NASA-CR-152306
19810003600

A Reproduced Copy

OF

Reproduced for NASA

by the

NASA Scientific and Technical Information Facility

SYSTEMS TECHNOLOGY, INC.

13766 SO. HAWTHORNE BOULEVARD • HAWTHORNE, CALIFORNIA 90250

Technical Report No. 1127-1

PRACTICAL OPTIMAL FLIGHT CONTROL SYSTEM DESIGN FOR HELICOPTER AIRCRAFT

Volume II

Software User's Guide

March 1979

(NASA-CR-152306) PRACTICAL OPTIMAL FLIGHT CONTROL SYSTEM DESIGN FOR HELICOPTER AIRCRAFT. VOLUME 2: SOFTWARE USER'S GUIDE Final Report (Systems Technology, Inc.) 254 p HC A12/MF A01 CSCL 01C G3/08

N81-12110

Unclas 40334

N91-12110 #

NASA CR-152306

SYSTEMS TECHNOLOGY, INC.

13766 SOUTH HAWTHORNE BOULEVARD • HAWTHORNE, CALIFORNIA 90250 • PHONE [213] 679-2281

Technical Report No. 1127-1

PRACTICAL OPTIMAL FLIGHT CONTROL SYSTEM DESIGN FOR HELICOPTER AIRCRAFT

Volume II

Software User's Guide

Susan A. Riedel

March 1979

Contract NAS2-9946

National Aeronautics and Space Administration
Ames Research Center
Moffett Field, CA 94035

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-152306		

4. Title and Subtitle	5. Report Date
PRACTICAL OPTIMAL FLIGHT CONTROL SYSTEM DESIGN FOR HELICOPTER AIRCRAFT Volume II: Software User's Guide	March 1979
	6. Performing Organization Code
7. Author(s) Susan A. Riedel	8. Performing Organization Report No. TR-1127-1-II
9. Performing Organization Name and Address Systems Technology, Inc. 13766 South Hawthorne Boulevard Hawthorne, California 90250	10. Work Unit No. (TRAIS)
	11. Contract or Grant No. NAS2-9946
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Ames Research Center Moffett Field, California 94035	13. Type of Report and Period Covered Final Report
	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A method by which modern and classical control theory techniques may be integrated in a synergistic fashion and used in the design of practical flight control systems is presented here. A general procedure is developed, and several illustrative examples are included. Emphasis is placed not only on the synthesis of the design, but on the assessment of the results as well.

The first step here is to establish the differences, distinguishing characteristics and connections between the modern and classical control theory approaches. Ultimately, this uncovers a relationship between bandwidth goals familiar in classical control and cost function weights in the equivalent optimal system. In order to obtain a practical optimal solution, it is also necessary to formulate the problem very carefully, and each choice of state, measurement and output variable must be judiciously considered. These so-called "engineering art" matters allow us to bridge the gap between the optimal control theory and its practical application.

Once design goals have been established and problem formulation completed, the control system is synthesized in a straightforward manner. Three steps are involved: filter-observer solution, regulator solution and the combination of those two into the controller. Assessment of the controller, which is often the bulk of the task at hand, permits an examination and expansion of the synthesis results. Often, the composite picture which results may lead to a revised design which is simpler and more practical.

(continued on following page)

17. Key Words	18. Distribution Statement
Optimal Control-Theory, Helicopter Flight Control Systems, Singular Kalman Filter, Optimal Control-Software, Optimal Control-Application, Cost Function Weights, Classical Control Analysis	

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

Abstract (concluded)

Key contributions in this work include the solution of the singular Kalman filter problem and the development of a user-oriented computer software package for the design and assessment of optimal control systems. Volume I of this report is the technical report.

FOREWORD

This report was prepared by Systems Technology, Inc., Hawthorne, California, under National Aeronautics and Space Administration Contract NAS2-9946. The program was administered by the NASA/Ames Research Center, Moffett Field, California. The NASA technical monitor was Dr. Heinz Erzberger.

The contract work was performed during the period May 1978 to May 1979. The draft of this report was submitted in April 1979.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES (Concluded)

LIST OF TABLES

SECTION I

INTRODUCTION

This manual describes a software system which integrates the design principles from optimal control theory and those from classical control theory to allow a user to design and analyze a control system. The interactive and very modular approach used here permits synthesis to proceed in exactly that sequence of steps which the user deems most appropriate for the particular problem. Figure 1 illustrates the block structure which forms the backbone of the design system. Each block is a separate executable file which performs the specific task noted in the figure. The user interaction with each block is minimal; in some cases, all that is required is input of two file names; in other cases, additional data is needed. The file system structure allows the different blocks to communicate information to one another. At each block, a "problem file" is read and/or written. The problem files are unformatted binary files (to conserve disk space) which store all intermediate results. The user accesses the problem files via a service routine, which selectively reads and formats to the line printer any user-requested elements in a given problem file.

There are two types of blocks shown in Fig. 1a. The first type (Blocks 1, 3, 4, 5, 6, 9, 10) implements various aspects of the optimal control design process using software adapted from Ref. 1. All output from these blocks is stored in problem files. The second type of block (Blocks 2, 7, 8) provides links between the optimal control design and the classical control analysis techniques. These blocks produce TRFN files, which can then be used in the TRFN/USAM2 software, parts of which are indicated in Fig. 1b. For further information concerning use of this software, see Ref. 2. The typical sequence of steps in designing an optimal control system is to begin with Block 1 at the top of Fig. 1a and work clockwise through all the blocks, making the appropriate indicated excursions to the blocks in Fig. 1b to use the classical control analysis methods. Blocks can be repeated to

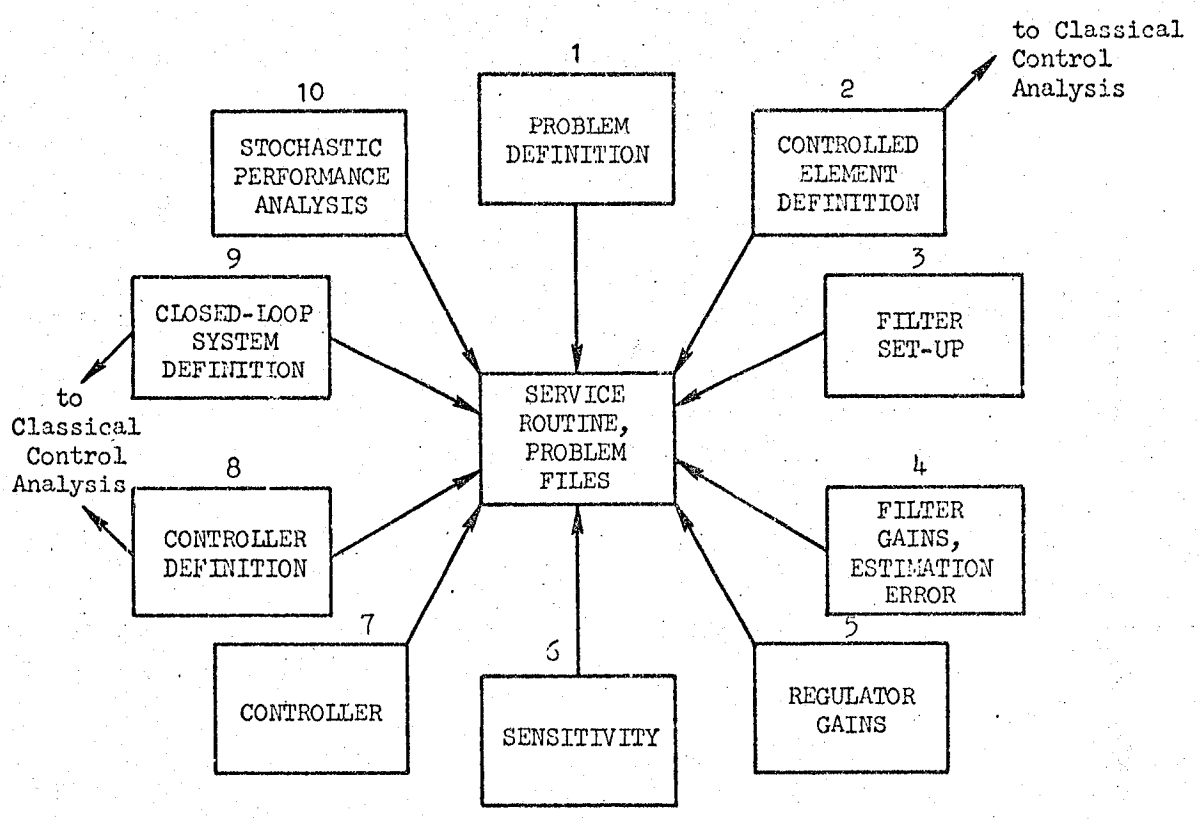

Figure 1a. Software Structure, Optimal Control

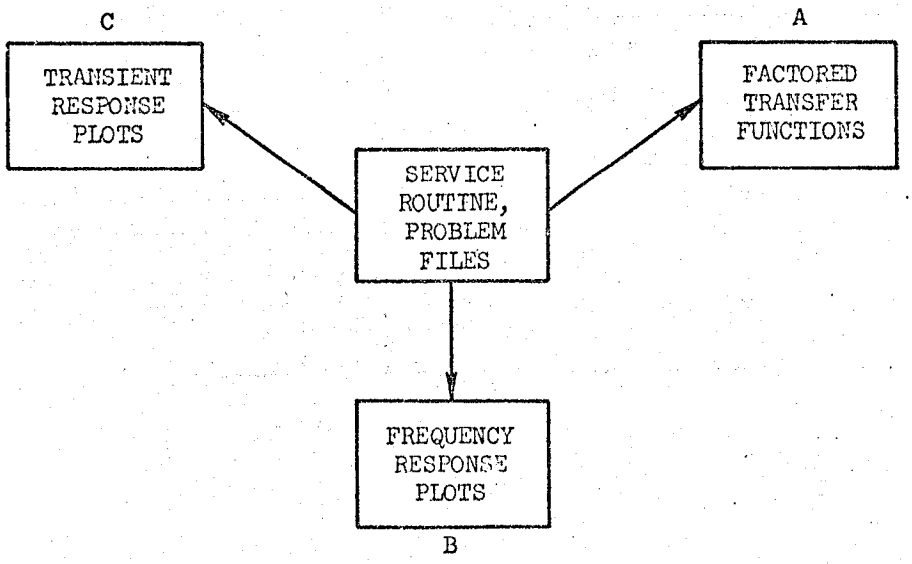

Figure 1b. Software Structure, Classical Control

cycle a design. Some blocks (such as the sensitivity block, for example) can be bypassed altogether.

This software was written in FORTRAN IV and developed on a PDP 11/10 under the RT-11 operating system. The system requires a single job monitor. The service routine and all the problem files reside on the DX0: disk; the executable files for the blocks reside on the DX1: disk. To execute a given block, simply type

RUN DX1:BLOCXX

where XX = 01, 02, ... 10. To run the service routine, type

RUN DX0:SERVIC

This manual is organized to correspond with the blocks — each of the sections describes a single block (and the service routine), including purpose, inputs required, computations performed and output, applicable restrictions, example input and output. A simple two state, two control point example is used to illustrate software use throughout. The appendix contains the source code listing for each mainline and subroutine in the optimal control package. For information on the theory and computational methods used in this software package, see Ref. 3.

SECTION II

PROBLEM DEFINITION — BLOCK 1

Purpose: This block accepts the dimensions, mnemonics and non-zero elements of all the matrices needed to define the problem's plant, outputs, and measurements in state vector form; it must always be accessed before attempting to solve any optimal control problem.

Input: The general state vector form assumed for the optimal control problem is stated below:

$$\dot{x} = Fx + Gu + \Gamma w, \; x(0) = X_o \qquad (1)$$

$$z = Hx + v \qquad (2)$$

$$y = H_R x \qquad (3)$$

where x = state vector

u = control vector

w = process noise vector

z = measurement vector

v = measurement noise vector

y = output vector

The particular form of the problem used here is shown in block diagram form in Fig. 2. The non-zero elements of the matrices shown must be input at this point in the problem, as well as certain dimensions, mnemonics, and file names. Notice that the state vector is comprised of two parts: the shaping filter states (x_S) and the controlled element states (x_C). The output vector, y, also has two components: those

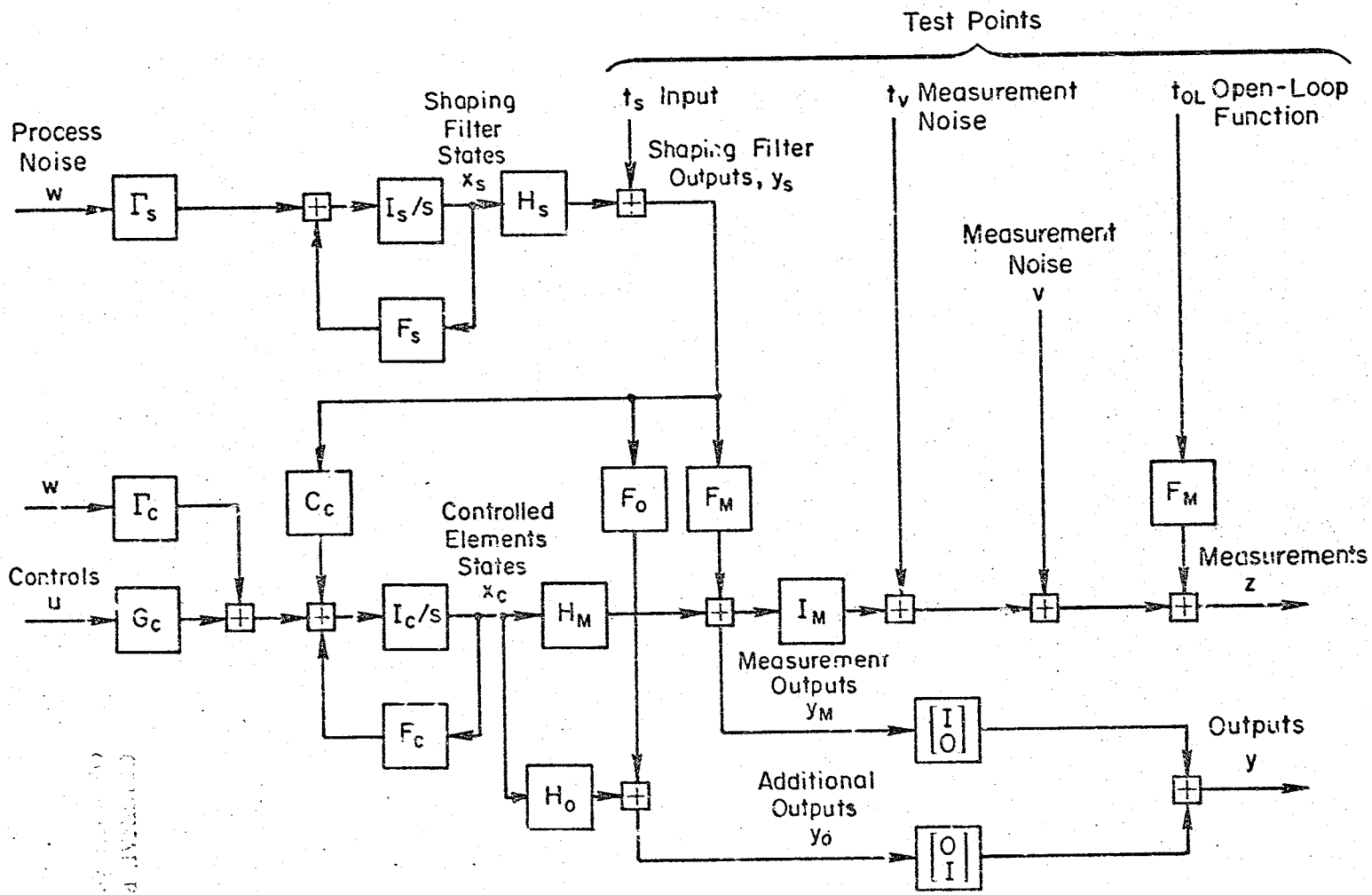

Figure 2. Form of Problem; Plant Measurements and Outputs

outputs which are, exclusive of measurement noise, the measurements (y_M), and any other outputs desired (y_O). Finally, provision is made for three vector test points: at the input point (t_S), at the measurement noise point (t_V), and for obtaining opened-loop transfer functions (t_{OL}).

Figure 3 presents the form of the data file which must be input in Block 1. The first line contains a 20 character title associated with this problem. The second line contains the dimensions of the problem vectors, in the order indicated. These dimensions are defined as follows:

n_{xS} - number of shaping filter states

n_{xC} - number of controlled element states

n_u - number of control inputs

n_w - number of process noise inputs

n_z - number of measurements

n_{yS} - number of shaping filter outputs

n_{yO} - number of additional outputs

All dimensions must be input, even if zero, and must be integers separated by spaces or commas.

The next several lines contain three-character mnemonics, separated by a single comma or space, associated with elements in the various vectors. The dimensions of these vectors are indicated in parentheses in Fig. 3. If a particular vector has zero dimension, no mnemonics are input.

Following the mnemonics are the non-zero elements of the matrices used to define the problem. The order of the matrices and their dimensions is shown in Fig. 3. The form of the input is:

i,j,XX.XX

AAA...	20 character title
$n_{xS}, n_{xC}, n_u, n_w, n_z, n_{yS}, n_{y_O}$	vector dimensions
MU1,MU2,...	u mnemonics (n_u)
MW1,MW2,...	w mnemonics (n_w)
MYS,MYS,...	y_S mnemonics ≡ t_S mnemonics (n_{yS})
MV1,MV2,...	v mnemonics ≡ t_V mnemonics (n_z)
MT1,MT2,...	t_{OL} mnemonics (n_{yS})
MZ1,MZ2,...	z mnemonics (n_z)
MYO,MYO,...	y_O mnemonics (n_{y_O})
i,j,Γ_S(i,j)	Γ_S elements (n_{xS}, n_w)
0	
i,j,F_S(i,j)	F_S elements (n_{xS}, n_{xS})
0	
i,j,H_S(i,j)	H_S elements (n_{yS}, n_{xS})
0	
i,j,Γ_C(i,j)	Γ_C elements (n_{xC}, n_w)
0	
i,j,G_C(i,j)	G_C elements (n_{xC}, n_u)
0	
i,j,C_C(i,j)	C_C elements (n_{xC}, n_{yS})
0	
i,j,F_C(i,j)	F_C elements (n_{xC}, n_{xC})
0	
i,j,H_M(i,j)	H_M elements (n_z, n_{xC})
0	
i,j,F_M(i,j)	F_M elements (n_z, n_{yS})
0	
i,j,H_O(i,j)	H_O elements (n_{y_O}, n_{xC})
0	
i,j,F_O(i,j)	F_O elements (n_{y_O}, n_{yS})
0	
DEV:FILE.EXT	output problem file name

Figure 3. Form of Input Data File for Block 1

where i is the integer representing the row number of the element, j is the integer representing the column number of the element, and XX.XX is the real number value (non-zero) of the element. Each non-zero element occupies a separate line, and the end of a particular matrix is signaled by the integer zero. If a matrix has zero dimensions, the integer zero must still be used.

The final line in this data file contains the name of the problem file to be used in outputting the data. This name must conform to RT-11 file name conventions.

Output: All data input is written to the output problem file. In addition, the input is combined into augmented arrays so that the problem is formulated in terms of the more conventional state vector equations. Those equations and the composition of the various matrices are given below. <u>All</u> matrices are written to the problem file.

$$\dot{x} = Fx + Gu + \Gamma w + E_3 t_S$$

$$y_M = Hx + E_2 t_S$$

$$z = I_M y_M + v + t_V + F_M t_{OL}$$

$$y = H_R x + E_1 t_S$$

where

$$x(n_{x_T}) = \begin{Bmatrix} x_S(n_{x_S}) \\ x_C(n_{x_C}) \end{Bmatrix} \quad \left[n_{x_T} = n_{x_S} + n_{x_C} \right]$$

$$F(n_{x_T}, n_{x_T}) = \begin{bmatrix} F_S(n_{x_S}, n_{x_S}) & 0(n_{x_S}, n_{x_C}) \\ C_C(n_{x_C}, n_{y_S}) \times H_S(n_{y_S}, n_{x_S}) & F_C(n_{x_C}, n_{x_C}) \end{bmatrix}$$

$$G(n_{x_T}, n_u) = \begin{bmatrix} 0(n_{x_S}, n_u) \\ G_C(n_{x_C}, n_u) \end{bmatrix}$$

$$\Gamma(n_{x_T}, n_w) = \begin{bmatrix} \Gamma_S(n_{x_S}, n_w) \\ \Gamma_C(n_{x_C}, n_w) \end{bmatrix}$$

$$E_3(n_{x_T}, n_{x_S}) = \begin{bmatrix} 0(n_{x_S}, n_{x_S}) \\ C_C(n_{x_C}, n_{x_S}) \end{bmatrix}$$

$$H(n_z, n_{x_T}) = \begin{bmatrix} F_M(n_z, n_{y_S}) \times H_S(n_{y_S}, n_{x_S}) & H_M(n_z, n_{x_C}) \end{bmatrix}$$

$$E_2(n_z, n_{y_S}) = F_M(n_z, n_{y_S})$$

$$y(n_{y_T}) = \begin{Bmatrix} y_M(n_z) \\ y_0(n_{y_0}) \end{Bmatrix} \quad \begin{bmatrix} n_{y_T} = n_z + n_{y_0} \end{bmatrix}$$

$$H_R(n_{y_T}, n_{x_T}) = \begin{bmatrix} F_M(n_z, n_{y_S}) \times H_S(n_{y_S}, n_{x_S}) & H_M(n_z, n_{x_C}) \\ F_0(n_{y_0}, n_{y_S}) \times H_S(n_{y_S}, n_{x_S}) & H_0(n_{y_0}, n_{x_C}) \end{bmatrix}$$

$$E_1(n_{y_T}, n_{y_S}) = \begin{bmatrix} F_M(n_z, n_{y_S}) \\ F_0(n_{y_0}, n_{y_S}) \end{bmatrix}$$

Restrictions: The following restrictions apply to data input for this portion of the problem:

1. The error message

   DIMENSIONS TOO LARGE OR Z IS 0

   is printed when any or all of the following occur

   $n_{x_S} > 5$

   $n_{x_C} > 10$

   $n_u > 5$

$n_w > 15$

$n_z > 15$, $n_z = 0$

$n_{y_S} > 5$

$n_{y_O} > 5$

2. The error message

   DIMENSIONS OUT OF RANGE FOR i,j,X(i,j)

   is printed if a matrix element is specified with an i or j which exceeds prescribed dimensions.

3. The error message

   WRONG FORMAT

   is printed if a matrix element is not recognizable (e.g., too many commas).

4. The RT-11 operating system will print fatal error messages if the file name is specified incorrectly, or if there is not enough room on the disk to write the file.

5. Certain other restrictions are not specifically flagged in the software. Their satisfaction is the responsibility of the user:

   - The software is restricted to that class of singular filter problems which do not require differentiation of any measurement.
   - The above requirement can be met via augmentation of the Γ matrix. This augmentation is provided by the user. The user must specify a process noise vector dimension (n_w) which is at least equal to the total number of states

$(n_{x_T})^*$. Augmentation of the process noise vector is considered to occur in the last n_{x_C} elements. Augmentation is operative only for filter problem solution (Blocks 3 and 4). In all other blocks, the effects of the last n_{x_C} process noise components are deleted.

- The user must verify independently that the plant is both detectable and stabilizable. This may be verified by requiring stability of the shaping filters and by examining various arrays of factored controlled element transfer functions.

Example: Figure 4 is the computer dialog and input file listing for this two state, two control example problem. Figure 5 presents the F,G, Γ, and H matrices which are output to the problem file. Note that all data in the example output in this manual have been read from the problem file and formatted via the service routine. Further notice that the first controlled element state is X06. This is the case because the first five plant state vector elements are reserved for shaping filter states.

* Any (or all) of the augmenting components may be zeroed by appropriate specification of the Γ matrix and/or the process noise intensities.

```
PROBLEM DEFINITION — BLOCK 1

INPUT DATA FILE NAME? DXO:2BY2.DAT*
```

Figure 4a. Input for Block 1

```
2X2 EXAMPLE CASE          - Problem Definition Title
Ø,2,2,4,2,Ø,Ø             - Dimensions
UØ1,UØ2
WØ1,WØ2,WX1,WX2           - MNEMONICS
VØ1,VØ2
ZØ1,ZØ2
Ø                         - Γ_S
Ø                         - F_S
Ø                         - H_S
1,1,1.
2,2,1.
1,3,1.                    - Γ_C
2,4,1.
Ø
1,1,1.
2,2,1.                    - G_C
Ø                         - C_C
Ø
1,2,1.
2,1,-2.                   - F_C
2,2,-3.
Ø
1,1,1.
2,2,1.                    - H_M
Ø
Ø                         - F_M
Ø                         - H_O
Ø                         - F_O
DXO: 2BY2.B1              - OUTPUT FILE NAME
```

Figure 4b. Listing of DXO:2BY2.DAT

*In this report, all user responses in computer dialog are underlined.

```
VECTOR DIMENSIONS :
  NXS = 0 NXC =  2 NU =  2 NW =  2 NZ =  2 NYS =  0 NYO =  0
```

F MATRIX

	1 (X06)	2 (X07)	
1 (X06)	0.000	1.00	1 X06
2 (X07)	-2.00	-3.00	2 X07

G MATRIX

	1 (U01)	2 (U02)	
1 (X06)	1.00	0.000	1 X06
2 (X07)	0.000	1.00	2 X07

GAMMA MATRIX

	1 (W01)	2 (W02)	3 (WX1)	4 (WX2)	
1 (X06)	1.00	0.000	1.00	0.000	1 X06
2 (X07)	0.000	1.00	0.000	1.00	2 X07

H MATRIX

	1 (X06)	2 (X07)	
1 (Z01)	1.00	0.000	1 Z01
2 (Z02)	0.000	1.00	2 Z02

Figure 5. Output from Block 1; Problem Definition

SECTION III

CONTROLLED ELEMENT DEFINITION — BLOCK 2

Purpose: This block uses the matrices input in Block 1 to compose a TRFN-compatible file. TRFN is an interactive software package for computing the factored transfer function characteristic polynomial and numerators of various kinds from Laplace transformed equations of the form $A_{TF}(s)x_{TF}(s) = B_{TF}(s)u_{TF}(s)$. Block 2 accesses data output from Block 1.

Input: The input to this block consists of three items:

1. The name of the problem file from which the data are to be read.
2. The name of the output file to which the TRFN file is written.
3. A sixty character title for the TRFN file.

Output: The output from this block is a TRFN-compatible file. (For a complete description of TRFN see Ref. 2.) The general form of the file is

$$Ax_{TF} = Bu_{TF}$$

where

$$x_{TF}(n_{xC} + n_{yT}) = \begin{Bmatrix} x_C(n_{xC}) \\ y(n_{yT}) \end{Bmatrix}$$

$$u_{TF}(n_u + n_{yS} + n_w) \qquad \begin{Bmatrix} u(n_u) \\ t_S(n_{yS}) \\ w(n_w) \end{Bmatrix}$$

$$A(n_{x_C}+n_{y_T}, n_{x_C}+n_{y_T}) = \begin{bmatrix} sI-F_C(n_{x_C}, n_{x_C}) & 0(n_{x_C}, n_{y_T}) \\ -H_M(n_z, n_{x_C}) & I(n_{y_T}, n_{y_T}) \\ -H_O(n_{y_O}, n_{x_C}) & \end{bmatrix}$$

$$B(n_u+n_{y_S}+n_w) = \begin{bmatrix} G_C(n_{x_C}, n_u) & C_C(n_{x_C}, n_{y_S}) & \Gamma_C(n_{x_C}, n_w) \\ 0(n_{y_T}, n_u) & E_1(n_{y_T}, n_{y_S}) & 0(n_{y_T}, n_w) \end{bmatrix}$$

Restrictions: The RT-11 operating system will halt with a fatal error if the format of either file name is incorrect, or if there is not enough room on the disk for the file.

Example: Figure 6 shows the computer dialog for this block; Fig. 7 presents the listing of the resulting TRFN file.

```
CONTROLLED ELEMENT DEFINITION -- BLOCK 2

INPUT PROBLEM FILE NAME? DX0:2BY2.B1

OUTPUT TRFN FILE NAME? DX0:2BY2.B2

TRFN TITLE (60 CHARS):

2BY2 EXAMPLE CONTROLLED ELEMENT TRANSFER FUNCTIONS
```

Figure 6. Input for Block 2

```
          1       1   0.00000000E+00   0.10000000E+01   0.00000000E+00       0
          1       2   0.00000000E+00   0.00000000E+00  -0.10000000E+01       0
          2       1   0.00000000E+00   0.00000000E+00   0.27000000E+01       0
          2       2   0.00000000E+00   0.10000000E+01   0.30000000E+01       0
          3       1   0.00000000E+00   0.00000000E+00  -0.10000000E+01       0
          4       2   0.00000000E+00   0.00000000E+00  -0.10000000E+01       0
          3       3   0.00000000E+00   0.00000000E+00   0.10000000E+01       0
          4       4   0.00000000E+00   0.00000000E+00   0.10000000E+01       0
          0
          1       1   0.00000000E+00   0.00000000E+00   0.10000000E+01       0
          2       2   0.00000000E+00   0.00000000E+00   0.10000000E+01       0
          1       3   0.00000000E+00   0.00000000E+00   0.10000000E+01       0
          2       4   0.00000000E+00   0.00000000E+00   0.10000000E+01       0
          0
2BY2 EXAMPLE   CONTROLLED ELEMENT TRANSFER FUNCTIONS
          4       4       0
X03 X07 Z01 Z02
U01 U02 W01 W02
```

This output file format is described below:

I , J , A_2 , A_1 , A_0 , K ! 0	Nonzero elements of left-hand matrix*
I , J , B_2 , B_1 , B_0 , K ! 0	Nonzero elements of right-hand matrix*
Title: 60 characters of User's choice to identify the job.	
N , M , ID	Matrix sizes and code for Δ output†
AAA_1 , AAA_2 , ··· , AAA_N	Column code, left-hand side, 3 characters†§**
BBB_1 , BBB_2 , ··· , BBB_M	Column code, right-hand side, 3 characters†§

*If K = 0, the element in the Ith row, Jth column is set to: $A_2s^2 + A_1s + A_0$. If K = 1 and $A_2 \neq 0$, the element in the Ith row, Jth column is set to: $A_2[(s + A_1)(s + A_0)]$. If K = 1 and $A_2 = 0$, the element in the Ith row, Jth column is set to: $A_1[(s + A_0)]$. If K = 2, the element in the Ith row, Jth column is set to: $A_2[s^2 + 2A_1A_0s + A_0^2]$.

†If M = 0, the line of data will terminate the file read. If ID = 0, the Δ equation will be output. If ID ≠ 0, the denominator polynomial is set to unity and not printed.

†There must be at least N left-hand column identifiers and M right-hand identifiers. Each identifier must be three characters long.

§Commas and semicolons between column identifiers may be replaced with any other character.

**If there is more than 18 column identifiers, place a + in column 72 and continue on the next line.

Figure 7. Output File Listing for Block 2

SECTION IV

FILTER SET-UP — BLOCK 3

Purpose: This block is used to form various matrices which will be used by Block 4 to solve for the filter gains. Block 3 accesses data output from Block 1. The non-zero elements of the diagonal Q and R matrices are input for this block. The routine determines whether or not the problem is singular (one or more diagonal elements of R is zero). The Euler-Lagrange system matrix is computed accordingly in one of two ways, depending on whether or not the filter problem is singular. Results are written to the output problem file.

Input: The input to this routine consists of the following information in response to prompting by the software:

1. Input problem file, in standard RT-11 format.
2. Output problem file, in standard RT-11 format.
3. Filter problem title, limited to 20 characters.
4. Augmentation constant, a real number.
5. The non-zero diagonal elements of the Q matrix, in the form

   i, Q(i) $(n_w \geq n_{x_C})$

   where i is the integer row and column of the real number Q(i). The end of the Q input is signaled by an integer zero.
6. The non-zero diagonal elements of the R matrix, in the form

   i, R(i) (n_z)

where i is the integer row and column of the real number R(i). The end of the R input is signaled by an integer zero.

Output: Once input is completed, a check is made to determine whether the problem is singular. If all the diagonal elements of R are non-zero, the Euler-Lagrange system matrix is formed as follows:

$$EL_F = \begin{bmatrix} F & -GQG' \\ -H'R^{-1}H & -F' \end{bmatrix}$$

It, as well as the Q and R matrices (and some other data required for compatibility with the singular case) is written to the problem file.

If one or more of the diagonal elements of R are zero, the measurements are reordered so that all of the noise-free measurements are in the lower partition of z. This reordering affects the measurement and measurement noise mnemonics and transforms the H and R matrices. A T_z matrix is generated which transforms the original measurements to the reordered measurements:

$$T_z(n_z, n_z) \times z(n_z) = \begin{bmatrix} z_1(m_1) \\ z_2(m_2) \end{bmatrix}$$

where z_1 is the vector of noisy measurements

z_2 is the vector of noise-free measurements

m_1 is the number of noisy measurements

m_2 is the number of noise-free measurements.

Next, the state vector is transformed, if necessary, so that the lower partition of H is [0 I]. This results in a transformation of the state vector, changes in the mnemonics,

and appropriate transformations of the F, G, and Γ matrices, as well as the H matrix. A T matrix is defined which transforms the original state vector to the reordered state vector:

$$T(n_{x_T}, n_{x_T}) \times x(n_{x_T}) = \begin{bmatrix} x_1(m_1') \\ z_2(m_2) \end{bmatrix}$$

where x_1 is the vector of the remaining original states

m_1' is the number of remaining original states

Notice that the noise-free measurements become states as the result of this transformation. The remaining states (x_1) are a subset of the original plant states. The transformed F, G, Γ, R, and H matrices are partitioned as follows:

$$F(n_{x_T}, n_{x_T}) \longrightarrow \begin{bmatrix} F_{11}(m_1', m_1') & F_{12}(m_1', m_2) \\ F_{21}(m_2, m_1') & F_{22}(m_2, m_2) \end{bmatrix}$$

$$G(n_{x_T}, n_u) \longrightarrow \begin{bmatrix} G_1(m_1', n_u) \\ G_2(m_2, n_u) \end{bmatrix}$$

$$\Gamma(n_{x_T}, n_w) \longrightarrow \begin{bmatrix} \Gamma_1(m_1', n_w) \\ \Gamma_2(m_2, n_w) \end{bmatrix}$$

$$R(n_w) \longrightarrow \text{diag} \{ R_1(m_1) \quad Q(m_2) \}$$

$$H(n_{x_T}, n_z) \longrightarrow \begin{bmatrix} H_{11}(m_1', m_1) & H_{12}(m_1', m_2) \\ 0(m_2, m_1) & I(m_2, m_2) \end{bmatrix}$$

The reordered mnemonics, the T_z and T matrices, the partitions of the F, G, Γ, and H matrices, and the Q and transformed R matrices are all written to the problem file.

