0
      CALL DLIN(EF,EE,NS,NS,NS2,NS2,EPS,IER,ICODE)
      L=0
      DO 1880 I=1,NS
      DO 1880 J=1,NS
      L=L+1
 1880 RICI(J,I)=EF(L)
C     DUM=RIC*P
      DO 1882 I=1,NS
      DO 1882 J=1,NS
      DUM(I,J)=0.0
      DO 1882 L=1,NS
 1882 DUM(I,J)=DUM(I,J)+RICI(I,L)*P(L,J,1)
      COST1=0.0
      DO 1885 I=1,NS
 1885 COST1=COST1+DUM(I,I)
      COST1=COST1/NS
      CALL HESSEN(DUM,CT,NS,L3)
      IPRNT=0
      CALL QREIG(CT,NS,RR,RI,IPRNT,L3)
      COST2=0.0
      COST3=1.0E20
      DO 1890 I=1,NS
      COST=RR(I)
      IF(COST-COST2) 1894,1892,1892
 1892 COST2=COST
 1894 IF(COST-COST3) 1896,1896,1890
 1896 COST3=COST
 1890 CONTINUE
      WRITE(IO3,148) COST1,COST2,COST3
      WRITE(IO3,120)
      GO TO 6000
 840  CONTINUE
      WRITE(IO3,120)
      WRITE(IO3,119) ITA
      WRITE(IO3,114) KTOL,KSTOL
      WRITE(IO3,105)
      DO 842 I=1,NC
 842  WRITE(IO3,50) (K(J,I),J=1,NU)
      WRITE(IO3,121)
      DO 844 I=1,NC
```

```
  844 WRITE(IO3,50) (SAVK(J,I),J=1,NU)
      WRITE(IO3,130)
      WRITE(IO3,132) (RR(I),RI(I),I=L,NS)
      IF(IPRT-2) 845,849,845
 845  WRITE(IO3,116)
      DO 846 I=1,NC
 846  WRITE(IO3,50) (GFM(J,I),J=1,NU)
      WRITE(IO3,122)
      DO 848 I=1,NJ
 848  WRITE(IO3,50) (JAC(I,J),J=1,NJ)
      IF(IPRT-1) 852,849,852
 852  WRITE(IO3,117)
      DO 854 I=1,NS
  854 WRITE(IO3,50) (P(I,J,1),J=1,I)
      IF(NPA) 860,849,860
  860 NN=NOP+NPA
      DO 870 II=2,NN
      WRITE(IO3,135) II
      DO 870 I=1,NS
  870 WRITE(IO3,50) (P(I,J,II),J=1,I)
 849  WRITE(IO3,120)
C     CALCULATE JACOBIAN
C
C
  850 CONTINUE
C      CALCULATE BQI
      DO 900 I=1,NS
      DO 900 J=1,NC
 900  BQI(I,J)=B(I,J)*QI(J,1)
      ICT=NOP+NPA
      DO 909 L=1,ICT
      DO 909 JJ=1,NS
      DO 909 II=1,NC
      DUMC(II,JJ,L)=0.0E0
      DO 902 L1=1,NS
 902  DUMC(II,JJ,L)=DUMC(II,JJ,L)+B(L1,II)*P(L1,JJ,L)
      IF(L-NOP) 907,907,909
 907  IF(NU-JJ) 909,903,903
 903  KZ=IK(JJ,II)
      IF(KZ) 904,909,904
  904 DUMC(II,JJ,L)=DUMC(II,JJ,L)-Q(II,L)*K(JJ,II)
 909  CONTINUE
      IF(NPA) 922,923,922
 922  DO 921 L=1,NPA
      L1=NOP+L
      DO 921 II=1,NS
      DO 921 JJ=1,NC
      DUMY(II,JJ,L)=0.0
      DO 921 LL=1,NS
 921  DUMY(II,JJ,L)=DUMY(II,JJ,L)+P(II,LL,L1)*B(LL,JJ)
  923 L=0
      ICODE=1
      DO 916 J=1,NC
      DO 916 I=1,NU
```

```
C     CALCULATE PKIJ.
      KZ=IK(I,J)
      IF(KZ) 906,916,906
  906 L=L+1
      IF(NPA) 892,890,892
  890 I2=1
      I1=1
      GO TO 899
  892 I2=NOP
      I1=2
      DO 898 II=1,NS
      DO 898 JJ=1,NS
  898 DUME(II,JJ)=0.0E0
      DO 897 II=1,NS
  897 DUME(I,II)=DUMC(J,II,1)
  899 DO 911 IC=I1,I2
      DO 901 J1=1,NS
      DO 901 K1=1,NS
  901 DUMA(J1,K1)=0.0E0
      DO 905 L1=1,NS
  905 DUMA(I,L1)=DUMC(J,L1,IC)
      DO 910 L1=1,NS
      DO 910 L2=1,NS
  910 DUM(L1,L2)=DUMA(L1,L2)+DUMA(L2,L1)
      ISYM=0
      INT=INT+1
      CALL LYPEIG(AK,DUM,RR,RI,V,DUMA,INT,ISS,TTT,NS,NS2,CKTOL,ICK,ISYM,IRAY)
     1IRAY)
      IF(NPA) 912,919,912
  912 DO 948 II=1,NS
      DO 948 JJ=1,NS
 948  DUM(II,JJ)=0.0
      ICC=IC+NPA
      ID=IC-1
      DO 949 II=1,NS
      DUM(I,II)=DUMC(J,II,ICC)
 949  DUM(II,I)=DUM(II,I)+DUMY(II,J,ID)
      DO 951 II=1,NS
      DO 951 JJ=1,NS
      DO 951 LL=1,NS
 951  DUM(II,JJ)=DUM(II,JJ)-DUMA(II,LL)*AQ(LL,JJ,ID)
      ISYM=1
      INT=INT+1
      CALL LYPEIG(AK,DUM,RR,RI,V,DUMA,INT,ISS,TTT,NS,NS2,CKTOL,ICK,ISYM,IRAY
     1IRAY)
      DO 952 II=1,NS
      DO 952 JJ=1,NS
      DO 952 LL=1,NS
 952  DUME(II,JJ)=DUME(II,JJ)-DUMA(LL,II)*AQ(LL,JJ,ID)
  911 CONTINUE
      IF(NPA) 955,919,955
  955 DO 960 II=1,NS
      DO 960 JJ=1,NS
  960 DUM(II,JJ)=DUME(II,JJ)+DUME(JJ,II)
```

```
      ISYM=0
      INT=INT+1
      CALL LYPEIG(AK,DUM,RR,RI,V,DUMA,INT,ISS,TTT,NS,NS2,CKTOL,ICK,ISYM,IRAY)
     1IRAY)
C     CALCULATE L-TH COLUMN OF JAC.
 919  M=0
      DO 915 N1=1,NC
      DO 915 N2=1,NU
      KZ=IK(N2,N1)
      IF(KZ) 914,915,914
  914 M=M+1
      JAC(M,L)=0.0
      DO 917 L1=1,NS
  917 JAC(M,L)=JAC(M,L)+DUMA(N2,L1)*BQI(L1,N1)
  915 CONTINUE
  916 CONTINUE
      DO 920 I=1,NJ
  920 JAC(I,I)=JAC(I,I)-1.0
C     CALCULATE NEW GAINS.
C     JAC*DK=-F(K)
      K1=0
      K2=0
      DO 932 I=1,NC
      DO 932 J=1,NU
      KZ=IK(J,I)
      IF(KZ) 929,932,929
  929 K1=K1+1
      EF(K1)=-GFM(J,I)
      DO 930 L=1,NJ
      K2=K2+1
  930 EE(K2)=JAC(L,K1)
  932 CONTINUE
      NEQ=1
      ICODE=0
      CALL DLIN(EF,EE,NJ,NEQ,NJ2,NJ,EPS,IER,ICODE)
C     GAIN CHANGE IS SAVK.
C     CALCULATE NEW GAINS. K=K+SAVK
      K1=0
      DO 940 I=1,NC
      DO 940 J=1,NU
      KZ=IK(J,I)
      IF(KZ) 938,940,938
  938 K1=K1+1
      SAVK(J,I)=EF(K1)
      K(J,I)=K(J,I)+SAVK(J,I)
  940 CONTINUE
      INT=-1
 1000 CONTINUE
 6000 CONTINUE
      GO TO 610
      END
```

```
      SUBROUTINE DLIN(R,A,M,N,MM,NM,EPS,IER,ICODE)
      DIMENSION A(100),R(100)
      DOUBLE PRECISION R,A,PIV,TB,TOL,PIVI
      DOUBLE PRECISION DABS,DSIGN,DSQRT,DMAX1
      DOUBLE PRECISION PSAV(10,1),PSS(60)
      DIMENSION IPSA(10,1)
  100 FORMAT(9D14.5)
      IF(M) 23,23,1
C      SEARCH FOR LARGEST ELEMENT IN A.
    1  IER=0
      PIV=0.D0
      KK1=0
      IF(ICODE-1) 850,840,850
  850 CONTINUE
      DO 3 L=1,MM
      TB=DABS(A(L))
      IF(TB-PIV) 3,3,2
    2 PIV=TB
      I=L
    3 CONTINUE
      TOL=EPS*PIV
C       A(I) IS THE PIVOT ELEMENT. PIV CONTAINS THE ABSOLUTE VALUE OF
C        A(I).
C
C       START ELIMINATION LOOP.
      LST=1
      DO 17 K=1,M
C
C       TEST ON SINGULARITY.
      IF(PIV) 23,23,4
    4  IF(IER) 7,5,7
    5 IF(PIV-TOL) 6,6,7
    6 IER=K-1
    7 PIVI=1.D0/A(I)
      J=(I-1)/M
      I=I-J*M-K
      J=J+1-K
C      I+K   ROW INDEX,    J+K COLUMN INDEX OF PIVOT ELEMENT.
C
C
C   & IVOT ROW REDUCTION AND ROW INTERCHANGE IN
C      RIGHT HAND SIDE R.
      DO 8 L=K,NM,M
      LL=L+I
      TB=PIVI*R(LL)
      R(LL)=R(L)
    8 R(L)=TB
      PSAV(K,1)=PIVI
      IPSA(K,1)=I
C
C      IS ELIMINATICN TERMINATED...
      IF(K-M) 9,18,18
C
C      COLUMN INTERCHANGE IN A...
```