For this singular filter problem, the Euler-Lagrange system matrix is formed as follows:

$$EL_F = \begin{bmatrix} (F_{11} - \Gamma_1 Q \Gamma_2' A F_{21}) & (-\Gamma_1 Q \Gamma_1' + \Gamma_1 Q \Gamma_2' A \Gamma_2 Q \Gamma_1') \\ (-H_{11}' R^{-1} H_{11} - F_{21}' A F_{21}) & (-F_{11}' + F_{21}' A \Gamma_2 Q \Gamma_1') \end{bmatrix}$$

where $A = (\Gamma_2 Q \Gamma_2')^{-1}$

If necessary, $\Gamma_2 Q \Gamma_2'$ is augmented to have full rank as follows:

$$\Gamma_2 Q \Gamma_2' + \begin{bmatrix} AC_1 & & 0 \\ & AC_2 & \\ 0 & & \ddots \end{bmatrix} = A^{-1}$$

where the AC's have the value of the augmentation constant (an input at the beginning of this block) or zero. The augmentation proceeds by beginning with the upper left element of the $\Gamma_2 Q \Gamma_2'$ matrix and progressively building up the dimension of this upper left partition. At each stage of the buildup, the rank of the upper left partition is tested. This is shown schematically below:

$$\Gamma_2 Q \Gamma_2' = \begin{bmatrix} \text{-} \text{-} \text{-} & \text{-} & \\ \text{-} \text{-} \text{-} & \text{-} & \\ & & \ddots \end{bmatrix} \begin{matrix} \\ 1 \\ 2 \\ n \end{matrix}$$

If the nth partition of $\Gamma_2 Q \Gamma_2'$ is not of full rank, AC_n has the value of the augmentation constant. Otherwise, it has a zero value. This is a protective measure to insure that A exists. If A exists, the filter solutions obtained do not require differentiation of any measurement. This automatic augmentation of $\Gamma_2 Q \Gamma_2'$ can be avoided if the user specifies Γ_s, Γ_c, and Q such that $\Gamma_2 Q \Gamma_2'$ is nonsingular prior to augmentation. Both the EL_F and A matrices are written to the problem file, as well as the new dimensions (m_1, m_2, m_1'). For further information regarding the singular filter problem, see Ref. 4.

Restrictions: The following restrictions apply to data input in this portion of the problem:

1. RT-11 operating system will halt on a fatal error if either file name does not conform to its standard, or if there is not enough room on the disk to which to write the data.

2. The error message

   DIMENSIONS OUT OF RANGE FOR i, X(i)

   will be printed if a matrix diagonal element is specified with an i which exceeds the specified dimension.

3. The error message

   WRONG FORMAT

   is printed if a matrix element is unrecognizable (e.g., too many commas).

The following errors or warnings may occur in the course of the filter set up:

1. The following advisory information is printed if the G, Γ, or H matrices do not have full rank:

   G MATRIX DOES NOT HAVE FULL RANK
   GM MATRIX DOES NOT HAVE FULL RANK*
   H MATRIX DOES NOT HAVE FULL RANK

   Full rank is sometimes desired for each of these matrices so these advisories indicate possible problem ambiguities which should be considered before proceeding.

2. The following warning is printed if the matrix product $\Gamma_2 Q \Gamma_2'$ does not have full rank:

   GM2*QD*GM2T HAS BEEN AUGMENTED

   indicating that the augmentation constant was used to give $\Gamma_2 Q \Gamma_2'$ full rank so that it could be inverted.

* Rank test requires rank $n_w - n_{x_c}$ or greater.

Example: Figure 8 presents the input dialog for this block. Figure 9 presents some of the data which is output to the problem file. Notice that this problem is singular and requires reordering of the measurements and the states.

```
                FILTER SET-UP -- BLOCK 3

INPUT PROBLEM FILE NAME ? DX0:2BY2.B1
OUTPUT PROBLEM FILE NAME ? DX0:2BY2.B3
PROBLEM TITLE (20 CHARS) : FILTER SOLUTION
AUGMENTATION CONSTANT : 0.01
NON-ZERO DIAGONAL Q ELEMENTS
2,12.0
3,.01
0
NON-ZERO DIAGONAL R ELEMENTS :
2,4.
0
```

Figure 8. Input to Block 3

```
MEASUREMENT PARTITION : M1 =   1 M2 =   1
NUMBER OF FILTER STATES : M1P =  1
```

Q MATRIX DIAGONAL, FILTER 1

0.000	W01
12.0	W02
0.100E-01	WX1
0.000	WX2

R MATRIX DIAGONAL, FILTER 1

4.00	Z02
0.000	Z01

Notice Reordering of Measurements

F11 MATRIX 1 E07

-3.00	E07

F12 MATRIX 1 Z01

-2.00	E07

F21 MATRIX 1 E07

1.00	Z01

F22 MATRIX 1 Z01

0.000	Z01

Partioned F Matrix

EULER LAGRANGE SYS MATRIX, FILTER

1 E07	2 E07	
-3.00	-12.0	E07
-100.	3.00	E07

Closed-loop System Matrix from root square locus

Figure 9. Output from Block 3; Filter Solution Set-Up

SECTION V

FILTER GAINS, ESTIMATION ERROR — BLOCK 4

Purpose: This routine uses an eigenvector decomposition method to solve the algebraic Riccati equation using the Euler-Lagrange system matrix computed in Block 3. This results in closed loop eigenvalues and eigenvectors and Riccati equation solutions from which the filter gains can be computed. Also computed are the rms vector and correlation matrix for the state estimation error.

Input: The input to this block consists of the input problem file title and the output problem file title — both entries are prompted by the software.

Output: The filter solution is obtained via eigenvector decomposition using the QR algorithm (see Ref. 5). Open-loop plant eigenvalues and closed loop eigenvalues and eigenvectors are computed and written to the problem file (eigenvectors are normalized). The Riccati matrix, P_x, is computed and from this the filter gains are computed:

$$K_{11} = P_x H_{11}' R^{-1}$$

$$K = (P_x F_{21}' + \Gamma_1 Q \Gamma_2')A$$

The K_{11} gains operate on the noisy measurements, while the K_{12} gains operate on the noise-free measurements in generating estimates of the plant states. Figure 10 shows the form of the filter solution in terms of K_{11}, K_{12}, and the partitions of the plant matrices. Finally, the equation

$$\sigma_i = \sqrt{(T^{-1} P_x T^{-T})_{ii}}$$

is used to compute the rms estimation error distributed across the original states — the corresponding correlation

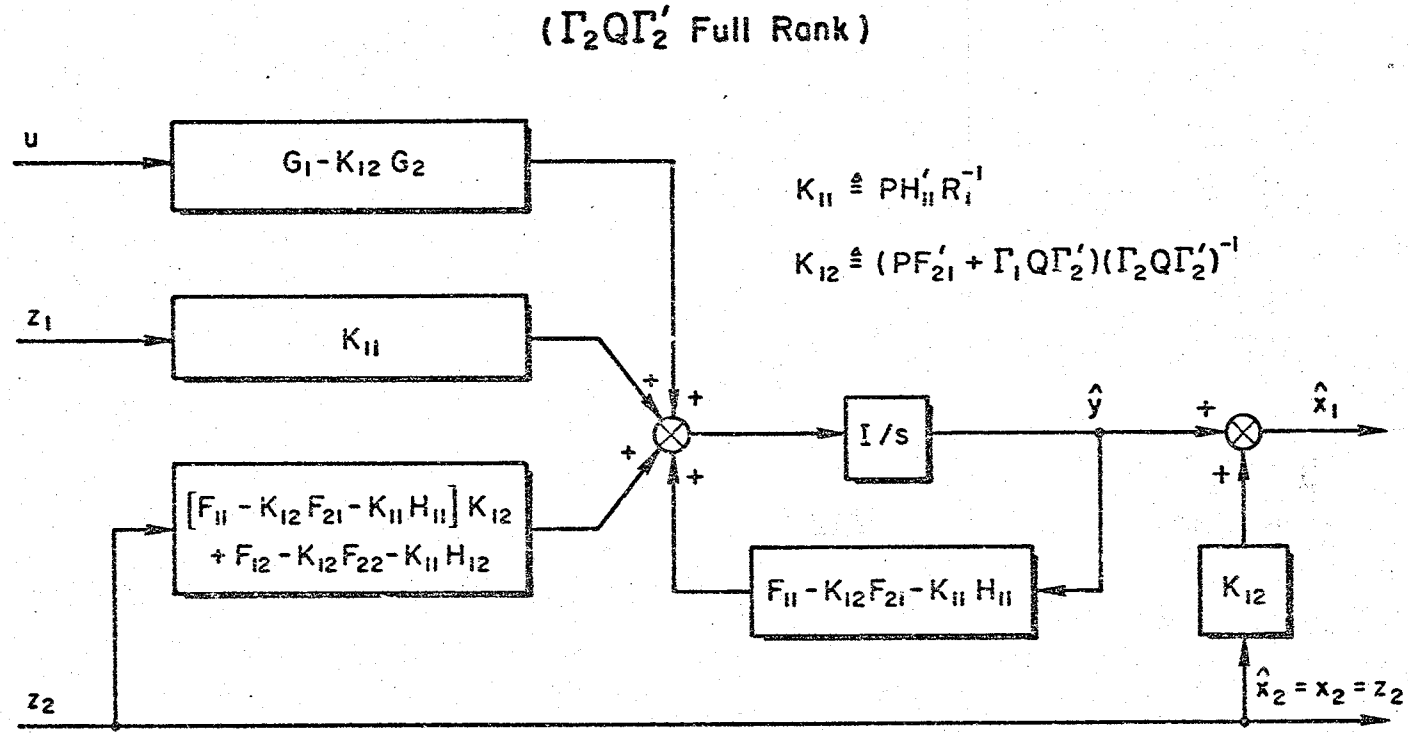

Figure 10. Filter-Observer Structure

matrix is also computed. The Riccati matrix, filter gains, rms vector, and correlation matrix are all written to the problem file.

Restrictions: On input a fatal error will be trapped by the RT-11 operating system if the filter names specified are not in accordance with the standard, or if there is not enough free space on the disk to write the output problem file. During the eigenvector decomposition and subsequent computations, the following errors can occur:

1. The error message

   ERROR COMPUTING CLOSED LOOP EIGENSYSTEM

   will be printed and execution halted if the eigensystem subroutine failed to converge on a given eigenvalue.

2. The warning

   REAL PART OF C.L. EIGENVALUE = 0.0

   will be printed if the real part of a closed-loop eigenvalue is identically zero. If this occurs, the selection of eigenvectors used in eigenvector decomposition will probably not be correct.*

3. The error message

   ERROR IN EIGENVALUE SELECTION

   will be printed and execution halted if there are not m_1 eigenvalues with positive real parts.*

4. The error message

   ERROR IN EIGENVECTOR DECOMPOSITION

   will be printed and execution halted if the matrix used to compute the Riccati matrix is singular.

---

* This is a limitation of the current software rather than a theoretical limition of the computational method.

5. The warning

FILTER RICCATI MATRIX NOT NON-NEGATIVE

will be printed if any diagonal element of P_x is less than zero. That element will be set to zero (as well as the intersecting row and column) elements and execution resumed.

Example: Figure 11 presents the input dialog for this block; Fig. 12 contains some of the matrices which are computed. Note that the closed loop filter eigenvalues will appear as a closed loop roots in the final closed loop system transfer functions.

```
FILTER GAINS, ESTIMATION ERROR — BLOCK 4
 INPUT PROBLEM FILE NAME ? DX0:2BY2.B3
 OUTPUT PROBLEM FILE NAME ? DX0:2BY2.B4
```

Figure 11. Input for Block 4

CLOSED LOOP EIGENVALUES, FILTER
1

34.8 180.	1 E07

Eigenvalues are in polar form with argument in degrees

K11 GAIN MATRIX, FILTER
1
Z02

0.793E-01	1 E07

K12 GAIN MATRIX, FILTER
1
Z01

31.7	1 E07

EXX denotes a filter state variable

RMS STATE EST ERROR, FILTER
1

0.000	1 X06
0.563	2 X07

Figure 12. Output from Block 4; Singular Filter Solution

SECTION VI

REGULATOR GAINS — BLOCK 5

Purpose: This block accepts user input of the Q_R and R_R non-zero diagonal elements and uses this information and information computed in Block 1 to solve the regulator problem. Note that solution of the regulator problem is in no way dependent upon the filter solution. The regulator problem, like the filter problem, is solved via eigenvector decomposition.

Input: The input to this routine is analogous to Block 3 and is prompted by the software. It includes:

1. Input problem file name.
2. Output problem file name.
3. Title for regulator solution, 20 characters.
4. The non-zero diagonal elements of the Q_R matrix, in the format

   $$i,\ Q_R(i)\quad (n_{yT})$$

   where i is the row and column integer of the real element $Q_R(i)$. The end of the Q_R input is signaled by an integer zero.
5. The non-zero diagonal elements of the R_R matrix, in the format

   $$i,\ R_R(i)\quad (n_u)$$

   where i is the row and column integer of the real element $R_R(i)$. The end of the R_R input is signaled by the integer zero.

Output: The first step in computing the regulator gains is to form the Euler-Lagrange system matrix, as follows:

$$EL_R = \begin{bmatrix} F' & H'Q_RH \\ GR_R^{-1}G' & -F \end{bmatrix}$$

Using this matrix, the open loop eigenvalues, closed loop eigenvalues, and normalized eigenvectors are computed. All are written to the output problem file, along with the Q_R, R_R, and EL_R matrices. The closed loop eigenvectors are used to compute the Riccati matrix via eigenvector decomposition. The Riccati matrix, S, is used in turn to compute the regulator gains:

$$C = R_R^{-1}G'S$$

The S and C matrices are written to the output problem file.

Restrictions: The following errors will be signaled if found in the input data:

1. RT-11 operating system will halt execution with a fatal error if the problem file names are not input in accordance with the proper file name specification, or if there is not enough contiguous free space on the disk to accommodate the output file.

2. The error message

   DIMENSIONS OUT OF RANGE FOR i, X(i)

   will be printed if i exceeds the specified dimension of X.

3. The error message

   WRONG FORMAT

   will be printed if the matrix entry is unrecognizable [e.g., decimal point missing from X(i)].

The following errors may be flagged during eigenvector decomposition and regulator gain computation:

1. The error message

   RRD DOES NOT HAVE FULL RANK

   will be printed and execution halted if any diagonal element of R_R is zero.

2. The error message

   ERROR COMPUTING CLOSED LOOP EIGENSYSTEM

   will be printed and execution halted if the eigensystem subroutine failed to converse on a given eigenvalue.

3. The warning

   REAL PART OF C.L. EIGENVALUE = 0.0

   will be printed if the real part of a closed loop eigenvalue is identically zero. If this occurs, the selection of eigenvectors used in eigenvector decomposition will probably not be correct.*

4. The error message

   ERROR IN EIGENVALUE SELECTION

   will be printed and execution halted if there are not n_{xT} eigenvalues with positive real parts.*

5. The error message

   ERROR IN EIGENVECTOR DECOMPOSITION

   will be printed and execution halted if the matrix used to compute the Riccati matrix is singular.

Example: Figure 13 is the computer dialog for Block 5; Fig. 14 is a sample of the output from this routine. Notice that the closed loop eigenvalues from the regulator solution are closed loop roots of the overall system.

---

* This characteristic of the software currently denies the possibility of designing optimal regulators having one or more closed loop eigenvalues having zero or negative real part. This software characteristic can be modified to accommodate these cases, as this limitation is not a theoretical one.

```
                REGULATOR GAINS — BLOCK 5

INPUT PROBLEM FILE NAME ? DXO:2BY2.B4
OUTPUT PROBLEM FILE NAME ? DXO:2BY2.B5
PROBLEM TITLE (20 CHARS) : REGULATOR SOLUTION
NON-ZERO DIAGONAL QR ELEMENTS:
2,5.
0
NON-ZERO DIAGONAL RR ELEMENTS :
1,.14286
2,.25
```

Figure 13. Input for Block 5

Q MATRIX DIAGONAL, REGULATOR 1

	1
0.000	Z01
	2
5.00	Z02

R MATRIX DIAGONAL, REGULATOR 1

	1
0.143	U01
	2
0.250	U02

CLOSED LOOP EIGENVALUES, REGULATOR 1

3.00	1
180.	X06
4.00	2
180.	X07

REGULATOR GAIN MATRIX

	1 X06	2 X07	
			1
	1.75	-0.875	U01
			2
	-0.500	2.25	U02

Figure 14. Output from Block 5; Regulator Solution

SECTION VII

SENSITIVITY — BLOCK 6

Purpose: This routine computes the normalized sensitivity of the closed loop regulator eigenvalues to changes in the F, G, and C matrices. Such sensitivity calculations can be used to identify the important elements of a given gain matrix or parameters in the plant matrices to which the controller design is highly sensitive. The normalized sensitivity can be considered a measure of the fractional eigenvalue shift per unit fractional change in a matrix element. For further treatment of this topic, see Ref. 6. Execution of this block must be preceded by execution of Block 5.

Input: Input to this block consists of three elements: the input problem file name, the output problem file name, and the 20 character title to be associated with this portion of the problem.

Output: Mathematically, the following normalized sensitivities are computed for each distinct eigenvalue and stored in the output problem file:

$$\frac{\partial \lambda_i}{\partial F_{jk}} = \frac{\frac{\partial \lambda_i}{\partial F_{jk}} F_{jk}}{\lambda_i}$$

$$\frac{\partial \lambda_i}{\partial G_{jk}} = \frac{\frac{\partial \lambda_i}{\partial G_{jk}} G_{jk}}{\lambda_i}$$

$$\frac{\partial \lambda_i}{\partial C_{jk}} = \frac{\frac{\partial \lambda_i}{\partial C_{jk}} C_{jk}}{\lambda_i}$$

where

$\hat{(\cdot)}$ is normalized of $(\cdot)$

λ_i is the ith eigenvalue

X_{jk} (where X = F, G, or C) is the element of X in the jth row, kth column

Thus, for each eigenvalue three matrices are computed; each matrix contains complex numbers which indicate the normalized sensitivity of the particular eigenvalue to the corresponding element in the original gain matrix. Small numbers in the sensitivity matrix indicate insensitivity of the appropriate regulator eigenvalue (and thus, the appropriate closed loop root) to the element in the corresponding matrix. All sensitivity matrices are written to the output problem file.

Restrictions: RT-11 operating system will halt with a fatal error if the problem file names are incorrectly specified, or if contiguous disk space is not available for the output file.

The error message

ERROR INVERTING EIGENVECTOR MATRIX

will be printed and execution halted if the eigenvector matrix is singular.

Example: Figure 15 is the input dialog for this block. Figure 16 presents illustrative sensitivity matrices for the example problem.

```
        SENSITIVITY — BLOCK 6

INPUT PROBLEM FILE NAME ? DX0:2BY2.B5

OUTPUT PROBLEM FILE NAME ? DX0:2BY2.B6

PROBLEM TITLE (20 CHARS) : SENSITIVITY
```

Figure 15. Input for Block 6

NORMALIZED F MATRIX SENSITIVITY TO EIGENVALUE : -4.00 0.000

1 X06	2 X07	
0.000 90.0	0.500 0.000	1 X06
1.25 0.000	1.25 180.	2 X07

Sensitivity vectors are in polar form with the argument in degrees

NORMALIZED G MATRIX SENSITIVITY TO EIGENVALUE : -4.00 0.000

1 U01	2 U02	
1.75 180.	0.000 90.0	1 X06
0.000 90.0	1.25 0.000	2 X07

NORMALIZED C MATRIX SENSITIVITY TO EIGENVALUE : -4.00 0.000

1 X06	2 X07	
1.31 180.	0.437 180.	1 U01
0.312 0.000	0.937 0.000	2 U02

Figure 16. Output for Block 6; Closed-Loop Regulator Eigenvalue Sensitivities

SECTION VIII

CONTROLLER --- BLOCK 7

Purpose: This routine combines the filter and the regulator solutions to compute coefficient matrices which define the optimal controller. Use of this block presumes a filter solution; a regulator solution is optional. If a regulator solution is not provided, the regulator gain matrix, C, is set to zero.

Input: Inputs to this block consist of the input problem file name, the output problem file name, and a 20 character problem title. All inputs are prompted by the software.

Output: The controller structure is defined by the following equations.

$$\dot{\hat{y}}(m_1') = A_F(m_1', m_1') \times \hat{y}(m_1') + B_F(m_1', n_z) \times z(n_z)$$

$$u(n_u) = C_F(n_u, m_1') \times \hat{y}(m_1') + D_F(n_u, n_z) \times z(n_z)$$

where $\hat{y}$ = filter states

The coefficient matrices in the controller equations are defined by:

$$A_F = \left[F_{11}-K_{12}F_{21}-K_{11}H_{11}-(G_1-K_{12}G_2)CT^{-1}\begin{bmatrix}I\\0\end{bmatrix}\right]$$

$$B_F = \left[K_{11} \;\vdots\; (F_{11}-K_{12}F_{21}-K_{11}H_{11})K_{12}+F_{12}-K_{12}F_{22}-K_{11}H_{12}\right] T_z$$
$$-\left[(G_1-K_{12}G_2)CT^{-1}\begin{bmatrix}0 & K_{12}\\0 & I\end{bmatrix}\right]T_z$$

$$C_F = -CT^{-1}\begin{bmatrix}I\\0\end{bmatrix}$$

$$D_F = -CT^{-1}\begin{bmatrix}0 & K_{12}\\0 & I\end{bmatrix}T_z$$

In addition, a controller system matrix, A_c, is computed to be used in the closed loop system performance analysis, Block 10:

$$A_c(n_{xT} + m_i', n_{xT} + m_i') = \begin{bmatrix} F + G[D_F \mid 0]H & GC_F \\ [B_F \mid 0]H & A_F \end{bmatrix}$$

The controller coefficient matrices and the A_c matrix are written to the output problem file.

Restrictions: RT-11 operating system will halt with a fatal error if the input and output problem file names are incorrectly specified, or if there is not enough contiguous disk space for the output problem file.

Example: Figure 17 is the input dialog for this block; Fig. 18 presents the controller coefficient matrices for the example problem.

```
CONTROLLER --- BLOCK 7

INPUT PROBLEM FILE NAME ? DX0:2BY2.B6
OUTPUT PROBLEM FILE NAME ? DX0:2BY2.B7
PROBLEM TITLE (20 CHARS) : CLOSED LOOP PA
```

Figure 17. Input for Block 7

AF MATRIX

1 E07	
-64.8	1 E07

BF MATRIX

1 Z01	2 Z02	
-0.200E+04	0.793E-01	1 E07

CF MATRIX

1 E07	
0.875	1 U01
-2.25	2 U02

DF MATRIX

1 Z01	2 Z02	
26.0	0.000	1 U01
-70.9	0.000	2 U02

Figure 18. Output for Block 7; Controller Coefficient Matrices

SECTION IX

CONTROLLER DEFINITION --- BLOCK 8

Purpose: This routine uses the controller equations generated in Block 7 to produce a TRFN-compatible file. Controller transfer functions can be computed using this file and the TRFN program (Ref. 2).

Input: The inputs to this block consist of the input problem file name, the output TRFN file name, and a 60 character title for the TRFN file. All are prompted by the software.

Output: The output from this block is a TRFN-compatible file — for a complete description, see Ref. 2. The general form of the controller representation is:

$$A_{TF}x_{TF} = B_{TF}u_{TF}$$

where

$$x_{TF}(m_1'+n_u) = \begin{Bmatrix} \hat{y}(m_1') \\ u(n_u) \end{Bmatrix}$$

$$u_{TF}(n_z) = \begin{Bmatrix} z_1(m_1) \\ z_2(m_2) \end{Bmatrix}$$

$$A(m_1'+n_u, m_1'+n_u) = \begin{bmatrix} (sI-[F_{11}-K_{12}F_{21}-K_{11}H_{11}]) & (-G_1+K_{12}G_2) \\ C_T^{-1}\begin{bmatrix} I \\ 0 \end{bmatrix} & I \end{bmatrix}$$

$$B(m_1'+n_u, n_z) = \begin{bmatrix} K_{11} & ([F_{11}-K_{12}F_{21}-K_{11}H_{11}]K_{12}+F_{12}-K_{12}F_{22}-K_{11}H_{12}) \\ 0 & -C_T^{-1}\begin{bmatrix} 0 & K_{12} \\ 0 & I \end{bmatrix} \end{bmatrix}$$

Restrictions: RT-11 operating system will halt with a fatal error if either file name is specified incorrectly, or if there is not enough contiguous space on the disk to accommodate the output TRFN file.

Example: Figure 19 is the dialog used on input to this block; Fig. 20 lists the resulting TRFN file.

```
        CONTROLLER DEFINITION --- BLOCK 8

INPUT PROBLEM FILE NAME ? DX0:2BY2.B7

OUTPUT TRFN FILE NAME ? DX0:2BY2.B8

TRFN TITLE (60 CHARS) :

2BY2 EXAMPLE CONTROLLER TRANSFER FUNCTIONS
```

Figure 19. Input for Block 8

```
      1       1  0.0000000E+00  0.1000000E+01  0.3489312E+02       0
      2       1  0.0000000E+00  0.0000000E+00 -0.8749831E+00       0
      3       1  0.0000000E+00  0.0000000E+00  0.2250000E+01       0
      1       2  0.0000000E+00  0.0000000E+00  0.3173445E+02       0
      1       3  0.0000000E+00  0.0000000E+00 -0.1000000E+01       0
      2       2  0.0000000E+00  0.0000000E+00  0.1000000E+01       0
      3       3  0.0000000E+00  0.0000000E+00  0.1000000E+01       0
      0
      1       1  0.0000000E+00  0.0000000E+00  0.7933613E-01       0
      1       2  0.0000000E+00  0.0000000E+00 -0.1106796E+04       0
      2       2  0.0000000E+00  0.0000000E+00  0.2601713E+02       0
      3       2  0.0000000E+00  0.0000000E+00 -0.7090252E+02       0
      0
2BY2 EXAMPLE   CONTROLLER TRANSFER FUNCTIONS
      3       2       0
E07 U01 U02
Z02 Z01
```

This output file format is described in Fig. 7.

Figure 20. Output File Listing for Block 8

SECTION X

CLOSED LOOP SYSTEM DEFINITION — BLOCK 9

Purpose: This routine uses the controlled element equation developed in Block 1 and the controller equations from Block 7 to compute a TRFN-compatible file from which closed loop system transfer functions can be obtained. In addition, the I_M identity matrix (refer to Fig. 2) can have some or all of its diagonal elements zeroed. This has the effect of selectively opening loops at the point of measurement. This, in turn, allows computation of any desired opened loop transfer functions.

Input: The inputs to this routine consist of the following:

1. The input problem file name.
2. The output TRFN file name.
3. The n_z diagonal elements of I_M. These are all input on one line, separated by commas. For the closed loop system, all have a value of 1.0. If opened loop transfer functions are desired, the elements of the I_M diagonal corresponding to the opened loops should have a value of 0.0.
4. The 60 character title for the TRFN file.

All inputs are prompted.

Output: The output from this block is a TRFN-compatible file — for a complete description, see Ref. 2. The general form of the file is:

$$A_{TF}x_{TF} = B_{TF}u_{TF}$$

where

$$x_{TF}(n_{xC}+m_1'+n_{yT}) = \begin{Bmatrix} x_C(n_{xC}) \\ \hat{y}(m_1') \\ y(n_{yT}) \end{Bmatrix}$$

$$u_{TF}(n_{yS}+n_z+n_{yS}+n_{yS}) = \begin{Bmatrix} t_S(n_{yS}) \\ t_V(n_z) \\ t_{OL}(n_{yS}) \\ w(n_{yS}) \end{Bmatrix}$$

$$A_{TF}(n_{xC} + m_1' + n_{yT},\ n_{xC} + m_1' + n_{yT})$$

$$= \begin{bmatrix} sI - (F_C + G_C D_F I_M H_M) & -G_C C_F & 0 \\ -B_F I_M H_M & sI - A_F & 0 \\ H_{TF} & 0 & I \end{bmatrix}$$

$$H_{TF} = \begin{bmatrix} H_M \\ H_O \end{bmatrix}$$

$$B_{TF}(n_{xC} + m_1 + n_{yT},\ n_{yS} + n_z + n_{yS} + n_{yS})$$

$$= \begin{bmatrix} C_C + G_C D_F I_M F_M & G_C D_F & G_C D_F F_M & \Gamma_C \\ B_F I_M F_M & B_F & B_F F_M & 0 \\ E_1 & 0 & 0 & 0 \end{bmatrix}$$

Restrictions: RT-11 operating system will halt with a fatal error if either file name is specified incorrectly, or if there is not enough contiguous disk space available for the TRFN file output.

Example: Figure 21 is the dialog used with this block; Fig. 22 lists the resulting TRFN file generated. This output file format is described in Fig. 7.

```
      CLOSED LOOP SYSTEM DEFINITION — BLOCK 9
INPUT PROBLEM FILE NAME ? DX0:2BY2.B7
OUTPUT TRFN FILE NAME ? DX0:2BY2.B9
TRFN TITLE (60 CHARS):
2BY2 EXAMPLE CLOSED LOOP SYSTEM TRANSFER FUNCTIONS
IM DIAGONAL ELEMENTS (NZ-OF-THEM):
1.,1.
```

Figure 21. Input for Block 9

```
     1        1   0.0000000E+00  0.1000000E+01 -0.2601713E+02        0
     1        2   0.0000000E+00  0.0000000E+00 -0.1000000E+01        0
     2        1   0.0000000E+00  0.0000000E+00  0.7290252E+02        0
     2        2   0.0000000E+00  0.1000000E+01  0.3000000E+01        0
     3        1   0.0000000E+00  0.0000000E+00  0.2003338E+04        0
     3        2   0.0000000E+00  0.0000000E+00 -0.7933613E-01        0
     4        1   0.0000000E+00  0.0000000E+00 -0.1000000E+01        0
     5        2   0.0000000E+00  0.0000000E+00 -0.1000000E+01        0
     1        3   0.0000000E+00  0.0000000E+00 -0.8749831E+00        0
     2        3   0.0000000E+00  0.0000000E+00  0.2250000E+01        0
     3        3   0.0000000E+00  0.1000000E+01  0.6483090E+02        0
     4        4   0.0000000E+00  0.0000000E+00  0.1200000E+01        0
     5        5   0.0000000E+00  0.0000000E+00  0.1000000E+01        0
     0
     1        1   0.0000000E+00  0.0000000E+00  0.2601713E+02        0
     2        1   0.0000000E+00  0.0000000E+00 -0.7290252E+02        0
     3        1   0.0000000E+00  0.0000000E+00 -0.2003338E+04        0
     3        2   0.0000000E+00  0.0000000E+00  0.7933613E-01        0
     1        3   0.0000000E+00  0.0000000E+00  0.1000000E+01        0
     2        4   0.0000000E+00  0.0000000E+00  0.1000000E+01        0
     0
2BY2 EXAMPLE  CLOSED LOOP SYSTEM TRANSFER FUNCTIONS
     5        4        0
X01 X02 E01 Z01 Z02
V01 V02 W01 W02
```

Figure 22. Output File Listing for Block 9

SECTION XI

STOCHASTIC PERFORMANCE ANALYSIS — BLOCK 10

Purpose: This routine computes the rms and covariance matrices arising from process noise, measurement noise, and the total for both noise components. These matrices are computed for both the closed loop system state vector and a vector composed of the outputs and controls. For the state vector this is accomplished by solving the equation

$$A(COV) + (COV)A' + Q = 0$$

for the covariance, COV. In addition, the correlation matrix for output and controls is computed. Execution of this block must be preceded by execution of Block 7.

Input: The input to this block consists of the problem file name for the input, and the problem file name for the output. Both are prompted by the software.

Output: A steady state covariance routine is used to solve the equation

$$A(COV) + (COV)A' + Q = 0$$

for COV given A for a stable system. The A_C matrix used here is computed and output by Block 6:

$$A_C = \begin{bmatrix} F + G[D_F \mid 0]H & GC_F \\ [B_F \mid 0]H & A_F \end{bmatrix}$$

The Q_W matrix used to compute the process noise contribution to the covariance matrix is

$$Q_W = \begin{bmatrix} \Gamma \\ 0 \end{bmatrix} Q \begin{bmatrix} \Gamma' & 0 \end{bmatrix}$$

The Q_V matrix used to compute the measurement noise contribution to the covariance matrix is

$$Q_v = \begin{bmatrix} 0 \\ B_F \end{bmatrix} T_Z^{-1} R T_Z^{-T} [0 \quad B_F']$$

From these Q's and A_c, the covariance matrices COV_w and COV_r are computed. The total covariance is the sum:

$$COV_{TOT} = COV_w + COV_v$$

The covariance of the output and controls is obtained as follows:

$$COV_{yu} = \begin{bmatrix} H_R & 0 \\ D_F H & C_F \end{bmatrix} COV_{TOT} \begin{bmatrix} H_R' & H' D_F' \\ 0 & C_F' \end{bmatrix}$$

The rms vectors are computed from the covariance matrices. All rms vectors and covariance matrices are written to the output file.

Restrictions: RT-11 operating system will halt with a fatal error if either file name is not input according to convention, or if disk space is not available for the output file.

Example: Figure 23 presents the input dialog for this block; Fig. 24 contains some rms vectors which are computed for the example problem.

```
STOCHASTIC PERFORMANCE ANALYSIS — BLOCK 10

  INPUT PROBLEM FILE NAME ? DX0:2BY2.B7

  OUTPUT PROBLEM FILE NAME ? DX0:2BY2.B10
```

Figure 23. Input to Block 10

RMS, PROCESS NOISE
1

	1
0.497	X06
	2
1.10	X07
	3
15.3	E07

RMS, MEASUREMENT NOISE
1

	1
8.72	X06
	2
27.6	X07
	3
279.	E07

RMS, TOTAL
1

	1
8.74	X06
	2
27.7	X07
	3
280.	E07

RMS, OUTPUT+CONTROLS
1

	1
8.74	Z01
	2
27.7	Z02
	3
138.	U01
	4
364.	U02

Figure 24. Output for Block 10

SECTION XII

SERVICE ROUTINE

Purpose: This routine allows the user to access the output problem files and format the data in a report-ready form.