```
    9 LEND=LST+M-K
      IF(J) 12,12,10
   10 II=J*M
      DO 11 L=LST,LEND
      TB=A(L)
      LL=L+II
      A(L)=A(LL)
   11 A(LL)=TB
C
C
C       ROW INTERCHANGE AND PIVOT ROW REDUCTION IN A.
   12 DO 13 L=LST,MM,M
      LL=L+I
      TB=PIVI*A(LL)
      A(LL)=A(L)
   13 A(L)=TB
C
C       SAVE COLUMN INTERCHANGE INFORMATION...
      A(LST)=J
C
C     ELEMENT REDUCTION AND NEXT PIVOT SEARCH....
      PIV=0.D0
      LST=LST+1
      J=0
      DO 16 II=LST,LEND
      PIVI=-A(II)
      KK1=KK1+1
      PSS(KK1)=PIVI
      IST=II+M
      J=J+1
      DO 15 L=IST,MM,M
      LL=L-J
      A(L)=A(L)+PIVI*A(LL)
      TB=DABS(A(L))
      IF(TB-PIV) 15,15,14
   14 PIV=TB
      I=L
   15  CONTINUE
      DO 16 L=K,NM,M
      LL=L+J
   16 R(LL)=R(LL)+PIVI*R(L)
   17 LST=LST+M
C     END OF ELIMINATION LOOP.****
C
C     BACK SUBSTITUTION AND BACK INTERCHANGE.
   18 IF(M-1) 23,22,19
   19 IST=MM+M
      LST=M+1
      DO 21 I=2,M
      II=LST-I
      IST=IST-LST
      L=IST-M
      L=A(L)+.5D0
      DO 21 J=II,NM,M
```

```
      TB=R(J)
      LL=J
      DO 20 K=IST,MM,M
      LL=LL+1
   20 TB=TB-A(K)*R(LL)
      K=J+L
      R(J)=R(K)
   21 R(K)=TB
   22 RETURN
C
C     ERROR RETURN $$$$$******
C
C
C     LET EPS=1.0E-15
   23 IER=-1
      RETURN
  840 CONTINUE
      MY=M-1
      LA=0
      DO 800 K=1,MY
      LL=K+IPSA(K,1)
      TB=PSAV(K,1)*R(LL)
      R(LL)=R(K)
      R(K)=TB
      K1=K+1
      DO 800 KK=K1,M
      LA=LA+1
  800 R(KK)=R(KK)+PSS(LA)*R(K)
      LL=M+IPSA(M,1)
      TB=PSAV(M,1)*R(LL)
      R(LL)=R(M)
      R(M)=TB
      GO TO 18
      END
```

```
      SUBROUTINE LYPEIG(F,S,RR,RI,V,P,INT,ISS,T,N,N2,CKTOL,ICK,ISYM,IRAY)
     1)
C     IF INT=0  INITIALIZE
C     IF  ISS=0  THEN STEADY STATE CALCULATIONS
C     ISS=1   NEW  S
C     ISS=2    NEW   T
C     ISS=3    BOTH
      DOUBLE PRECISION KB(10,10),DDUM(10,10),ETA(10,10,2),DDUMA(10,10),V
     1IN(10,10),V(10,10),SB(10,10)
      DOUBLE PRECISION EE(100),EF(100)
      DIMENSION DUM(10,10),DUMA(10,10),RR(10),RI(10),WW(10,4),XR(10),XI(
     110),VR(10),VI(10)
      DOUBLE PRECISION RR,RI,XR,XI,WW,VR,VI,TT
      DOUBLE PRECISION DUM,DUMA
      DOUBLE PRECISION ROOTR,ROOTI
      DOUBLE PRECISION SW1
      DOUBLE PRECISION A1,A2,A3,A4,A5,A6
      DOUBLE PRECISION SJJR,SJJI,A,B,C,D
      DOUBLE PRECISION S1,S1I,S2,S2I,T1,T2
      DIMENSION S(10,10),F(10,10),P(10,10),IROW(10,2)
      EQUIVALENCE  (DUM(1,1),DDUM(1,1)),(DUMA(1,1),DDUMA(1,1))
 800  FORMAT(1X,10G12.4)
 850  FORMAT(1X,10I5)
 900  FORMAT(1X,10D13.5)
      IF(INT) 87,5,87
    5 CONTINUE
      IAA=N
C     CALCULATE EIGENVALUES
      DO 7 I=1,N
      DO 7 J=1,N
    7 DUM(I,J)=F(I,J)
      L3=10
      CALL HESSEN(F,DUM,N,L3)
      L1=30
      IPRNT=0
      CALL QREIG(DUM,N,RR,RI,IPRNT,L1)
C     REORDER EIGENVALUES, FIRST NR WILL BE REAL.
      NR=0
      NRR=N+1
      DO 20 I=1,N
      TT=RI(I)
      IF(TT) 15,10,15
   10 NR=NR+1
      DUM(NR,1)=RR(I)
      DUM(NR,2)=RI(I)
      GO TO 20
   15 NRR=NRR-1
      DUM(NRR,1)=RR(I)
      DUM(NRR,2)=RI(I)
   20 CONTINUE
      DO 22 I=1,N
      RR(I)=DUM(I,1)
 22   RI(I)=DUM(I,2)
C     CALCULATE EIGENVALUES AND FORMULATE V.
```

```
C       SQUARE F.
        NTAPE=0
        NMAX=30
        MMAX=NMAX
        CALL ASQR(NTAPE,F,DUMA,N,NMAX,MMAX)
        IF(IRAY-1) 270,280,270
  270   IVC=3
        GO TO 290
  280   IVC=1
 290    MM=2
        IF(NR) 25,30,25
 25     ROOTI=0.0D0
        DO 50 I=1,NR
        DO 50 LL=1,IRAY
        ROOTR=RR(I)
        DO 37 II=1,N
        DO 37 JJ=1,N
 37     DUM(II,JJ)=DUMA(II,JJ)
        CALL EIGVEC(IVC,F,DUM,WW,IROW,XR,XI,VR,VI,ROOTR,ROOTI,N,NMAX,
       1NTAPE,SW1,ITER,DIF,MM)
        IF(IRAY-1) 175,285,175
 175    CALL RAYL(F,ROOTR,ROOTI,XR,XI,VR,VI,N)
        RR(I)=ROOTR
 285    DO 50 J=1,N
   50   V(J,I)=XR(J)
 30     IF(NR-N) 35,40,35
 35     DO 60 I=NRR,N,2
        DO 60 LL=1,IRAY
        ROOTR=RR(I)
        ROOTI=RI(I)
        DO 56 II=1,N
        DO 56 JJ=1,N
 56     DUM(II,JJ)=DUMA(II,JJ)
        CALL EIGVEC(IVC,F,DUM,WW,IROW,XR,XI,VR,VI,ROOTR,ROOTI,N,NMAX,
       1NTAPE,SW1,ITER,DIF,MM)
        IF(IRAY-1) 210,205,210
 210    CALL RAYL(F,ROOTR,ROOTI,XR,XI,VR,VI,N)
        RR(I)=ROOTR
        RI(I)=ROOTI
 205    DO 60 J=1,N
        V(J,I)=XR(J)
   60   V(J,I+1)=XI(J)
   40   CONTINUE
C       CALCULATE VIN.
        DO 70 I=1,N
        DO 70 J=1,N
 70     VIN(I,J)=0.0D0
        DO 72 I=1,N
 72     VIN(I,I)=1.0D0
        L=0
        DO 75 I=1,N
        DO 75 J=1,N
        L=L+1
        EF(L)=VIN(J,I)
```

```
   75 EE(L)=V(J,I)
      EPS=1.0E-15
      IER=0
      ICODE=0
      CALL DLIN(EF,EE,N,N,N2,N2,EPS,IER,ICODE)
      L=0
      DO 80 I=1,N
      DO 80 J=1,N
      L=L+1
   80 VIN(J,I)=EF(L)
   85 IF(INT) 87,100,87
   87 ITY=ISS+1
      GO TO (105,105,100,100) , ITY
 100  IJF=1
      CALL CALETA(RR,RI,ETA,N,NR,NRR,ISS,INT,T,IJF)
      IF(ISS-2) 105,109,105
C     CALCULATE SBAR=SB=-V'SV
 105  DO 90 I=1,N
      DO 90 J=1,N
      DDUM(I,J)=0.0D0
      DO 90 L=1,N
 90   DDUM(I,J)=DDUM(I,J)-S(I,L)*V(L,J)
      DO 95 I=1,N
      IF(ISYM) 92,93,92
 92   ISUB=N
      GO TO 94
 93   ISUB=I
 94   DO 95 J=1,ISUB
      SB(I,J)=0.0D0
      DO 95 L=1,N
 95   SB(I,J)=SB(I,J)+V(L,I)*DDUM(L,J)
C     CALCULATE ETA.
C     CALCULATE KBAR=KB.
 109  IF(NR) 110,148,110
C     KB(I,J) I=1,NR, J=1,I
  110 DO 120 I=1,NR
      IF(ISYM) 122,124,122
 122  ISUB=NR
      GO TO 123
 124  ISUB=I
 123  DO 120 J=1,ISUB
  120 KB(I,J)=SB(I,J)*ETA(I,J,1)
C     CALCULATE KB(I,J) I=NRR,N,2 , J=1,NR
      IF(NR-N) 125,500,125
  125 DO 130 I=NRR,N,2
      DO 130 J=1,NR
      A1=SB(I,J)
      A2=SB(I+1,J)
      A3=ETA(I,J,1)
      A4=ETA(I,J,2)
      CALL RIMULT(A1,A2,A3,A4,A5,A6)
      KB(I,J)=A5
  130 KB(I+1,J)=A6
      IF(ISYM) 134,150,134
```

```
 134  DO 136 I=1,NR
      DO 136 J=NRR,N,2
      A1=SB(I,J)
      A2=SB(I,J+1)
      A3=ETA(I,J,1)
      A4=ETA(I,J,2)
      CALL RIMULT(A1,A2,A3,A4,A5,A6)
      KB(I,J)=A5
 136  KB(I,J+1)=A6
 148  IF(ISYM) 149,150,149
  149 NN=NRR
      GO TO 172
C     CALCULATE KB COMPLEX DIAGONAL TERMS.
  150 DO 160 J=NRR,N,2
      SJJR=SB(J,J)-SB(J+1,J+1)
      SJJI=2.0*SB(J+1,J)
      SJ11=(SB(J,J)+SB(J+1,J+1))*ETA(J+1,J,1)
      A=ETA(J,J,1)
      B=ETA(J,J,2)
      CALL RIMULT(SJJR,SJJI,A,B,C,D)
      KB(J,J)=0.5*(C+SJ11)
      KB(J+1,J)=0.5*D
  160 KB(J+1,J+1)=0.5*(-C+SJ11)
      IF(NR+2-N) 170,500,500
C     KB(I,J) I=NR+3,N-1,2 , J=NR+3,I,2 .
  170 NN=NR+3
 172  DO 180 I=NN,N,2
      IF(ISYM) 176,174,176
 174  ISUB=I-1
      GO TO 178
 176  ISUB=N
 178  DO 180 J=NRR,ISUB,2
      S1=SB(I,J)-SB(I+1,J+1)
      S1I=SB(I+1,J)+SB(I,J+1)
      S2=SB(I,J)+SB(I+1,J+1)
      S2I=-SB(I+1,J)+SB(I,J+1)
      T1=ETA(I,J,1)
      T2=ETA(I,J,2)
      CALL RIMULT(S1,S1I,T1,T2,A1,A2)
      T1=ETA(I+1,J,1)
      T2=ETA(I+1,J,2)
      CALL RIMULT(S2,S2I,T1,T2,A3,A4)
      KB(I,J)=0.5*(A1+A3)
      KB(I,J+1)=0.5*(A2+A4)
      KB(I+1,J)=0.5*(A2-A4)
  180 KB(I+1,J+1)=0.5*(-A1+A3)
  500 CONTINUE
C     KB CALCULATIONS ARE COMPLETE.
C
C     CALCULATE P.
C     P=VIN*KB*
C     P=VIN'*KB*VIN
      DO 190 I=1,N
      DO 190 J=1,N
```

```
      DDUMA(I,J)=0.0D0
      DO 190 L=1,N
      IF(ISYM) 182,184,182
 184  IF(I-L) 185,182,182
 182  DDUMA(I,J)=DDUMA(I,J)+KB(I,L)*VIN(L,J)
      GO TO 190
 185  DDUMA(I,J)=DDUMA(I,J)+KB(L,I)*VIN(L,J)
 190  CONTINUE
      DO 200 I=1,N
      DO 200 J=1,N
      DDUM(I,J)=0.0D0
      DO 195 L=1,N
 195  DDUM(I,J)=DDUM(I,J)+VIN(L,I)*DDUMA(L,J)
 200  P(I,J)=DDUM(I,J)
      IF(ICK) 300,350,300
 300  DO 320 I=1,N
      DO 320 J=1,N
      DDUMA(I,J)=-S(I,J)
      DO 320 L=1,N
 320  DDUMA(I,J)=DDUMA(I,J)+F(L,I)*DDUM(L,J)+DDUM(I,L)*F(L,J)
C     CKTOL = MAX ABS ROW SUM
      CKTOL=0.0
      DO 330 I=1,N
      RSUM=0.0
      DO 335 J=1,N
      CT=DDUMA(I,J)
 335  RSUM=RSUM+ABS(CT)
      IF(RSUM-CKTOL) 330,330,328
 328  CKTOL=RSUM
 330  CONTINUE
 350  RETURN
      END
```

```
      SUBROUTINE RAYL(A,E,EI,X,XI,V,VI,N)
      DOUBLE PRECISION E,EI,X(10),XI(10),V(10),VI(10),DVXR,DVXI,DVAXR,DV
     1AXI,DXDR(10),DXDI(10),A1,A2,A3
      DIMENSION A(10,10),DA(10,10)
800   FORMAT(1X,5G12.4)
      DO 10 I=1,N
      DO 10 J=1,N
10    DA(I,J)=A(I,J)
      DO 20 I=1,N
20    DA(I,I)=DA(I,I)-E
      DVXR=0.0
      DO 30 I=1,N
      DVXR=DVXR+V(I)*X(I)
      DXDR(I)=0.0
      DO 30 L=1,N
30    DXDR(I)=DXDR(I)+DA(I,L)*X(L)
      DVAXR=0.0
      DO 40 I=1,N
 40   DVAXR=DVAXR+V(I)*DXDR(I)
      IF(EI) 60,50,60
50    E=E+DVAXR/DVXR
      RETURN
60    DVXI=0.0
      DO 70 I=1,N
      DVXR=DVXR-VI(I)*XI(I)
70    DVXI=DVXI+VI(I)*X(I)+V(I)*XI(I)
      DO 80 I=1,N
      DXDI(I)=0.0
      DO 80 J=1,N
80    DXDI(I)=DXDI(I)+DA(I,J)*XI(J)
      A1=0.0
      A2=0.0
      A3=0.0
      DO 90 I=1,N
      A1=A1+VI(I)*DXDI(I)
      A2=A2+VI(I)*DXDR(I)
90    A3=A3+V(I)*DXDI(I)
      DVAXR=DVAXR+EI*DVXI-A1
      DVAXI=A2+A3-EI*DVXR
      A1=DVXR*DVXR+DVXI*DVXI
      E=E+(DVXR*DVAXR+DVXI*DVAXI)/A1
      EI=EI+(DVAXI*DVXR-DVAXR*DVXI)/A1
      RETURN
      END
```

```
      SUBROUTINE PSCLV(AK,P,S,IRAY)
      DOUBLE PRECISION V(10,10),RR(10),RI(10)
      REAL K(10,10)
      DIMENSION AK(10,10),S(10,10),P(10,10,7)
      COMMON K
      COMMON DUM(10,10),RR,RI,V,AQ(10,10,3),B(10,10),DUMA(10,10),
     1Q(10,10),DUMC(10,10,7),TTT
      COMMON IK(10,10),NOP,NS,NC,ISYM,INT,NS2,NU,NPA
C
 1000 FORMAT(1X,5G16.6)
C     FORMULATE RHS
      ICK=0
      ISS=0
      ICK=1
      DO 599 LL=1,NOP
      DO 580 I=1,NS
      DO 580 J=1,NC
      DUMA(J,I)=0.0E0
      IF(NU-I) 580,575,575
 575  KZ=IK(I,J)
      IF(KZ) 576,580,576
 576  DUMA(J,I)=Q(J,LL)*K(I,J)
  580 CONTINUE
      DO 599 I=1,NS
      DO 599 J=1,NS
      IF(I-J) 598,597,598
  597 DUMC(I,J,LL)=-S(I,LL)
      GO TO 596
  598 DUMC(I,J,LL)=0.0
  596 DO 599 L=1,NC
      IF(NU-I) 599,650,650
 650  KZ=IK(I,L)
      IF(KZ) 594,599,594
  594 DUMC(I,J,LL)=DUMC(I,J,LL)-K(I,L)*DUMA(L,J)
  599 CONTINUE
C     IF NPA G.T. 0,CALCULATE LOWER DIAGONAL BLOCKS.
      IF(NPA-1) 593,592,592
  592 DO 612 L=1,NPA
      ISYM=0
      LL=L+1
      DO 614 I=1,NS
      DO 614 J=1,NS
  614 DUM(I,J)=DUMC(I,J,LL)
      INT=INT+1
      CALL LYPEIG(AK,DUM,RR,RI,V,DUMA,INT,ISS,TTT,NS,NS2,CKTOL,ICK,ISYM,IRA
     1IRAY)
      DO 616 I=1,NS
      DO 616 J=1,NS
  616 P(I,J,LL)=DUMA(I,J)
C     CALCULATE LOWER 1ST COLUMN BLOCKS.
      DO 602 I=1,NS
      DO 602 J=1,NS
      DUM(I,J)=0.0
      DO 602 LL=1,NS
```

```
  602 DUM(I,J)=DUM(I,J)-DUMA(I,LL)*AQ(LL,J,L)
      ISYM=1
      INT=INT+1
      LK=NPA+1+L
      CALL LYPEIG(AK,DUM,RR,RI,V,DUMA,INT,ISS,TTT,NS,NS2,CKTOL,ICK,ISYM,IRAY)
     1IRAY)
      DO 604 I=1,NS
      DO 604 J=1,NS
  604 P(I,J,LK)=DUMA(I,J)
  612 CONTINUE
C     CALCULATE MAIN P BLOCK.
  593 DO 615 I=1,NS
      DO 615 J=1,NS
  615 DUM(I,J)=DUMC(I,J,1)
      IF(NPA) 618,632,618
  618 DO 621 I=1,NS
      DO 621 J=1,NS
      DUMA(I,J)=0.0
      DO 621 LL=1,NPA
      LC=NPA+1+LL
      DO 621 KL=1,NS
  621 DUMA(I,J)=DUMA(I,J)-P(KL,I,LC)*AQ(KL,J,LL)
      DO 623 I=1,NS
      DO 623 J=1,NS
  623 DUM(I,J)=DUM(I,J)+DUMA(I,J)+DUMA(J,I)
  632 CONTINUE
      INT =INT+1
      ISYM=0
      CALL LYPEIG(AK,DUM,RR,RI,V,DUMA,INT,ISS,TTT,NS,NS2,CKTOL,ICK,ISYM,IRAY)
     1IRAY)
      DO 637 I=1,NS
      DO 637 J=1,NS
  637 P(I,J,1)=DUMA(I,J)
      RETURN
      END
```