Input: The user inputs the name of the output problem file to be accessed, and then the numbers identifying the elements in that file which are to be printed. Table 1 lists the elements available for printout. Notice that the complete list is only applicable when all blocks have been run. Otherwise, only an appropriate subset of the elements can be located in a given problem file. An index of zero terminates the file element requests.

Output: The output of the service routine is a formatted listing of the desired elements of the problem file.

Restrictions: RT-11 operating system will halt with a fatal error if the problem file name is incorrectly specified.

Example: Figure 25 is the dialog used to obtain the file element shown in Fig. 24.

```
PROBLEM FILE NAME ? DXC:2BY2.B10
INPUT INDEX OF EACH ELEMENT TO BE LISTED
ONE PER LINE -- LAST INDEX SHOULD BE 0
122
124
126
128
0
```

Figure 25. Service Routine Input

TABLE 1. INDEX TO PROBLEM FILE ELEMENTS

INDEX	BLOCK CREATED	ELEMENT	ROW DIM	COLUMN DIM
1	Block 1	Title	—	—
2		Input dimensions	—	—
3		u mnemonics	1	n_u
4		w mnemonics	1	n_w
5		y_S mnemonics	1	n_{yS}
6		v mnemonics	1	n_z
7		t_{OL} mnemonics	1	n_{yS}
8		y_M mnemonics	1	n_z
9		y_O mnemonics	1	n_{yO}
10		z mnemonics	1	n_z
11		x_S mnemonics	1	n_{xS}
12		x_C mnemonics	1	n_{xC}
13		Γ_S	n_{xS}	n_w
14		F_S	n_{xS}	n_{xS}
15		H_S	n_{yS}	n_{xS}
16		Γ_C	n_{xC}	n_w
17		G_C	n_{xC}	n_u
18		C_C	n_{xC}	n_{yS}
19		F_C	n_{xC}	n_{xC}
20		H_M	n_z	n_{xC}
21		F_M	n_z	n_{yS}
22		H_O	n_{yO}	n_{xC}
23		F_O	n_{yO}	n_{yS}
24		F	n_{xT}	n_{xT}
25		G	n_{xT}	n_u
26		Γ	n_{xT}	n_w
27		H	n_z	n_{xT}
28		H_R	n_{yT}	n_{xT}
29		E_1	n_{yT}	n_{yS}
30		E_2	n_z	n_{yS}

TABLE 1 (Continued)

INDEX	BLOCK CREATED	ELEMENT	ROW DIM	COLUMN DIM
31	Block 1	E_3	n_{xT}	n_{yS}
32	Block 3	v mnemonics (reordered)	1	n_z
33		z mnemonics (reordered)	1	n_z
34		x mnemonics (reordered)	1	n_{xT}
35		T_z	n_z	n_z
36		T	n_{xT}	n_{xT}
37		T^{-1}	n_{xT}	n_{xT}
38		F_{11}	m_1'	m_1'
39		F_{12}	m_1'	m_2
40		F_{21}	m_2	m_1'
41		F_{22}	m_2	m_2
42		G_1	m_1'	n_u
43		G_2	m_2	n_u
44		Γ_1	m_1'	n_w
45		Γ_2	m_2	n_w
46		H^*T	n_z	n_{xT}
47		H_{11}	m_1	m_1'
48		H_{12}	m_1	m_2
49		H_{22}	m_2	m_2
50		Q	1	n_w
51		R	1	n_z
52		EL_F	$2m_1'$	$2m_1'$
53		A	m_2	m_2
54		filter dimensions: m_1, m_1', m_2	—	—
55	Block 4	λ_{OL}	1	m_1'
56		λ_{CL}	1	m_1'
57		$W_{21}(\lambda)_{CL}$ (eigenvector partition for λ_{CL})	m_1'	m_1'
58		P	m_1'	m_1'
59		K_{11}	m_1'	m_1
60		K_{12}	m_1'	m_2

* σ ≡ rms vector.

TABLE 1 (Continued)

INDEX	BLOCK CREATED	ELEMENT	ROW DIM	COLUMN DIM
61	Block 4	σ_F^*	1	m_1'
62	↓	$\rho_F^\dagger$	m_1'	m_1'
63	Block 5	Q_R	1	n_{yT}
64		R_R	1	n_u
65		EL_R	$2n_{xT}$	$2n_{xT}$
66		λ_{OL}	1	n_{xT}
67		λ_{CL}	1	n_{xT}
68		$W_{21}(\lambda)_{CL}$ (eigenvector partition for λ_{CL})	n_{xT}	n_{xT}
69		S	n_{xT}	n_{xT}
70	↓	C	n_u	n_{xT}
71	Block 6	$\partial\lambda_1/\partial F$	n_{xT}	n_{xT}
72		$\partial\lambda_1/\partial G$	n_{xT}	n_u
73		$\partial\lambda_1/\partial C$	n_u	n_{xT}
74		$\partial\lambda_2/\partial F$	n_{xT}	n_{xT}
75		$\partial\lambda_2/\partial G$	n_{xT}	n_u
76		$\partial\lambda_2/\partial C$	n_u	n_{xT}
77		$\partial\lambda_3/\partial F$	n_{xT}	n_{xT}
78		$\partial\lambda_3/\partial G$	n_{xT}	n_u
79		$\partial\lambda_3/\partial C$	n_u	n_{xT}
80		$\partial\lambda_4/\partial F$	n_{xT}	n_{xT}
81		$\partial\lambda_4/\partial G$	n_{xT}	n_u
82		$\partial\lambda_4/\partial C$	n_u	n_{xT}
83		$\partial\lambda_5/\partial F$	n_{xT}	n_{xT}
84		$\partial\lambda_5/\partial G$	n_{xT}	n_u
85		$\partial\lambda_5/\partial C$	n_u	n_{xT}
86		$\partial\lambda_6/\partial F$	n_{xT}	n_{xT}
87		$\partial\lambda_6/\partial G$	n_{xT}	n_u
88	↓	$\partial\lambda_6/\partial C$	n_u	n_{xT}

* σ ≡ rms vector.
† ρ ≡ correlation matrix.

TABLE 1 (Continued)

INDEX	BLOCK CREATED	ELEMENT	ROW DIM	COLUMN DIM
89	Block 6	$\partial\lambda_7/\partial F$	n_{xT}	n_{xT}
90		$\partial\lambda_7/\partial G$	n_{xT}	n_u
91		$\partial\lambda_7/\partial C$	n_u	n_{xT}
92		$\partial\lambda_8/\partial F$	n_{xT}	n_{xT}
93		$\partial\lambda_8/\partial G$	n_{xT}	n_u
94		$\partial\lambda_8/\partial C$	n_u	n_{uT}
95		$\partial\lambda_9/\partial F$	n_{xT}	n_{xT}
96		$\partial\lambda_9/\partial G$	n_{xT}	n_u
97		$\partial\lambda_9/\partial C$	n_u	n_{xT}
98		$\partial\lambda_{10}/\partial F$	n_{xT}	n_{xT}
99		$\partial\lambda_{10}/\partial G$	n_{xT}	n_u
100		$\partial\lambda_{10}/\partial C$	n_u	n_{xT}
101		$\partial\lambda_{11}/\partial F$	n_{xT}	n_{xT}
102		$\partial\lambda_{11}/\partial G$	n_{xT}	n_u
103		$\partial\lambda_{11}/\partial C$	n_u	n_{xT}
104		$\partial\lambda_{12}/\partial F$	n_{xT}	n_{xT}
105		$\partial\lambda_{12}/\partial G$	n_{xT}	n_u
106		$\partial\lambda_{12}/\partial C$	n_u	n_{xT}
107		$\partial\lambda_{13}/\partial F$	n_{xT}	n_{xT}
108		$\partial\lambda_{13}/\partial G$	n_{xT}	n_u
109		$\partial\lambda_{13}/\partial C$	n_u	n_{xT}
110		$\partial\lambda_{14}/\partial F$	n_{xT}	n_{xT}
111		$\partial\lambda_{14}/\partial G$	n_{xT}	n_u
112		$\partial\lambda_{14}/\partial C$	n_u	n_{xT}
113		$\partial\lambda_{15}/\partial F$	n_{xT}	n_{xT}
114		$\partial\lambda_{15}/\partial G$	n_{xT}	n_u
115		$\partial\lambda_{15}/\partial C$	n_u	n_{xT}
116	Block 7	A_F	m_1'	m_1'
117		B_F	m_1'	n_z
118		C_F	n_u	m_1'
119		D_F	n_u	n_z

TABLE 1 (Concluded)

INDEX	BLOCK CREATED	ELEMENT	ROW DIM	COLUMN DIM
120	Block 7	A_c	$n_{xT}+m_1'$	$n_{xT} + m_1'$
121	Block 10	COV_w^*	$n_{xT}+m_1'$	$n_{xT} + m_1'$
122	↓	σ_w	1	$n_{xT} + m_1'$
123		COV_v^*	$n_{xT}+m_1'$	$n_{xT} + m_1'$
124		σ_v	1	$n_{xT} + m_1'$
125		C_{TOT}	$n_{xT}+m_1$	$n_{xT} + m_1$
126		σ_{TOT}	1	$n_{xT} + m_1$
127		C_{yu}	$n_{yT}+n_u$	$n_{yT} + n_u$
128		σ_{yu}	1	$n_{yT} + n_u$
129		ρ_{yu}	$n_{yT}+n_u$	$n_{yT} + n_u$

* COV ≡ covariance matrix.

SECTION XIII

CLASSICAL CONTROL PROCEDURES

For the sake of completeness, Figs. 26, 27, and 28 are included to present example closed loop transfer functions (computed using TRFN), frequency response plots (computed using USAM2), and transient response plots (computed using USAM2). Each of these STI-proprietary software packages have public domain counterparts.

```
GO (Y7LIB)ETRFN

                              1-Feb-79    17:24

  INPUT DATA FILE NAME:  2X2

  CASE:  2X2 EXAMPLE CASE CLOSED LOOP SYSTEM TRANSFER FUNCTIONS

DENOMINATOR:

   1.0000
 ( 3.0000     ) ( 4.0000     ) ( 34.014     )
 < 417.76     >


NUMERATOR:  X06/V01

   26.017
 ( 2.3806     ) (-4.6493     )
 <-287.96     >


NUMERATOR:  X07/V01

   -70.903
 (-.69486     ) ( 2.6863     )
 < 132.35     >


NUMERATOR:  E07/V01

   -2003.3
 (  .99926    ) ( 2.0036     )
 <-4010.8     >


NUMERATOR:  Z01/V01

   26.017
 ( 2.3806     ) (-4.6493     )
 <-287.96     >


NUMERATOR:  Z02/V01

   -70.903
 (-.69486     ) ( 2.6863     )
 < 132.35     >
```


---

*The following shorthand notation is used to represent transfer functions:

$$\frac{K(s+z)(s^2+2\zeta_1\omega_1 s+\omega_1^2)}{(s+p)(s^2+2\zeta_2\omega_2 s+\omega_2^2)} \equiv \frac{K(z)[\zeta_1;\ \omega_1]}{(p)[\zeta_2;\ \omega_2]}$$

Figure 26. Factored Transfer Functions

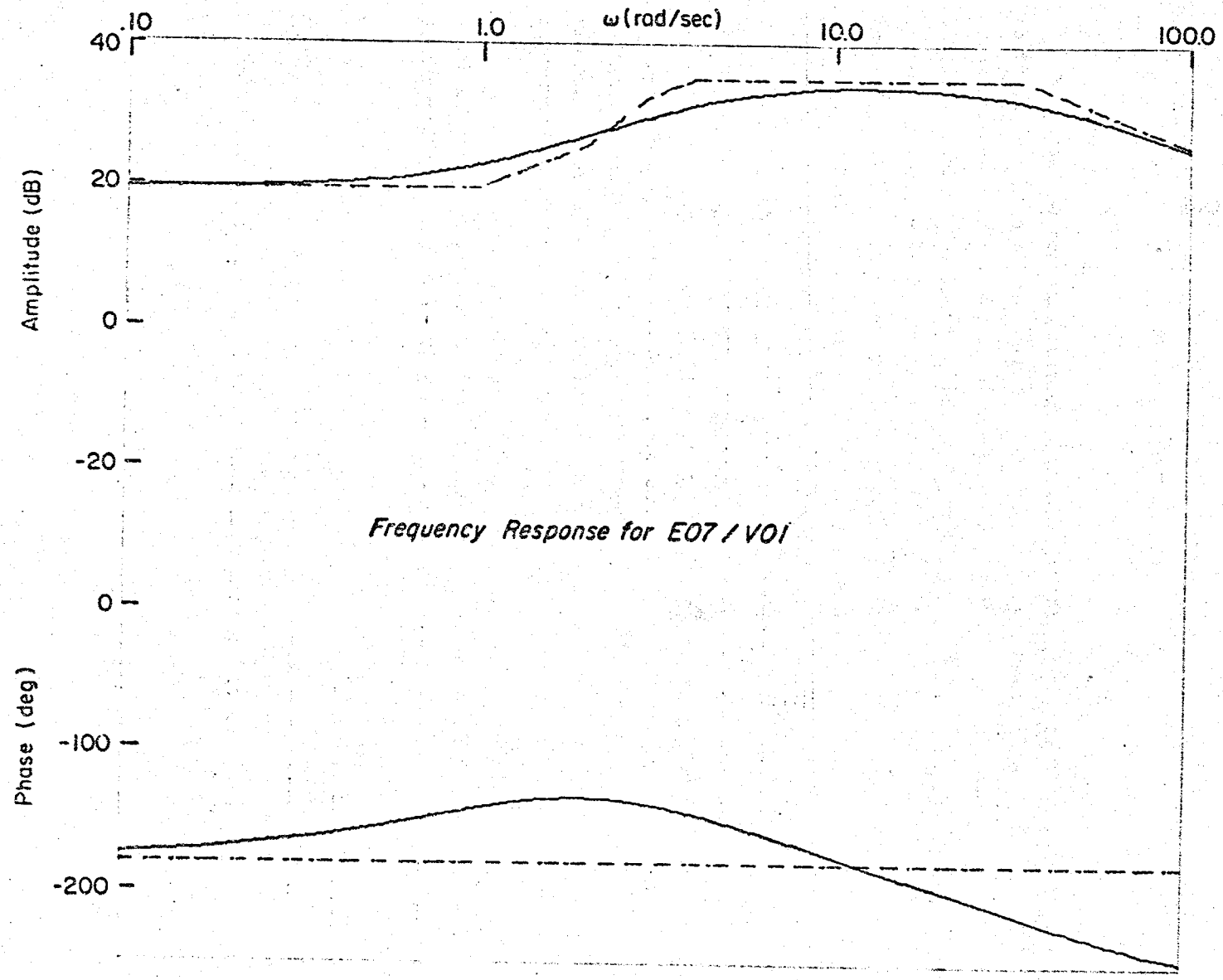

Figure 27. Frequency Response

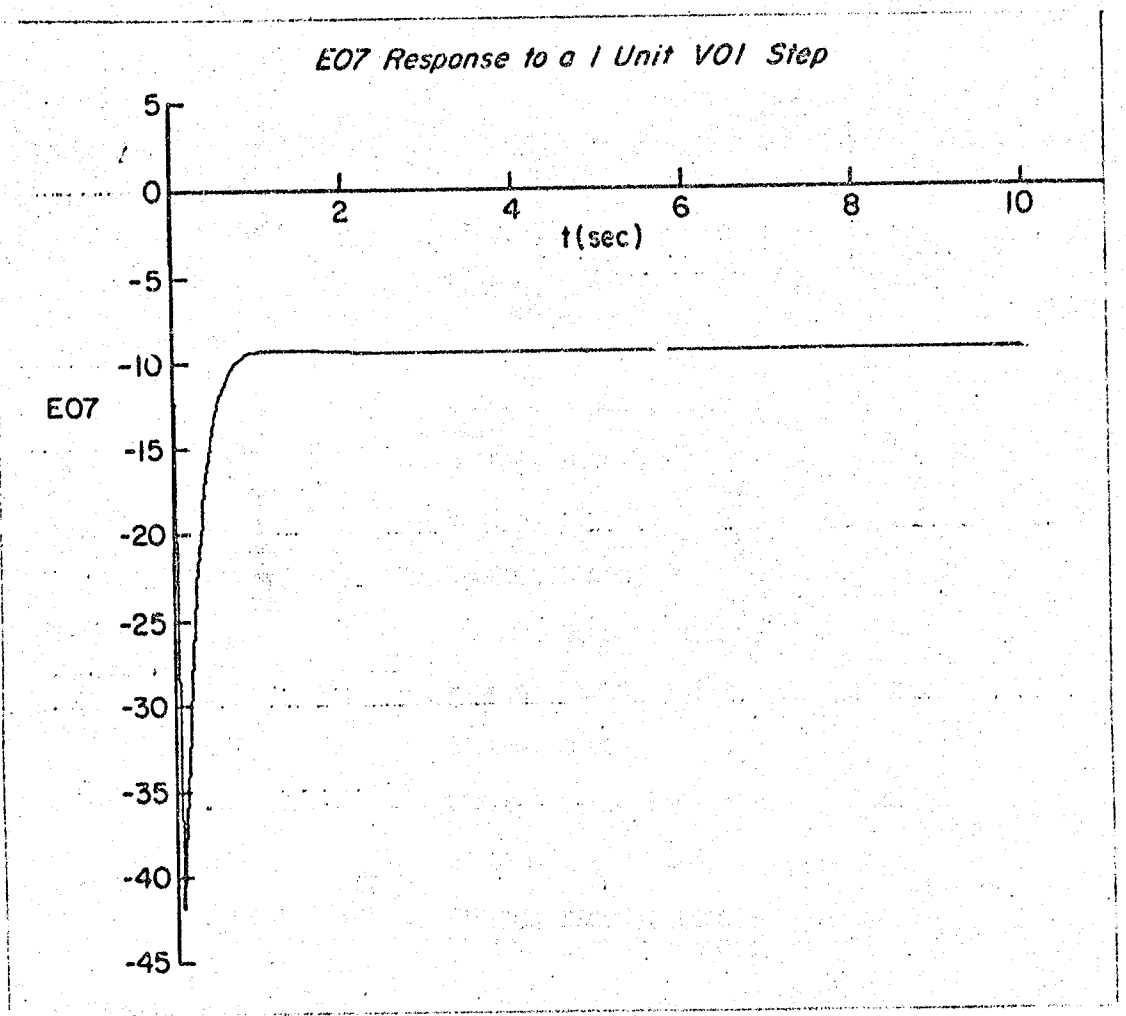

Figure 28. Transient Response

REFERENCES

1. Bryson, Arthur E., Jr., and W. Earl Hall, Jr., Optimal Control and Filter Synthesis by Eigenvector Composition, Stanford University SUDAAR No. 436, Nov. 1971.

2. Stapleford, R. L., Users' Manual for STI Control System Design Programs, Systems Technology, Inc., TR-407-1, July 1970 (Revised Aug. 1976).

3. Riedel, Susan A., and L. G. Hofmann, Practical Optimal Control for Flight Control Application. Vol. I: Technical Report, Systems Technology, Inc., TR-1127-1, forthcoming.

4. Hofmann, L. G., Transformations for Singular Filter Problems Enabling Eigenvector Decomposition, Systems Technology, Inc., WP-1127-1, Aug. 1978.

5. Wilkinson, J. H. and C. Reinsch, Linear Algebra, New York, Springer-Verlag, 1971.

6. Slater, G. L., "Approach Guidance for a STOL A/C," Univ. of Cincinnati, Dept. of Aerospace Engineering and Applied Mechanics, unpublished, circa 1976.

7. Smith, B. T., et al., Lecture Notes in Computer Science. Vol. 6: Matrix Eigensystem Routines - EISPACK Guide, New York, Springer-Verlag, 2nd ed., 1976.

8. Sandell, Nils R. and Michael Athans, Modern Control Theory. A Self-Study Subject Computer Manual, MIT, Center for Advanced Engineering Study, 1974.

9. Hofmann, L. G., Topics on Practical Application of Optimal Control to Single and Multiple Control-Point Flight Control Problems, AFFDL-TR-70-52, Feb. 1971.

10. Forsythe, George E., Michael A. Malcolm, and Cleve B. Moler, Computer Methods for Mathematical Computations, Englewood Cliffs, N.J., Prentice-Hall, Inc., 1977.

APPENDIX

PROGRAM LISTINGS

This appendix contains complete listings for all of the software used to implement the optimal control design package. The mainline routines which solve the filter and regulator problems were adapted from the Bach/Slater version of OPTSYS (Ref. 1). All of the programs which solve the eigenvalue problem were taken from EISPAK (Ref. 7). The generalized matrix inverse routine is from the CAES optimal control package (Ref. 8). The covariance equation routine was adapted from Kleinman (Ref. 9). Subroutines used to solve a set of complex linear equations were taken from Forsythe, et al. (Ref. 10).

Table 2 lists all the subroutines used, where they are used, and the source of the code. Table 3 lists all of the RT-11 specific features employed in this software, and a description of their functions. Appropriate replacements should be found if this software is to be run under another operating system. Table 4 lists the mnemonics used in the cross-reference listings.

The mainlines for each block follow. Listings for the two routines which comprise the service routine software are given next. Finally, a listing of each subroutine is given.

TABLE 2. SUBROUTINE LIBRARY

SUBROUTINE	BLOCK USED	REF
BALANC	BLOCO4, BLOCO5, BLOC10	7
BALBAK	BLOCO4, BLOCO5	7
DECOMP	BLOCO4, BLOCO5, BLOCO6	10
SOLVE	BLOCO4, BLOCO5, BLOCO6	10
EIGRF	BLOCO4, BLOCO5	—
ELMHES	BLOCO4, BLOCO5	7
ELTRAN	BLOCO4, BLOCO5	7
GMINV	BLOCO3, BLOCO7, BLOC10	8
HQR	BLOCO4, BLOCO5	7
HQR2	BLOCO4, BLOCO5	7
LEQT2C	BLOCO6	—
LINEQ1	BLOC10	9
REWR	BLOCO3, BLOCO4, BLOCO5 BLOCO6, BLOCO7, BLOC10	—

TABLE 3. RT-11 SPECIFIC FEATURES

CALL ASSIGN (lun, fname)	— assigns the file name fname to logical unit number lun, and opens file on first READ
CALL CLOSE (lun)	— writes an end-of-file to the file name assigned to lun, and frees lun for further assignment
CALL DATE (array)	— returns correct date as nine ASC11 characters
CALL TIME (array)	— returns correct system time as eight ASC11 characters
CALL TRANSL (in,out,r,p)	— replaces character string in with character string out after modifying all occurrences of substring r by substring p