```
      SUBROUTINE RICKA(A,B,SS,PS,F,QS,QIS,W,FBG,N,NC,STOL,FNORM,MAXIT,
     1NUMIT,IPRT,IRAY)
C     SUBROUTINE SOLVES STEADY STATE RICATTI EQUATION.
C     SPECIAL VERSICN OF RICATTI EQUATION SOLVER FOR USE IN SOC SYSTEM.
C     PROGRAM RESTRICTED TO DIAGONAL WEIGHTING.
C     LYAPUNOV EQUATION SOLVED VIA EIGENSYSTEM APPROACH.
      DOUBLE PRECISION RR(10),RI(10),V(10,10)
      DIMENSION A(10,10),B(10,10),S(10,10),Q(1C,10),QI(10,10),FBG(10,10)
     1,P(10,10),SFBG(10,10),DUM(10,10),AK(10,10),W2(10,10),F(10,10),
     1SS(10,10),PS(10,10,7),QS(10,10),QIS(10,10),W2K(10,10),W(10,10)
C     FORMAT STATEMENTS
  105 FORMAT(//T40,'*     K',1H',5X,'*'//)
 110  FORMAT(1X,10G13.5)
 117  FORMAT(//T40,'**     LOWER TRIANGULAR PORTION OF RICATTI MATRIX.
     1   **'///)
 121  FORMAT(//T40,'*   TRANSPOSED GAIN INCREMENT MATRIX   *'//)
  130 FORMAT(////T30,'***     CHARACTERISTIC RCOTS OF CLOSED LOOP SYSTEM
     1.     ***'//T15,'REAL',T27,'IMAGINARY',T55,'REAL',T67,'IMAGINARY'/
     2/)
 132  FORMAT(T10,G13.5,T25,G13.5,T50,G13.5,T65,G13.5)
  200 FORMAT (/5(1P10E13.4))
  400 FORMAT (//T10,'NOTE  **  GAIN TOLERANCE WAS NOT MET AFTER ',I3,
     B'ITERATIONS.'///)
  500 FORMAT(///T2,'*******************')
  501 FORMAT(///T2,'*',T10,'ITERATION ',I3)
  502 FORMAT (/T2,'*',T15,'GAIN TOLERANCE * CURRENT/STOPPING = ',
     C1PE12.4,'/',1PE12.4)
 510  FORMAT(///T20,'THE NORM OF THE RICATTI FUNCTION = ',G12.4///)
 520  FORMAT(//T30,'THE ALLSTATE PROBLEM WILL BE SOLVED.')
 535  FORMAT(//T30,'THE ALLSTATE PROBLEM SOLUTION.')
  600 FORMAT (1X,'*',1X,5(1PE12.4,'/',1PE12.4))
       IO3=3
      WRITE(IO3,500)
      WRITE(IO3,520)
      WRITE(IO3,500)
      DO 205 I=1,NC
      DO 205 J=1,NC
      Q(I,J)=0.0
  205 QI(I,J)=0.0
      DO 210 I=1,NC
      Q(I,I)=QS(I,1)
 210  QI(I,I)=QIS(I,1)
      DO 220 I=1,N
      DO 220 J=1,N
 220  S(I,J)=0.0
      DO 230 I=1,N
 230  S(I,I)=SS(I,1)
      ISYM=0
      ICK=0
      N2=N*N
      T=0.0
      ISS=0
      INT=0
      IO1=1
```

```
C     WRITE INITIAL FEEDBACK GAINS (TRANSPOSED)
      DO 12 J=1,NC
      DO 12 I=1,N
   12 W2(I,J)=0.0
C     ** MAIN LOOP **
      DO 99 NUMIT=1,MAXIT
C     FORMULATE RHS OF RECURSIVE EQUATION
      DO 20 I=1,N
      DO 20 J=1,N
      AK(I,J)=A(I,J)
      DUM(I,J)=0.0E0
      W2K(I,J)=0.0
      DO 15 II=1,NC
      W2K(I,J)=W2K(I,J)+W2(I,II)*FBG(J,II)
      AK(I,J)=AK(I,J)-B(I,II)*FBG(J,II)
      DO 15 JJ=1,NC
   15 DUM(I,J)=DUM(I,J)+FBG(I,II)*Q(II,JJ)*FBG(J,JJ)
   20 DUM(I,J)=-DUM(I,J)-S(I,J)
      DO 22 I=1,N
      DO 22 J=1,N
 22   DUM(I,J)=DUM(I,J)+W2K(I,J)+W2K(J,I)
      ICK=0
      ICODE=0
      CALL LYPEIG(AK,DUM,RR,RI,V,P,INT,ISS,T,N,N2,CKTOL,ICK,ISYM,IRAY)
C     CALCULATE NEW FEEDBACK GAINS AND CHECK TOLERANCE.
C     FBGTOL=MAX /K(I+1)-K(I)/  /  /K(I+1)/
C     FBG=QI*(B'P+W2)
      DKTOL=0.0E0
      DO 24 I=1,N
      DO 24 J=1,NC
      DUM(I,J)=W2(I,J)
      DO 24 K=1,N
   24 DUM(I,J)=DUM(I,J)+B(K,J)*P(K,I)
      DO 30 I=1,N
      DO 30 J=1,NC
      SFBG(I,J)=FBG(I,J)
      FBG(I,J)=0.0E0
      DO 25 L=1,NC
   25 FBG(I,J)=FBG(I,J)+QI(J,L)*DUM(I,L)
      SFBG(I,J)=FBG(I,J)-SFBG(I,J)
      CC=SFBG(I,J)/FBG(I,J)
      DC=ABS(CC)
      IF(DC-DKTOL) 30,30,28
   28 DKTOL=DC
   30 CONTINUE
C
      IF(IPRT-2) 32,38,32
   32 WRITE (IO3,500)
      WRITE (IO3,501) NUMIT
      WRITE (IO3,502) DKTOL,STOL
      WRITE(IO3,130)
      WRITE(IO3,132) (RR(I),RI(I),I=1,N)
      IF(IPRT-1) 250,38,250
 250  WRITE(IO3,105)
```

```
      DO 842 I=1,NC
 842  WRITE(IO3,110) (FBG(J,I),J=1,N)
      WRITE(IO3,121)
      DO 844 I=1,NC
 844  WRITE(IO3,110) (SFBG(J,I),J=1,N)
 38   WRITE(IO3,500)
      IF(STOL-DKTOL) 70,40,40
C     CALCULATE FUNCTION TOLERANCE
   40 FNORM=0.0E0
      DO 42 I=1,N
      DO 42 J=1,N
      AK(I,J)=A(I,J)
      DO 42 L=1,NC
   42 AK(I,J)=AK(I,J)-B(I,L)*FBG(J,L)
      DO 60 I=1,N
      RWSUM=0.0E0
      DO 50 J=1,N
      F(I,J)=S(I,J)
      DO 45 K=1,NC
      DO 45 L=1,NC
   45 F(I,J)=F(I,J)+FBG(I,K)*Q(K,L)*FBG(J,L)
      DO 48 II=1,N
   48 F(I,J)=F(I,J)+AK(II,I)*P(II,J)+P(I,II)*AK(II,J)
      CC=F(I,J)
   50 RWSUM=RWSUM+ABS(CC)
      IF(RWSUM-FNORM) 60,60,55
   55 FNORM=RWSUM
   60 CONTINUE
      WRITE(IO3,500)
      WRITE(IO3,535)
      WRITE(IO3,500)
      WRITE(IO3,510) FNORM
      IF(IPRT) 65,150,65
 65   WRITE(IO3,117)
      DO 870 I=1,N
 870  WRITE(IO3,110) (P(I,J),J=1,I)
      GO TO 150
 70   IF(IPRT) 99,80,99
 80   WRITE(IO3,117)
      DO 1870 I=1,N
 1870 WRITE(IO3,110) (P(I,J),J=1,I)
   99 CONTINUE
      WRITE(3,400) MAXIT
  150 DO 175 I=1,N
      DO 175 J=1,N
  175 PS(I,J,1)=P(I,J)
      WRITE(IO3,500)
      WRITE(IO3,500)
      RETURN
      END
```

```
      SUBROUTINE CALETA(RR,RI,ETA,N,NR,NRR,ISS,INT,T,ISYM)
      DIMENSION RR(10),RI(10)
      DOUBLE PRECISION A
      DOUBLE PRECISION RR,RI
      DOUBLE PRECISION ETA(10,10,2),DUM(10,10,2)
      DOUBLE PRECISION A1,A2,A3,A4,A5,A6
      IT=INT*ISS
      IF(IT) 60,20,60
   20 IF(NR) 25,40,25
   25 DO 30 I=1,NR
      DO 30 J=1,I
      DUM(I,J,1) =-1.0/(RR(I)+RR(J))
      ETA(I,J,2)=0.0E0
   30 ETA(I,J,1)=DUM(I,J,1)
      IF(NR-N) 40,50,40
   40 DO 45 I=NRR,N
      DO 45 J=1,I
      A=(RR(I)+RR(J))*(RR(I)+RR(J))+(RI(I)+RI(J))*(RI(I)+RI(J))
      DUM(I,J,1)=-(RR(I)+RR(J))/A
      DUM(I,J,2)=(RI(I)+RI(J))/A
      ETA(I,J,1)=DUM(I,J,1)
   45 ETA(I,J,2)=DUM(I,J,2)
   50 IF(ISS) 60,55,60
55    GO TO 90
   60 IF(NR) 65,70,65
   65 DO 68 I=1,NR
      DO 68 J=1,I
      TAU=(RR(I)+RR(J))*T
   68 ETA(I,J,1)=(1.0-EXP(TAU))*DUM(I,J,1)
   70 IF(NR-N) 75,90,75
   75 DO 80 I=NRR,N
      DO 80 J=NRR,I
      TAU=(RR(I)+RI(J))*T
      BETA=(RI(I)+RI(J))*T
      A1=1.0-EXP(TAU)*COS(BETA)
      A2=SIN(BETA)
      A3=DUM(I,J,1)
      A4=DUM(I,J,2)
      CALL RIMULT(A1,A2,A3,A4,A5,A6)
      ETA(I,J,1)=A5
   80 ETA(I,J,2)=A6
90    IF(ISYM) 95,99,95
95    LL=N-1
      DO 98 I=1,LL
      LA=I+1
      DO 98 J=LA,N
      ETA(I,J,1)=ETA(J,I,1)
98    ETA(I,J,2)=ETA(J,I,2)
99    RETURN
      END
```

```
      SUBROUTINE RIMULT(A1,A2,A3,A4,A5,A6)
      DOUBLE PRECISION A1,A2,A3,A4,A5,A6
C     A5=RE((A1+JA2)*(A3+JA4))
C     A6=IM((A1+JA2)*(A3+JA4))
      A5=A1*A3-A4*A2
      A6=A1*A4+A2*A3
      RETURN
      END
```

```
      SUBROUTINE QRT(A,N,R,SIG,D,M)                                     ESY5   0
      DIMENSION A(10,10),PSI(2),G(3)
      DOUBLE PRECISION DABS,DSIGN,DSQRT,DMAX1
      DOUBLE PRECISION A,PSI,G,R,SIG,D,DEN,XK,AL,C,E
      N1 = N - 1                                                        ESY5  20
      IA = N - 2                                                        ESY5  30
      IP = IA                                                           ESY5  40
      IF(N-3) 101,10,60                                                 ESY5  50
60    DO 12 J = 3,N1                                                    ESY5  60
      J1 = N - J                                                        ESY5  70
      IF(DABS(A(J1+1,J1))-D) 10,10,11
   11 DEN = A(J1+1,J1+1)*(A(J1+1,J1+1)-SIG)+A(J1+1,J1+2)*A(J1+2,J1+1)+R ESY5  90
      IF(DEN) 61,12,61                                                  ESY5 100
61    IF(DABS(A(J1+1,J1)*A(J1+2,J1+1)*(DABS(A(J1+1,J1+1)+A(J1+2,J1+2)-
     1SIG)+DABS(A(J1+3,J1+2)))/DEN)-D) 10,10,12
12    IP=J1                                                             ESY5 130
10    DO  14  J=1,IP                                                    ESY5 140
      J1=IP-J+1                                                         ESY5 150
      IF(DABS(A(J1+1,J1))-D) 13,13,14
14    IQ=J1                                                             ESY5 170
13    DO  100  I=IP,N1                                                  ESY5 180
      IF(I-IP)   16,15,16                                               ESY5 190
15    G(1)=A(IP,IP)*(A(IP,IP)-SIG)+A(IP,IP+1)*A(IP+1,IP)+R              ESY5 200
      G(2)=A(IP+1,IP)*(A(IP,IP)+A(IP+1,IP+1)-SIG)                       ESY5 210
      G(3)=A(IP+1,IP)*A(IP+2,IP+1)                                      ESY5 220
      A(IP+2,IP)=0.0D0
      GO TO 19                                                          ESY5 240
16    G(1)=A(I,I-1)                                                     ESY5 250
      G(2)=A(I+1,I-1)                                                   ESY5 260
      IF(I-IA)   17,17,18                                               ESY5 270
17    G(3)=A(I+2,I-1)                                                   ESY5 280
      GO TO 19                                                          ESY5 290
18    G(3)=0.0D0
19    XK=DSIGN(DSQRT(G(1)**2+G(2)**2+G(3)**2),G(1))
22    IF(XK)   23,24,23                                                 ESY5 320
23    AL=G(1)/XK+1.0                                                    ESY5 330
      PSI(1)=G(2)/(G(1)+XK)                                             ESY5 340
      PSI(2)=G(3)/(G(1)+XK)                                             ESY5 350
      GO TO 25                                                          ESY5 360
24    AL=2.0D0
      PSI(1)=0.0D0
      PSI(2)=0.0D0
25    IF(I-IQ)   26,27,26                                               ESY5 400
26    IF(I-IP)    29,28,29                                              ESY5 410
28    A(I,I-1)=-A(I,I-1)                                                ESY5 420
      GO TO 27                                                          ESY5 430
29    A(I,I-1)=-XK                                                      ESY5 440
27    DO 30  J=I,N                                                      ESY5 450
      IF(I-IA)  31,31,32                                                ESY5 460
31    C=PSI(2)*A(I+2,J)                                                 ESY5 470
      GO TO 33                                                          ESY5 480
32    C=0.0D0
33    E=AL*(A(I,J)+PSI(1)*A(I+1,J)+C)                                   ESY5 500
      A(I,J)=A(I,J)-E                                                   ESY5 510
```

```
      A(I+1,J)=A(I+1,J)-PSI(1)*E                                        ESY5 520
      IF(I-IA)    34,34,30                                              ESY5 530
34    A(I+2,J)=A(I+2,J)-PSI(2)*E                                        ESY5 540
30    CONTINUE                                                          ESY5 550
      IF(I-IA)    35,35,36                                              ESY5 560
35    L=I+2                                                             ESY5 570
      GO TO  37                                                         ESY5 580
36    L=N                                                               ESY5 590
37    DO  40  J=IQ,L                                                    ESY5 600
      IF(I-IA)  38,38,39                                                ESY5 610
38    C=PSI(2)*A(J,I+2)                                                 ESY5 620
      GO TO  41                                                         ESY5 630
39    C=0.0D0
41    E=AL*(A(J,I)+PSI(1)*A(J,I+1)+C)                                   ESY5 650
      A(J,I)=A(J,I)-E                                                   ESY5 660
      A(J,I+1)=A(J,I+1)-PSI(1)*E                                        ESY5 670
      IF(I-IA)  42,42,40                                                ESY5 680
42    A(J,I+2)=A(J,I+2)-PSI(2)*E                                        ESY5 690
40    CONTINUE                                                          ESY5 700
      IF(I-N+3)    43,43,100                                            ESY5 710
43    E=AL*PSI(2)*A(I+3,I+2)                                            ESY5 720
      A(I+3,I)=-E                                                       ESY5 730
      A(I+3,I+1)=-PSI(1)*E                                              ESY5 740
      A(I+3,I+2)=A(I+3,I+2)-PSI(2)*E                                    ESY5 750
100   CONTINUE                                                          ESY5 760
101   RETURN                                                            ESY5 770
      END
```

```
      SUBROUTINE QREIG(A,M,ROOTR,ROOTI,IPRNT,L)                         ESY4   0
C       PROGRAM TO CALL QR TRANSFORMATION, MAXIMUM ITER IS 50.          ESY4  10
      DIMENSION A(10,10),ROOTR(10),ROOTI(10)
      DOUBLE PRECISION DABS,DSIGN,DSQRT,DMAX1
      DOUBLE PRECISION VQ
      DOUBLE PRECISION A,ROOTR,ROOTI,XNN,XN2,AA,B,C,DD,R,SIG,X,S,E,G
      DOUBLE PRECISION F,H,SQ,D,ET,GT,Z1,Z2
      IO1=1
      IO3=3
      N = M                                                             ESY4  30
      IF(IPRNT)  80,81,80                                               ESY4  40
   80 WRITE(IO3,104)
 81   ZERO=0.0D0
      JJ=1                                                              ESY4  70
 177  XNN=0.0D0
      XN2=0.0D0
      AA=0.0D0
      B=0.0D0
      C=0.0D0
      DD=0.0D0
      R=0.0D0
      SIG=0.0D0
      ITER = 0                                                          ESY4 160
 17   IF(N-2)     13,14,12                                              ESY4 170
 13   IF(IPRNT)  82,83,82                                               ESY4 180
   82 WRITE(IO3,105) A(1,1)
 83   ROOTR(1) = A(1,1)                                                 ESY4 200
      ROOTI(1)=0.0D0
 1    RETURN
 14   JJ=-1                                                             ESY4 230
   12 X = (A(N-1,N-1) - A(N,N))**2                                      ESY4 240
      S = 4.0*A(N,N-1)*A(N-1,N)                                         ESY4 250
      ITER = ITER + 1                                                   ESY4 260
      IF(X .EQ. 0.0D0) GO TO 15
      IF(DABS(S/X) .GT. 1.0D-8) GO TO 15
 16   IF(DABS(A(N-1,N-1))-DABS(A(N,N))) 32,32,31
   31 E = A(N-1,N-1)                                                    ESY4 300
      G = A(N,N)                                                        ESY4 310
      GO TO 33                                                          ESY4 320
   32 G = A(N-1,N-1)                                                    ESY4 330
      E = A(N,N)                                                        ESY4 340
  33  F=0.0D0
      H=0.0D0
      GO TO 24                                                          ESY4 370
   15 S = X + S                                                         ESY4 380
      X = A(N-1,N-1) + A(N,N)                                           ESY4 390
      IF(S)     18,19,19                                                ESY4 400
 19   SQ=DSQRT(S)
      F=0.0D0
      H=0.0D0
      IF (X) 21,21,22                                                   ESY4 440
 21   E=(X-SQ)/2.0                                                      ESY4 450
      G=(X+SQ)/2.0                                                      ESY4 460
      GO TO 24                                                          ESY4 470
```