TABLE 4. MNEMONICS USED IN CROSS-REFERENCE LISTINGS

	USED IN ARITHMETIC LINE
=	VARIABLE ASSIGNED VALUE
'	LABEL LOCATION DEFINED
AC	ACCEPT
AG	ARGUMENT IN FUNCTION, SUBROUTINE OR CALL
AS	ASSIGN LABEL NUMBER TO VARIABLE
BD	BLOCK DATA
BS	BACKSPACE
BY	BYTE
CE	CLOSE
CH	CHARACTER
CL	CALL
CM	COMMON VARIABLE
CN	COMMON NAME
CX	COMPLEX
DA	DATA
DE	DECODE
DF	DEFINE FILE
DI	DIMENSION
DP	DOUBLE PRECISION
DO	DO
EF	ENDFILE
EN	ENCODE
EQ	EQUIVALENCE
EX	EXTERNAL
EY	ENTRY
FB	BYTE FUNCTION
FC	COMPLEX FUNCTION
FD	DOUBLE PRECISION FUNCTION
FH	CHARACTER FUNCTION
FI	INTEGER FUNCTION
FL	LOGICAL FUNCTION
FN	FIND
FO	FORMAT
FR	REAL FUNCTION
FU	FUNCTION
GT	GO TO
IF	IF
IN	INTEGER
LG	LOGICAL
OP	OPEN
PA	PARAMETER
PG	PROGRAM
PR	PRINT
RD	READ
RC	READ
RL	REAL
RW	REWIND
TY	TYPE
VI	VIRTUAL
WR	WRITE

```
      C       BLOCK 1 - THIS MAINLINE ACCEPTS INPUT OF THE DIMENSIONS,
      C                 MNEMONICS AND NON-ZERO ARRAY ELEMENTS FOR THE
      C                 "FIXED" PORTION OF THE CONTROL PROBLEM.  IT
      C                 ASSEMBLES MATRICES NECESSARY FOR USE IN THE
      C                 FILTER AND REGULATOR BLOCKS DOWNSTREAM.
00001         DIMENSION TITLE(5),FNAME(4)
      C         MNEMONIC ARRAYS
00002         REAL MNU(5),MNV(15),MNYS(5),MNV(15),MNTOL(5),
     +        MNYM(15),MNYO(15),MNXS(5),MNXC(10),MNZ(15)
      C         INPUT MATRIX ARRAYS
00003         DIMENSION GMS(5,15),FS(5,5),HS(5,5),GMC(10,15),GC(10,5),
     +             CC(10,5),FC(10,10),HM(15,10),FM(15,5),HO(15,15),
     +             FO(15,5)
      C         OUTPUT MATRIX ARRAYS
00004         DIMENSION F(15,15),G(15,5),GM(15,15),H(15,15),HR(30,15),
     +             E1(30,5),E2(15,5),E3(15,5)
      C         INITIALIZE ARRAYS
00005         COMPLEX C0
00006         DATA GMS,FS,HS,GMC,GC,CC,FC,HM,FM,HO,FO/1050*0.0/
00007         DATA F,G,GM,H,HR,E1,E2,E3/1500*0.0/
      C         CONSTRUCT MNEMONICS FOR XS,XC AND Z
00008         DATA MNXS /'X01','X02','X03','X04','X05'/
00009         DATA MNXC /'X06','X07','X08','X09','X10',
     +              'X11','X12','X13','X14','X15'/
      C         OTHER DATA
00010         DATA IR1,R0,C0 /1,0.,(0.,0.)/
00011         DATA IIN,IOUT,IFIN,IFOUT,NDIM1,NDIM2,NDIM3/5,7,20,21,5,10,15/
00012         COMMON/IO/ IIN,IOUT,IFIN,IFOUT
      C         GET FILENAME FOR INPUT DATA
00013         WRITE(IOUT,10)
00014 10      FORMAT(10X,'PROBLEM DEFINITION - BLOCK 1',/,1X,
     +            'INPUT DATA FILE NAME ? ',$)
00015         READ(IIN,20) (FNAME(I),I=1,4)
00016 20      FORMAT(4A4)
00017         CALL ASSIGN(IFIN,FNAME)
      C         READ PROBLEM TITLE
00018         READ(IFIN,30) (TITLE(I),I=1,5)
00019 30      FORMAT(5A4)
      C         READ DIMENSION AND CHECK
00020         READ(IFIN,40) NXS,NXC,NU,NV,NZ,NYS,NYO
00021 40      FORMAT(7I)
00022         IF(NXS.LE.5.AND.NXC.LE.10.AND.NU.LE.5.AND.NV.LE.15
     +     .AND.NZ.LE.15.AND.NYS.LE.5.AND.NYO.LE.15
     +     .AND.NZ.NE.0)
     +      GO TO 50
00024         WRITE(IOUT,45)
00025 45      FORMAT(5X,'DIMENSIONS TOO LARGE OR Z IS 0')
00026         GO TO 1000
      C         READ MNEMONICS
00027 50      CONTINUE
00028         IF(NU.NE.0) READ(IFIN,60) (MNU(I),I=1,NU)
00030 60      FORMAT(15(A3,1X))
00031         IF(NV.NE.0) READ(IFIN,60) (MNV(I),I=1,NV)
00033         IF(NYS.NE.0) READ(IFIN,60) (MNYS(I),I=1,NYS)
```

```
00035          READ(IFIN,60) (MNV(I),I=1,NZ)
00036          IF(NYS.NE.0) READ(IFIN,60) (MNTOL(I),I=1,NYS)
00038          READ(IFIN,60) (MNYM(I),I=1,NZ)
00039          IF(NYO.NE.0) READ(IFIN,60) (MNYO(I),I=1,NYO)
       C         READ THE NON-ZERO ELEMENTS OF THE ARRAYS
00041 70       CALL READMX(NXS,NN,NDIM1,GMS,IERR)
00042          IF(IERR.NE.1) GO TO 70
00044 80       CALL READMX(NXS,NXS,NDIM1,FS,IERR)
00045          IF(IERR.NE.1) GO TO 80
00047 90       CALL READMX(NYS,NXS,NDIM1,HS,IERR)
00048          IF(IERR.NE.1) GO TO 90
00050 100      CALL READMX(NXC,NN,NDIM2,GMC,IERR)
00051          IF(IERR.NE.1) GO TO 100
00053 110      CALL READMX(NXC,NU,NDIM2,GC,IERR)
00054          IF(IERR.NE.1) GO TO 110
00056 120      CALL READMX(NXC,NYS,NDIM2,CC,IERR)
00057          IF(IERR.NE.1) GO TO 120
00059 130      CALL READMX(NXC,NXC,NDIM2,FC,IERR)
00060          IF(IERR.NE.1) GO TO 130
00062 140      CALL READMX(NZ,NXC,NDIM3,HM,IERR)
00063          IF(IERR.NE.1) GO TO 140
00065 150      CALL READMX(NZ,NYS,NDIM3,FM,IERR)
00066          IF(IERR.NE.1) GO TO 150
00068 160      CALL READMX(NYO,NXC,NDIM3,HO,IERR)
00069          IF(IERR.NE.1) GO TO 160
00071 170      CALL READMX(NYO,NYS,NDIM3,FO,IERR)
00072          IF(IERR.NE.1) GO TO 170
       C         READ FILENAME FOR OUTPUT FILE
00074          READ(IFIN,20) (FNAME(I),I=1,4)
00075          CALL ASSIGN(IFOUT,FNAME)
```

```
      C                    ( FS      0)
      C        FORM F = (             )
      C                    (CC*HS   FC)
00076         IF(NXS.EQ.0) GO TO 210
00078         DO 180 I=1,NXS
00079         DO 180 J=1,NXS
00080 180     F(I,J)=FS(I,J)
00081         IF(NXC.EQ.0.OR.NYS.EQ.0) GO TO 210
00083         DO 200 I=1,NXC
00084         L=I+NXS
00085         DO 200 J=1,NXS
00086         DUM=0.
00087         DO 190 K=1,NYS
00088 190     DUM=DUM+CC(I,K)*HS(K,J)
00089 200     F(L,J)=DUM
00090 210     CONTINUE
00091         IF(NXC.EQ.0) GO TO 230
00093         DO 220 I=1,NXC
00094         II=I+NXS
00095         DO 220 J=1,NXC
00096         JJ=J+NXS
00097 220     F(II,JJ)=FC(I,J)
00098 230     CONTINUE
      C                    ( 0 )
      C        FORM G = (    )
      C                    (GC )
00099         IF(NU.EQ.0) GO TO 250
00101         DO 240 I=1,NXC
00102         II=I+NXS
00103         DO 240 J=1,NU
00104 240     G(II,J)=GC(I,J)
00105 250     CONTINUE
      C                    (GMS)
      C        FORM GM = (    )
      C                    (GMC)
00105         IF(NW.EQ.0) GO TO 290
00108         IF(NXS.EQ.0) GO TO 270
00110         DO 260 I=1,NXS
00111         DO 260 J=1,NW
00112 260     GM(I,J)=GMS(I,J)
00113 270     CONTINUE
00114         IF(NXC.EQ.0) GO TO 290
00115         DO 280 I=1,NXC
00117         II=I+NXS
00118         DO 280 J=1,NW
00119 280     GM(II,J)=GMC(I,J)
00120 290     CONTINUE
      C         FORM H = (FM*HS    H4)
00121         IF(NYS.EQ.0.OR.NXS.EQ.0) GO TO 320
00123         DO 310 I=1,NZ
00124         DO 310 J=1,NXS
00125         DUM=0.
00126         DO 300 K=1,NYS
00127 300     DUM=DUM+FM(I,K)*HS(K,J)
```

```
00128 310    H(I,J)=DUM
00129 320    CONTINUE
00130        IF(NXC.EQ.0) GO TO 340
00132        DO 330 I=1,NZ
00133        DO 330 J=1,NXC
00134        JJ=J+NXS
00135 330    H(I,JJ)=HM(I,J)
00136 340    CONTINUE
             C                  (FM*HS   HM)
             C       FORM HR = (          )
             C                  (FO*HS   HO)
00137        IF(NYS.EQ.0.OR.NXS.EQ.0) GO TO 390
00139        DO 360 I=1,NZ
00140        DO 360 J=1,NXS
00141        DUM=0.
00142        DO 350 K=1,NYS
00143 350    DUM=DUM+FM(I,K)*HS(K,J)
00144 360    HR(I,J)=DUM
00145        IF(NYO.EQ.0) GO TO 390
00147        DO 380 I=1,NYO
00148        II=I+NZ
00149        DO 380 J=1,NXS
00150        DUM=0.
00151        DO 370 K=1,NYS
00152 370    DUM=DUM+FO(I,K)*HS(K,J)
00153 380    HR(II,J)=DUM
00154 390    CONTINUE
00155        IF(NXC.EQ.0) GO TO 420
00157        DO 400 I=1,NZ
00158        DO 400 J=1,NXC
00159        JJ=J+NXS
00160 400    HR(I,JJ)=HM(I,J)
00161        IF(NYO.EQ.0) GO TO 420
00163        DO 410 I=1,NYO
00164        II=I+NZ
00165        DO 410 J=1,NXC
00166        JJ=J+NXS
00167 410    HR(II,JJ)=HO(I,J)
00168 420    CONTINUE
             C                  (FM)
             C       FORM E1 = (  )
             C                  (FO)
00169        IF(NYS.EQ.0) GO TO 450
00171        DO 430 I=1,NZ
00172        DO 430 J=1,NYS
00173 430    E1(I,J)=FM(I,J)
00174        IF(NYO.EQ.0) GO TO 450
00176        DO 440 I=1,NYO
00177        II=I+NZ
00178        DO 440 J=1,NYS
00179 440    E1(II,J)=FO(I,J)
00180 450    CONTINUE
             C       FORM E2 = FM
00181        IF(NYS.EQ.0) GO TO 470
```

```
00183         DO 450 I=1,NZ
00184         DO 450 J=1,NYS
00185 460     E2(I,J)=FM(I,J)
00186 470     CONTINUE
      C                    ( 0 )
      C       FORM E3 =    (   )
      C                    (CC )
00187         IF(NXS.EQ.0.OR.NYS.EQ.0) GO TO 490
00189         DO 480 I=1,NXS
00190         II=I+NXC
00191         DO 480 J=1,NYS
00192 480     E3(II,J)=CC(I,J)
00193 490     CONTINUE
      C         OUTPUT AREA
00194         IDEN=1
00195         WRITE(IFOUT) IDEN,NDIM1,(TITLE(I),I=1,5)
00196         IDEN=2
00197         WRITE(IFOUT) IDEN,NXS,NXC,NU,NN,NZ,NYS,NYO
00198         IDEN=3
00199         IF(NU.EQ.0) WRITE(IFOUT) IDEN,IR1,R0
00201         IF(NU.NE.0) WRITE(IFOUT) IDEN,NU,(MNU(I),I=1,NU)
00203         IDEN=4
00204         IF(NN.EQ.0) WRITE(IFOUT) IDEN,IR1,R0
00206         IF(NN.NE.0) WRITE(IFOUT) IDEN,NN,(MNN(I),I=1,NN)
00208         IDEN=5
00209         IF(NYS.EQ.0) WRITE(IFOUT) IDEN,IR1,R0
00211         IF(NYS.NE.0) WRITE(IFOUT) IDEN,NYS,(MNYS(I),I=1,NYS)
00213         IDEN=6
00214         WRITE(IFOUT) IDEN,NZ,(MNV(I),I=1,NZ)
00215         IDEN=7
00216         IF(NYS.EQ.0) WRITE(IFOUT) IDEN,IR1,R0
00218         IF(NYS.NE.0) WRITE(IFOUT) IDEN,NYS,(MNTOL(I),I=1,NYS)
00220         IDEN=8
00221         WRITE(IFOUT) IDEN,NZ,(MNYM(I),I=1,NZ)
00222         IDEN=9
00223         IF(NYO.EQ.0) WRITE(IFOUT) IDEN,IR1,R0
00225         IF(NYO.NE.0) WRITE(IFOUT) IDEN,NYO,(MNYO(I),I=1,NYO)
00227         IDEN=10
00228         WRITE(IFOUT) IDEN,NZ,(MNYM(I),I=1,NZ)
00229         IDEN=11
00230         IF(NXS.EQ.0) WRITE(IFOUT) IDEN,IR1,R0
00232         IF(NXS.NE.0) WRITE(IFOUT) IDEN,NXS,(MNXS(I),I=1,NXS)
00234         IDEN=12
00235         IF(NXC.EQ.0) WRITE(IFOUT) IDEN,IR1,R0
00237         IF(NXC.NE.0) WRITE(IFOUT) IDEN,NXC,(MNXC(I),I=1,NXC)
00239         IDEN=13
00240         IF(NXS.EQ.0.OR.NN.EQ.0) WRITE(IFOUT) IDEN,IR1,IR1,R0
00242         IF(NXS.NE.0.AND.NN.NE.0)
             +      WRITE(IFOUT) IDEN,NXS,NN,((GMS(I,J),J=1,NN),I=1,NXS)
00244         IDEN=14
00245         IF(NXS.EQ.0) WRITE(IFOUT) IDEN,IR1,IR1,R0
00247         IF(NXS.NE.0)
             + WRITE(IFOUT) IDEN,NXS,NXS,((FS(I,J),J=1,NXS),I=1,NXS)
00249         IDEN=15
```

```
00250       IF(NYS.EQ.0.OR.NXS.EQ.0) WRITE(IFOUT) IDEN,IR1,IR1,R0
00252       IF(NYS.NE.0.AND.NXS.NE.0)
           +       WRITE(IFOUT) IDEN,NYS,NXS,((HS(I,J),J=1,NXS),I=1,NYS)
00254       IDEN=16
00255       IF(NXC.EQ.0.OR.NN.EQ.0) WRITE(IFOUT) IDEN,IR1,IR1,R0
00257       IF(NXC.NE.0.AND.NN.NE.0)
           +       WRITE(IFOUT) IDEN,NXC,NN,((GMC(I,J),J=1,NN),I=1,NXC)
00259       IDEN=17
00260       IF(NXC.EQ.0.OR.NU.EQ.0) WRITE(IFOUT) IDEN,IR1,IR1,R0
00262       IF(NXC.NE.0.AND.NU.NE.0)
           +       WRITE(IFOUT) IDEN,NXC,NU,((GC(I,J),J=1,NU),I=1,NXC)
00264       IDEN=18
00265       IF(NXC.EQ.0.OR.NYS.EQ.0) WRITE(IFOUT) IDEN,IR1,IR1,R0
00267       IF(NXC.NE.0.AND.NYS.NE.0)
           +       WRITE(IFOUT) IDEN,NXC,NYS,((CC(I,J),J=1,NYS),I=1,NXC)
00269       IDEN=19
00270       IF(NXC.EQ.0) WRITE(IFOUT) IDEN,IR1,IR1,R0
00272       IF(NXC.NE.0)
           + WRITE(IFOUT) IDEN,NXC,NXC,((FC(I,J),J=1,NXC),I=1,NXC)
00274       IDEN=20
00275       IF(NXC.EQ.0) WRITE(IFOUT) IDEN,IR1,IR1,R0
00277       IF(NXC.NE.0)
           + WRITE(IFOUT) IDEN,NZ,NXC,((HM(I,J),J=1,NXC),I=1,NZ)
00279       IDEN=21
00280       IF(NYS.EQ.0) WRITE(IFOUT) IDEN,IR1,IR1,R0
00282       IF(NYS.NE.0)
           + WRITE(IFOUT) IDEN,NZ,NYS,((EM(I,J),J=1,NYS),I=1,NZ)
00284       IDEN=22
00285       IF(NYO.EQ.0.OR.NXC.EQ.0) WRITE(IFOUT) IDEN,IR1,IR1,R0
00287       IF(NYO.NE.0.AND.NXC.NE.0)
           +   WRITE(IFOUT) IDEN,NYO,NXC,((HO(I,J),J=1,NXC),I=1,NYO)
00289       IDEN=23
00290       IF(NYO.EQ.0.OR.NYS.EQ.0) WRITE(IFOUT) IDEN,IR1,IR1,R0
00292       IF(NYO.NE.0.AND.NYS.NE.0)
           +  WRITE(IFOUT) IDEN,NYO,NYS,((FO(I,J),J=1,NYS),I=1,NYO)
00294       IDEN=24
00295       NXT=NXC+NXS
00296       WRITE(IFOUT) IDEN,NXT,NXT,((F(I,J),J=1,NXT),I=1,NXT)
00297       IDEN=25
00298       IF(NU.EQ.0) WRITE(IFOUT) IDEN,IR1,IR1,R0
00300       IF(NU.NE.0)
           + WRITE(IFOUT) IDEN,NXT,NU,((G(I,J),J=1,NU),I=1,NXT)
00302       IDEN=26
00303       IF(NN.EQ.0) WRITE(IFOUT) IDEN,IR1,IR1,R0
00305       IF(NN.NE.0)
           + WRITE(IFOUT) IDEN,NXT,NN,((GM(I,J),J=1,NN),I=1,NXT)
00307       IDEN=27
00308       WRITE(IFOUT) IDEN,NZ,NXT,((H(I,J),J=1,NXT),I=1,NZ)
00309       IDEN=28
00310       NZT=NZ+NYO
00311       WRITE(IFOUT) IDEN,NZT,NXT,((HR(I,J),J=1,NXT),I=1,NZ+NYO)
00312       IDEN=29
00313       IF(NYS.EQ.0) WRITE(IFOUT) IDEN,IR1,IR1,R0
00315       IF(NYS.NE.0)
```

```
                +WRITE(IFOUT) IDEN,NZT,NYS,((E1(I,J),J=1,NYS),I=1,NZT)
00317           IDEN=30
00318           IF(NYS.EQ.0) WRITE(IFOUT) IDEN,IR1,IR1,R0
00320           IF(NYS.NE.0)
                +WRITE(IFOUT) IDEN,NZ,NYS,((E2(I,J),J=1,NYS),I=1,NZ)
00322           IDEN=31
00323           IF(NYS.EQ.0) WRITE(IFOUT) IDEN,IR1,IR1,R0
00325           IF(NYS.NE.0)
                +WRITE(IFOUT) IDEN,NXT,NYS,((E3(I,J),J=1,NYS),I=1,NXT)
00327           DO 500 I=32,130
00328 500       WRITE(IFOUT) I,IR1,IR1,C0
        C         THE END
00329 1000      STOP
00330           END
```

```
ASSIGN  00017CL 00075CL
CC      00003DI 00006DA 00256AG 00088   00192   00268WR
C0      00005CX 00010DA 00328WR
DUM     00085=  00088=  00089   00125=  00127=  00128   00141=  00143=
        00144   00150=  00152=  00153
E1      00004DI 00007DA 00173=  00179=  00316WR
E2      00004DI 00007DA 00185=  00321WR
E3      00004DI 00007DA 00192=  00326WR
F       00004DI 00007DA 00080=  00089=  00097=  00295WR
FC      00003DI 00005DA 00059AG 00097   00273WR
FM      00003DI 00006DA 00065AG 00127   00143   00173   00185   00283WR
FNAME   00001DI 00015RD 00017AG 00074RD 00075AG
FO      00003DI 00006DA 00071AG 00152   00179   00293WR
FS      00003DI 00006DA 00044AG 00080   00248WR
G       00004DI 00007DA 00104=  00301WR
GC      00003DI 00006DA 00053AG 00104   00263WR
GM      00004DI 00007DA 00112=  00119=  00305WR
GMC     00003DI 00006DA 00050AG 00119   00258WR
GMS     00003DI 00006DA 00041AG 00112   00243WR
H       00004DI 00007DA 00128=  00135=  00308WR
HM      00003DI 00006DA 00062AG 00135   00160   00278WR
HO      00003DI 00006DA 00068AG 00167   00288WR
HR      00004DI 00007DA 00144=  00153=  00160=  00167=  00311WR
HS      00003DI 00006DA 00047AG 00088   00127   00143   00152   00253WR
I       00015RD 00018RD 00029RD 00032RD 00034RD 00035RD 00037RD 00039RD
        00040RD 00074RD 00078DO 00080   00083DO 00084   00088   00093DO
        00094   00097   00101DO 00102   00104   00110DO 00112   00116DO
        00117   00119   00123DO 00127   00128   00132DO 00135   00139DO
        00143   00144   00147DO 00148   00152   00157DO 00160   00163DO
        00164   00167   00171DO 00173   00176DO 00177   00179   00183DO
        00185   00189DO 00190   00192   00195WR 00202WR 00207WR 00212WR
        00214WR 00219WR 00221WR 00226WR 00228WR 00233WR 00236WR 00243WR
        00246WR 00253WR 00258WR 00263WR 00268WR 00273WR 00278WR 00283WR
        00288WR 00293WR 00298WR 00301WR 00306WR 00308WR 00311WR 00316WR
        00321WR 00326WR 00327DO 00328WR
IDEN    00194=  00195WR 00196=  00197WR 00198=  00200WR 00202WR 00203=
        00205WR 00207WR 00208=  00210WR 00212WR 00213=  00214WR 00215=
        00217WR 00219WR 00220=  00221WR 00222=  00226WR 00226WR 00227=
        00228WR 00229=  00231WR 00233WR 00234=  00236WR 00238WR 00239=
        00241WR 00243WR 00244=  00246WR 00248WR 00249=  00251WR 00253WR
        00254=  00256WR 00258WR 00259=  00261WR 00263WR 00264=  00266WR
        00268WR 00269=  00271WR 00273WR 00274=  00276WR 00278WR 00279=
        00281WR 00283WR 00284=  00286WR 00288WR 00289=  00291WR 00293WR
        00294=  00296WR 00297=  00299WR 00301WR 00302=  00304WR 00306WR
        00307=  00308WR 00309=  00311WR 00312=  00314WR 00316WR 00317=
        00319WR 00321WR 00322=  00324WR 00326WR
IERR    00041AG 00042IF 00044AG 00045IF 00047AG 00048IF 00050AG 00051IF
        00053AG 00054IF 00056AG 00057IF 00059AG 00062IF 00062AG 00063IF
        00065AG 00066IF 00068AG 00069IF 00071AG 00072IF
IFIN    00011DA 00012CM 00017AG 00018RD 00029RD 00029RD 00032RD 00034RD
        00035RD 00037RD 00038RD 00040RD 00074RD
IFOUT   00011DA 00012CM 00075AG 00195WR 00197WR 00200WR 00202WR 00205WR
        00207WR 00210WR 00212WR 00214WR 00217WR 00219WR 00221WR 00226WR
        00226WR 00228WR 00231WR 00233WR 00236WR 00238WR 00241WR 00243WR
```

```
370     00151DO 00152*
380     00147DO 00149DO 00153*
390     00138GT 00146GT 00154*
40      00020RD 00021*
400     00157DO 00158DO 00160*
410     00163DO 00165DO 00167*
420     00156GT 00162GT 00168*
430     00171DO 00172DO 00173*
440     00176DO 00178DO 00179*
45      00024NR 00025*
450     00170GT 00175GT 00180*
460     00183DO 00184DO 00185*
470     00182GT 00186*
480     00189DO 00191DO 00192*
490     00188GT 00193*
50      00023GT 00027*
500     00327DO 00328*
60      00029RD 00030*  00032RD 00034RD 00035RD 00037RD 00038RD 00040RD
70      00041*  00043GT
80      00044*  00046GT
90      00047*  00049GT
```

```
00001       SUBROUTINE READMX(IMAX,JMAX,NROW,ARRAY,IERR)
      C       ROUTINE READ MATRIX ELEMENT OF THE FORM:
      C                I,J,VALUE
      C       CHECKS FOR DIMENSIONS OUT OF RANGE AND FORMS:
      C                ARRAY(I,J)=VALUE
00002       COMMON/IO/ IIN,IOUT,IFIN,IFOUT
00003       DIMENSION ARRAY(NROW,JMAX)
00004       READ(IFIN,10,ERR=50) I,J,VALUE
00005 10    FORMAT(2I,E)
00006       IF(I.GT.IMAX.OR.I.LT.0.OR.J.GT.JMAX.OR.J.LT.0) GO TO 30
00008       IF(I.EQ.0) GO TO 20
00010       ARRAY(I,J)=VALUE
00011       IERR=0
00012       RETURN
      C       ERRORS:   IERR=1 - I=0 INDICATING TERMINATION OF DATA FOR
      C                           THIS MATRIX
      C                 IERR=2 - DIMENSIONS OUT OF RANGE
      C                 IERR=3 - WRONG FORMAT ON INPUT
00013 20    IERR=1
00014       RETURN
00015 30    IERR=2
00016       WRITE(IOUT,40) I,J,VALUE
00017 40    FORMAT(5X,'DIMENSIONS OUT OF RANGE FOR ',3X,2I3,E15.6)
00018       RETURN
00019 50    IERR=3
00020       WRITE(IOUT,60)
00021 60    FORMAT(5X,'WRONG FORMAT')
00022       RETURN
00023       END
```

```
ARRAY   00001AG 00003DI 00010=
I       00004RD 00006IF 00008IF 00010   00016WR
IERR    00001AG 00011=  00013=  00015=  00019=
IFIN    00002CM 00004RD
IFOUT   00002CM
IIN     00002CM
IMAX    00001AG 00006IF
IO      00002CN
IOUT    00002CM 00016WR 00020WR
J       00004RD 00006IF 00010   00016WR
JMAX    00001AG 00003DI 00006IF
NROW    00001AG 00003DI
READMX  00001SU
VALUE   00004RD 00010   00016WR
10      00004RD 00005*
20      00009GT 00013*
30      00007GT 00015*
40      00016WR 00017*
50      00004RD 00019*
60      00020WR 00021*
```

```
      C         BLOCK2 - CONTROLLED ELEMENT ANALYSIS
      C           SETS UP A TREN FILE FOR TRANSFER FUNCTION ANALYSIS
      C           OF THE CONTROLLED ELEMENT.
00001         DIMENSION FNAME(5),TITLE(15)
      C         PROBLEM FILE INPUT MATRICES
00002         REAL MNU(5),MNY(15),MNYS(5),MNZ(15),MNYO(15),MNXC(10)
00003         DIMENSION GMC(10,15),GC(10,5),CC(10,5),FC(10,10),
     +                  HM(15,10),HO(15,15),E1(30,5)
      C         WORKING SPACE
00004         REAL MN(400),MM(25)
      C         DATA
00005         DATA IIN,IOUT,IPFIN,ITREN /5,7,20,21/
00006         DATA KO,IO,A2,A1,IPLUS,X1 /2*0,2*0.,'+',1./
      C         READ PROBLEM FILE NAME, TITLE AND TREN FILE NAME
00007         WRITE(IOUT,10)
00008 10      FORMAT(10X,'CONTROLLED ELEMENT DEFINITION - BLOCK 2',/,
     +       2X,'INPUT PROBLEM FILE NAME ? ',$)
00009         READ(IIN,20) (FNAME(I),I=1,4)
00010         CALL ASSIGN(IPFIN,FNAME)
00011         WRITE(IOUT,12)
00012 12      FORMAT(2X,'OUTPUT TREN FILE NAME ? ',$)
00013         READ(IIN,20) (FNAME(I),I=1,4)
00014         CALL ASSIGN(ITREN,FNAME)
00015         WRITE(IOUT,14)
00016 14      FORMAT(2X,'TREN TITLE (60 CHARS) : ')
00017         READ(IIN,20) (TITLE(I),I=1,15)
00018 20      FORMAT(15A4)
      C         READ THE FOLLOWING FROM THE PROBLEM FILE:
      C           DIMENSIONS  IDEN=2
      C           U MNEMONICS IDEN=3
      C           W MNEMONICS IDEN=4
      C           YS MNEMONICS IDEN=5
      C           Y MNEMONICS IDEN=9&10
      C           XC MNEMONICS IDEN=12
      C           GMC MATRIX  IDEN=15
      C           CC MATRIX   IDEN=17
      C           CC MATRIX   IDEN=18
      C           FC MATRIX   IDEN=19
      C           HM MATRIX   IDEN=20
      C           HO MATRIX   IDEN=22
      C           E1 MATRIX   IDEN=29
00019         DO 23 K=1,200
00020         READ(IPFIN,END=35) IDEN
00021         IF(IDEN.EQ.1) READ(IPFIN) IDEN,NXS,NXC,NU,NW,NZ,NYS,NYO
00023         IF(IDEN.EQ.2) READ(IPFIN) IDEN,NX,(MNU(I),I=1,NX)
00025         IF(IDEN.EQ.3) READ(IPFIN) IDEN,NX,(MNW(I),I=1,NX)
00027         IF(IDEN.EQ.5) READ(IPFIN) IDEN,NX,(MNYS(I),I=1,NX)
00029         IF(IDEN.EQ.8) READ(IPFIN) IDEN,NX,(MNYO(I),I=1,NX)
00031         IF(IDEN.EQ.9) READ(IPFIN) IDEN,NX,(MNZ(I),I=1,NX)
00033         IF(IDEN.EQ.11) READ(IPFIN) IDEN,NX,(MNXC(I),I=1,NX)
00035         IF(IDEN.EQ.15) READ(IPFIN) IDEN,NX,NY,((GMC(I,J),J=1,NZ),I=1,NX)
00037         IF(IDEN.EQ.16) READ(IPFIN) IDEN,NX,NY,((GC(I,J),J=1,NY),I=1,NX)
00039         IF(IDEN.EQ.17) READ(IPFIN) IDEN,NX,NY,((CC(I,J),J=1,NY),I=1,NX)
00041         IF(IDEN.EQ.18) READ(IPFIN) IDEN,NX,NY,((FC(I,J),J=1,NY),I=1,NX)
```

```
00043          IF(IDEN.EQ.19) READ(IPFIN) IDEN,NX,NY,((HM(I,J),J=1,NY),I=1,NX)
00045          IF(IDEN.EQ.21) READ(IPFIN) IDEN,NX,NY,((HO(I,J),J=1,NY),I=1,NX)
00047          IF(IDEN.EQ.28) READ(IPFIN) IDEN,NX,NY,((E1(I,J),J=1,NY),I=1,NX)
00049 30       CONTINUE
00050 35       NY=NZ+NYO
00051          NX=NX-NXC
00052          CALL CLOSE(IPFIN)
               C                    (SI-FC   0)
               C       FORM LHS = (          ) = (NXC,NXC   NXC,NY)
               C                  ( -HM    I)   (NY,NXC     NY,NY)
               C                  ( -HO     )
00053          IF(NXC.EQ.0) GO TO 65
00055          DO 40 I=1,NXC
00056          DO 40 J=1,NXC
00057          A1=0.
00058          IF(I.EQ.J) A1=1.
00060          X=-FC(I,J)
00061          IF(A1.EQ.0..AND.X.EQ.0.) GO TO 40
00063          WRITE(ITRFN,80) I,J,A2,A1,X,K0
00064 40       CONTINUE
00065          A1=0.
00066          DO 50 I=1,NZ
00067          II=I+NXC
00068          DO 50 J=1,NXC
00069          IF(HM(I,J).EQ.0.) GO TO 50
00071          X=-HM(I,J)
00072          WRITE(ITRFN,80) II,J,A2,A1,X,K0
00073 50       CONTINUE
00074          IF(NYO.EQ.0) GO TO 65
00076          DO 60 I=1,NYO
00077          II=I+NXC+NZ
00078          DO 60 J=1,NXC
00079          IF(HO(I,J).EQ.0.) GO TO 60
00081          X=-HO(I,J)
00082          WRITE(ITRFN,80) II,J,A2,A1,X,K0
00083 60       CONTINUE
00084 65       DO 70 I=1,NY
00085          II=I+NXC
00086 70       WRITE(ITRFN,80) II,II,A2,A1,X1,K0
00087 80       FORMAT(2I,3E,I)
00088          WRITE(ITRFN,80) I0
               C                    (GC   CC   G*C)     (NXC,NU   NXC,NYS   NXC,NV)
               C       FORM RHS = (                ) = (                          )
               C                    (0    E1    0 )     (NY,NU    NY,NYS    NY,NV)
00089          IF(NXC.EQ.0) GO TO 105
00091          IF(NU.EQ.0) GO TO 95
00093          DO 90 I=1,NXC
00094          DO 90 J=1,NXC
00095          IF(GC(I,J).EQ.0.) GO TO 90
00097          X=GC(I,J)
00098          WRITE(ITRFN,80) I,J,A2,A1,X,K0
00099 90       CONTINUE
00100 95       CONTINUE
00101          IF(NYS.EQ.0) GO TO 115
```

```
00103          DO 100 I=1,NXC
00104          DO 100 J=1,NYS
00105          JJ=J+NU
00106          IF(CC(I,J).EQ.0.) GO TO 100
00108          X=CC(I,J)
00109          WRITE(ITRFN,80) I,JJ,A2,A1,X,K0
00110 100      CONTINUE
00111 105      DO 110 I=1,NY
00112          II=I+NXC
00113          DO 110 J=1,NYS
00114          JJ=J+NU
00115          IF(E1(I,J).EQ.0.) GO TO 110
00117          X=E1(I,J)
00118          WRITE(ITRFN,80) II,JJ,A2,A1,X,K0
00119 110      CONTINUE
00120 115      CONTINUE
00121          IF(NW.EQ.0.OR.NXC.EQ.0) GO TO 125
00123          DO 120 I=1,NXC
00124          DO 120 J=1,NW
00125          JJ=J+NU+NYS
00126          IF(GMC(I,J).EQ.0.) GO TO 120
00128          X=GMC(I,J)
00129          WRITE(ITRFN,80) I,JJ,A2,A1,X,K0
00130 120      CONTINUE
00131 125      WRITE(ITRFN,80) I0
               C     COMPOSE COLUMN CODES
00132          IF(NXC.EQ.0) GO TO 140
00134          DO 130 I=1,NXC
00135 130      MN(I)=MNXC(I)
00136 140      CONTINUE
00137          DO 150 I=1,NZ
00138          II=I+NXC
00139 150      MN(II)=MNZ(I)
00140          IF(NYO.EQ.0) GO TO 170
00142          DO 160 I=1,NYO
00143          II=I+NXC+NZ
00144 160      MN(II)=MNYO(I)
00145 170      CONTINUE
00146          IF(NU.EQ.0) GO TO 190
00148          DO 180 I=1,NU
00149 180      MM(I)=MNU(I)
00150 190      CONTINUE
00151          IF(NYS.EQ.0) GO TO 210
00153          DO 200 I=1,NYS
00154          II=I+NU
00155 200      MM(II)=MNYS(I)
00156 210      CONTINUE
00157          IF(NW.EQ.0) GO TO 230
00159          DO 220 I=1,NW
00160          II=I+NXS+NU
00161 220      MM(II)=MNW(I)
               C     WRITE TITLE, MATRIX DIMENSIONS AND COLUMN CODES
00162 230      WRITE(ITRFN,20) (TITLE(I),I=1,15)
00163          N=NXC+NY
```

```
00164          M=NU+NYS+NW
00165          WRITE(ITRFN,240) N,M,I0
00166 240      FORMAT(3I)
00167          IF(N.LE.18) GO TO 270
00169          WRITE(ITRFN,250) (MN(I),I=1,18),IPLUS
00170 250      FORMAT(17A4,A3,A1)
00171          IF(N.LE.36) GO TO 260
00173          WRITE(ITRFN,250) (MN(I),I=19,36),IPLUS
00174          WRITE(ITRFN,250) (MN(I),I=37,N)
00175          GO TO 280
00176 260      WRITE(ITRFN,250) (MN(I),I=19,N)
00177          GO TO 280
00178 270      WRITE(ITRFN,250) (MN(I),I=1,N)
00179 280      CONTINUE
00180          IF(M.LE.18) GO TO 290
00182          WRITE(ITRFN,250) (M4(I),I=1,18),IPLUS
00183          WRITE(ITRFN,250) (M4(I),I=19,M)
00184          GO TO 300
00185 290      WRITE(ITRFN,250) (M4(I),I=1,M)
               C          THE END
00185 300      STOP
00187          END
```

```
ASSIGN  000010CL 000014CL
A1      000006DA 000057=  000059=  000061IF 000063#R 000065=  000072#R 000082#R
        000093#R 000098#R 000109#R 000118#R 000129#R
A2      000105DA 000063#R 000072#R 000082#R 000093#R 000098#R 000109#R 000118#R
        000129#R
CC      000003DI 000040RD 000106IF 000108
CLOSE   000052CL
E1      000043DI 000048RD 000115IF 000117
FC      000033DI 000042RD 000060
FNAME   000001DI 000009RD 000014AG 000013RD 000014AG
GC      000003DI 000039RD 000095IF 000097
GMC     000033DI 000036RD 000126IF 000128
HM      000003DI 000044RD 000069IF 000071
HO      000033DI 000046RD 000079IF 000081
I       000049RD 000013RD 000017RD 000024RD 000026RD 000028RD 000030RD 000032RD
        000034RD 000036RD 000039RD 000040RD 000042RD 000044RD 000046RD 000048RD
        000055DO 000058IF 000060  000063#R 000066DO 000067  000069IF 000071
        000076DO 000077  000079IF 000081  000084DO 000085  000093DO 000095IF
        000097  000098#R 000103DO 000106IF 000108  000109#R 000111DO 000112
        000115IF 000117  000123DO 000126IF 000128  000129#R 000134DO 000135
        000137DO 000138  000139  000142DO 000143  000144  000148DO 000149
        000153DO 000154  000155  000159DO 000160  000161  000162#R 000169#R
        000173#R 000174#R 000176#R 000178#R 000182#R 000183#R 000185#R
IDEN    000020RD 000021IF 000022RD 000023IF 000024RD 000025IF 000026RD 000027IF
        000028RD 000029IF 000030RD 000031IF 000032RD 000033IF 000034RD 000035IF
        000036RD 000037IF 000038RD 000039IF 000040RD 000041IF 000042RD 000043IF
        000044RD 000045IF 000046RD 000047IF 000048RD
II      000057=  000072#R 000077=  000082#R 000085=  000086#R 000112=  000118#R
        000138=  000139  000143=  000144  000154=  000155  000160=  000161
IIN     000005DA 000009RD 000013RD 000017RD
IOUT    000005DA 000007#R 000011#R 000015#R
IPFIN   000025DA 000010AG 000028RD 000032RD 000036RD 000038RD 000040RD 000042RD
        000044RD 000046RD 000048RD 000052AG
IPLUS   000006DA 000169#R 000173#R 000182#R
ITRFN   000005DA 000014AG 000063#R 000072#R 000082#R 000086#R 000093#R 000098#R
        000109#R 000118#R 000129#R 000131#R 000162#R 000165#R 000169#R 000173#R
        000174#R 000176#R 000178#R 000182#R 000183#R 000185#R
IO      000005DA 000086#R 000131#R 000165#R
J       000036RD 000038RD 000040RD 000042RD 000044RD 000046RD 000048RD 000056DO
        000058IF 000060  000063#R 000069DO 000069IF 000071  000072#R 000078DO
        000079IF 000081  000082#R 000094DO 000095IF 000097  000098#R 000104DO
        000105  000106IF 000108  000113DO 000114  000115IF 000117  000124DO
        000125  000126IF 000128
JJ      000105=  000109#R 000114=  000118#R 000125=  000129#R
K       000119DO
KB      000006DA 000063#R 000072#R 000082#R 000086#R 000098#R 000109#R 000118#R
        000129#R
M       000164=  000169#R 000180IF 000183#R 000185#R
MM      000004RL 000149=  000155=  000161=  000182#R 000183#R 000185#R
MN      000004RL 000135=  000139=  000144=  000169#R 000173#R 000176#R 000178#R
        000178#R
MNU     000002RL 000024RD 000149
MNN     000002RL 000026RD 000161
```

```
MNXC    00002RL 00034RD 00135
MNYO    00002RL 00038RD 00144
MNYS    00002RL 00028RD 00155
MNZ     00002RL 00032RD 00139
N       00163=  00165NR 00167IF 00171IF 00174NR 00175NR 00178NR
NU      00022RD 00091IF 00105   00114   00125   00146IF 00148DO 00154
        00160   00164
NV      00022RD 00051=  00121IF 00124DO 00157IF 00159DO 00164
NX      00024RD 00026RD 00028RD 00030RD 00032RD 00034RD 00036RD 00038RD
        00040RD 00042RD 00044RD 00046RD 00048RD
NXC     00022RD 00051   00053IF 00055DO 00056DO 00067   00068DO 00077
        00078DO 00085   00089IF 00093DO 00094DO 00103DO 00112   00121IF
        00123DO 00132IF 00134DO 00138   00143   00163
NXS     00022RD 00160
NY      00036RD 00038RD 00040RD 00042RD 00044RD 00046RD 00048RD 00050=
        00084DO 00111DO 00163
NYO     00022RD 00050   00074IF 00076DO 00140IF 00142DO
NYS     00022RD 00101IF 00104DO 00113DO 00125   00151IF 00153DO 00164
NZ      00022RD 00050   00066DO 00077   00137DO 00143
TITLE   00001DI 00017RD 00162NR
X       00056=  00061IF 00063NR 00071=  00072NR 00081=  00082NR 00097=
        00098NR 00108=  00109NR 00117=  00118NR 00128=  00129NR
X1      00005DA 00086NR
10      00007NR 00008*
103     00103DO 00104DO 00107GT 00110*
105     00090GT 00111*
110     00111DO 00113DO 00116GT 00119*
115     00102GT 00120*
12      00011NR 00012*
120     00123DO 00124DO 00127GT 00130*
125     00122GT 00131*
130     00134DO 00135*
14      00015NR 00016*
140     00133GT 00136*
150     00137DO 00139*
160     00142DO 00144*
170     00141GT 00145*
180     00148DO 00149*
190     00147GT 00150*
20      00009RD 00013RD 00017RD 00018*  00162NR
200     00153DO 00155*
210     00152GT 00156*
220     00159DO 00161*
230     00158GT 00162*
240     00165NR 00166*
250     00169NR 00170*  00173NR 00174NR 00175NR 00178NR 00182NR 00183NR
        00185NR
260     00172GT 00176*
270     00166GT 00178*
280     00175GT 00177GT 00179
290     00181GT 00185*
30      00019DO 00049*
300     00184GT 00186*
35      00020RD 00050*
```

```
40      00055DO 00056DO 00062GT 00064*
50      00066DO 00068DO 00070GT 00073*
60      00076DO 00078DO 00080GT 00083*
65      00054GT 00075GT 00084*
70      00084DO 00085*
80      00062NR 00072NR 00082NR 00088NR 00087*  00088NR 00093NR 00109NR
        00110NR 00129NR 00131NR
90      00093DO 00094DO 00095GT 00099*
95      00092GT 00100*
```

```
      C        BLOCK 3 - THE FILTER PROBLEM SET-UP
      C          THIS ROUTINE SETS UP THE SINGULAR OR NON-SINGULAR
      C          FILTER PROBLEM USING THE DATA GENERATED BY BLOCK 1
      C          AND ADDITIONAL INPUTS QD AND RD.
00001       DIMENSION FNAME(4),TTLNEW(5),SCALE(15)
      C        INPUT MATRIX ARRAYS
00002       DIMENSION QD(15),RD(15)
      C        PROBLEM FILE INPUT MATRIX ARRAYS
00003       REAL MNV(15),MNZ(15),MNXT(15),MNXS(5),MNXC(10)
00004       DIMENSION F(15,15),G(15,5),GM(15,15),H(15,15),IREOR(15,15)
      C        OUTPUT ARRAYS
00005       DIMENSION T1(15,15),T2(15,15),AR4(30,30)
      C        WORKING ARRAYS
00006       DIMENSION ARRAY(900),IARRAY(900),DUMMY(15,15),DUMMY2(15,15)
00007       DIMENSION XINVRT(15,15),T2I(15,15),A(15,15),ZETA(15),TZ(15,15)
      C        EQUIVALENCES
00008       EQUIVALENCE (ARRAY,IARRAY),(ARRAY,T1),(ARRAY,AR4),
     +                (ARRAY(226),T2),(ARRAY(451),T2I)
00009       EQUIVALENCE (XINVRT,A),(DUMMY2,TZ)
      C        COMMONS
00010       COMMON/IO/ IIN,IOUT
00011       COMMON/NONAME/ NDIM3
      C        INITIALIZE ARRAYS
00012       DATA F,G,GM,H,QD,RD,A,ARM,ZETA /1920*0./
      C        OTHER DATA
00013       DATA IIN,IOUT,IPFIN,IPFOUT /5,7,20,21/
00014       DATA I1,R0,I900,LASTID,NDIM3,NDIM6 /1,0.,900,31,15,30/
      C        READ PROBLEM FILE NAMES AND REGULATOR TITLE
00015       WRITE(IOUT,10)
00016 10    FORMAT(20X,'FILTER SET-UP - BLOCK 3',/,2X,
     +       'INPUT PROBLEM FILE NAME ? ',$)
00017       READ(IIN,20) (FNAME(I),I=1,4)
00018       CALL ASSIGN(IPFIN,FNAME)
00019       WRITE(IOUT,12)
00020 12    FORMAT(2X,'OUTPUT PROBLEM FILE NAME ? ',$)
00021       READ(IIN,20) (FNAME(I),I=1,4)
00022       CALL ASSIGN(IPFOUT,FNAME)
00023       WRITE(IOUT,14)
00024 14    FORMAT(2X,'PROBLEM TITLE (20 CHARS) : ',$)
00025       READ(IIN,20) (TTLNEW(I),I=1,5)
00026 20    FORMAT(5A4)
00027       WRITE(IOUT,22)
00028 22    FORMAT(2X,'AUGMENTATION CONSTANT : ',$)
00029       READ(IIN,25) ACONST
00030 25    FORMAT(E)
      C        GET THE FOLLOWING DATA FROM THE PROBLEM FILE:
      C          DIMENSIONS     IDEN=2
      C          V MNEMONICS    IDEN=6
      C          Z MNEMONICS    IDEN=10
      C          X MNEMONICS    IDEN=11&12
      C          F MATRIX       IDEN=24
      C          G MATRIX       IDEN=25
      C          GM MATRIX      IDEN=26
      C          H MATRIX       IDEN=27
```

```
00031        DO 40 K=1,200
00032        READ(IPFIN,END=42) IDEN
00033        IF(IDEN.EQ.1) READ(IPFIN) IDEN,NXS,NXC,NU,NM,NZ,NYS,NYO
00035        IF(IDEN.EQ.5) READ(IPFIN) IDEN,NX,(MNV(I),I=1,NX)
00037        IF(IDEN.EQ.9) READ(IPFIN) IDEN,NX,(MNZ(I),I=1,NX)
00039        IF(IDEN.EQ.10) READ(IPFIN) IDEN,NX,(MNXS(I),I=1,NX)
00041        IF(IDEN.EQ.11) READ(IPFIN) IDEN,NX,(MNXC(I),I=1,NX)
00043        IF(IDEN.EQ.23) READ(IPFIN) IDEN,NX,NY,((F(I,J),J=1,NY),I=1,NX)
00045        IF(IDEN.EQ.24) READ(IPFIN) IDEN,NX,NY,((G(I,J),J=1,NY),I=1,NX)
00047        IF(IDEN.EQ.25) READ(IPFIN) IDEN,NX,NY,((G4(I,J),J=1,NY),I=1,NX)
00049        IF(IDEN.EQ.26) READ(IPFIN) IDEN,NX,NY,((H(I,J),J=1,NY),I=1,NX)
00051 40     CONTINUE
      C         READ QD AND RD
00052 42     WRITE(IOUT,45)
00053 45     FORMAT(2X,'NON-ZERO DIAGONAL Q ELEMENTS : ')
00054 50     CALL READMX(NM,QD,IERR)
00055        IF(IERR.NE.1) GO TO 50
00057        WRITE(IOUT,55)
00058 55     FORMAT(2X,'NON-ZERO DIAGONAL R ELEMENTS : ')
00059 60     CALL READMX(NZ,RD,IERR)
00061        IF(IERR.NE.1) GO TO 60
      C         READ AND WRITE PROBLEM FILE, COMPOSEING NEW TITLE
00062        REWIND IPFIN
00063        CALL RENR(IPFIN,IPFOUT,TTLNEW,1000,LASTID,ARRAY,IARRAY)
00064        NXT=NXC+NXS
00065        IF(NXS.EQ.0) GO TO 64
00067        DO 62 I=1,NXS
00068 62     MNXT(I)=MNXS(I)
00069 64     CONTINUE
00070        IF(NXC.EQ.0) GO TO 68
00072        DO 66 I=1,NXC
00073        II=I+NXS
00074 66     MNXT(II)=MNXC(I)
00075 68     CONTINUE
```

```
      C        SEE THAT G, GM AND H HAVE FULL RANK
00076         DO 70 I=1,NXT
00077         DO 70 J=1,NU
00078 70      DUMMY(I,J)=G(I,J)
00079         CALL GMINV(NXT,NU,DUMMY,XINVRT,MR,0)
00080         IF(MR.NE.NU) WRITE(IOUT,80)
00082 80      FORMAT(5X,'G MATRIX DOES NOT HAVE FULL RANK')
00083         DO 90 I=1,NXT
00084         DO 90 J=1,NM
00085 90      DUMMY(I,J)=GM(I,J)
00086         CALL GMINV(NXT,NM,DUMMY,XINVRT,MR,0)
00087         IF(MR.LT.NM-NXC) WRITE(IOUT,100)
00089 100     FORMAT(5X,'GM MATRIX DOES NOT HAVE FULL RANK')
00090         DO 110 I=1,NZ
00091         DO 110 J=1,NXT
00092 110     DUMMY(I,J)=H(I,J)
00093         CALL GMINV(NZ,NXT,DUMMY,XINVRT,MR,0)
00094         IF(MR.NE.NZ) WRITE(IOUT,120)
00096 120     FORMAT(5X,'H MATRIX DOES NOT HAVE FULL RANK')
      C        HERE WE SHOULD TEST FOR THE DETECTABILITY OF (F  H),
      C        BUT WE DON'T DO THAT.  IF (F  H) IS NOT DETECTABLE,
      C        THERE WILL BE CLOSED LOOP ROOTS IN THE RIGHT-HALF PLANE.
      C        IS THIS A SINGULAR FILTERING PROBLEM?  IF SO,
      C        REORDER RD, H, MNV AND MNZ.
00097         DO 125 I=1,NZ
00098         DO 125 J=1,NZ
00099         TZ(I,J)=0.
00102 125     TZ(I,I)=1.
00101         M2P=0
00102         DO 140 I=1,NZ
00103         II=NZ-I+1
00104         IF(RD(II).NE.0.) GO TO 140
00106         M2P=M2P+1
00107         IF(I.EQ.M2P) GO TO 140
00109         DUM=RD(II)
00110         RD(II)=RD(II+1)
00111         RD(II+1)=DUM
00112         DO 130 J=1,NXT
00113         DUM=H(II,J)
00114         H(II,J)=H(II+1,J)
00115         HREOR(II,J)=H(II+1,J)
00116         H(II+1,J)=DUM
00117 130     HREOR(II+1,J)=DUM
00118         DO 135 J=1,NZ
00119         DUM=TZ(II,J)
00120         TZ(II,J)=TZ(II+1,J)
00121 135     TZ(II+1,J)=DUM
00122         DUM=MNV(II)
00123         MNV(II)=MNV(II+1)
00124         MNV(II+1)=DUM
00125         DUM=MNZ(II)
00126         MNZ(II)=MNZ(II+1)
00127         MNZ(II+1)=DUM
00128 140     CONTINUE
```

```
00129          IF(M2P.EQ.0) GO TO 700
      C           WRITE MNEMONICS AND TZ TO OUTPUT PROBLEM FILE
00131          IDEN=LASTID+1
00132          WRITE(IPFOUT) IDEN,NZ,(MNV(I),I=1,NZ)
00133          IDEN=IDEN+1
00134          WRITE(IPFOUT) IDEN,NZ,(MNZ(I),I=1,NZ)
00135          M1=NZ-M2P
00136          M2=M2P
00137          M1P=NXT-M2
      C           MAKE T1 THE IDENTITY MATRIX
00138          DO 144 I=1,NXT
00139          DO 144 J=1,NXT
00140          T1(I,J)=0.
00141  144     T1(I,I)=1.0
      C           SINGULAR FILTERING PROBLEM - REORDER STATE TO CAUSE
      C           H22 TO HAVE FULL RANK.
00142          DO 200 NCOL=1,M2P
00143          NBCOL=NXT-NCOL+1
00144          NC=NCOL
00145          DO 146 I=1,M2P
00146          II=I+M1
00147          DO 146 J=1,NCOL
00148          JB=NXT-J+1
00149          JC=NCOL-J+1
00150  146     DUMMY(I,JC)=H(II,JB)
00151          NC=NCOL
00152          CALL GMINV(M2P,NC,DUMMY,XINVRT,MR,0)
00153          IF(MR.EQ.NCOL) GO TO 200
00155          DO 160 L=1,NZ
00156          DUM=H(L,NBCOL)
00157          DO 150 M=1,NBCOL-1
00158          MBACK=NBCOL-M+1
00159  150     H(L,MBACK)=H(L,MBACK-1)
00160  160     H(L,1)=DUM
00161          DUMN=MNXT(NBCOL)
00162          DO 170 L=1,NBCOL-1
00163          LBACK=NBCOL-L+1
00164  170     MNXT(LBACK)=MNXT(LBACK-1)
00165          MNXT(1)=DUMN
00166          DO 190 L=1,NXT
00167          DUM=T1(L,NBCOL)
00168          DO 180 M=1,NBCOL-1
00169          MBACK=NBCOL-M+1
00170  180     T1(L,MBACK)=T1(L,MBACK-1)
00171  190     T1(L,1)=DUM
00172  200     CONTINUE
      C                        ( I    0 )
      C           FORM T2 =    (        )
      C                        (H21  H22)
00173  210     CONTINUE
00174          DO 215 I=1,NXT
00175          DO 215 J=1,NXT
00176  215     T2(I,J)=0.
00177          IF(M1P.EQ.0) GO TO 245
```

```
00179        DO 220 I=1,M1P
00180 220    T2(I,I)=1.0
00181        DO 240 I=1,M2P
00182        II=I+M1
00183        IK=I+M1P
00184        DO 240 J=1,M1P
00185 240    T2(IK,J)=H(II,J)
00186 245    DO 250 I=1,M2P
00187        II=I+M1
00188        IK=I+M1P
00189        DO 250 J=1,M2P
00190        JJ=J+NXT-M2P
00191 250    T2(IK,JJ)=H(II,JJ)
      C         COMPUTE T=T2*T1 AND TI
00192        DO 255 I=1,NXT
00193        DO 255 J=1,NXT
00194        DUM=0.
00195        DO 253 K=1,NXT
00196 253    DUM=DUM+T2(I,K)*T1(J,K)
00197 255    DUMMY(I,J)=DUM
00198        DO 257 I=1,NXT
00199        DO 257 J=1,NXT
00200 257    T2(I,J)=DUMMY(I,J)
00201        CALL GMINV(NXT,NXT,DUMMY,T2I,MR,0)
00202        DO 258 I=1,M1P
00203 258    CALL TRANSL(MNXT,MNXT,'E','X')
00204        IDEN=IDEN+1
00205        WRITE(IPFOUT) IDEN,NXT,(MNXT(I),I=1,M1P),
            +      (MNZ(I),I=1+M1,M1+M2)
00206        IDEN=IDEN+1
00207        WRITE (IPFOUT) IDEN,NZ,NZ,((TZ(I,J),J=1,NZ),I=1,NZ)
00208        IDEN=IDEN+1
00209        WRITE(IPFOUT) IDEN,NXT,NXT,((T2(I,J),J=1,NXT),I=1,NXT)
00210        IDEN=IDEN+1
00211        WRITE(IPFOUT) IDEN,NXT,NXT,((T2I(I,J),J=1,NXT),I=1,NXT)
00212        IF(M1P.EQ.0) GO TO 645
      C         TRANSFORM F, G, GM:
      C           F=T2*F*T2I
      C           G=T2*G
      C           GM=T2*GM
00214        DO 270 I=1,NXT
00215        DO 270 J=1,NXT
00216        DUM=0.
00217        DO 260 K=1,NXT
00218 260    DUM=DUM+T2(I,K)*F(K,J)
00219 270    DUMMY(I,J)=DUM
00220        DO 290 I=1,NXT
00221        DO 290 J=1,NXT
00222        DUM=0.
00223        DO 280 K=1,NXT
00224 280    DUM=DUM+DUMMY(I,K)*T2I(K,J)
00225 290    F(I,J)=DUM
00226        DO 310 I=1,NXT
00227        DO 310 J=1,NU
```

```
00228          DUM=0.
00229          DO 300 K=1,NXT
00230  300     DUM=DUM+T2(I,K)*G(K,J)
00231  310     DUMMY(I,J)=DUM
00232          DO 320 I=1,NXT
00233          DO 320 J=1,NU
00234  320     G(I,J)=DUMMY(I,J)
00235          DO 340 I=1,NXT
00236          DO 340 J=1,NN
00237          DUM=0.
00238          DO 330 K=1,NXT
00239  330     DUM=DUM+T2(I,K)*GM(K,J)
00240  340     DUMMY(I,J)=DUM
00241          DO 350 I=1,NXT
00242          DO 350 J=1,NN
00243  350     GM(I,J)=DUMMY(I,J)
       C          DETERMINE THE RANK OF GM2*QD*GM2T AND IF RANK IS NOT
       C          FULL, AUGMENT.
00244          DO 410 ISQR=1,M2
00245          DO 400 I=1,ISQR
00246          II=I+M1P
00247          DO 400 J=1,ISQR
00248          JJ=J+M1P
00249          DUM=0.
00250          DO 390 K=1,NN
00251  390     DUM=DUM+GM(II,K)*GM(JJ,K)*QD(K)
00252          IF(I.EQ.J) DUM=DUM+ZETA(I)
00254  400     DUMMY(I,J)=DUM
00255          IS=ISQR
00256          CALL GMINV(IS,IS,DUMMY,A,MR,0)
00257          IF(MR.LT.ISQR) ZETA(ISQR)=ACONST
00259  410     CONTINUE
00260          DO 425 I=1,M2
00261          II=I+M1P
00262          DO 425 J=1,M2
00263          JJ=J+M1P
00264          DUM=0.
00265          DO 420 K=1,NN
00266  420     DUM=DUM+GM(II,K)*GM(JJ,K)*QD(K)
00267          IF(I.EQ.J) DUM=DUM+ZETA(I)
00269  425     DUMMY(I,J)=DUM
00270          CALL GMINV(M2,M2,DUMMY,A,MR,0)
00271          IS=0
00272          DO 428 I=1,M2
00273          IF(ZETA(I).GT.0.) IS=1
00275  428     CONTINUE
00276          IF(IS.EQ.1) WRITE(IOUT,429)
00278  429     FORMAT(5X,'GM2*QD*GM2T HAS BEEN AUGMENTED')
```

```
      C         FORM UPPER LEFT QUADRANT OF ARM:
      C          F11-GM1*QD*GM2T*(GM2*QD*GM2T)I*F21
00279           DO 440 I=1,M1P
00280           DO 440 J=1,M2
00281           JJ=J+M1P
00282           DUM=0.
00283           DO 430 K=1,NN
00284 430       DUM=DUM+GM(I,K)*GM(JJ,K)*QD(K)
00285 440       DUMMY(I,J)=DUM
00286           DO 460 I=1,M1P
00287           DO 460 J=1,M2
00288           DUM=0.
00289           DO 450 K=1,M2
00290 450       DUM=DUM+DUMMY(I,K)*A(K,J)
00291 460       DUMMY2(I,J)=DUM
00292           DO 480 I=1,M1P
00293           DO 480 J=1,M1P
00294           DUM=0.
00295           DO 470 K=1,M2
00296           KK=K+M1P
00297 470       DUM=DUM+DUMMY2(I,K)*F(KK,J)
00298 480       ARM(I,J)=F(I,J)-DUM
      C            FORM THE LOWER RIGHT QUADRANT OF ARM, WHICH IS THE NEGATIVE
      C            TRANSPOSE OF THE UPPER LEFT QUADRANT.
00299           DO 490 I=1,M1P
00300           DO 490 J=1,M1P
00301 490       ARM(I+M1P,J+M1P)=-ARM(J,I)
      C            FORM THE LOWER LEFT QUADRANT OF ARM:
      C             -H11T*R1I*H11 - F21T*(GM2*QD*GM2T)I*F21
00302           DO 510 I=1,M1P
00303           DO 510 J=1,M2
00304           DUM=0.
00305           DO 500 K=1,M2
00306           KK=K+M1P
00307 500       DUM=DUM+F(KK,I)*A(K,J)
00308 510       DUMMY(I,J)=DUM
00309           DO 525 I=1,M1P
00310           II=I+M1P
00311           DO 525 J=1,M1P
00312           DUM=0.
00313           DO 520 K=1,M2
00314           KK=K+M1P
00315 520       DUM=DUM+DUMMY(I,K)*F(KK,J)
00316 525       ARM(II,J)=-DUM
00317           IF(M1.EQ.0) GO TO 545
00319           DO 540 I=1,M1P
00320           II=I+M1P
00321           DO 540 J=1,M1P
00322           DUM=0.
00323           DO 530 K=1,M1
00324 530       DUM=DUM-H(K,I)*H(K,J)/RD(K)
00325 540       ARM(II,J)=DUM+ARM(II,J)
      C            FORM THE UPPER RIGHT QUADRANT OF ARM:
      C             -GM1*QD*GM1T + GM1*QD*GM2T*(GM2*QD*GM2T)I*GM2*QD*GM1T
```

```
00326 545    DO 560 I=1,M1P
00327        DO 550 J=1,M2
00328        JJ=I+M1P
00329        DUM=0.
00330        DO 550 K=1,NN
00331 550    DUM=DUM+GM(I,K)*GM(JJ,K)*QD(K)
00332 560    DUMMY(I,J)=DUM
00333        DO 580 I=1,M1P
00334        DO 580 J=1,M2
00335        DUM=0.
00336        DO 570 K=1,M2
00337 570    DUM=DUM+DUMMY(I,K)*A(K,J)
00338 580    DUMMY2(I,J)=DUM
00339        DO 600 I=1,M1P
00340        DO 600 J=1,NN
00341        DUM=0.
00342        DO 590 K=1,M2
00343        KK=K+M1P
00344 590    DUM=DUM+DUMMY2(I,K)*GM(KK,J)*QD(J)
00345 600    DUMMY(I,J)=DUM
00346        DO 620 I=1,M1P
00347        DO 620 J=1,M1P
00348        DUM=0.
00349        DO 610 K=1,NN
00350 610    DUM=DUM+DUMMY(I,K)*GM(J,K)
00351 620    DUMMY2(I,J)=DUM
00352        DO 640 I=1,M1P
00353        DO 640 J=1,M1P
00354        JJ=J+M1P
00355        DUM=0.
00356        DO 630 K=1,NN
00357 630    DUM=DUM-GM(I,K)*GM(I,K)*QD(K)
00358 640    ARM(I,JJ)=DUM+DUMMY2(I,J)
      C        WRITE THE FOLLOWING TO THE PROBLEM FILE:
      C          F11,F12,F21,F22,G1,G2,GM1,GM2,HREOR,H11,H12,H22,QD,RD,ARM
00359        IDEN=IDEN+1
00360        WRITE(IPFOUT) IDEN,M1P,M1P,((F(I,J),J=1,M1P),I=1,M1P)
00361        IDEN=IDEN+1
00362        WRITE(IPFOUT) IDEN,M1P,M2,((F(I,J),J=M1P+1,NXT),I=1,M1P)
00363        IDEN=IDEN+1
00364        WRITE(IPFOUT) IDEN,M2,M1P,((F(I,J),J=1,M1P),I=M1P+1,NXT)
00365        IDEN=IDEN+1
00366        WRITE(IPFOUT) IDEN,M2,M2,((F(I,J),J=M1P+1,NXT),I=M1P+1,NXT)
00367        IDEN=IDEN+1
00368        WRITE(IPFOUT) IDEN,M1P,NU,((G(I,J),J=1,NU),I=1,M1P)
00369        IDEN=IDEN+1
00370        WRITE(IPFOUT) IDEN,M2,NU,((G(I,J),J=1,NU),I=M1P+1,NXT)
00371        IDEN=IDEN+1
00372        WRITE(IPFOUT) IDEN,M1P,NN,((GM(I,J),J=1,NN),I=1,M1P)
00373        IDEN=IDEN+1
00374        WRITE(IPFOUT) IDEN,M2,NN,((GM(I,J),J=1,NN),I=M1P+1,NXT)
00375        IDEN=IDEN+1
00376        NX=M1+M2P
00377        NY=M1P+M2
```