```
22    G=(X-SQ)/2.0                                                    ESY4 480
      E=(X+SQ)/2.0                                                    ESY4 490
      GO TO 24                                                        ESY4 500
18    F=DSQRT(-S)/2.0
      E=X/2.0                                                         ESY4 520
      G=E                                                             ESY4 530
      H=-F                                                            ESY4 540
24    IF(JJ)  28,70,70                                                ESY4 550
70    D=1.0D-10*(DABS(G)+F)
      IF(DABS(A(N-1,N-2)) .GT. D) GO TO 26
28    IF(IPRNT)  84,85,84                                             ESY4 580
 84   WRITE(IO3,105) E,F,ITER
      WRITE(IO3,105) G,H
85    ROOTR(N) = E                                                    ESY4 610
      ROOTI(N) = F                                                    ESY4 620
      ROOTR(N-1) = G                                                  ESY4 630
      ROOTI(N-1) = H                                                  ESY4 640
      N=N-2                                                           ESY4 650
      IF(JJ)  1,177,177                                               ESY4 660
26    IF(DABS(A(N,N-1)) .GT. 1.0D-10*DABS(A(N,N))) GO TO 50
29    IF(IPRNT)  86,87,86                                             ESY4 680
 86   WRITE(IO3,105) A(N,N),ZERO,ITER
87    ROOTR(N) = A(N,N)                                               ESY4 700
      ROOTI(N)=0.0D0
      N=N-1                                                           ESY4 720
      GO TO 177                                                       ESY4 730
50    IF(DABS(DABS(XNN/A(N,N-1))-1.0D0)-1.0D-6) 63,63,62
62    IF(DABS(DABS(XN2/A(N-1,N-2))-1.0D0)-1.0D-6) 63,63,700
63    VQ=DABS(A(N,N-1))-DABS(A(N-1,N-2))
      IF (ITER-15) 53,164,64                                          ESY4 770
 164  IF(VQ) 165,165,166                                              ESY4 780
 165  R = A(N-1,N-2)**2                                               ESY4 790
      SIG = 2.0*A(N-1,N-2)                                            ESY4 800
      GO TO 60                                                        ESY4 810
 166  R = A(N,N-1)**2                                                 ESY4 820
      SIG = 2.0*A(N,N-1)                                              ESY4 830
      GO TO 60                                                        ESY4 840
64    IF(VQ)   67,67,66                                               ESY4 850
66    IF(IPRNT)  88,85,88                                             ESY4 860
 88   WRITE(IO3,107) A(N-1,N-2)
      GO TO 84                                                        ESY4 880
67    IF(IPRNT)  89,87,89                                             ESY4 890
 89   WRITE(IO3,107) A(N,N-1)
      GO TO 86                                                        ESY4 910
 700  IF(ITER .GT. 50) GO TO 63                                       ESY4 920
      IF(ITER .GT. 5 ) GO TO 53                                       ESY4 930
 701  ET = E*E + F*F                                                  ESY4 940
      IF(ET)702,601,702                                               ESY4 950
 702  GT = G*G + H*H                                                  ESY4 960
      IF(GT)703,601,703                                               ESY4 970
 703  Z1 = ((E-AA)**2 + (F-B)**2) / ET                                ESY4 980
      Z2 = ((G-C)**2 + (H-DD)**2) / GT                                ESY4 990
      IF(Z1-0.25)  51,51,52                                           ESY41000
51    IF(Z2-0.25)   53,53,54                                          ESY41010
```

```
53    R=E*G-F*H                                                        ESY41020
      SIG=E+G                                                          ESY41030
      GO TO 60                                                         ESY41040
54    R=E*E                                                            ESY41050
      SIG=E+E                                                          ESY41060
      GO TO 60                                                         ESY41070
52    IF(Z2-0.25) 55,55,601                                            ESY41080
55    R=G*G                                                            ESY41090
      SIG=G+G                                                          ESY41100
      GO TO 60                                                         ESY41110
601   R=0.0D0
      SIG=0.0D0
60    XNN=A(N,N-1)                                                     ESY41140
      XN2=A(N-1,N-2)                                                   ESY41150
      CALL QRT(A,N,R,SIG,D,L)                                          ESY41160
      AA=E                                                             ESY41170
      B=F                                                              ESY41180
      C=G                                                              ESY41190
      DD=H                                                             ESY41200
      GO TO 12                                                         ESY41210
104   FORMAT(////5X, 9HREAL PART  6X  14HIMAGINARY PART, 20X,          ESY41220
     1  13HTAKEN AS ZERO,6X,'ITERATION NO.'//)                          ESY41230
105   FORMAT(1X,D15.8,3X,D15.8,42X,I3)
107   FORMAT(52X,D17.8)
       END                                                             ESY41260
```

```
      SUBROUTINE HESSEN(AA,A,M,L)
C     SUBROUTINE TO PUT MATRIX IN UPPER HESSENBERG FORM.                ESY3  10
      DIMENSION A(10,10),B(200),AA(10,10)
      DOUBLE PRECISION DABS,DSIGN,DSQRT,DMAX1
      DOUBLE PRECISION A,B,DIV,SUM
      DO 50 I=1,M
      DO 50 J=1,M
 50   A(I,J)=AA(I,J)
      IF (M - 2) 30,30,32                                               ESY3  30
   32 DO 40 LC = 3,M                                                    ESY3  40
      N = M - LC + 3                                                    ESY3  50
      N1 = N - 1                                                        ESY3  60
      N2 = N - 2                                                        ESY3  70
      NI = N1                                                           ESY3  80
      DIV=DABS(A(N,N-1))
      DO 2 J = 1,N2                                                     ESY3 100
      IF(DABS(A(N,J))-DIV) 2,2,1
    1 NI = J                                                            ESY3 120
      DIV=DABS(A(N,J))
    2 CONTINUE                                                          ESY3 140
      IF(DIV) 3,40,3                                                    ESY3 150
    3 IF(NI - N1) 4, 7,4                                                ESY3 160
    4 DO 5 J = 1,N                                                      ESY3 170
      DIV = A(J,NI)                                                     ESY3 180
      A(J,NI) = A(J,N1)                                                 ESY3 190
    5 A(J,N1) = DIV                                                     ESY3 200
      DO 6 J = 1,M                                                      ESY3 210
      DIV = A(NI,J)                                                     ESY3 220
      A(NI,J) = A(N1,J)                                                 ESY3 230
    6 A(N1,J) = DIV                                                     ESY3 240
    7 DO 26 K = 1, N1                                                   ESY3 250
   26 B(K) = A(N,K)/A(N,N-1)                                            ESY3 260
      DO 45 J = 1,M                                                     ESY3 270
      SUM = 0.0                                                         ESY3 280
      IF (J - N1) 46,43,43                                              ESY3 290
   46 IF(B(J))  41,43,41                                                ESY3 300
   41 A(N,J) = 0.0                                                      ESY3 310
      DO 42 K = 1,N1                                                    ESY3 320
      A(K,J) = A(K,J) - A(K,N1)*B(J)                                    ESY3 330
   42 SUM = SUM + A(K,J)*B(K)                                           ESY3 340
      GO TO 45                                                          ESY3 350
   43 DO 44 K = 1,N1                                                    ESY3 360
   44 SUM = SUM + A(K,J)*B(K)                                           ESY3 370
   45 A(N1,J) = SUM                                                     ESY3 380
   40 CONTINUE                                                          ESY3 390
   30 RETURN                                                            ESY3 400
      END
```

```
      SUBROUTINE ASQR(T2, A, B, NN, NMAX, MMAX)                         ESY2   0
C                                                                       ESY2  10
C     PROGRAM TO SQUARE THE A MATRIX. THE MATRIX MUST BE ORIGINALLY     ESY2  20
C     IN A. IF T2 IS NON-ZERO, A SQUARED WILL BE STORED ON T2, AND B    ESY2  30
C     WILL BE A BUFFER WHICH MUST HAVE AT LEAST 250 WORDS.  IF T2 IS    ESY2  40
C     ZERO, A SQUARED WILL BE IN B WHICH MUST BE NMAX BY NMAX.          ESY2  50
C                                                                       ESY2  60
      DIMENSION B(10,10),AA(10,10),A(10,10)
      DOUBLE PRECISION B,AA
      INTEGER T2                                                        ESY2  80
      N = NN                                                            ESY2  90
      DO 50 I=1,N
      DO 50 J=1,N
 50   AA(I,J)=A(I,J)
   11 IF(T2) 12, 13, 12                                                 ESY2 100
   12 REWIND T2                                                         ESY2 110
      JJ = 0                                                            ESY2 120
      DO 3 I=1,N                                                        ESY2 130
      DO 3 J=1,N                                                        ESY2 140
      JJ = JJ + 1                                                       ESY2 150
      B(JJ,1)=0.0D0
      DO 1 K=1,N                                                        ESY2 170
 1    B(JJ,1)=B(JJ,1)+AA(I,K)*AA(K,J)
      IF(JJ-250) 3,2,2                                                  ESY2 190
    2 JJ = 0                                                            ESY2 200
      WRITE (T2) (B(L,1), L = 1,250)                                    ESY2 210
    3 CONTINUE                                                          ESY2 220
      IF(JJ .NE. 0) WRITE (T2) (B(L,1), L = 1,250)                      ESY2 230
      REWIND T2                                                         ESY2 240
      RETURN                                                            ESY2 250
   13 DO 14 I = 1, N                                                    ESY2 260
      DO 14 J = 1,N                                                     ESY2 270
      B(I,J)=0.0D0
      DO 14 K = 1,N                                                     ESY2 290
 14   B(I,J)=B(I,J) + AA(I,K)*AA(K,J)
      RETURN                                                            ESY2 310
      END
```

```
      SUBROUTINE EIGVEC(IVC,  A, B, W, IROW, XR, XI, VR, VI, ROOTRE,     ESY1
     1   ROOTIE, NE, NMAX, T2, SW1, COUNTE, ERR,MMM)                     ESY1
C        SUBROUTINE TO FIND THE EIGENVECTORS OF A NON-SYMMETRIC MATRIX  ESY1
C        BY A MODIFIED WILKINSON'S INVERSE ITERATION METHOD.            ESY1
C        CONTROL IVC CODE IS                                            ESY1
C             1  FIND ONLY THE REGULAR EIGENVECTORS   (A X = LAMBDA X)  ESY1
C             2  FIND ONLY THE TRANSPOSED EIGENVECTORS (AT V = LAMBDA V)ESY1
C             3  FIND BOTH TYPES OF EIGENVECTORS.                       ESY1
      DIMENSION B(10,10),W(10,4),XR(10),XI(10),VR(10),VI(10)
      DOUBLE PRECISION DABS,DSIGN,DSQRT,DMAX1
      DOUBLE PRECISION ROOTR,ROOTI,ROOTRE,ROOTIE,TEMP,TEMP2
      DOUBLE PRECISION AMAX,C1,C2,SW1
      DOUBLE PRECISION W,XR,XI,VR,VI,B
      DOUBLE PRECISION ZERO,DCERR
      DIMENSION A(10,10),IROW(10,4)
      INTEGER COUNT, COUNTE, T2                                          ESY1
      IO1=1
      IO3=3
      ROOTR = ROOTRE                                                     ESY1
      ROOTI = ROOTIE                                                     ESY1
      N = NE                                                             ESY1
      MM = MMM - 1                                                       ESY1
      N1 = N - 1                                                         ESY1
      NP1 = N + 1                                                        ESY1
      IVC1 = IVC - 1                                                     ESY1
      IVC2 = IVC1 - 1                                                    ESY1
      COUNT = 1                                                          ESY1
      DO 400 I=1,N
      W(I,1)=0.0D0
      XR(I)=0.0D0
  400 CONTINUE
      CLIM = 1.0E-4                                                      ESY1
      IF(ROOTI) 1, 60, 1                                                 ESY1
C                                                                        ESY1
C        COMPLEX EIGENVALUE.                                             ESY1
C                                                                        ESY1
    1 TEMP = - ROOTR - ROOTR                                             ESY1
      ISW = 2                                                            ESY1
      TEMP2=ROOTR*ROOTR+ROOTI*ROOTI
      JJ = 300                                                           ESY1
      DO 606 I = 1, N                                                    ESY1
      IF(T2) 600, 603, 600                                               ESY1
  600 DO 602 J = 1, N                                                    ESY1
      JJ = JJ + 1                                                        ESY1
      IF(JJ - 251) 602, 601, 601                                         ESY1
  601 JJ = 1                                                             ESY1
      READ (T2) (W(LL,1), LL = 1,250)                                    ESY1
  602 B(I,J) = A(I,J)*TEMP + W(JJ,1)                                     ESY1
      GO TO 605                                                          ESY1
  603 DO 604 J = 1, N                                                    ESY1
  604 B(I,J) = A(I,J)*TEMP + B(I,J)                                      ESY1
  605 B(I,I) = B(I,I) + TEMP2                                            ESY1
  606 A(I,I) = A(I,I) - ROOTR                                            ESY1
      IF(T2 .NE. 0) REWIND T2                                            ESY1
```