C-2

```
00378        WRITE(IPFOUT) IDEN,NX,NY,((HREOR(I,J),J=1,NY),I=1,NX)
00379        IDEN=IDEN+1
00380        IF(M1.EQ.0) WRITE(IPFOUT) IDEN,I1,I1,R0
00382        IF(M1.NE.0)
            + WRITE(IPFOUT) IDEN,M1,M1P,((H(I,J),J=1,M1P),I=1,M1)
00384        IDEN=IDEN+1
00385        IF(M1.EQ.0) WRITE(IPFOUT) IDEN,I1,I1,R0
00387        IF(M1.NE.0)
            + WRITE(IPFOUT) IDEN,M1,M2,((H(I,J),J=M1P+1,NXT),I=1,M1)
00389        IDEN=IDEN+1
00390        WRITE(IPFOUT) IDEN,M2,M2,((H(I,J),J=M1P+1,NXT),I=M1+1,NZ)
00391        IDEN=IDEN+1
00392        WRITE(IPFOUT) IDEN,NN,(QD(I),I=1,NN)
00393        IDEN=IDEN+1
00394        WRITE(IPFOUT) IDEN,NZ,(RD(I),I=1,NZ)
00395        IDEN=IDEN+1
00396        NUM=M1P*2
00397        WRITE(IPFOUT) IDEN,NUM,NUM,((ARM(I,J),J=1,NUM),I=1,NUM)
00398        IDEN=IDEN+1
00399        WRITE(IPFOUT) IDEN,M2,M2,((A(I,J),J=1,M2),I=1,M2)
00400        IDEN=IDEN+1
00401        WRITE(IPFOUT) IDEN,M1,M1P,M2,M2P
00402        GO TO 900
     C          FOR R IDENTICALLY SINGULAR
00403 655    DO 655 K=1,3
00404        IDEN=IDEN+1
00405 655    WRITE(IPFOUT) IDEN,I1,I1,R0
00406        IDEN=IDEN+1
00407        WRITE(IPFOUT) IDEN,M2,M2,((F(I,J),J=1,M2),I=1,M2)
00408        IDEN=IDEN+1
00409        WRITE(IPFOUT) IDEN,I1,I1,R0
00410        IDEN=IDEN+1
00411        WRITE(IPFOUT) IDEN,M2,NU,((G(I,J),J=1,NU),I=1,M2)
00412        IDEN=IDEN+1
00413        WRITE(IPFOUT) IDEN,I1,I1,R0
00414        IDEN=IDEN+1
00415        WRITE(IPFOUT) IDEN,M2,NN,((GM(I,J),J=1,NN),I=1,M2)
00416        DO 660 K=1,3
00417        IDEN=IDEN+1
00418 660    WRITE(IPFOUT) IDEN,I1,I1,R0
00419        IDEN=IDEN+1
00420        WRITE(IPFOUT) IDEN,M2,M2,((H(I,J),J=1,M2),I=1,M2)
00421        IDEN=IDEN+1
00422        WRITE(IPFOUT) IDEN,NN,(QD(I),I=1,NN)
00423        IDEN=IDEN+1
00424        WRITE(IPFOUT) IDEN,NZ,(RD(I),I=1,NZ)
00425        IDEN=IDEN+1
00426        WRITE(IPFOUT) IDEN,I1,I1,R0
00427        IDEN=IDEN+1
00428        WRITE(IPFOUT) IDEN,I1,I1,R0
00429        IDEN=IDEN+1
00430        WRITE(IPFOUT) IDEN,M1,M1P,M2,M2P
00431        GO TO 900
```

```
      C        NON-SINGULAR FILTER
      C                          (   F      -GQGT)
      C        ASSEMBLE ARM = (                  )
      C                          (-HTRIH    -FT )
00432 700    DO 710 I=1,NXT
00433        DO 710 J=1,NXT
00434 710    ARM(I,J)=F(I,J)
00435        DO 730 I=1,NXT
00436        II=I+NXT
00437        DO 730 J=1,NXT
00438        DUM=0.
00439        DO 720 K=1,NZ
00440 720    DUM=DUM+H(K,I)*H(K,J)/RD(K)
00441 730    ARM(II,J)=DUM
00442        DO 750 I=1,NXT
00443        DO 750 J=1,NXT
00444        JJ=J+NXT
00445        DUM=0.
00446        DO 740 K=1,NU
00447 740    DUM=DUM-G(I,K)*G(J,K)*QD(K)
00448 750    ARM(I,JJ)=DUM
00449        DO 760 I=1,NXT
00450        II=I+NXT
00451        DO 760 J=1,NXT
00452        JJ=J+NXT
00453 760    ARM(II,JJ)=-F(J,I)
      C        WRITE THE FOLLOWING TO THE PROBLEM FILE:
      C          MNEMONICS,T1,T,TI,F,G,GM,H,QD,RD AND ARM
00454        DO 770 I=1,NXT
00455        DO 770 J=1,NXT
00456        DUMMY(I,J)=0.
00457 770    DUMMY(I,I)=1.0
00458        IDEN=LASTID+1
00459        WRITE(IPFOUT) IDEN,NZ,(MNV(I),I=1,NZ)
00460        IDEN=IDEN+1
00461        WRITE(IPFOUT) IDEN,NZ,(MNZ(I),I=1,NZ)
00462        IDEN=IDEN+1
00463        DO 775 I=1,NXT
00464 775    CALL TRANSL(MNXT,MNXT,'E','X')
00465        WRITE(IPFOUT) IDEN,NXT,(MNXT(I),I=1,NXT)
00466        IDEN=IDEN+1
00467        WRITE(IPFOUT) IDEN,NXT,NXT,((DUMMY(I,J),J=1,NXT),I=1,NXT)
00468        IDEN=IDEN+1
00469        WRITE(IPFOUT) IDEN,NXT,NXT,((DUMMY(I,J),J=1,NXT),I=1,NXT)
00470        IDEN=IDEN+1
00471        WRITE(IPFOUT) IDEN,NXT,NXT,((DUMMY(I,J),J=1,NXT),I=1,NXT)
00472        IDEN=IDEN+1
00473        WRITE(IPFOUT) IDEN,NXT,NXT,((F(I,J),J=1,NXT),I=1,NXT)
00474        DO 780 I=1,3
00475        IDEN=IDEN+1
00476 780    WRITE(IPFOUT) IDEN,I1,I1,R0
00477        IDEN=IDEN+1
00478        WRITE(IPFOUT) IDEN,NXT,NU,((G(I,J),J=1,NU),I=1,NXT)
00479        IDEN=IDEN+1
```

```
00480        WRITE(IPFOUT) IDEN,I1,I1,R0
00481        IDEN=IDEN+1
00482        WRITE(IPFOUT) IDEN,NXT,NN,((GM(I,J),J=1,NN),I=1,NXT)
00483        IDEN=IDEN+1
00484        WRITE(IPFOUT) IDEN,I1,I1,R0
00485        IDEN=IDEN+1
00486        WRITE(IPFOUT) IDEN,NZ,NXT,((H(I,J),J=1,NXT),I=1,NZ)
00487        IDEN=IDEN+1
00488        WRITE(IPFOUT) IDEN,NZ,NXT,((H(I,J),J=1,NXT),I=1,NZ)
00489        DO 790 I=1,2
00490        IDEN=IDEN+1
00491  790   WRITE(IPFOUT) IDEN,I1,I1,R0
00492        IDEN=IDEN+1
00493        WRITE(IPFOUT) IDEN,NN,(QD(I),I=1,NN)
00494        IDEN=IDEN+1
00495        WRITE(IPFOUT) IDEN,NZ,(RD(I),I=1,NZ)
00496        IDEN=IDEN+1
00497        NUM=2*NXT
00498        WRITE(IPFOUT) IDEN,NUM,NUM,((ARM(I,J),J=1,NUM),I=1,NUM)
00499        IDEN=IDEN+1
00500        WRITE(IPFOUT) IDEN,I1,I1,R0
00501        IDEN=IDEN+1
00502        WRITE(IPFOUT) IDEN,NZ,NXT,M2P,M2P
     C          THE END
00503  900   DO 910 I=IDEN+1,130
00504  910   WRITE(IPFOUT) I,I1,I1,R0
00505  1300  STOP
00506        END
```

```
        00383#R 00388#R 00390#R 00392#R 00394#R 00397#R 00399#R 00407#R
        00411#R 00415#R 00423#R 00422#R 00424#R 00432DO 00434   00435DO
        00436   00440   00442DO 00447   00448   00449DO 00450   00453
        00454DO 00456   00457   00459#R 00461#R 00463DO 00465#R 00467#R
        00469#R 00471#R 00473#R 00474DO 00478#R 00482#R 00486#R 00488#R
        00489DO 00493#R 00495#R 00498#R 00503DO 00504#R
IARRAY  00026DI 00008EQ 00063AG
IDEN    00032RD 00033IF 00034RD 00035IF 00036RD 00037IF 00038RD 00039IF
        00042RD 00041IF 00042RD 00043IF 00044RD 00045IF 00046RD 00047IF
        00048RD 00049IF 00050RD 00131=  00132#R 00133=  00134#R 00204=
        00205#R 00206=  00207#R 00208=  00209#R 00210=  00211#R 00359=
        00360#R 00361=  00362#R 00363=  00364#R 00365=  00366#R 00367=
        00368#R 00369=  00370#R 00371=  00372#R 00373=  00374#R 00375=
        00378#R 00379=  00381#R 00383#R 00384=  00386#R 00388#R 00389=
        00390#R 00391=  00392#R 00393=  00394#R 00395=  00397#R 00398=
        00399#R 00400=  00401#R 00404=  00405#R 00406=  00407#R 00408=
        00409#R 00410=  00411#R 00412=  00413#R 00414=  00415#R 00417=
        00418#R 00419=  00420#R 00421=  00422#R 00423=  00424#R 00425=
        00426#R 00427=  00428#R 00429=  00430#R 00458=  00459#R 00460=
        00461#R 00462=  00465#R 00466=  00467#R 00468=  00469#R 00470=
        00471#R 00472=  00473#R 00475=  00476#R 00477=  00478#R 00479=
        00480#R 00481=  00482#R 00483=  00484#R 00485=  00486#R 00487=
        00488#R 00490=  00491#R 00492=  00493#R 00494=  00495#R 00496=
        00498#R 00499=  00500#R 00501=  00502#R 00503DO
IERR    00054AG 00055IF 00059AG 00060IF
II      00073=  00074   00103=  00104IF 00109   00110   00111   00113
        00114   00115   00116   00117   00119   00120   00121   00122
        00123   00124   00125   00126   00127   00146=  00150   00182=
        00185   00187=  00191   00246=  00251   00261=  00266   00310=
        00316   00320=  00325   00436=  00441   00453=  00453
IIN     00010CM 00013DA 00017RD 00021RD 00025RD 00029RD
IK      00183=  00185   00183=  00191
IO      00010CN
IOUT    00010CM 00013DA 00019#R 00019#R 00023#R 00027#R 00052#R 00057#R
        00081#R 00088#R 00095#R 00277#R
IPFIN   00013DA 00018AG 00032RD 00034RD 00036RD 00038RD 00040RD 00042RD
        00044RD 00046RD 00048RD 00050RD 00052RD 00063AG
IPFOUT  00013DA 00022AG 00063AG 00132#R 00134#R 00205#R 00207#R 00209#R
        00211#R 00360#R 00362#R 00364#R 00365#R 00368#R 00370#R 00372#R
        00374#R 00376#R 00381#R 00383#R 00386#R 00388#R 00390#R 00392#R
        00394#R 00397#R 00399#R 00401#R 00405#R 00407#R 00409#R 00411#R
        00413#R 00415#R 00418#R 00420#R 00422#R 00424#R 00426#R 00428#R
        00430#R 00459#R 00461#R 00465#R 00467#R 00469#R 00471#R 00473#R
        00476#R 00478#R 00480#R 00482#R 00484#R 00486#R 00488#R 00491#R
        00493#R 00495#R 00498#R 00500#R 00502#R 00504#R
IS      00255=  00255AG 00271=  00274=  00276IF
ISQR    00244DO 00245DO 00247DQ 00255   00257IF 00259
I1      00014DA 00381#R 00386#R 00405#R 00409#R 00413#R 00418#R 00426#R
        00428#R 00476#R 00480#R 00484#R 00491#R 00502#R 00504#R
I9P0    00014DA 00063AG
J       00044RD 00046RD 00048RD 00050RD 00077DO 00078   00084DO 00085
        00091DO 00092   00098DO 00099   00112DO 00113   00114   00115
        00116   00117   00118DO 00119   00120   00121   00139DO 00140
        00147DO 00148   00149   00175DO 00176   00184DO 00185   00189DO
```

```
M2P       00101=  00106=  00107IF 00129IF 00135   00136   00142DO 00145DO
          00152AG 00181DO 00186DO 00189DO 00190   00376   00401WR 00430WR
          00502WR
NBCOL     00143=  00156   00157DO 00159   00161   00162DO 00163   00167
          00168DO 00169
NC        00144=  00151=  00152AG
NCOL      00142DO 00143   00144   00147DO 00149   00151   00153IF
NDIM3     00011CM 00014DA
NDIM6     00014DA
NONAME    00011CN
NU        00034RD 00077DO 00079AG 00080IF 00227DO 00233DO 00368WR 00372WR
          00411WR 00446DO 00478WR
NUM       00396=  00397WR 00497=  00498WR
NN        00034RD 00054AG 00084DO 00086AG 00087IF 00236DO 00242DO 00250DO
          00265DO 00283DO 00330DO 00340DO 00349DO 00356DO 00372WR 00374WR
          00392WR 00415WR 00422WR 00482WR 00493WR
NX        00036RD 00039RD 00040RD 00042RD 00044RD 00046RD 00048RD 00050RD
          00376=  00378WR
NXC       00034RD 00064   00070IF 00072DO 00087IF
NXS       00034RD 00064   00065IF 00067DO 00073
NXT       00064=  00076DO 00079AG 00083DO 00086AG 00091DO 00093AG 00112DO
          00137   00138DO 00139DO 00143   00148   00166DO 00174DO 00175DO
          00190   00192DO 00193DO 00195DO 00198DO 00199DO 00201AG 00205WR
          00209WR 00211WR 00214DO 00215DO 00217DO 00220DO 00221DO 00223DO
          00226DO 00229DO 00232DO 00235DO 00238DO 00241DO 00362WR 00364WR
          00368WR 00370WR 00374WR 00388WR 00390WR 00432DO 00433DO 00435DO
          00436   00437DO 00442DO 00443DO 00444   00449DO 00450   00451DO
          00452   00454DO 00455DO 00463DO 00465WR 00467WR 00469WR 00471WR
          00473WR 00478WR 00482WR 00488WR 00489WR 00497   00502WR
NY        00044RD 00046RD 00048RD 00050RD 00377=  00378WR
NYO       00034RD
NYS       00034RD
NZ        00034RD 00059AG 00090DO 00093AG 00094IF 00097DO 00098DO 00102DO
          00103   00118DO 00132WR 00134WR 00135   00155DO 00207WR 00390WR
          00394WR 00424WR 00439DO 00459WR 00461WR 00486WR 00488WR 00495WR
          00502WR
QD        00022DI 00012DA 00054AG 00251   00266   00284   00331   00344
          00357   00392WR 00422WR 00447   00493WR
RD        00022DI 00012DA 00059AG 00104IF 00109   00110=  00111=  00324
          00394WR 00424WR 00440   00499WR
READ4X    00054CL 00059CL
RENR      00063CL
R0        00014DA 00301WR 00386WR 00435WR 00469WR 00413WR 00418WR 00426WR
          00426WR 00476WR 00482WR 00494WR 00491WR 00507WR 00504WR
SCALE     00001DI
TRANSL    00203CL 00464CL
TTLNEW    00001DI 00025RD 00053AG
TZ        00007DI 00009EQ 00099=  00100=  00119   00120=  00121=  00207WR
T1        00005DI 00008EQ 00140=  00141=  00167   00170=  00171=  00196
T2        00005DI 00008EQ 00176=  00180=  00185=  00191=  00196   00200=
          00209WR 00218   00230   00239
T2I       00007DI 00008EQ 00201AG 00211WR 00224
XINVRT    00007DI 00009EQ 00079AG 00086AG 00093AG 00152AG
ZETA      00007DI 00012DA 00253   00258=  00268   00273IF
```

```
10      00015NR 00016*
100     00089NR 00089*
1000    00505*
110     00090DO 00091DO 00092*
12      00019NR 00020*
120     00095NR 00096*
125     00097DO 00098DO 00100*
130     00112DO 00117*
135     00118DO 00121*
14      00023NR 00024*
140     00102DO 00105GT 00109GT 00120*
144     00133DO 00139DO 00141*
146     00145DO 00147DO 00150*
150     00157DO 00159*
160     00155DO 00160*
170     00162DO 00164*
180     00168DO 00170*
190     00166DO 00171*
20      00017RD 00021RD 00025RD 00026*
200     00142DO 00154GT 00172*
210     00173*
215     00174DO 00175DO 00176*
22      00027NR 00028*
220     00179DO 00180*
240     00181DO 00184DO 00185*
245     00179GT 00186*
25      00029RD 00030*
250     00186DO 00189DO 00191*
253     00195DO 00196*
255     00192DO 00193DO 00197*
257     00198DO 00199DO 00200*
258     00202DO 00203*
260     00217DO 00218*
270     00214DO 00215DO 00219*
280     00223DO 00224*
290     00220DO 00221DO 00225*
300     00229DO 00230*
310     00226DO 00227DO 00231*
320     00232DO 00233DO 00234*
330     00238DO 00239*
340     00235DO 00236DO 00240*
350     00241DO 00242DO 00243*
390     00250DO 00251*
40      00031DO 00051*
400     00245DO 00247DO 00254*
410     00244DO 00259*
42      00032RD 00052*
420     00265DO 00266*
425     00260DO 00262DO 00269*
428     00272DO 00275*
429     00277NR 00278*
430     00203DO 00234*
440     00279DO 00280DO 00285*
45      00052NR 00053*
```

```
450     00289DO 00290*
460     00285DO 00287DO 00291*
470     00295DO 00297*
480     00292DO 00293DO 00298*
490     00299DO 00300DO 00301*
50      00054*  00056GT
500     00305DO 00307*
510     00302DO 00303DO 00308*
520     00313DO 00315*
525     00309DO 00311DO 00316*
530     00323DO 00324*
540     00319DO 00321DO 00325*
545     00310GT 00326*
55      00057MR 00058*
550     00330DO 00331*
560     00326DO 00327DO 00332*
570     00336DO 00337*
580     00333DO 00334DO 00338*
590     00342DO 00344*
60      00059*  00061GT
600     00339DO 00340DO 00345*
610     00349DO 00350*
62      00067DO 00068*
620     00346DO 00347DO 00351*
630     00356DO 00357*
64      00066GT 00069*
640     00352DO 00353DO 00358*
645     00213GT 00403*
655     00403DO 00405*
66      00072DO 00074*
660     00416DO 00418*
68      00071GT 00075*
70      00076DO 00077DO 00078*
700     00130GT 00432*
710     00432DO 00433DO 00434*
720     00439DO 00440*
730     00435DO 00437DO 00441*
740     00446DO 00447*
750     00442DO 00443DO 00448*
760     00449DO 00451DO 00453*
770     00454DO 00455DO 00457*
775     00463DO 00464*
780     00474DO 00476*
790     00489DO 00491*
80      00081MR 00082*
90      00083DO 00084DO 00085*
900     00442GT 00431GT 00503*
910     00503DO 00504*
```

```
00001        SUBROUTINE READMX(IMAX,ARRAY,IERR)
             C     ROUTINE READS MATRIX ELEMENT OF THE FORM:
             C          I,VALUE
             C     CHECKS FOR DIMENSIONS OUT OF RANGE AND FORMS:
             C          ARRAY(I)=VALUE
             C
00002        COMMON/IO/ IIN,IOUT
00003        DIMENSION ARRAY(IMAX)
00004        READ(IIN,10,ERR=50) I,VALUE
00005  10    FORMAT(I,E)
00006        IF(I.GT.IMAX.OR.I.LT.0) GO TO 30
00007        IF(I.EQ.0) GO TO 20
00010        ARRAY(I)=VALUE
00011        IERR=0
00012        RETURN
             C     ERRORS:    IERR=1 - I=0 INDICATING TERMINATION OF DATA
             C                         FOR THIS MATRIX
             C                IERR=2 - DIMENSIONS OUT OF RANGE
             C                IERR=3 - WRONG FORMAT ON INPUT
             C
00013  20    IERR=1
00014        RETURN
00015  30    IERR=2
00016        WRITE(IOUT,40) I,VALUE
00017  40    FORMAT(5X,'DIMENSIONS OUT OF RANGE FOR ',3X,I3,E15.6)
00018        RETURN
00019  50    IERR=3
00020        WRITE(IOUT,60)
00021  60    FORMAT(5X,'WRONG FORMAT')
00022        RETURN
00023        END
```

```
ARRAY   000001AG 000003DI 00010=
I       000004RD 000006IF 000008IF 00010   00010NR
IERR    000001AG 00011=  00013=  00015=  00019=
IIN     000002CM 000004RD
IMAX    000001AG 000003DI 000006IF
IO      000002CN
IOUT    000002CM 00010NR 00020NR
READMX  000001SU
VALUE   000004RD 00010   00010NR
10      000004RD 00005*
20      000009GT 00013*
30      000007GT 00015*
40      00010NR 00017*
50      000004RD 00019*
60      00020NR 00021*
```

```
      C          BLOCK 4 - FILTER GAINS AND ESTIMATION ERROR
      C        , UTILIZES FILTER SET-UP (BLOCK 3) AND PERFORMS
      C          EIGENVALUE DECOMPOSITION (QR ALGORITHM) TO OBTAIN
      C          FILTER GAINS. ALSO COMPUTES RMS AND CORRELATION.
00001       DIMENSION FNAME(4),TTLNEW(5)
      C        PROBLEM FILE INPUT MATRIX ARRAYS
00002       DIMENSION F21(15,15),GM(15,15),GM2(15,15),H(15,15)
00003       DIMENSION ARM(30,30),QD(15),RD(15)
      C        OUTPUT ARRAYS
00004       COMPLEX EVAL(30),EVEC(30,30),W21(15,15)
      C        WORKING ARRAYS
00005       DIMENSION ARRAY(900),IARRAY(900),DUMMY(15,15),DUMMY2(15,15)
00006       DIMENSION A(15,15),WK(30,30),INT(30),SCALE(30),WR(30),WI(30)
00007       COMPLEX DD,CD,C(15),WA(225),W11(15,15)
      C        EQUIVALENCES
00008       EQUIVALENCE (ARM,W11),(ARM(1,16),W21)
00009       EQUIVALENCE (EVEC,DUMMY,ARRAY,GM2),(EVEC(1,16),DUMMY2,IARRAY)
00010       EQUIVALENCE (WK,WA),(C,SCALE)
      C        INITIALIZE ARRAYS
00011       DATA F21,GM,H,ARM,PX,QD,RD,SCALE,WR,WI /1920*0./
00012       DATA INT /30*0/,TTLNEW /' "" ',4*' '/
00013       DATA EVAL,EVEC /930*(0.,0.)/
      C        OTHER DATA
00014       DATA IIN,IOUT,IPFIN,IPFOUT /5,7,20,21/
00015       DATA I1,R0,IWK,I900,LASTID,NDIM3,NDIM6 /1,0.,225,900,54,15,30/
00016       DATA CD /(0.,0.)/
      C        READ PROBLEM FILE NAMES
00017       WRITE(IOUT,10)
00018 10    FORMAT(20X,'FILTER GAINS, EST ERROR - BLOCK 4',/,2X,
           +     'INPUT PROBLEM FILE NAME ? ',$)
00019       READ(IIN,20) (FNAME(I),I=1,4)
00020       CALL ASSIGN(IPFIN,FNAME)
00021       WRITE(IOUT,15)
00022 15    FORMAT(2X,'OUTPUT PROBLEM FILE NAME ? ',$)
00023       READ(IIN,20) (FNAME(I),I=1,4)
00024       CALL ASSIGN(IPFOUT,FNAME)
00025 20    FORMAT(4A4)
      C        GET THE FOLLOWING FROM THE PROBLEM FILE:
      C          DIMENSIONS  IDEN=2
      C          F21 MATRIX  IDEN=40
      C          GM1 MATRIX  IDEN=44
      C          GM2 MATRIX  IDEN=45
      C          H11 MATRIX  IDEN=47
      C          QD  MATRIX  IDEN=50
      C          RD  MATRIX  IDEN=51
      C          ARM MATRIX  IDEN=52
      C          A   MATRIX  IDEN=53
      C          DIMENSSIONS IDEN=54
00026       DO 30 K=1,200
00027       READ(IPFIN,END=40) IDEN
00028       IF(IDEN.EQ.1) READ(IPFIN) IDEN,NXS,NXC,NU,NM,NZ,NYS,NYO
00030       IF(IDEN.EQ.39) READ(IPFIN) IDEN,NX,NY,((F21(I,J),J=1,NY),I=1,NX)
00032       IF(IDEN.EQ.43) READ(IPFIN) IDEN,NX,NY,((GM(I,J),J=1,NY),I=1,NX)
00034       IF(IDEN.EQ.44) READ(IPFIN) IDEN,NX,NY,((GM2(I,J),J=1,NY),I=1,NX)
```

```
00035        IF(IDEN.EQ.46) READ(IPFIN) IDEN,NX,NY,((H(I,J),J=1,NY),I=1,NX)
00038        IF(IDEN.EQ.49) READ(IPFIN) IDEN,NX,(OD(I),I=1,NX)
00040        IF(IDEN.EQ.50) READ(IPFIN) IDEN,NX,(ID(I),I=1,NX)
00042        IF(IDEN.EQ.51) READ(IPFIN) IDEN,NX,NY,((ARM(I,J),J=1,NY),I=1,NX)
00044        IF(IDEN.EQ.52) READ(IPFIN) IDEN,NX,NY,((A(I,J),J=1,NY),I=1,NX)
00046        IF(IDEN.EQ.53) READ(IPFIN) IDEN,M1,M1P,M2,M2P
00048 30     CONTINUE
00049 40     DO 45 I=1,M2
00050        II=I+M1P
00051        DO 45 J=1,NY
00052 45     G4(II,J)=G42(I,J)
     C          READ AND WRITE PROBLEM FILE
00053        REWIND IPFIN
00054        CALL RDWR(IPFIN,IPFOUT,TTLNEW,1900,LASTID,ARRAY,IARRAY)
00055        NTOT=M1P
00056        IDEN=LASTID
00057        IF(M1P.EQ.0) GO TO 1500
     C          COMPUTE OPEN LOOP EIGENVALUES AND WRITE THEM
00059        DO 50 I=1,NTOT
00060        DO 50 J=1,NTOT
00061 50     DUMMY(I,J)=ARM(I,J)
00062        CALL EIGRF(DUMMY,NTOT,NDIM3,0,EVAL,EVEC,INT,SCALE,WR,WI,WK,IERR)
00063        IDEN=IDEN+1
00064        WRITE(IPFOUT) IDEN,NTOT,(EVAL(I),I=1,NTOT)
     C          COMPUTE CLOSED LOOP EIGENVALUES AND EIGENVECTORS
00065        CALL EIGRF(ARM,2*NTOT,NDIM6,1,EVAL,EVEC,INT,SCALE,WR,WI,WK,IERR)
00066        IF(IERR.EQ.0) GO TO 790
00068        WRITE(IOUT,780)
00069 780    FORMAT(5X,'ERROR COMPUTING CLOSED LOOP EIGENSYSTEM')
00070        GO TO 1500
     C               (W11) WITH EIGENVECTORS CORRESPONDING
     C          FORM (   ) TO EIGENVALUES WITH POSITIVE
     C               (W21) REAL PARTS.
00071 790    J=0
00072        DO 810 K=1,2*NTOT
00073        IF(REAL(EVAL(K)).LT.0.) GO TO 810
00075        IF(REAL(EVAL(K)).EQ.0.) WRITE(IOUT,795)
00077 795    FORMAT(5X,'REAL PART OF C.L. EIGENVALUE = 0.0')
00078        J=J+1
00079        EVAL(J)=-EVAL(K)
00080        DO 800 I=1,NTOT
00081        L=NTOT+I
00082        W11(J,I)=EVEC(I,K)
00083        W21(J,I)=EVEC(L,K)
00084 800    EVEC(I,J)=EVEC(I,K)
00085        IF(J.EQ.NTOT) GO TO 830
00087 810    CONTINUE
00088        WRITE(IOUT,820)
00089 820    FORMAT(5X,'ERROR IN EIGENVALUE SELECTION')
00090        GO TO 1500
     C          COMPLEX SOLUTION COV=W11*W21**-1
00091 830    CALL LEQT2C(W21,NTOT,NDIM3,W11,NTOT,NDIM3,INT,C,WK,WA,IERR)
00092        IF(IERR.EQ.0) GO TO 850
00094        WRITE(IOUT,840)
```

```
00095 840   FORMAT(5X,'ERROR IN EIGENVECTOR DECOMPOSITION')
00096       GO TO 1500
00097 850   CONTINUE
00098       DO 860 I=1,NTOT
00099       DO 860 J=1,NTOT
00100 860   DUMMY(I,J)=-REAL(W11(I,J))
      C        COMPUT NORMALIZED EIGENVECTORS
00101       DO 890 I=1,NTOT
00102       SF=0.
00103       DO 870 J=1,NTOT
00104       SFF=CABS(W21(I,J))
00105       IF(SF.GT.SFF) GO TO 870
00107       JMAX=J
00108       SF=SFF
00109 870   CONTINUE
00110       DD=W21(I,JMAX)
00111       DO 880 J=1,NTOT
00112 880   W21(I,J)=W21(I,J)/DD
00113 890   CONTINUE
      C        WRITE CLOSED LOOP EIGENVALUES, NORMALIZED
      C        EIGENVECTORS AND TRANSITION MATRIX
00114       IDEN=IDEN+1
00115       WRITE(IPFOUT) IDEN,NTOT,(EVAL(I),I=1,NTOT)
00116       IDEN=IDEN+1
00117       WRITE(IPFOUT) IDEN,NTOT,NTOT,((W21(I,J),J=1,NTOT),I=1,NTOT)
00118       IDEN=IDEN+1
00119       WRITE(IPFOUT) IDEN,NTOT,NTOT,((DUMMY(I,J),J=1,NTOT),I=1,NTOT)
      C        COMPUTE FILTER GAINS AND WRITE THEM:
      C          K11 = PX*H11T*RI
      C          K12 = (PX*P21T + G41*QD*G42T)*(G42*QD*G42T)I
00120       IF(M1.NE.0) GO TO 895
00122       IDEN=IDEN+1
00123       WRITE(IPFOUT) IDEN,I1,I1,R0
00124       GO TO 920
00125 895   DO 910 I=1,NTOT
00126       DO 910 J=1,M1
00127       DUM=0.
00128       DO 900 K=1,NTOT
00129 900   DUM=DUM+REAL(W11(I,K))*H(J,K)/RD(J)
00130 910   DUMMY(I,J)=DUM
00131       IDEN=IDEN+1
00132       WRITE(IPFOUT) IDEN,NTOT,M1,((DUMMY(I,J),J=1,M1),I=1,NTOT)
00133 920   CONTINUE
00134       IF(M2.EQ.1.AND.A(1,1).EQ.0.) GO TO 1010
00136       DO 960 I=1,NTOT
00137       DO 960 J=1,M2
00138       JJ=J+NTOT
00139       DUM=0.
00140       DO 950 K=1,NW
00141 950   DUM=DUM+GM(I,K)*GM(JJ,K)*QD(K)
00142 960   DUMMY(I,J)=DUM
00143       DO 980 I=1,NTOT
00144       DO 980 J=1,M2
00145       DUM=0.
```

```
00146          DO 970 K=1,NTOT
00147 970      DUM=DUM-REAL(W11(I,K))*F21(J,K)
00148 980      DUMMY2(I,J)=DUM+DUMMY(I,J)
00149          DO 1000 I=1,NTOT
00150          DO 1000 J=1,M2
00151          DUM=0.
00152          DO 990 K=1,M2
00153 990      DUM=DUM+DUMMY2(I,K)*A(K,J)
00154 1000     DUMMY(I,J)=DUM
00155          IDEN=IDEN+1
00156          WRITE(IPFOUT) IDEN,NTOT,M2,((DUMMY(I,J),J=1,M2),I=1,NTOT)
00157          GO TO 1015
00158 1010     IDEN=IDEN+1
00159          WRITE(IPFOUT) IDEN,I1,I1,R0
       C           GET TI
00160 1015     LASTID=IDEN
00161          REWIND IPFIN
00162          DO 1020 K=1,200
00163          READ(IPFIN,END=1030) IDEN
00164          IF(IDEN.EQ.36) READ(IPFIN) IDEN,NX,NY,((DUMMY2(I,J),J=1,NY),
              +     I=1,NX)
00165 1020     CONTINUE
       C           TRANSFORM P - TI*P*TI'
00167 1030     NXT=NX
00168          DO 1050 I=1,NXT
00169          DO 1050 J=1,NTOT
00170          DUM=0.
00171          DO 1040 K=1,NTOT
00172 1040     DUM=DUM+DUMMY2(I,K)*(-REAL(W11(K,J)))
00173 1050     DUMMY(I,J)=DUM
00174          DO 1070 I=1,NXT
00175          DO 1070 J=1,NXT
00176          DUM=0.
00177          DO 1060 K=1,NTOT
00178 1060     DUM=DUM+DUMMY(I,K)*DUMMY2(J,K)
00179 1070     W11(I,J)=CMPLX(-DUM,0.)
       C           COMPUTE RMS AND CORRELATION
00180          DO 1080 I=1,NXT
00181          DUM=-REAL(W11(I,I))
00182          IF(DUM.GE.0.) GO TO 1080
  '34          WRITE(IOUT,1075)
  _35 1075     FORMAT(5X,'FILTER RICCATI MATRIX NOT NON-NEGATIVE')
  .36 1080     WR(I)=SQRT(DUM)
00187          DO 1100 I=1,NXT
00188          DO 1100 J=1,NXT
00189          IF(WR(I).EQ.0..OR.WR(J).EQ.0.) GO TO 1090
00191          DUMMY(I,J)=-REAL(W11(I,J))/(WR(I)*WR(J))
00192          GO TO 1100
00193 1090     DUMMY(I,J)=0.
00194          DUMMY(J,I)=0.
00195          DUMMY(I,I)=1.
00196 1100     CONTINUE
00197          IDEN=LASTID+1
00198          WRITE(IPFOUT) IDEN,NXT,(WR(I),I=1,NXT)
```

```
00199          IDEN=IDEN+1
00200          WRITE(IPFOUT) IDEN,NXT,NXT,((DUMMY(I,J),J=1,NXT),I=1,NXT)
     C            THE END
00201 1500     DO 1510 I=IDEN+1,130
00202 1510     WRITE(IPFOUT) I,I1,I1,C0
00203 2000     STOP
00204          END
```

```
A       00006DI 00045RD 00134IF 00153
ARM     00003DI 00028EQ 00011DA 00043RD 00061   00065AG
ARRAY   00005DI 00009EQ 00054AG
ASSIGN  00020CL 00024CL
C       00007CX 00010EQ 00091AG
CABS    00104
CMPLX   00179
CØ      00007CX 00016DA 00202NR
DD      00007CX 00110=  00112
DUM     00127=  00129=  00130   00139=  00141=  00142   00145=  00147=
        00148   00151=  00153=  00154   00170=  00172=  00173   00176=
        00178=  00179   00181=  00182IF 00186
DUMMY   00005DI 00009EQ 00051=  00052AG 00100=  00119NR 00130=  00132NR
        00142=  00148   00154=  00156NR 00173=  00178   00191=  00193=
        00194=  00195=  00202NR
DUMMY2  00005DI 00009EQ 00148=  00153   00165RD 00172   00178
EIGRF   00062CL 00065CL
EVAL    00004CX 00013DA 00052AG 00064NR 00065AG 00073IF 00075IF 00079=
        00115NR
EVEC    00004CX 00009EQ 00013DA 00052AG 00065AG 00082   00083   00084=
FNAME   00031DI 00019RD 00020AG 00023RD 00024AG
F21     00002DI 00011DA 00031RD 00147
GM      00002DI 00011DA 00033RD 00052=  00141
G42     00002DI 00009EQ 00035RD 00052
H       00002DI 00011DA 00037RD 00129
I       00019RD 00023RD 00031RD 00033RD 00035RD 00037RD 00039RD 00041RD
        00043RD 00045RD 00049DO 00050   00052   00059DO 00061   00064NR
        00080DO 00081   00082   00083   00084   00099DO 00100   00101DO
        00104   00110   00112   00115NR 00117NR 00119NR 00125DO 00129
        00130   00132NR 00136DO 00141   00142   00143DO 00147   00148
        00149DO 00153   00154   00156NR 00165RD 00168DO 00172   00173
        00174DO 00178   00179   00182DO 00181   00186   00187DO 00189IF
        00191   00193   00194   00195   00198NR 00202NR 00201DO 00202NR
IARRAY  00005DI 00009EQ 00054AG
IDEN    00027RD 00028IF 00029RD 00030IF 00031RD 00032IF 00033RD 00034IF
        00035RD 00036IF 00037RD 00038IF 00039RD 00040IF 00041RD 00042IF
        00043RD 00044IF 00045RD 00046IF 00047RD 00056=  00063=  00064NR
        00114=  00115NR 00116=  00117NR 00118=  00119NR 00122=  00123NR
        00131=  00132NR 00155=  00156NR 00158=  00159NR 00160   00163RD
        00164IF 00165RD 00197=  00198NR 00199=  00200NR 00201DO
IERR    00062AG 00065AG 00066IF 00091AG 00092IF
II      00050=  00052
IIN     00014DA 00019RD 00023RD
INT     00005DI 00012DA 00062AG 00065AG 00091AG
IOUT    00014DA 00017NR 00021NR 00068NR 00076NR 00096NR 00094NR 00104NR
IPFIN   00014DA 00020AG 00027RD 00029RD 00031RD 00033RD 00035RD 00037RD
        00039RD 00041RD 00043RD 00045RD 00047RD 00053RD 00054AG 00161RD
        00163RD 00165RD
IPFOUT  00014DA 00024AG 00054AG 00064NR 00119NR 00117NR 00119NR 00123NR
        00132NR 00156NR 00159NR 00198NR 00200NR 00202NR
IWK     00015DA 00091AG
I1      00015DA 00123NR 00156NR 00202NR
I900    00015DA 00054AG
J       00031RD 00033RD 00035RD 00037RD 00043RD 00045RD 00051DO 00052
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
        00056DO 00061   00071=  00078=  00079   00082   00083   00084
        00085IF 00099DO 00100   00103DO 00104   00107   00111DO 00112
        00117NR 00119NR 00125DO 00129   00130   00132NR 00137DO 00138
        00142   00144DO 00147   00148   00150DO 00153   00154   00156NR
        00165RD 00169DO 00172   00173   00175DO 00178   00179   00188DO
        00189IF 00191   00193   00194   00200NR
JJ      00132=  00141
JMAX    00107=  00110
K       00020DO 00072DO 00073IF 00075IF 00079   00082   00083   00084
        00120DO 00129   00140DO 00141   00146DO 00147   00152DO 00153
        00162DO 00171DO 00172   00177DO 00178
L       00081=  00093
LASTID  00015DA 00054AG 00056   00160=  00197
LEQT2C  00091CL
M1      00047RD 00120IF 00126DO 00132NR
M1P     00047RD 00050   00055   00057IF
M2      00047RD 00049DO 00134IF 00137DO 00144DO 00150DO 00152DO 00156NR
M2P     00047RD
NDIM3   00015DA 00062AG 00091AG
NDIM6   00015DA 00065AG
NTOT    00055=  00059DO 00060DO 00062AG 00064NR 00065AG 00072DO 00080DO
        00081   00085IF 00091AG 00098DO 00099DO 00101DO 00103DO 00111DO
        00113NR 00117NR 00119NR 00125DO 00128DO 00132NR 00136DO 00138
        00143DO 00146DO 00149DO 00156NR 00169DO 00171DO 00177DO
NU      00029RD
NW      00029RD 00051DO 00142DO
NX      00031RD 00033RD 00035RD 00037RD 00039RD 00041RD 00043RD 00045RD
        00165RD 00167
NXC     00029RD
NXS     00029RD
NXT     00167=  00168DO 00174DO 00175DO 00180DO 00187DO 00188DO 00190NR
        00200NR
NY      00031RD 00033RD 00035RD 00037RD 00043RD 00045RD 00165RD
NYO     00029RD
NYS     00029RD
NZ      00029RD
PX      00011DA
QD      00003DI 00011DA 00039RD 00141
RD      00003DI 00011DA 00041RD 00129
REAL    00073IF 00075IF 00100   00129   00147   00172   00181   00191
REWR    00054CL
R0      00015DA 00123NR 00159NR
SCALE   00006DI 00010EQ 00011DA 00062AG 00065AG
SF      00102=  00105IF 00108=
SFF     00104=  00105IF 00108
SQRT    00186
TTLNEW  00001DI 00012DA 00054AG
WA      00007CX 00010EQ 00091AG
WI      00006DI 00011DA 00062AG 00065AG
WK      00006DI 00010EQ 00062AG 00065AG
WR      00006DI 00011DA 00062AG 00065AG 00186=  00189IF 00191   00193NR
W11     00007CX 00008EQ 00082=  00091AG 00100   00129   00147   00172
        00179=  00181   00191
W21     00004CX 00008EQ 00083=  00091AG 00104   00110   00112=  00117NR
```

```
10      00017NR 00018*
1000    00149DO 00150DO 00154*
1010    00135GT 00158*
1015    00157GT 00160*
1020    00162DO 00166*
1030    00163RD 00167*
1040    00171DO 00172*
1050    00168DO 00169DO 00173*
1060    00177DO 00178*
1070    00174DO 00175DO 00179*
1075    00184NR 00185*
1080    00180DO 00183GT 00186*
1090    00190GT 00193*
1100    00187DO 00188DO 00192GT 00196*
15      00021NR 00022*
1500    00059GT 00070GT 00090GT 00096GT 00201*
1510    00201DO 00202*
20      00019RD 00023RD 00025*
2000    00203*
30      00026DO 00048*
40      00027RD 00049*
45      00049DO 00051DO 00052*
50      00059DO 00062DO 00061*
700     00058NR 00069*
790     00067GT 00071*
795     00076NR 00077*
800     00080DO 00084*
810     00072DO 00074GT 00087*
820     00083NR 00089*
830     00085GT 00091*
840     00094NR 00095*
850     00093GT 00097*
860     00098DO 00099DO 00100*
870     00103DO 00106GT 00109*
880     00111DO 00112*
890     00101DO 00113*
895     00121GT 00125*
900     00128DO 00129*
910     00125DO 00126DO 00130*
920     00124GT 00133*
950     00140DO 00141*
950     00136DO 00137DO 00142*
970     00146DO 00147*
980     00143DO 00144DO 00148*
990     00152DO 00153*
```

```
      C         BLOCK 5 - THE REGULATOR PROBLEM
      C          THIS ROUTINE COMPUTES THE REGULATOR GAINS USING THE
      C          PROBLEM FILE GENERATED IN BLOCK 1 AND ADDITIONAL
      C          INPUTS OF QRD AND RRD.
00001         DIMENSION FNAME(4),TTLNEW(5)
      C          INPUT MATRIX ARRAYS
00002         DIMENSION RRD(5),QRD(30)
      C          PROBLEM FILE INPUT MATRIX ARRAYS
00003         DIMENSION F(15,15),G(15,5),HR(15,15)
      C          OUTPUT ARRAYS
00004         DIMENSION CG(5,15),PX(15,15),ARM(30,30)
00005         COMPLEX W21(15,15),EVAL(30),EVEC(30,30)
00006         EQUIVALENCE (CG,ARM)
      C          WORKING ARRAYS
00007         DIMENSION WK(30,30),WI(30),WR(30),SCALE(30),INT(30)
00008         DIMENSION ARRAY(900),IARRAY(900)
00009         COMPLEX DD,C0,W11(15,15),C(15),WA(225)
00010         EQUIVALENCE (SCALE,C),(EVEC,WA,F),(WK,W21,HR),(ARM,W11)
00011         EQUIVALENCE (ARM,ARRAY,IARRAY)
      C          INITIALIZE ARRAYS
00012         DATA RRD,QRD,F,G,HR,CG,PX,ARM,SCALE,WR,WI/1850*0./
00013         DATA INT /30*0/
00014         DATA W21,EVAL,EVEC,DD,W11,C /1396*(0.,0.)/
      C          OTHER DATA
00015         DATA IIN,IOUT,IPFIN,IPFOUT /5,7,20,21/
00016         DATA NK,I900,LASTID,NDIM3,NDIM6 /225,900,62,15,30/
00017         DATA I1,R0,C0 /1,0.,(0.,0.)/
00018         COMMON/IO/IIN,IOUT
      C          READ PROBLEM FILE NAMES AND REGULATOR TITLE
00019         WRITE(IOUT,10)
00020 10      FORMAT(20X,'REGULATOR GAINS - BLOCK 5 ',/,2X,
             +     'INPUT PROBLEM FILE NAME ? ',$)
00021         READ(IIN,20) (FNAME(I),I=1,4)
00022         CALL ASSIGN(IPFIN,FNAME)
00023         WRITE(IOUT,12)
00024 12      FORMAT(2X,'OUTPUT PROBLEM FILE NAME ? ',$)
00025         READ(IIN,20) (FNAME(I),I=1,4)
00026         CALL ASSIGN(IPFOUT,FNAME)
00027         WRITE(IOUT,14)
00028 14      FORMAT(2X,'PROBLEM TITLE (20 CHARS) : ',$)
00029         READ(IIN,20) (TTLNEW(I),I=1,5)
00030 20      FORMAT(5A4)
      C          GET DATA FROM PROBLEM FILE:
      C             TITLE       IDEN=1
      C             DIMENSIONS  IDEN=2
      C             F MATRIX    IDEN=24
      C             G MATRIX    IDEN=25
      C             HR MATRIX   IDEN=28
00031         DO 30 K=1,31
00032         READ(IPFIN,END=40) IDEN
00033         IF(IDEN.EQ.1) READ(IPFIN) IDEN,NXS,NXC,NU,NW,NZ,NYS,NYO
00035         IF(IDEN.EQ.23) READ(IPFIN) IDEN,NX,NY,((F(I,J),J=1,NY),I=1,NX)
00037         IF(IDEN.EQ.24) READ(IPFIN) IDEN,NX,NY,((G(I,J),J=1,NY),I=1,NX)
00039         IF(IDEN.EQ.27) READ(IPFIN) IDEN,NX,NY,((HR(I,J),J=1,NY),I=1,NX)
```

```
00041 30    CONTINUE
      C        READ IPFIN, WRITE IPFOUT, COMPOSING NEW TITLE
00042 40    REWIND IPFIN
00043       CALL RENR(IPFIN,IPFOUT,TTLNEW,1900,LASTID,ARRAY,IARRAY)
      C        READ QRDAND RRD MATRICES FROM INPUT FILE
00044       WRITE(IOUT,45)
00045 45    FORMAT(2X,'NON-ZERO DIAGONAL QR ELEMENTS :')
00046 50    CALL READMX(NZ+NYO,QRD,IERR)
00047       IF(IERR.NE.1) GO TO 50
00049       WRITE(IOUT,55)
00050 55    FORMAT(2X,'NON-ZERO DIAGONAL RR ELEMENTS :')
00051 60    CALL READMX(NU,RRD,IERR)
00052       IF(IERR.NE.1) GO TO 60
      C        AT THIS POINT, A CHECK SHOULD BE PERFORMED TO SEE IF (F  G)
      C        IS STABILIZABLE.  IN THIS VERSION THIS CHECK IS NOT MADE.
      C        IF (F   G) IS NOT STABILIZABLE, IT WILL BE APPARENT FROM
      C        THE TRANSFER FUNCTIONS OF THE CLOSED LOOP SYSTEM, SINCE THERE
      C        WILL BE UNSTABLE POLES.
      C        CHECK IF RRD HAS FULL RANK
00054       DO 80 I=1,NU
00055       IF(RRD(I).NE.0.0) GO TO 80
00057       WRITE(IOUT,70)
00058 70    FORMAT(5X,'RRD DOES NOT HAVE FULL RANK')
00059       GO TO 285
00060 80    CONTINUE
      C                              (  FT      HTQH)
      C        ASSEMBLE ARM =        (               )
      C                              (GRIST     -F  )
00061       NXT=NXC+NXS
00062       DO 90 I=1,NXT
00063       DO 90 J=1,NXT
00064       ARM(I,J)=F(J,I)
00065 90    PX(I,J)=F(I,J)
00066       DO 110 I=1,NXT
00067       II=I+NXT
00068       DO 110 J=1,NXT
00069       DUM=0.0
00070       DO 100 K=1,NU
00071 100   DUM=DUM+G(I,K)*G(J,K)/RRD(K)
00072 110   ARM(II,J)=DUM
00073       DO 130 I=1,NXT
00074       DO 130 J=1,NXT
00075       JJ=J+NXT
00076       DUM=0.0
00077       DO 120 K=1,NZ+NYO
00078 120   DUM=DUM+HR(K,I)*HR(K,J)*QRD(K)
00079 130   ARM(I,JJ)=DUM
00080       DO 140 I=1,NXT
00081       II=I+NXT
00082       DO 140 J=1,NXT
00083       JJ=J+NXT
00084 140   ARM(II,JJ)=-F(I,J)
      C        WRITE QRD,RRD,AND,ARM TO PROBLEM FILE
00085       IDEN=LASTID+1
```

```
00086        NQT=NZ+NYO
00087        WRITE(IPFOUT) IDEN,NQT,(QRD(I),I=1,NQT)
00088        IDEN=IDEN+1
00089        WRITE(IPFOUT) IDEN,MU,(RRD(I),I=1,MU)
00090        IDEN=IDEN+1
00091        NXT2=2*NXT
00092        WRITE(IPFOUT) IDEN,NXT2,NXT2,((ARM(I,J),J=1,NXT2),I=1,NXT2)
     C          COMPUTE OPEN LOOP EIGENVALUES AND WRITE THEM
00093        CALL EIGRF(PX,NXT,NDIM3,0,EVAL,EVEC,INT,SCALE,WR,WI,WK,IERR)
00094        IDEN=IDEN+1
00095        WRITE(IPFOUT) IDEN,NXT,(EVAL(I),I=1,NXT)
     C          COMPUTE CLOSED LOOP EIGENVALUES AND EIGENVECTORS
00096        CALL EIGRF(ARM,2*NXT,NDIM6,1,EVAL,EVEC,INT,SCALE,WR,WI,WK,IERR)
00097        IF(IERR.EQ.0) GO TO 160
00099        WRITE(IOUT,150)
00100 150    FORMAT(5X,'ERROR COMPUTING CLOSED LOOP EIGENSYSTEM')
00101        GO TO 285
     C                (W11) WITH EIGENVECTORS CORRESPONDING TO
     C          FORM (   ) EIGENVALUES WITH POSITIVE REAL
     C                (W21) PARTS
00102 160    J=0
00103        DO 180 K=1,2*NXT
00104        IF(REAL(EVAL(K)).LT.0.0) GO TO 180
00106        IF(REAL(EVAL(K)).EQ.0.0) WRITE(IOUT,165)
00108 165    FORMAT(5X,'REAL PART OF C.L. EIGENVALUE = 0.0')
00109        J=J+1
00110        EVAL(J)=-EVAL(K)
00111        DO 170 I=1,NXT
00112        L=NXT+I
00113        W11(J,I)=EVEC(I,K)
00114        W21(J,I)=EVEC(L,K)
00115 170    EVEC(I,J)=EVEC(I,K)
00116        IF(J.EQ.NXT) GO TO 200
00118 180    CONTINUE
00119        WRITE(IOUT,190)
00120 190    FORMAT(5X,'ERROR IN EIGENVALUE SELECTION')
00121        GO TO 285
     C          COMPLEX SOLUTION COV = W11*W21**-1
00122 200    CALL LEQT2C(W21,NXT,NDIM3,W11,NXT,NDIM3,INT,C,IWK,WA,IERR)
00123        IF(IERR.EQ.0) GO TO 220
00125        WRITE(IOUT,210) IERR
00126 210    FORMAT(5X,I5,'ERROR IN EIGENVECTOR DECOMPOSITION')
00127        GO TO 285
00128 220    CONTINUE
00129        DO 230 I=1,NXT
00130        DO 230 J=1,NXT
00131 230    PX(I,J)=REAL(W11(I,J))
     C          COMPUTE NORMALIZED EIGENVECTORS
00132        DO 260 I=1,NXT
00133        SF=0.
00134        DO 240 J=1,NXT
00135        SFF=CABS(W21(I,J))
00136        IF(SF.GT.SFF) GO TO 240
00138        JMAX=J
```

```
00139          SF=SFF
00140 240      CONTINUE
00141          DD=W21(I,JMAX)
00142          DO 250 J=1,NXT
00143 250      W21(I,J)=W21(I,J)/DD
00144 260      CONTINUE
       C          COMPUTE REGULATOR GAIN MATRIX
00145          DO 280 I=1,NU
00146          DO 280 J=1,NXT
00147          DUM=0.0
00148          DO 270 K=1,NXT
00149 270      DUM=DUM+G(K,I)*PX(K,J)/RRD(I)
00150 280      CG(I,J)=DUM
       C          WRITE CLOSED LOOP EIGENVALUES AND NORMALIZED EIGENVECTORS
       C          AND TRANSITION MATRIX AND REGULATOR GAIN MATRIX
00151          IDEN=IDEN+1
00152          WRITE(IPFOUT) IDEN,NXT,(EVAL(I),I=1,NXT)
00153          IDEN=IDEN+1
00154          WRITE(IPFOUT) IDEN,NXT,NXT,((W21(I,J),J=1,NXT),I=1,NXT)
00155          IDEN=IDEN+1
00156          WRITE(IPFOUT) IDEN,NXT,NXT,((PX(I,J),J=1,NXT),I=1,NXT)
00157          IDEN=IDEN+1
00158          WRITE(IPFOUT) IDEN,NU,NXT,((CG(I,J),J=1,NXT),I=1,NU)
       C          THE END
00159 285      DO 290 I=IDEN+1,130
00160 290      WRITE(IPFOUT) I,I1,I1,C0
00161 1000     STOP
00162          END
```

```
L       00112=  00114
LASTID  00016DA 00043AG 00085
LEQT2C  00122CL
NDIM3   00016DA 00093AG 00122AG
NDIM6   00016DA 00096AG
NQT     00086=  00087WR
NU      00034RD 00051AG 00054DO 00073DO 00089WR 00145DO 00158WR
NW      00034RD
NX      00036RD 00038RD 00042RD
NXC     00034RD 00061
NXG     00034RD 00051
NXT     00051=  00052DO 00063DO 00066DO 00067   00068DO 00073DO 00074DO
        00075   00080DO 00081   00082DO 00083   00091   00093AG 00095WR
        00096AG 00103DO 00111DO 00112   00116IF 00122AG 00129DO 00130DO
        00132DO 00134DO 00142DO 00146DO 00148DO 00152WR 00154WR 00156WR
        00158WR
NXT2    00091=  00092WR
NY      00036RD 00038RD 00040RD
NYO     00034RD 00046AG 00077DO 00086
NYS     00034RD
NZ      00034RD 00046AG 00077DO 00086
PX      00004DI 00012DA 00065=  00093AG 00131=  00149   00158WR
QRD     00002DI 00012DA 00046AG 00078   00087WR
READMX  00046CL 00051CL
REAL    00104IF 00106IF 00131
RETWR   00043CL
RRD     00002DI 00012DA 00051AG 00055IF 00071   00089WR 00149
R0      00017DA
SCALE   00007DI 00010EQ 00012DA 00093AG 00096AG
SF      00133=  00136IF 00139=
SFF     00135=  00136IF 00139
TTLNEW  00001DI 00029RD 00043AG
WA      00009CX 00010EQ 00122AG
WI      00007DI 00012DA 00093AG 00096AG
WK      00007DI 00010EQ 00093AG 00096AG
WR      00007DI 00012DA 00093AG 00096AG
W11     00009CX 00010EQ 00014DA 00113=  00122AG 00131
W21     00005CX 00010EQ 00014DA 00114=  00122AG 00135   00141   00143=
        00154WR
10      00019WR 00020*
100     00027DO 00071*
1000    00161*
110     00066DO 00068DO 00072*
12      00023WR 00024*
120     00077DO 00078*
130     00073DO 00074DO 00079*
14      00027WR 00028*
140     00080DO 00082DO 00084*
150     00099WR 00100*
160     00098GT 00102*
165     00127WR 00108*
170     00111DO 00115*
180     00103DO 00105GT 00118*
190     00119WR 00120*
```

RT-11 INDEX V03 CROSS REFERENCE LISTING 10-APR-79 02:34:32 PAGE 00007

```
20      00021RD 00025RD 00029RD 00030*
200     00117GT 00122*
210     00125WR 00126*
220     00124GT 00128*
230     00129DO 00130DO 00131*
240     00134DO 00137GT 00140*
250     00142DO 00143*
260     00132DO 00144*
270     00148DO 00149*
280     00145DO 00146DO 00150*
285     00059GT 00101GT 00121GT 00127GT 00159*
290     00159DO 00160*
30      00031DO 00041*
40      00032RO 00042*
45      00244WR 00245*
50      00043*  00048GT
55      00049WR 00050*
60      00051*  00053GT
70      00057WR 00058*
80      00054DO 00056GT 00060*
90      00062DO 00063DO 00065*
```

```
00001        SUBROUTINE READMX(IMAX,ARRAY,IERR)
             C     ROUTINE READS MATRIX ELEMENT OF THE FORM:
             C           I,VALUE
             C     CHECKS FOR DIMENSIONS OUT OF RANGE AND FORMS:
             C           ARRAY(I)=VALUE
00002        COMMON/IO/ IIN,IOUT
00003        DIMENSION ARRAY(IMAX)
00004        READ(IIN,10,ERR=50) I,VALUE
00005  10    FORMAT(I,E)
00006        IF(I.GT.IMAX.OR.I.LT.0) GO TO 30
00008        IF(I.EQ.0) GO TO 20
00010        ARRAY(I)=VALUE
00011        IERR=0
00012        RETURN
             C     ERRORS:   IERR=1 - I=0 INDICATING TERMINATION OF DATA FOR
             C                        THIS MATRIX
             C               IERR=2 - DIMENSIONS OUT OF RANGE
             C               IERR=3 - WRONG FORMAT ON INPUT
00013  20    IERR=1
00014        RETURN
00015  30    IERR=2
00016        WRITE(IOUT,40) I,VALUE
00017  40    FORMAT(5X,'DIMENSIONS OUT OF RANGE FOR ',3X,I3,E15.6)
00018        RETURN
00019  50    IERR=3
00020        WRITE(IOUT,60)
00021  60    FORMAT(5X,'WRONG FORMAT')
00022        RETURN
00023        END
```