```
      GO TO 700                                                         ESY1 430
  607 IF(ICC) 622, 608, 622                                             ESY1 440
C                                                                       ESY1 450
C        MATRIX SINGULAR.                                               ESY1 460
C                                                                       ESY1 470
 622  IF(IVC2) 623, 625, 623
  623 DO 624 LL = 1, N                                                  ESY1 490
      W(LL,2)=0.0D0
 624  XI(LL)=0.0D0
      IF(IVC1) 625, 514, 625                                            ESY1 510
  625 DO 626 LL = 1, N                                                  ESY1 520
      W(LL,4)=0.0D0
 626  VI(LL)=0.0D0
      GO TO 511                                                         ESY1 540
C                                                                       ESY1 550
C        MATRIX NOT SINGULAR.                                           ESY1 560
C                                                                       ESY1 570
  608 DO 609 LL = 1, N                                                  ESY1 580
      W(LL,1)=1.0D0
      W(LL,2)=1.0D0
      W(LL,3)=1.0D0
 609  W(LL,4)=1.0D0
  699 IF(IVC2) 610, 612, 610                                            ESY1 600
  610 DO 611 I = 1, N                                                   ESY1 610
      I2 = IROW(I,2)                                                    ESY1 620
      XI(I2) = W(I,1)*ROOTI                                             ESY1 630
      DO 611 J = 1, N                                                   ESY1 640
  611 XI(I2) = XI(I2) + A(I,J)*W(J,2)                                   ESY1 650
      IF(IVC1) 612, 500, 612                                            ESY1 660
  612 DO 613 I = 1, N                                                   ESY1 670
      VI(I) = W(I,3)*ROOTI                                              ESY1 680
      DO 613 J = 1,N                                                    ESY1 690
  613 VI(I) = VI(I) + A(J,I)*W(J,4)                                     ESY1 700
      GO TO 499                                                         ESY1 710
  615 CERR = 0.0                                                        ESY1 720
      DCERR=0.0D0
      IF(IVC2) 616, 619, 616                                            ESY1 730
  616 DO 618 I = 1, N                                                   ESY1 740
      XR(I) = -W(I,2)                                                   ESY1 750
      DO 617 J = 1, N                                                   ESY1 760
  617 XR(I) = XR(I) + A(I,J)*XI(J)                                      ESY1 770
  618 XR(I) = XR(I)/ROOTI                                               ESY1 780
      IF(IVC1) 619, 633, 619                                            ESY1 790
  619 DO 621 I = 1, N                                                   ESY1 800
      VR(I) = -W(I,4)                                                   ESY1 810
      DO 620 J = 1, N                                                   ESY1 820
  620 VR(I) = VR(I) + A(J,I)*VI(J)                                      ESY1 830
  621 VR(I) = VR(I)/ROOTI                                               ESY1 840
C                                                                       ESY1 850
C        SEARCH VECTORS FOR LARGEST ELEMENT AND NORMALIZE.              ESY1 860
C                                                                       ESY1 870
 627  AMAX=0.0D0
      DO 629 L = 1, N                                                   ESY1 890
      TEMP = VR(L)**2 + VI(L)**2                                        ESY1 900
```

```
      IF(TEMP - AMAX) 629, 629, 628                                     ESY1 910
  628 AMAX = TEMP                                                       ESY1 920
      I2 = L                                                            ESY1 930
  629 CONTINUE                                                          ESY1 940
      C1 = VR(I2)/AMAX                                                  ESY1 950
      C2 = -VI(I2)/AMAX                                                 ESY1 960
      DO 630 L = 1, N                                                   ESY1 970
      TEMP = VI(L)                                                      ESY1 980
      VI(L) = VR(L)*C2 + TEMP*C1                                        ESY1 990
  630 VR(L) = VR(L)*C1 - TEMP*C2                                        ESY11000
      IF(COUNT .EQ. 1) GO TO 632                                        ESY11010
      DO 631 LL = 1, N                                                  ESY11020
 631  DCERR=DMAX1(DCERR,DABS(VR(LL)-W(LL,3)),DABS(VI(LL)-W(LL,4)))
  632 IF(IVC2) 633, 638, 633                                            ESY11040
 633  AMAX=0.0D0
      DO 635 L = 1, N                                                   ESY11060
      TEMP = XR(L)**2 + XI(L)**2                                        ESY11070
      IF(TEMP - AMAX) 635, 635, 634                                     ESY11080
  634 AMAX = TEMP                                                       ESY11090
      I2 = L                                                            ESY11100
  635 CONTINUE                                                          ESY11110
      C1 = XR(I2)/AMAX                                                  ESY11120
      C2 = -XI(I2)/AMAX                                                 ESY11130
      DO 636 L = 1, N                                                   ESY11140
      TEMP = XI(L)                                                      ESY11150
      XI(L) = XR(L)*C2 + TEMP*C1                                        ESY11160
  636 XR(L) = XR(L)*C1 - TEMP*C2                                        ESY11170
      IF(COUNT .EQ. 1) GO TO 646                                        ESY11180
      DO 637 LL = 1, N                                                  ESY11190
 637  DCERR=DMAX1(DCERR,DABS(XR(LL)-W(LL,1)),DABS(XI(LL)-W(LL,2)))
C                                                                       ESY11210
C        TEST FOR CONVERGENCE.                                          ESY11220
C                                                                       ESY11230
  638 IF(COUNT .EQ. 1) GO TO 646                                        ESY11240
      CERR=DCERR
      IF(CERR .GE. 1.0E-4) GO TO 639                                    ESY11250
      IF(CERR .GE. CLIM) GO TO 648                                      ESY11260
      CLIM = CERR                                                       ESY11270
      IF(CLIM .LE. 1.0E-8) GO TO 648                                    ESY11280
  639 IF(COUNT .GE. 15) GO TO 68                                        ESY11290
  647 COUNT = COUNT + 1                                                 ESY11300
      IF(ROOTI) 642, 673, 642                                           ESY11310
  642 IF(IVC2) 640, 644, 640                                            ESY11320
  640 DO 641 LL = 1, N                                                  ESY11330
      W(LL,1) = XR(LL)                                                  ESY11340
  641 W(LL,2) = XI(LL)                                                  ESY11350
      IF(IVC1) 644, 610, 644                                            ESY11360
  644 DO 645 LL = 1, N                                                  ESY11370
      W(LL,3) = VR(LL)                                                  ESY11380
  645 W(LL,4) = VI(LL)                                                  ESY11390
      GO TO 699                                                         ESY11400
  646 CERR = 0.0                                                        ESY11410
      DCERR=0.0D0
      IF(ICC) 648, 647, 648                                             ESY11420
```

```
  648 ERR = CERR                                                        ESY11430
      COUNTE = COUNT                                                    ESY11440
      IF(ROOTI) 667, 668, 667                                           ESY11450
  667 DO 649 I = 1, N                                                   ESY11460
  649 A(I,I) = A(I,I) + ROOTR                                           ESY11470
      RETURN                                                            ESY11480
 68   WRITE(IO3,101) ROOTR,ROOTI,CERR
      GO TO 648                                                         ESY11500
C                                                                       ESY11510
C        REAL EIGENVECTORS.                                             ESY11520
C                                                                       ESY11530
   60 ISW = 1                                                           ESY11540
      DO 651 I = 1, N                                                   ESY11550
      DO 650 J = 1, N                                                   ESY11560
  650 B(I,J) = A(I,J)                                                   ESY11570
  651 B(I,I) = B(I,I) - ROOTR                                           ESY11580
      GO TO 700                                                         ESY11590
  652 IF(ICC) 680, 685, 680                                             ESY11600
C                                                                       ESY11610
C        SINGULAR MATRIX.                                               ESY11620
C                                                                       ESY11630
  680 IF(IVC2) 681, 683, 681                                            ESY11640
  681 DO 682 L = 1, N                                                   ESY11650
 682  XI(L)=0.0D0
      IF(IVC1) 683, 514, 683                                            ESY11670
  683 DO 684 L = 1, N                                                   ESY11680
 684  VI(L)=0.0D0
      GO TO 511                                                         ESY11700
C                                                                       ESY11710
C        MATRIX NOT SINGULAR.                                           ESY11720
C                                                                       ESY11730
  685 IF(IVC2) 653, 656, 653                                            ESY11740
  653 DO 654 L = 1, N                                                   ESY11750
 654  XI(L)=1.0D0
      IF(IVC1) 656, 500, 656                                            ESY11770
  656 DO 657 L = 1, N                                                   ESY11780
 657  VI(L)=1.0D0
      GO TO 499                                                         ESY11800
C                                                                       ESY11810
C        NORMALIZE REAL VECTORS.                                        ESY11820
C                                                                       ESY11830
 655  CERR=0.0
      DCERR=0.0D0
      IF(IVC2) 658, 662, 658                                            ESY11850
 658  C1=0.0D0
      C2=0.0D0
      DO 660 L = 1, N                                                   ESY11870
      TEMP=DABS(XI(L))
      IF(TEMP - C1) 660, 660, 659                                       ESY11890
  659 C1 = TEMP                                                         ESY11900
      C2 = XI(L)                                                        ESY11910
  660 CONTINUE                                                          ESY11920
      DO 661 L = 1, N                                                   ESY11930
      XI(L) = XI(L)/C2                                                  ESY11940
```