RT-11 INDEX V03 CROSS REFERENCE LISTING 10-APR-79 02:34:32 PAGE 00002

```
ARRAY   000001AG 00003DI 00010=
I       00004RD 00006IF 00008IF 00010   00010NR
IERR    00001AG 00011=  00013=  00015=  00019=
IIN     00002CM 00004RD
IMAX    00001AG 00003DI 00006IF
IO      00002CN
IOUT    00002CM 00010NR 00020NR
READIX  00031SU
VALUE   00004RD 00010   00010NR
1$      00004RD 00005*
2$      00009GT 00013*
3$      00007GT 00015*
4$      00010NR 00017*
5$      00004RD 00019*
6$      00020NR 00021*
```

```
      C         BLOCK 6 - SENSITIVITY CALCULATIONS
      C          THIS ROUTINE READS THE CLOSED LOOP EIGENVALUES AND
      C          EIGENVECTORS, THE REGULATOR GAINS , THE F AND G
      C          MATRICES FROM THE PROBLEM FILE AND COMPUTES THE
      C          NORMALIZED SENSITIVITIES W.R.T. F, G, AND K FOR
      C          EACH EIGENVALUE.
00001         DIMENSION FNAME(4),TTLNEW(5)
      C          INPUT MATRICES
00002         DIMENSION F(15,15),G(15,5),CG(5,15)
00003         COMPLEX EVAL(15),EVEC(15,15)
      C          OUTPUT MATRICES
00004         COMPLEX SIII(15,15)
      C          WORKING STORAGE
00005         COMPLEX DD,WA(225),W21I(15,15),C(15)
00006         DIMENSION ARRAY(900),IARRAY(900)
00007         DIMENSION IPVT(15)
00008         EQUIVALENCE (ARRAY,IARRAY),(ARRAY,WA),(ARRAY(451),W21I)
      C          INITIALIZE
00009         DATA IPVT /15*0/
00010         DATA F,G,CG /450*0./
00011         DATA EVAL,EVEC,SIII,WA,W21I,C /930*(0.,0.)/
      C          OTHER DATA
00012         DATA IIN,IOUT,IPFIN,IPFOUT /5,7,20,21,/
00013         DATA NDIM3,LASTID,I225,I900 /15,70,225,900/
00014         COMPLEX C0
00015         DATA I1,C0 /1,(0.,0.)/
      C          READ PROBLEM FILE NAME AND NEW TITLE
00016         WRITE(IOUT,10)
00017 10      FORMAT(20X,'SENSITIVITY - BLOCK 6 ',/,2X,
     +          'INPUT PROBLEM FILE NAME ? ',$)
00018         READ(IIN,20) (FNAME(I),I=1,4)
00019         CALL ASSIGN(IPFIN,FNAME)
00020         WRITE(IOUT,12)
00021 12      FORMAT(2X,'OUTPUT PROBLEM FILE NAME ? ',$)
00022         READ(IIN,20) (FNAME(I),I=1,4)
00023         CALL ASSIGN(IPFOUT,FNAME)
00024         WRITE(IOUT,14)
00025 14      FORMAT(2X,'PROBLEM TITLE (20 CHARS) : ',$)
00026         READ(IIN,20) (TTLNEW(I),I=1,5)
00027 20      FORMAT(5A4)
      C          READ IPFIN, WRITE IPFOUT, COMPOSING NEW TITLE
00028         CALL RENR(IPFIN,IPFOUT,TTLNEW,I900,LASTID,ARRAY,IARRAY)
00029         REWIND IPFIN
      C          READ THE FOLLOWING FROM THE PROBLEM FILE:
      C            DIMENSIONS - IDEN=2
      C            F MATRIX   - IDEN=24
      C            G MATRIX   - IDEN=25
      C            CL EVALS   - IDEN=67
      C            CL EVECS   - IDEN=68
      C            REG GAINS  - IDEN=70
00030         DO 30 K=1,LASTID
00031         READ(IPFIN,END=35) IDEN
00032         IF(IDEN.EQ.1) READ(IPFIN) IDEN,NXS,NXC,NU,NW,NZ,NYS,NYO
00034         IF(IDEN.EQ.23) READ(IPFIN) IDEN,NX,NY,((F(I,J),J=1,NY),I=1,NX)
```

```
00036          IF(IDEN.EQ.21) READ(IPFIN) IDEN,NX,NY,((G(I,J),J=1,NY),I=1,NX)
00039          IF(IDEN.EQ.66) READ(IPFIN) IDEN,NX,(EVAL(I),I=1,NX)
00040          IF(IDEN.EQ.67) READ(IPFIN) IDEN,NX,NY,((EVEC(I,J),J=1,NY),I=1,NX)
00042          IF(IDEN.EQ.69) READ(IPFIN) IDEN,NX,NY,((CG(I,J),J=1,NY),I=1,NX)
00044 30       CONTINUE
00045 35       NXT=NXS+NXC
00046          IDEN=LASTID
          C       GET LEFT EIGENVALUES FROM EVEC
          C       RIGHT EIGENVALUES - ROWS OF EVEC
          C       LEFT EIGENVALUES - COLUMNS FOR W21I
00047          DO 40 I=1,NXT
00048 40       W21I(I,I)=1.0
00049          CALL LEQT2C(EVEC,NXT,NDIM3,W21I,NXT,NDIM3,IPVT,C,I225,WA,IERR)
00050          IF(IERR.EQ.0) GO TO 60
00052          WRITE(IOUT,50)
00053 50       FORMAT(5X,'ERROR INVERTING EIGENVECTOR MATRIX')
00054          GO TO 1000
00055 60       CONTINUE
          C       CALCULATE SENSITIVITIES
00056          DO 150 I=1,NXT
00057          AIM=AIMAG(EVAL(I))
00058          IF(AIM) 150,70,90
          C       ZERO OUT COMPLEX ROUND-OFF ERROR
00059 70       CONTINUE
00060          DO 80 J=1,NXT
00061 80       W21I(J,I)=CMPLX(REAL(W21I(J,I)),0.)
00062 90       CONTINUE
          C       NORMALIZED SENSITIVITY W.R.T. F
00063          DO 100 J=1,NXT
00064          DO 100 K=1,NXT
00065 100      SIII(J,K)=W21I(J,I)*EVEC(I,K)*F(J,K)/EVAL(I)
00066          IDEN=IDEN+1
00067          WRITE(IPFOUT) IDEN,NXT,NXT,((SIII(II,JJ),JJ=1,NXT),II=1,NXT)
          C       NORMALIZED SENSITIVITY W.R.T. G
00068          DO 120 J=1,NXT
00069          DO 120 K=1,NU
00070          DD=0.
00071          DO 110 L=1,NXT
00072 110      DD=DD+W21I(J,I)*EVEC(I,L)*CG(K,L)
00073 120      SIII(J,K)=DD*G(J,K)/EVAL(I)
00074          IDEN=IDEN+1
00075          WRITE(IPFOUT) IDEN,NXT,NU,((SIII(II,JJ),JJ=1,NU),II=1,NXT)
          C       NORMALIZED SENSITIVITY W.R.T. K
00076          DO 140 J=1,NU
00077          DO 140 K=1,NXT
00078          DD=0.
00079          DO 130 L=1,NXT
00080 130      DD=DD+G(L,J)*W21I(L,I)*EVEC(I,K)
00081 140      SIII(J,K)=DD*CG(J,K)/EVAL(I)
00082          IDEN=IDEN+1
00083          WRITE(IPFOUT)IDEN,NU,NXT,((SIII(II,JJ),JJ=1,NXT),II=1,NU)
00084 150      CONTINUE
00085          DO 160 I=IDEN+1,130
00086 160      WRITE(IPFOUT) I,II,II,C3
```

```
      C        THE END
00087 1000     STOP
00088          END
```

```
AIM     0057=  0058IF
AIMAG   0057
ARRAY   0006DI 0008EQ 0028AG
ASSIGN  0019CL 0023CL
C       0005CX 0011DA 0049AG
CG      0002DI 0010DA 0043RD 0072   0081
CMPLX   0061
C0      0014CX 0015DA 0089NR
DD      0005CX 0070=  0072=  0073   0078=  0080=  0081
EVAL    0003CX 0011DA 0039RD 0057   0065   0073   0081
EVEC    0003CX 0011DA 0041RD 0049AG 0065   0072   0080
F       0002DI 0016DA 0035RD 0065
FNAME   0001DI 0018RD 0019AG 0022RD 0023AG
G       0002DI 0014DA 0037RD 0073   0080
I       0018RD 0022RD 0026RD 0035RD 0037RD 0039RD 0041RD 0043RD
        0047DO 0048   0056DO 0057   0061   0065   0072   0073
        0080   0081   0085DO 0089NR
IARRAY  0006DI 0008EQ 0028AG
IDEN    0031RD 0032IF 0033RD 0034IF 0035RD 0036IF 0037RD 0038IF
        0039RD 0043IF 0041RD 0042IF 0043RD 0046=  0056=  0067NR
        0074=  0075NR 0082=  0083NR 0085DO
IERR    0049AG 0050IF
II      0067NR 0075NR 0083NR
IIN     0012DA 0018RD 0022RD 0026RD
IOUT    0012DA 0016NR 0020NR 0024NR 0052NR
IPFIN   0012DA 0019AG 0028AG 0029RN 0031RD 0033RD 0035RD 0037RD
        0039RD 0041RD 0043RD
IPFOUT  0012DA 0023AG 0028AG 0067NR 0075NR 0083NR 0089NR
IPVT    0007DI 0009DA 0049AG
I1      0015DA 0089NR
I225    0013DA 0049AG
I900    0013DA 0028AG
J       0035RD 0037RD 0041RD 0043RD 0052DO 0061   0063DO 0065
        0069DO 0072   0073   0076DO 0080   0081
JJ      0067NR 0075NR 0083NR
K       0063DO 0064DO 0065   0069DO 0072   0073   0077DO 0080
        0081
L       0071DO 0072   0079DO 0080
LASTID  0013DA 0028AG 0030DO 0046
LEQT2C  0049CL
NDIM3   0013DA 0049AG
NU      0033RD 0069DO 0075NR 0077DO 0083NR
NN      0033RD
NX      0035RD 0037RD 0039RD 0041RD 0043RD
NXC     0033RD 0045
NXS     0033RD 0045
NXT     0045=  0047DO 0049AG 0056DO 0062DO 0063DO 0064DO 0067NR
        0068DO 0071DO 0075NR 0077DO 0079DO 0083NR
NY      0035RD 0037RD 0041RD 0043RD
NYO     0033RD
NYS     0033RD
NZ      0033RD
REAL    0061
RETR    0028CL
```

```
 RT-11 INDEX V03  CROSS REFERENCE LISTING 18-APR-79 02:43:09 PAGE 00005

SIII     00004CX 00011DA 00065=  00067#R 00073=  00075#R 00081=  00083#R
TTLNEW   00001DI 00026RD 00028AG
WA       00005CX 00008EQ 00011DA 00049AG
W2II     00005CX 00008EQ 00011DA 00048=  00049AG 00061=  00065   00072
         00080
10       00019#R 00017*
100      00063DO 00064DO 00065*
1000     00054GT 00087*
110      00071DO 00072*
12       00023#R 00021*
120      00068DO 00069DO 00073*
130      00079DO 00080*
14       00024#R 00025*
140      00076DO 00077DO 00081*
150      00056DO 00056IF 00084*
160      00085DO 00086*
20       00018RD 00022RD 00026RD 00027*
30       00033DO 00044*
35       00031RD 00045*
40       00047DO 00048*
50       00052#R 00053*
60       00051GT 00055*
70       00058IF 00059*
80       00060DO 00061*
90       00058IF 00052*
```

```
      C          BLOCK 7 - CONTROLLER
      C          THIS ROUTINE SETS UP THE CONTROLLER EQUATIONS USING
      C          THE FILTER AND REGULATOR SOLUTIONS COMPUTED
      C          PREVIOUSLY.
00001       DIMENSION FNAME(4),TTLNEW(5)
      C        INPUT MATRIX ARRAYS
00002       DIMENSION F(15,15),G(15,5),GM(15,15),H(15,15),TZ(15,15),
     +           XK(15,15),C(5,15),TI(15,15),F12(15,15),F21(15,15),
     +           F22(15,15),G2(15,5),H2(15,15),XK2(15,15)
      C        OUTPUT MATRIX ARRAYS
00003       DIMENSION AF(15,15),BF(15,15),CF(5,15),DF(5,15)
      C        WORKING ARRAYS
00004       DIMENSION DUMMY(15,15),DUMMY2(15,15),ARRAY(900),IARRAY(900),
     +           A(30,30)
      C        EQUIVALENCES
00005       EQUIVALENCE (A,ARRAY,IARRAY,XK),(A(16,2),TI),(A(1,16),TZ),
     +           (A(16,23),DUMMY2,H2),(AF,F12),(BF,F21),(CF,G2),
     +           (DUMMY,F22),(XK2,DF)
      C        DATA
00006       DATA IIN,IOUT,IPFIN,IPFOUT /5,7,20,21/
00007       DATA I900,LASTID,I1,R0 /900,115,1,0./
      C        READ INPUT AND OUTPUT PROBLEM FILE NAMES AND TITLE
00008       WRITE(IOUT,10)
00009 10    FORMAT(20X,'CONTROLLER - BLOCK 7',/,2X,
     +     'INPUT PROBLEM FILE NAME ? ',$)
00010       READ(IIN,20) (FNAME(I),I=1,4)
00011       CALL ASSIGN(IPFIN,FNAME)
00012       WRITE(IOUT,12)
00013 12    FORMAT(2X,'OUTPUT PROBLEM FILE NAME ? ',$)
00014       READ(IIN,20) (FNAME(I),I=1,4)
00015       CALL ASSIGN(IPFOUT,FNAME)
00016       WRITE(IOUT,14)
00017 14    FORMAT(2X,'PROBLEM TITLE (20 CHARS) : ',$)
00018       READ(IIN,20) (TTLNEW(I),I=1,5)
00019 20    FORMAT(5A4)
      C        READ AND WRITE PROBLEM FILE, COMPOSING TITLE
00020       CALL REWR(IPFIN,IPFOUT,TTLNEW,I900,LASTID,ARRAY,IARRAY)
00021       REWIND IPFIN
      C        GET THE FOLLOWING FROM THE PROBLEM FILE:
      C          DIMENSIONS  IDEN=2
      C          TZ MATRIX   IDEN=35
      C          TI MATRIX   IDEN=37
      C          F MATRIX    IDEN=38,39,40,41
      C          G MATRIX    IDEN=42,43
      C          H MATRIX    IDEN=47,48
      C          DIMENSIONS  IDEN=54
      C          K MATRIX    IDEN=59,60
      C          C MATRIX    IDEN=70
00022       DO 30 K=1,200
00023       READ(IPFIN,END=40) IDEN
00024       IF(IDEN.EQ.1) READ(IPFIN) IDEN,NXS,NXC,NU,NW,NZ,NYS,NYO
00026       IF(IDEN.EQ.34) READ(IPFIN) IDEN,NX,NY,((TZ(I,J),J=1,NY),I=1,NX)
00028       IF(IDEN.EQ.36) READ(IPFIN) IDEN,NX,NY,((TI(I,J),J=1,NY),I=1,NX)
00030       IF(IDEN.EQ.37) READ(IPFIN) IDEN,NX,NY,((F(I,J),J=1,NY),I=1,NX)
```

```
00032         IF(IDEN.EQ.38) READ(IPFIN) IDEN,NX,NY,((F12(I,J),J=1,NY),I=1,NX)
00034         IF(IDEN.EQ.39) READ(IPFIN) IDEN,NX,NY,((F21(I,J),J=1,NY),I=1,NX)
00036         IF(IDEN.EQ.40) READ(IPFIN) IDEN,NX,NY,((F22(I,J),J=1,NY),I=1,NX)
00038         IF(IDEN.EQ.41) READ(IPFIN) IDEN,NX,NY,((G(I,J),J=1,NY),I=1,NX)
00040         IF(IDEN.EQ.42) READ(IPFIN) IDEN,NX,NY,((G2(I,J),J=1,NY),I=1,NX)
00042         IF(IDEN.EQ.46) READ(IPFIN) IDEN,NX,NY,((H(I,J),J=1,NY),I=1,NX)
00044         IF(IDEN.EQ.47) READ(IPFIN) IDEN,NX,NY,((H2(I,J),J=1,NY),I=1,NX)
00046         IF(IDEN.EQ.53) READ(IPFIN) IDEN,M1,M1P,M2,M2P
00048         IF(IDEN.EQ.58) READ(IPFIN) IDEN,NX,NY,((XK(I,J),J=1,NY),I=1,NX)
00050         IF(IDEN.EQ.59) READ(IPFIN) IDEN,NX,NY,((XK2(I,J),J=1,NY),I=1,NX)
00052         IF(IDEN.EQ.69) READ(IPFIN) IDEN,NCX,NCY,((C(I,J),J=1,NCY),I=1,NCX)
00054 30      CONTINUE
00055 40      REWIND IPFIN
      C          COMPOSE F,G,H AND K
00056         IF(M1P.EQ.0) GO TO 60
00058         IF(M2.EQ.0) GO TO 85
00060         DO 50 I=1,M1P
00061         DO 50 J=1,M2
00062         JJ=J+M1P
00063         F(I,JJ)=F12(I,J)
00064 50      F(JJ,I)=F21(J,I)
00065         DO 55 I=1,M1P
00066         DO 55 J=1,M2
00067         JJ=J+M1
00068 55      XK(I,JJ)=XK2(I,J)
00069 60      DO 70 I=1,M2
00070         II=I+M1P
00071         DO 70 J=1,M2
00072         JJ=J+M1P
00073 70      F(II,JJ)=F22(I,J)
00074         DO 75 I=1,M2
00075         II=I+M1P
00076         DO 75 J=1,NU
00077 75      G(II,J)=G2(I,J)
00078         IF(M1.EQ.0) GO TO 85
00080         DO 80 I=1,M1
00081         DO 80 J=1,M2
00082         JJ=J+M1P
00083 80      H(I,JJ)=H2(I,J)
00084 85      CONTINUE
00085         NV=NYS
00086         NXT=NXC+NXS
00087         IF(M1P.EQ.0) GO TO 440
00089         IF(NCX.NE.1.OR.NCY.NE.1) GO TO 95
00091         DO 90 I=1,NU
00092         DO 90 J=1,NXT
00093 90      C(I,J)=0.
      C          COMPUTE AF = (F11-K12*F21-K11*H11-(G1-K12*G2)*C*TI)
00094 95      DO 110 I=1,M1P
00095         DO 110 J=1,M1P
00096         DUM=0.
00097         IF(M1.EQ.0) GO TO 110
00099         DO 100 K=1,M1
00100 100     DUM=DUM+XK(I,K)*H(K,J)
```

```
00101 110   AF(I,J)=F(I,J)-DUM
00102       IF(M2.EQ.0) GO TO 200
00104 115   DO 130 I=1,M1P
00105       DO 130 J=1,M1P
00106       DUM=0.
00107       DO 120 K=1,M2
00108       KK=K+M1
00109       KF=K+M1P
00110 120   DUM=DUM+XK(I,KK)*F(KF,J)
00111 130   AF(I,J)=AF(I,J)-DUM
00112       IF(NU.EQ.0) GO TO 200
00114       DO 150 I=1,M1P
00115       DO 150 J=1,NU
00116       DUM=0.
00117       DO 140 K=1,M2
00118       KK=K+M1
00119       KG=K+M1P
00120 140   DUM=DUM+XK(I,KK)*G(KG,J)
00121 150   DUMMY(I,J)=G(I,J)-DUM
00122       DO 170 I=1,M1P
00123       DO 170 J=1,NXT
00124       DUM=0.
00125       DO 160 K=1,NU
00126 160   DUM=DUM+DUMMY(I,K)*C(K,J)
00127 170   DUMMY2(I,J)=DUM
00128       DO 190 I=1,M1P
00129       DO 190 J=1,M1P
00130       DUM=0.
00131       DO 180 K=1,NXT
00132 180   DUM=DUM+DUMMY2(I,K)*TI(K,J)
00133 190   AF(I,J)=AF(I,J)-DUM
00134 200   CONTINUE
      C        COMPUTE BF = (K11   (F11-K12*F21-K11*H11)*K12+F12-K12*F22
      C                              -K11*H12-(G1-K12*G2)*C*TI*K12  )*TZ
00135       IF(M1.EQ.0) GO TO 215
00137       DO 210 I=1,M1P
00138       DO 210 J=1,M1
00139 210   BF(I,J)=XK(I,J)
00140       IF(M2.EQ.0) GO TO 380
00142 215   DO 240 I=1,M1P
00143       DO 240 J=1,M1P
00144       DUM=0.
00145       DO 220 K=1,M2
00146       KK=K+M1
00147       KF=K+M1P
00148 220   DUM=DUM+XK(I,KK)*F(KF,J)
00149       DUMMY(I,J)=F(I,J)-DUM
00150       DUM=0.
00151       IF(M1.EQ.0) GO TO 240
00153       DO 230 K=1,M1
00154 230   DUM=DUM+XK(I,K)*H(K,J)
00155 240   DUMMY(I,J)=DUMMY(I,J)-DUM
00156       DO 260 I=1,M1P
00157       DO 260 J=1,M2
```

```
00158          JJ=J+M1
00159          JF=J+M1P
00160          DUM=0.
00161          DO 250 K=1,M1P
00162 250      DUM=DUM+DUMMY(I,K)*XK(K,JJ)
00163 260      BF(I,JJ)=F(I,JF)+DUM
00164          DO 290 I=1,M1P
00165          DO 290 J=1,M2
00166          JJ=J+M1
00167          JF=J+M1P
00168          DUM=0.
00169          DO 270 K=1,M2
00170          KK=K+M1
00171          KF=K+M1P
00172 270      DUM=DUM+XK(I,KK)*F(KF,JF)
00173          BF(I,JJ)=BF(I,JJ)-DUM
00174          DUM=0.
00175          IF(M1.EQ.0) GO TO 290
00177          DO 280 K=1,M1
00178 280      DUM=DUM+XK(I,K)*U(K,JF)
00179 290      BF(I,JJ)=BF(I,JJ)-DUM
00180          IF(NU.EQ.0) GO TO 380
00182          DO 310 I=1,M1P
00183          DO 310 J=1,NU
00184          DUM=0.
00185          DO 300 K=1,M2
00186          KK=K+M1
00187          KG=K+M1P
00188 300      DUM=DUM+XK(I,KK)*G(KG,J)
00189 310      DUMMY(I,J)=G(I,J)-DUM
00190          DO 330 I=1,M1P
00191          DO 330 J=1,NXT
00192          DUM=0.
00193          DO 320 K=1,NU
00194 320      DUM=DUM+DUMMY(I,K)*C(K,J)
00195 330      DUMMY2(I,J)=DUM
00196          DO 350 I=1,M1P
00197          DO 350 J=1,NXT
00198          DUM=0.
00199          DO 340 K=1,NXT
00200 340      DUM=DUM+DUMMY2(I,K)*TI(K,J)
00201 350      DUMMY(I,J)=DUM
00202          DO 355 I=1,NXT
00203          DO 355 J=1,NZ
00204 355      DUMMY2(I,J)=0.
00205          DO 360 I=1,M1P
00206          DO 360 J=1,M2
00207          JJ=J+M1
00208 360      DUMMY2(I,JJ)=XK(I,JJ)
00209          DO 365 I=1,M2
00210          II=I+M1P
00211          III=I+M1
00212 365      DUMMY2(II,III)=1.
00213          DO 375 I=1,M1P
```

```
00214          DO 375 J=1,NZ
00215          DUM=0.
00216          DO 370 K=1,NXT
00217 370      DUM=DUM+DUMMY(I,K)*DUMMY2(K,J)
00218 375      BF(I,J)=BF(I,J)-DUM
00219 380      DO 400 I=1,M1P
00220          DO 400 J=1,NZ
00221          DUM=0.
00222          DO 390 K=1,NZ
00223 390      DUM=DUM+BF(I,K)*TZ(K,J)
00224 400      DUMMY(I,J)=DUM
00225          DO 410 I=1,M1P
00226          DO 410 J=1,NZ
00227 410      BF(I,J)=DUMMY(I,J)
       C          COMPUTE CF = -C*TI
00228          IF(NU.EQ.0) GO TO 510
00230          DO 430 I=1,NU
00231          DO 430 J=1,M1P
00232          DUM=0.
00233          DO 420 K=1,NXT
00234 420      DUM=DUM+C(I,K)*TI(K,J)
00235 430      CF(I,J)=-DUM
       C          COMPUTE DF = -C*TI*K12*TZ
00236 440      DO 460 I=1,NU
00237          DO 460 J=1,NXT
00238          DU1=0.
00239          DO 450 K=1,NXT
00240 450      DU1=DU1+C(I,K)*TI(K,J)
00241 460      DUMMY(I,J)=-DUM
00242          DO 470 I=1,NXT
00243          DO 470 J=1,NZ
00244 470      DUMMY2(I,J)=0.
00245          IF(M2.EQ.0) GO TO 485
00247          IF(M1P.EQ.0) GO TO 482
00249          DO 480 I=1,M1P
00250          DO 480,J=1,M2
00251          JJ=J+M1
00252 480      DUMMY2(I,JJ)=XK(I,JJ)
00253 482      DO 484 I=1,M2
00254          II=I+M1P
00255          IJ=I+M1
00256 484      DUMMY2(II,IJ)=1.
00257 485      DO 487 I=1,NU
00258          DO 487 J=1,NZ
00259          DUM=0.
00260          DO 486 K=1,NXT
00261 486      DUM=DUM+DUMMY(I,K)*DUMMY2(K,J)
00262 487      DF(I,J)=DUM
00263          DO 495 I=1,NU
00264          DO 495 J=1,NZ
00265          DUM=0.
00266          DO 490 K=1,NZ
00267 490      DUM=DU1+DF(I,K)*TZ(K,J)
00268 495      DUMMY(I,J)=DU1
```

```
00269          DO 500 I=1,MU
00270          DO 500 J=1,NZ
00271  500     DF(I,J)=DUMMY(I,J)
       C          WRITE AF,BF,CF,AND DF
00272  510     IDEN=LASTID+1
00273          IF(M1P.EQ.0) WRITE(IPFOUT) IDEN,I1,I1,R0
00275          IF(M1P.NE.0) WRITE(IPFOUT)
              +    IDEN,M1P,M1P,((AF(I,J),J=1,M1P),I=1,M1P)
00277          IDEN=IDEN+1
00278          IF(M1P.EQ.0) WRITE(IPFOUT) IDEN,I1,I1,R0
00280          IF(M1P.NE.0) WRITE(IPFOUT)
              +  IDEN,M1P,NZ,((BF(I,J),J=1,NZ),I=1,M1P)
00282          IDEN=IDEN+1
00283          IF(M1P.EQ.0.OR.MU.EQ.0) WRITE(IPFOUT) IDEN,I1,I1,R0
00285          IF(M1P.NE.0.AND.MU.NE.0) WRITE(IPFOUT)
              +  IDEN,MU,M1P,((CF(I,J),J=1,M1P),I=1,MU)
00287          IDEN=IDEN+1
00288          IF(MU.EQ.0) WRITE(IPFOUT) IDEN,I1,I1,R0
00290          IF(MU.NE.0) WRITE(IPFOUT)
              +  IDEN,MU,NZ,((DF(I,J),J=1,NZ),I=1,MU)
00292          LASTID=IDEN
       C          READ THE FOLLOWING FROM THE PROBLEM FILE:
       C            F MATRIX    IDEN=24
       C            G MATRIX    IDEN=25
       C            G1 MATRIX   IDEN=26
       C            H MATRIX    IDEN=27
00293          DO 520 K=1,200
00294          READ(IPFIN,END=530) IDEN
00295          IF(IDEN.EQ.23) READ(IPFIN) IDEN,NX,NY,((F(I,J),J=1,NY),I=1,NX)
00297          IF(IDEN.EQ.24) READ(IPFIN) IDEN,NX,NY,((G(I,J),J=1,NY),I=1,NX)
00299          IF(IDEN.EQ.25) READ(IPFIN) IDEN,NX,NY,((G1(I,J),J=1,NY),I=1,NX)
00301          IF(IDEN.EQ.26) READ(IPFIN) IDEN,NX,NY,((H(I,J),J=1,NY),I=1,NX)
00303  520     CONTINUE
00304  530     CALL CLOSE(IPFIN)
00305          IDEN=LASTID
       C                          (F+G*DF*H     G*CF)
       C            COMPUTE A =   (                 )
       C                          (  BF*H       AF  )
00306          DO 535 I=1,NXT+M1P
00307          DO 535 J=1,NXT+M1P
00308  535     A(I,J)=0.
00309          DO 540 I=1,NXT
00310          DO 540 J=1,NXT
00311  540     A(I,J)=F(I,J)
00312          IF(MU.EQ.0) GO TO 590
00314          DO 560 I=1,NXT
00315          DO 560 J=1,NZ
00316          DUM=0.
00317          DO 550 K=1,MU
00318  550     DUM=DUM+G(I,K)*DF(K,J)
00319  560     DUMMY(I,J)=DUM
00320          DO 580 I=1,NXT
00321          DO 580 J=1,NXT
00322          DUM=0.
```

```
00323         DO 570 K=1,NZ
00324  570    DUM=DUM+DUMMY(I,K)*H(K,J)
00325  580    A(I,J)=A(I,J)+DUM
00326  590    CONTINUE
00327         IF(M1P.EQ.0) GO TO 660
00329         DO 610 I=1,M1P
00330         II=I+NXT
00331         DO 610 J=1,NXT
00332         DUM=0.
00333         DO 600 K=1,NZ
00334  600    DUM=DUM+BF(I,K)*H(K,J)
00335  610    A(II,J)=DUM
00336         IF(NU.EQ.0) GO TO 640
00338         DO 630 I=1,NXT
00339         DO 630 J=1,M1P
00340         JJ=J+NXT
00341         DUM=0.
00342         DO 620 K=1,NU
00343  620    DUM=DUM+G(I,K)*CF(K,J)
00344  630    A(I,JJ)=DUM
00345  640    DO 650 I=1,M1P
00346         II=I+NXT
00347         DO 650 J=1,M1P
00348         JJ=J+NXT
00349  650    A(II,JJ)=AF(I,J)
00350  660    IDEN=IDEN+1
00351         NTOT=NXT+M1P
00352         WRITE(IPFOUT) IDEN,NTOT,NTOT,((A(I,J),J=1,NTOT),I=1,NTOT)
00353         DO 700 I=IDEN+1,130
00354  700    WRITE(IPFOUT) I,II,II,R0
      C          THE END
00355         STOP
00356         END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
            000186=  000188
LASTID      000007DA 0002JAG 000272   000292=  000305
M1          000047RD 000067  000078IF 000083DO 000097IF 000099DO 000103  000110
            000135IF 000139DO 000146  000151IF 000153DO 000159  000166  000170
            000175IF 000177DO 000186  000207  000211  000231  000255
M1P         000047RD 000056IF 000063DO 000062  000065DO 000070  000072  000075
            000082   000087IF 000094DO 000095DO 000104DO 000105DO 000109  000114DO
            000119   000122DO 000126DO 000129DO 000137DO 000142DO 000143DO 000147
            000150DO 000159   000161DO 000164DO 000167   000171   000182DO 000187
            000190DO 000196DO 000205DO 000210   000213DO 000219DO 000225DO 000231DO
            000247IF 000249DO 000254   000273IF 000275IF 000279NR 000281F 000286IF
            000288NR 000283IF 000285IF 000286NR 000306DO 000307DO 000327IF 000329DO
            000339DO 000345DO 000347DO 000351
M2          000047RD 000058IF 000061DO 000066DO 000069DO 000071DO 000074DO 000081DO
            000102IF 000107DO 000117DO 000140IF 000145DO 000157DO 000165DO 000169DO
            000185DO 000206DO 000209DO 000245IF 000250DO 000253DO
M2P         000047RD
NCX         000053RD 000089IF
NCY         000053RD 000089IF
NTOT        000351=  000352NR
NU          000025RD 000076DO 000091DO 000112IF 000115DO 000125DO 000180IF 000183DO
            000193DO 000220IF 000230DO 000236DO 000257DO 000263DO 000269DO 000283IF
            000285IF 000286NR 000288IF 000290IF 000291NR 000312IF 000317DO 000336IF
            000342DO
NV          000025RD 000085=
NX          000027RD 000029RD 000031RD 000033RD 000035RD 000037RD 000039RD 000041RD
            000043RD 000045RD 000049RD 000051RD 000296RD 000298RD 000300RD 000302RD
NXC         000025RD 000086
NXS         000025RD 000086
NXT         000095=  000092DO 000123DO 000131DO 000191DO 000197DO 000199DO 000202DO
            000216DO 000233DO 000237DO 000239DO 000242DO 000260DO 000306DO 000307DO
            000309DO 000310DO 000314DO 000320DO 000321DO 000324   000331DO 000339DO
            000340   000346   000348   000351
NY          000027RD 000029RD 000031RD 000033RD 000035RD 000037RD 000039RD 000041RD
            000043RD 000045RD 000049RD 000051RD 000295RD 000298RD 000304RD 000302RD
NYC         000025RD
NYS         000025RD 000085
NZ          000025RD 000203DO 000214DO 000220DO 000222DO 000228DO 000243DO 000250DO
            000264DO 000266DO 000270DO 000281NR 000291NR 000315DO 000323DO 000333DO
RENR        000202CL
R0          000047DA 000274NR 000279NR 000294NR 000289NR 000354NR
TI          000042DI 000035EQ 000029RD 000132   000200   000234   000240
TTLNEW      000041DI 000019RD 0002JAG
TZ          000042DI 000035EQ 000027RD 000223   000267
XR          000042DI 000035EQ 000049RD 000050=  000120   000110   000120   000139
            000148   000154   000162   000172   000178   000108   000208   000252
XR2         000042DI 000035EQ 000051RD 000063
10          000038NR 000039*
100         000099DO 000100*
110         000094DO 000095DO 000098GT 000101*
115         000104*
12          000112NR 000113*
120         000107DO 000110*
130         000104DO 000105DO 000111*
```

```
14      00019#R 00017*
140     00117DO 00120*
150     00114DO 00115DO 00121*
160     00125DO 00126*
170     00122DO 00123DO 00127*
180     00131DO 00132*
190     00128DO 00129DO 00133*
20      00013RD 00014RD 00018RD 00019*
200     00103GT 00113GT 00134*
210     00137DO 00138DO 00139*
215     00139GT 00142*
220     00145DO 00148*
230     00153DO 00154*
240     00142DO 00143DO 00152GT 00155*
250     00161DO 00162*
260     00156DO 00157DO 00163*
270     00169DO 00172*
280     00177DO 00178*
290     00164DO 00165DO 00179GT 00179*
30      00022DO 00454*
300     00185DO 00188*
310     00182DO 00183DO 00189*
320     00193DO 00194*
330     00196DO 00191DO 00195*
340     00199DO 00200*
350     00196DO 00197DO 00201*
355     00202DO 00203DO 00204*
360     00205DO 00206DO 00208*
365     00209DO 00212*
370     00216DO 00217*
375     00213DO 00214DO 00218*
380     00141GT 00181GT 00219*
390     00222DO 00223*
40      00023RD 00455*
400     00219DO 00220DO 00224*
410     00225DO 00226DO 00227*
420     00233DO 00234*
430     00230DO 00231DO 00235*
440     00088GT 00236*
450     00239DO 00240*
460     00236DO 00237DO 00241*
470     00242DO 00243DO 00244*
480     00249DO 00250DO 00252*
482     00240GT 00253*
484     00253DO 00256*
485     00246GT 00257*
486     00260DO 00261*
487     00257DO 00258DO 00262*
490     00264DO 00267*
495     00263DO 00264DO 00268*
50      00056DO 00051DO 00054*
500     00269DO 00270DO 00271*
510     00229GT 00272*
520     00293DO 00303*
```

```
530     003294RD 003304*
535     003305DO 003307DO 003308*
540     003309DO 003310DO 003311*
55      003065DO 003066DO 003068*
550     003317DO 003318*
560     003314DO 003315DO 003319*
570     003323DO 003324*
580     003320DO 003321DO 003325*
590     003313GT 003326*
60      003057GT 003269*
600     003333DO 003334*
610     003329DO 003331DO 003335*
620     003342DO 003343*
630     003339DO 003339DO 003344*
640     003337GT 003345*
650     003345DO 003347DO 003349*
660     003228GT 003350*
70      003069DO 003071DO 003073*
700     003353DO 003354*
75      003074DO 003076DO 003077*
80      003080DO 003081DO 003083*
85      003059GT 003079GT 003084*
90      003091DO 003092DO 003093*
95      003090GT 003094*
```

```
      C         BLOCK 8 - CONTROLLER ANALYSIS
      C           THIS ROUTINE SETS UP THE TREN FILE TO ALLOW TRANSFER
      C           FUNCTION ANALYSIS OF THE CONTROLLER
00001         DIMENSION FNAME(4),TITLE(15)
      C         INPUT MATRIX ARRAYS
00002         REAL MNU(5),MNZ(15),MNXT(15)
00003         DIMENSION F(15,15),G(15,5),H(15,15),TI(15,15),XK(15,15),C(5,15)
00004         DIMENSION F12(15,15),F21(15,15),F22(15,15),G2(15,5),H2(15,15),
     +                  XK2(15,15)
      C         WORKING SPACE
00005         REAL DUMMY(15,15),DUMMY2(15,15),MN(20)
00006         EQUIVALENCE (F21,DUMMY),(F12,DUMMY2)
      C         DATA
00007         DATA IIN,IOUT,IPFIN,ITREN /5,7,20,21/
00008         DATA A1,A2,K0,IPLUS,X1 /0.,0.,0,'+',1./
      C         READ FILE NAMES AND TITLE
00009         WRITE(IOUT,10)
00010 10      FORMAT(20X,'CONTROLLER DEFINITION - BLOCK 8',/,2X,
     +         'INPUT PROBLEM FILE NAME ? ',$)
00011         READ(IIN,20) (FNAME(I),I=1,4)
00012         CALL ASSIGN(IPFIN,FNAME)
00013         WRITE(IOUT,12)
00014 12      FORMAT(2X,'OUTPUT TREN FILE NAME ? ',$)
00015         READ(IIN,20) (FNAME(I),I=1,4)
00016         CALL ASSIGN(ITREN,FNAME)
00017         WRITE(IOUT,14)
00018 14      FORMAT(2X,'TREN TITLE (60 CHARS) :')
00019         READ(IIN,20) (TITLE(I),I=1,15)
00020 20      FORMAT(15A4)
      C         READ THE FOLLOWING FROM THE PROBLEM FILE:
      C           DIMENSIONS  IDEN=2
      C           U MNEMONICS IDEN=3
      C           Z MNEMONICS IDEN=33
      C           X MNEMONICS IDEN=34
      C           TI MATRIX   IDEN=37
      C           F MATRIX    IDEN=38,39,40,41
      C           G MATRIX    IDEN=42,43
      C           H MATRIX    IDEN=47,48
      C           DIMENSIONS  IDEN=54
      C           K MATRIX    IDEN=59,60
      C           C MATRIX    IDEN=70
00021         DO 30 K=1,200
00022         READ(IPFIN,END=40) IDEN
00023         IF(IDEN.EQ.1) READ(IPFIN) IDEN,NXS,NXC,NU,NW,NZ,NYS,NYO
00025         IF(IDEN.EQ.2) READ(IPFIN) IDEN,NX,(MNU(I),I=1,NX)
00027         IF(IDEN.EQ.32) READ(IPFIN) IDEN,NX,(MNZ(I),I=1,NX)
00029         IF(IDEN.EQ.33) READ(IPFIN) IDEN,NX,(MNXT(I),I=1,NX)
00031         IF(IDEN.EQ.36) READ(IPFIN) IDEN,NX,NY,((TI(I,J),J=1,NY),I=1,NX)
00033         IF(IDEN.EQ.37) READ(IPFIN) IDEN,NX,NY,((F(I,J),J=1,NY),I=1,NX)
00035         IF(IDEN.EQ.38) READ(IPFIN) IDEN,NX,NY,((F12(I,J),J=1,NY),I=1,NX)
00037         IF(IDEN.EQ.39) READ(IPFIN) IDEN,NX,NY,((F21(I,J),J=1,NY),I=1,NX)
00039         IF(IDEN.EQ.40) READ(IPFIN) IDEN,NX,NY,((F22(I,J),J=1,NY),I=1,NX)
00041         IF(IDEN.EQ.41) READ(IPFIN) IDEN,NX,NY,((G(I,J),J=1,NY),I=1,NX)
00043         IF(IDEN.EQ.42) READ(IPFIN) IDEN,NX,NY,((G2(I,J),J=1,NY),I=1,NX)
```

```
00045        IF(IDEN.EQ.46) READ(IPFIN) IDEN,NX,NY,((H(I,J),J=1,NY),I=1,NX)
00047        IF(IDEN.EQ.47) READ(IPFIN) IDEN,NX,NY,((H2(I,J),J=1,NY),I=1,NX)
00049        IF(IDEN.EQ.53) READ(IPFIN) IDEN,M1,M1P,M2,M2P
00051        IF(IDEN.EQ.58) READ(IPFIN) IDEN,NX,NY,((XK(I,J),J=1,NY),I=1,NX)
00053        IF(IDEN.EQ.59) READ(IPFIN) IDEN,NX,NY,((XK2(I,J),J=1,NY),I=1,NX)
00055        IF(IDEN.EQ.69) READ(IPFIN) IDEN,NX,NY,((C(I,J),J=1,NY),I=1,NX)
00057 30     CONTINUE
00058 40     NXT=NXC+NXS
00059        CALL CLOSE(IPFIN)
      C        COMPOSE F,G,H AND XK MATRICES
00060        IF(M1P.EQ.0) GO TO 50
00062        IF(M2.EQ.0) GO TO 85
00064        DO 50 I=1,M1P
00065        DO 50 J=1,M2
00066        JJ=J+M1P
00067        F(I,JJ)=F12(I,J)
00068 50     F(JJ,I)=F21(J,I)
00069 60     DO 70 I=1,M2
00070        II=I+M1P
00071        DO 70 J=1,M2
00072        JJ=J+M1P
00073 70     F(II,JJ)=F22(I,J)
00074        DO 75 I=1,M2
00075        II=I+M1P
00076        DO 75 J=1,NU
00077 75     G(II,J)=G2(I,J)
00078        DO 80 I=1,M1P
00079        DO 80 J=1,M2
00080        JJ=J+M1
00081        H(I,JJ)=H2(I,J)
00082 80     XK(I,JJ)=XK2(I,J)
00083 85     CONTINUE
00084        IF(M1P.EQ.0.AND.NU.NE.0) GO TO 185
00086        IF(M1P.EQ.0.AND.NU.EQ.0) GO TO 195
      C                    (SI-F11+K12*F21+K11*H11      -G1+K12*G2)
      C        WRITE AF = (                                        )
      C                    (          C*TI                    I    )
00088        DO 130 I=1,M1P
00089        DO 130 J=1,M1P
00090        DUM=0.
00091        IF(M1.EQ.0) GO TO 95
00093        DO 90 K=1,M1
00094 90     DUM=DUM+XK(I,K)*H(K,J)
00095 95     DUMMY(I,J)=-F(I,J)+DUM
00096        IF(M2.EQ.0) GO TO 110
00098        DUM=0.
00099        DO 100 K=1,M2
00100        KK=K+M1
00101        KF=K+M1P
00102 100    DUM=DUM+XK(I,KK)*F(KF,J)
00103        DUMMY(I,J)=DUMMY(I,J)+DUM
00104 110    A1=0.
00105        IF(I.EQ.J) A1=1.0
00107        IF(A1.EQ.0..AND.DUMMY(I,J).EQ.0.) GO TO 130
```

```
00109          WRITE(ITRFN,120) I,J,A2,A1,DUMMY(I,J),K0
00110 120      FORMAT(2I,3F,I)
00111 130      CONTINUE
00112          A1=0.
00113          IF(NU.EQ.0) GO TO 195
00115          DO 150 I=1,NU
00116          II=I+M1P
00117          DO 150 J=1,M1P
00118          DUM=0.
00119          DO 140 K=1,NXT
00120 140      DUM=DUM+C(I,K)*TI(K,J)
00121          IF(DUM.EQ.0.) GO TO 150
00123          WRITE(ITRFN,120) II,J,A2,A1,DUM,K0
00124 150      CONTINUE
00125          DO 180 I=1,M1P
00126          DO 180 J=1,NU
00127          JJ=J+M1P
00128          DUM=0.
00129          IF(M2.EQ.0) GO TO 170
00131          DO 160 K=1,M2
00132          KK=K+M1
00133          KG=K+M1P
00134 160      DUM=DUM+XK(I,KK)*G(KG,J)
00135 170      DUMMY(I,J)=-G(I,J)+DUM
00136          IF(DUMMY(I,J).EQ.0.) GO TO 180
00138          WRITE(ITRFN,120) I,JJ,A2,A1,DUMMY(I,J),K0
00139 180      CONTINUE
00140 195      DO 190 I=1,NU
00141          II=I+M1P
00142          DO 190 J=1,NU
00143          JJ=J+M1P
00144          IF(I.NE.J) GO TO 190
00146          WRITE(ITRFN,120) II,JJ,A2,A1,X1,K0
00147 190      CONTINUE
00148 195      WRITE(ITRFN,120) K0
               C                    (K11    (F11-K12*F21-K11*H11)*K12+F12-K12*F22-K11*H12)
               C       WRITE BF = (                                                        )
               C                    ( 0                    C*TI*K12                     )
00149          IF(M1.EQ.0) GO TO 205
00151          IF(M1P.EQ.0) GO TO 275
00153          DO 200 I=1,M1P
00154          DO 200 J=1,M1
00155          IF(XK(I,J).EQ.0.) GO TO 200
00157          WRITE(ITRFN,120) I,J,A2,A1,XK(I,J),K0
00158 200      CONTINUE
00159          IF(M2.EQ.0) GO TO 320
00161 205      DO 230 I=1,M1P
00162          DO 230 J=1,M1P
00163          DUM=0.
00164          DO 210 K=1,M2
00165          KK=K+M1
00166          KF=K+M1P
00167 210      DUM=DUM+XK(I,KK)*F(KF,J)
00168          DUMMY(I,J)=F(I,J)-DUM
```

```
00169          DUM=0.
00170          IF(M1.EQ.0) GO TO 230
00172          DO 220 K=1,M1
00173 220      DUM=DUM+XK(I,K)*H(K,J)
00174 230      DUMMY(I,J)=DUMMY(I,J)-DUM
00175          DO 245 I=1,M1P
00176          DO 245 J=1,M2
00177          JJ=J+M1
00178          JF=J+M1P
00179          DUM=0.
00180          DO 240 K=1,M1P
00181 240      DUM=DUM+DUMMY(I,K)*XK(K,JJ)
00182 245      DUMMY2(I,J)=F(I,JF)+DUM
00183          DO 265 I=1,M1P
00184          DO 265 J=1,M2
00185          JJ=J+M1
00186          JF=J+M1P
00187          DUM=0.
00188          DO 250 K=1,M2
00189          KK=K+M1
00190          KF=K+M1P
00191 250      DUM=DUM+XK(I,KK)*F(KF,JF)
00192          DUMMY2(I,J)=DUMMY2(I,J)-DUM
00193          DUM=0.
00194          IF(M1.EQ.0) GO TO 265
00196          DO 260 K=1,M1
00197 260      DUM=DUM+XK(I,K)*H(K,JF)
00198 265      DUMMY2(I,J)=DUMMY2(I,J)-DUM
00199          DO 270 I=1,M1P
00200          DO 270 J=1,M2
00201          JJ=J+M1
00202          IF(DUMMY2(I,J).EQ.0.) GO TO 270
00204          WRITE(ITRFN,120) I,JJ,M2,M1,DUMMY2(I,J),K0
00205 270      CONTINUE
00206 275      CONTINUE
00207          IF(NU.EQ.0) GO TO 320
00209          DO 290 I=1,NU
00210          DO 290 J=1,NXT
00211          DUM=0.
00212          DO 280 K=1,NXT
00213 280      DUM=DUM-C(I,K)*TI(K,J)
00214 290      DUMMY(I,J)=DUM
00215          DO 292 I=1,NXT
00216          DO 292 J=1,M2
00217 292      DUMMY2(I,J)=0.
00218          IF(M1P.EQ.0) GO TO 295
00220          DO 294 I=1,M1P
00221          DO 294 J=1,M2
00222          JJ=J+M1
00223 294      DUMMY2(I,J)=XK(I,JJ)
00224 295      DO 296 I=1,M2
00225          II=I+M1P
00226 296      DUMMY2(II,I)=1.
00227          DO 310 I=1,NU
```

```
00228          II=I+M1P
00229          DO 310 J=1,M2
00230          JJ=J+M1
00231          DUM=0.
00232          DO 300 K=1,NXT
00233          DUM=DUM+DUMMY(I,K)*DUMMY2(K,J)
00234 300      CONTINUE
00235          IF(DUM.EQ.0.) GO TO 310
00237          WRITE(ITRFN,120) II,JJ,A2,A1,DUM,K0
00238 310      CONTINUE
00239 320      WRITE(ITRFN,120) K0
       C          WRITE TITLE,DIMENSIONS AND MATRIX CODES
00240          WRITE(ITRFN,20) (TITLE(I),I=1,15)
00241          NX=M1P+NU
00242          NY=M1+M2
00243          WRITE(ITRFN,330) NX,NY,K0
00244 330      FORMAT(3I)
00245          IF(M1P.EQ.0) GO TO 345
00247          DO 340 I=1,M1P
00248 340      MN(I)=MNXT(I)
00249 345      CONTINUE
00250          IF(NU.EQ.0) GO TO 355
00252          DO 350 I=1,NU
00253          II=I+M1P
00254 350      MN(II)=MMU(I)
00255 355      CONTINUE
00256          IF(NX.LE.18) GO TO 370
00258          WRITE(ITRFN,360) (MN(I),I=1,18),IPLUS
00259 360      FORMAT(17A4,A3,A1)
00260          ISTART=19
00261          GO TO 380
00262 370      ISTART=1
00263 380      WRITE(ITRFN,360) (MN(I),I=ISTART,NX)
00264          WRITE(ITRFN,360) (MMZ(I),I=1,NY)
       C          THE END
00265          STOP
00266          END
```

```
         00048RD 00052RD 00054RD 00056RD 00065DO 00066   00067   00068
         00071DO 00072   00073   00076DO 00077   00079DO 00080   00081
         00082   00089DO 00094   00095   00102   00103   00105IF 00107IF
         00109WR 00117DO 00120   00123WR 00126DO 00127   00134   00135
         00136IF 00138WR 00142DO 00143   00144IF 00154DO 00155IF 00157WR
         00162DO 00167   00168   00173   00174   00176DO 00177   00178
         00182   00184DO 00185   00186   00192   00198   00200DO 00201
         00202IF 00204WR 00210DO 00213   00214   00216DO 00217   00221DO
         00222   00223   00229DO 00230   00233
JF       00178=  00182   00186=  00191   00197
JJ       00066=  00067   00068   00072=  00073   00080=  00081   00082
         00127=  00138WR 00143=  00143WR 00177=  00181   00185=  00201=
         00204WR 00222=  00223   00230=  00237WR
K        00021DO 00093DO 00094   00099DO 00100   00101   00119DO 00120
         00131DO 00132   00133   00164DO 00165   00166   00172DO 00173
         00180DO 00181   00188DO 00189   00190   00196DO 00197   00212DO
         00213   00232DO 00233
KF       00101=  00102   00166=  00167   00190=  00191
KG       00133=  00134
KK       00100=  00102   00132=  00134   00165=  00167   00189=  00191
KU       00008DA 00109WR 00123WR 00138WR 00146WR 00148WR 00157WR 00204WR
         00237WR 00239WR 00243WR
MN       00005RL 00248=  00254=  00258WR 00263WR
MNU      00002RL 00026RD 00254
MNXT     00042RL 00030RD 00248
MNZ      00002RL 00028RD 00254WR
MI       00050RD 00030   00091IF 00093DO 00100   00132   00149IF 00154DO
         00165   00170IF 00172DO 00177   00185   00189   00194IF 00196DO
         00201   00222   00230   00242
MIP      00050RD 00060IF 00064DO 00066   00070   00072   00075   00073DO
         00084IF 00086IF 00088DO 00089DO 00101   00116   00117DO 00125DO
         00127   00133   00141   00143   00151IF 00153DO 00161DO 00162DO
         00166   00175DO 00178   00180DO 00183DO 00186   00190   00199DO
         00218IF 00220DO 00225   00228   00241   00245IF 00247DO 00253
M2       00050RD 00062IF 00065DO 00069DO 00071DO 00074DO 00079DO 00096IF
         00099DO 00129IF 00131DO 00159IF 00164DO 00176DO 00184DO 00188DO
         00220DO 00216DO 00221DO 00224DO 00229DO 00242
M2P      00050RD
MU       00024RD 00076DO 00084IF 00086IF 00113IF 00115DO 00126DO 00140DO
         00142DO 00207IF 00209DO 00227DO 00241   00250IF 00252DO
MM       00024RD
MX       00026RD 00028RD 00030RD 00032RD 00034RD 00036RD 00038RD 00040RD
         00042RD 00044RD 00046RD 00048RD 00052RD 00054RD 00056RD 00241=
         00243WR 00255IF 00263WR
MXC      00024RD 00058
MXS      00024RD 00058
MXT      00058=  00119DO 00210DO 00212DO 00215DO 00232DO
MY       00032RD 00034RD 00036RD 00038RD 00040RD 00042RD 00044RD 00046RD
         00048RD 00052RD 00054RD 00056RD 00242=  00243WR 00264WR
MYO      00024RD
MYS      00024RD
MZ       00024RD
TI       00003DI 00032RD 00120   00213
TITLE    00001DI 00019RD 00242WR
```

```
RT-11 INDEX V03  CROSS REFERENCE LISTING 10-APR-79 03:02:02 PAGE 00008

XK      000003DI 000052RD 000082=  000094   000102   000134   000155IF 000157NR
        000167   000173   000181   000191   000197   000223
XK2     000004DI 000054RD 000082
X1      000038DA 000143NR
10      000039NR 000310*
100     000099DO 000102*
110     000097GT 000104*
12      000113NR 000014*
120     000139NR 000110*  000123NR 000138NR 000146NR 000148NR 000157NR 000204WR
        000237NR 000239NR
130     000088DO 000089DO 000108GT 000111*
14      000117NR 000118*
140     000119DO 000120*
150     000115DO 000117DO 000122GT 000124*
160     000131DO 000134*
170     000133GT 000135*
180     000125DO 000126DO 000137GT 000139*
185     000085GT 000140*
190     000143DO 000142DO 000145GT 000147*
195     000087GT 000114GT 000148*
20      000011RD 000015RD 000019RD 000020*  000240NR
200     000153DO 000154DO 000156GT 000158*
205     000156GT 000161*
210     000164DO 000167*
220     000172DO 000173*
230     000161DO 000162DO 000171GT 000174*
240     000180DO 000181*
245     000175DO 000176DO 000182*
250     000188DO 000191*
260     000196DO 000197*
265     000183DO 000184DO 000195GT 000198*
270     000199DO 000200DO 000203GT 000205*
275     000152GT 000206*
280     000212DO 000213*
290     000209DO 000210DO 000214*
292     000215DO 000216DO 000217*
294     000222DO 000221DO 000223*
295     000219GT 000224*
296     000224DO 000226*
30      000021DO 000057*
300     000232DO 000234*
310     000227DO 000229DO 000236GT 000238*
320     000216GT 000200GT 000239*
330     000243NR 000244*
340     000247DO 000240*
345     000246GT 000249*
350     000252DO 000254*
355     000251GT 000255*
360     000258NR 000259*  000263NR 000264NR
370     000257GT 000262*
380     000261GT 000263*
40      000022RD 000058*
50      000064DO 000065DO 000068*
60      000061GT 000069*
```