```
      DCERR=DMAX1(DCERR,DABS(XI(L)-XR(L)))
  661 XR(L) = XI(L)                                                 ESY11960
      IF(IVC1) 662, 638, 662                                        ESY11970
 662  C2=0.0D0
      C1=0.0D0
      DO 664 L = 1, N                                               ESY11990
      TEMP=DABS(VI(L))
      IF(TEMP - C1) 664, 664, 663                                   ESY12010
  663 C1 = TEMP                                                     ESY12020
      C2 = VI(L)                                                    ESY12030
  664 CONTINUE                                                      ESY12040
      DO 665 LL = 1, N                                              ESY12050
      VI(LL) = VI(LL)/C2                                            ESY12060
      DCERR=DMAX1(DCERR,DABS(VI(LL)-W(LL,1)))
      W(LL,1)=VI(LL)
  665 VR(LL)=W(LL,1)
      GO TO 638                                                     ESY12090
  668 IF(IVC2) 669, 671, 669                                        ESY12100
  669 DO 670 L = 1, N                                               ESY12110
 670  XI(L)=0.0D0
      IF(IVC1) 671, 70, 671                                         ESY12130
  671 DO 672 L = 1, N                                               ESY12140
 672  VI(L)=0.0D0
   70 RETURN                                                        ESY12160
  673 IF(IVC2) 674, 502, 674                                        ESY12170
  674 DO 675 I = 1, N                                               ESY12180
      I2 = IROW(I,2)                                                ESY12190
  675 XI(I2) = XR(I)                                                ESY12200
      GO TO 500                                                     ESY12210
C                                                                   ESY12220
C        BACK SUBSTITUTION SECTION.                                 ESY12230
C                                                                   ESY12240
  499 IF(IVC2) 500, 502, 500                                        ESY12250
  500 DO 501 I = 2, N                                               ESY12260
      I1 = I - 1                                                    ESY12270
      DO 501 J = 1, I1                                              ESY12280
  501 XI(I) = XI(I) - B(I,J)*XI(J)                                  ESY12290
  511 IF(IVC1) 502, 514, 502                                        ESY12300
  502 DO 510 I = 1, N                                               ESY12310
      I1 = I - 1                                                    ESY12320
      IF(I1) 503, 505, 503                                          ESY12330
  503 DO 504 J = 1, I1                                              ESY12340
  504 VI(I) = VI(I) - B(J,I)*VI(J)                                  ESY12350
      IF(ICC) 505, 506, 505                                         ESY12360
  505 IF(B(I,I)) 506, 507, 506                                      ESY12370
  506 VI(I) = VI(I)/B(I,I)                                          ESY12380
      GO TO 510                                                     ESY12390
  507 IF(VI(I)) 508, 509, 508                                       ESY12400
  508 VI(I) = VI(I)*1.0E+15                                         ESY12410
      GO TO 510                                                     ESY12420
 509  VI(I)=1.0D0
  510 CONTINUE                                                      ESY12440
      IF(IVC2) 514, 525, 514                                        ESY12450
  514 DO 522 I = 1, N                                               ESY12460
```

```
      IR = NP1 - I                                                      ESY12470
      IF(I - 1) 515, 517, 515                                           ESY12480
  515 I2 = IR + 1                                                       ESY12490
      DO 516 J = I2, N                                                  ESY12500
  516 XI(IR) = XI(IR) - B(IR,J)*XI(J)                                   ESY12510
      IF(ICC) 517, 518, 517                                             ESY12520
  517 IF(B(IR,IR)) 518, 519, 518                                        ESY12530
  518 XI(IR) = XI(IR)/B(IR,IR)                                          ESY12540
      GO TO 522                                                         ESY12550
  519 IF(XI(IR)) 520, 521, 520                                          ESY12560
  520 XI(IR) = XI(IR)*1.0E+15                                           ESY12570
      GO TO 522                                                         ESY12580
 521  XI(IR)=1.0D0
  522 CONTINUE                                                          ESY12600
      IF(IVC1) 525, 529, 525                                            ESY12610
  525 DO 526 I = 2, N                                                   ESY12620
      IR = NP1 - I                                                      ESY12630
      I2 = IR + 1                                                       ESY12640
      DO 526 J = I2, N                                                  ESY12650
  526 VI(IR) = VI(IR) - B(J,IR)*VI(J)                                   ESY12660
      DO 527 L = 1, N                                                   ESY12670
      I2 = IROW(L,1)                                                    ESY12680
  527 VR(I2) = VI(L)                                                    ESY12690
      DO 528 L = 1, N                                                   ESY12700
  528 VI(L) = VR(L)                                                     ESY12710
  529 IF(ROOTI) 615, 655, 615                                           ESY12720
C                                                                       ESY12730
C        FACTOR MATRIX.                                                 ESY12740
C                                                                       ESY12750
  700 ICC = 0                                                           ESY12760
      SW1=1.0D72
      DO 701 LL = 1, N                                                  ESY12780
  701 IROW(LL,1) = LL                                                   ESY12790
      DO 708 K = 1, N1                                                  ESY12800
      AMAX=DABS(B(K,K))
      IMAX = K                                                          ESY12820
      K1 = K + 1                                                        ESY12830
      DO 702 I = K1, N                                                  ESY12840
      IF(AMAX .GT. DABS(B(I,K))) GO TO 702
      AMAX=DABS(B(I,K))
      IMAX = I                                                          ESY12870
  702 CONTINUE                                                          ESY12880
      IF(AMAX .LT. SW1) SW1 = AMAX                                      ESY12890
      IF(AMAX .GE. 1.0D-25) GO TO 723
      B(K,K)=0.0D0
      ICC = ICC + 1                                                     ESY12920
      GO TO 708                                                         ESY12930
  723 IF(IMAX .EQ. K) GO TO 704                                         ESY12940
      DO 703 J = 1, N                                                   ESY12950
      AMAX = B(K,J)                                                     ESY12960
      B(K,J) = B(IMAX,J)                                                ESY12970
  703 B(IMAX,J) = AMAX                                                  ESY12980
      I2 = IROW(K,1)                                                    ESY12990
      IROW(K,1) = IROW(IMAX,1)                                          ESY13000
```

```
      IROW(IMAX,1) = I2                                                 ESY13010
  704 DO 707 I = K1, N                                                  ESY13020
      IF(B(I,K)) 705, 707, 705                                          ESY13030
  705 B(I,K) = B(I,K)/B(K,K)                                            ESY13040
      DO 706 J = K1, N                                                  ESY13050
  706 B(I,J) = B(I,J) - B(K,J)*B(I,K)                                   ESY13060
  707 CONTINUE                                                          ESY13070
  708 CONTINUE                                                          ESY13080
      AMAX= DABS(B(N,N))
      IF(AMAX- 1.0D-25) 712,712,713
712   B(N,N)=0.0D0
      SW1=0.0D0
      ICC = ICC + 1                                                     ESY13120
      GO TO 709                                                         ESY13130
  713 IF(AMAX .LT. SW1) SW1 = AMAX                                      ESY13140
  709 IF(ICC .LE. ISW) GO TO 710                                        ESY13150
      IF(MM) 1050,1050,1051
 1051 WRITE(I03,102) ICC
      COUNTE = 0                                                        ESY13180
      RETURN                                                            ESY13190
 1050 WRITE(I03,1052) ICC
  710 DO 711 LL = 1, N                                                  ESY13210
      I2 = IROW (LL,1)                                                  ESY13220
  711 IROW(I2,2) = LL                                                   ESY13230
      IF(ROOTI) 607, 652, 607                                           ESY13240
 1052 FORMAT(///23H ****** WARNING ******    ,'   SUBROUTINE EIGVEC HAS ESY13250
     1FOUND AN EIGENVALUE OF APPARENT MULTIPLICITY',                    ESY13260
     1                                   I4,/23X,'   COMPUTATION OF EIESY13270
     2GENVECTOR(S) CONTINUES AT USER S OPTION'//)                       ESY13280
  101 FORMAT(38HOMORE THAN 15 LOOPS FOR EIGENVECTOR OF,2E12.4,          ESY13290
     2   14H DIFFERENCE OF,E12.4)                                       ESY13300
  102 FORMAT(16HO****WARNING**** , I4, 71H ZEROS ON DIAGONAL OF FACTOREDESY13310
     1 MATRIX.  CHECK FOR MULTIPLE EIGENVALUES./20X,                    ESY13320
     2' SUBROUTINE EIGVEC WILL NOT PERFORM COMPUTATION FOR THIS EIGENVECESY13330
     3TOR '//)                                                          ESY13340
      END
```

```
      SUBROUTINE AQCAL (AQ,NS,NPA,IC1)
      DIMENSION AQ(10,10,3)
      DO 10 L=1,NPA
      DO 10 I=1,NS
      DO 10 J=1,NS
 10   AQ(I,J,L)=0.0
      GO TO (100,200,300,400,500),IC1
100   CONTINUE
      AQ(1,1,2)=1.0
      AQ(2,1,1)=-1.0
      RETURN
200   CONTINUE
      RETURN
300   CONTINUE
      RETURN
400   CONTINUE
      RETURN
500   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DINP (A,B,K,S,Q,NS,NC,NU,NOP)
      REAL K(10,10)
      DIMENSION A(10,10),B(10,10),S(10,10),Q(10,10)
15    FORMAT (5E15.5)
      IO1=1
      READ (IO1,15) ((A(I,J),I=1,NS),J=1,NS)
      READ (IO1,15) ((B(I,J),I=1,NS),J=1,NC)
      READ (IO1,15) ((K(I,J),I=1,NU),J=1,NC)
      READ (IO1,15) ((S(I,J),I=1,NS),J=1,NOP)
      READ (IO1,15) ((Q(I,J),I=1,NC),J=1,NOP)
      RETURN
      END

'DATA
```