RT-11 INDEX V03 CROSS REFERENCE LISTING 10-APR-79 03:02:02 PAGE 00009

```
70      000069DO 000071DO 000073*
75      000074DO 000076DO 000077*
80      000078DO 000079DO 000082*
85      000063GT 000083*
90      000093DO 000094*
95      000092GT 000095*
```

```
      C         BLOCK 9 - CLOSED LOOP SYSTEM ANALYSIS
      C           COMPUTES TREN FILE FROM PROBLEM FILE AND IM INPUT
      C           FOR CLOSED LOOP SYSTEM TRANSFER ANALYSIS.
00001         DIMENSION FNAME(4),TITLE(15)
      C           INPUT MATRIX ARRAYS
00002         REAL MNE(15),MNTS(5),MNV(15),MNTOL(5),MNYO(15),MNZ(15),
     +        MNXC(15),MNXT(15)
00003         DIMENSION GMC(15,15),CC(15,5),CC(15,5),FC(15,15),IM(15,15),
     +                  FM(15,5),FD(15,15),E1(30,5),AF(15,15),BF(15,15),
     +                  CF(5,15),DF(5,15),XIM(15)
      C           WORKING SPACE
00004         REAL MSN(55),MN4(45)
00005         DIMENSION DUMMY(15,15),DUMMY2(15,15)
      C           DATA
00006         DATA IIN,IOUT,IPFIN,ITREN /5,7,20,21/
00007         DATA A2,A1,K0,X1,IP /0.,0.,0,1.,'+'/
      C           READ FILENAMES, TITLE AND IM MATRIX
00008         WRITE(IOUT,10)
00009 10      FORMAT(1X,'CLOSED LOOP SYSTEM DEFINITION - BLOCK 9',/,2X,
     +        'INPUT PROBLEM FILE NAME ? ',$)
00010         READ(IIN,20) (FNAME(I),I=1,4)
00011         CALL ASSIGN(IPFIN,FNAME)
00012         WRITE(IOUT,12)
00013 12      FORMAT(2X,'OUTPUT TREN FILE NAME ? ',$)
00014         READ(IIN,20) (FNAME(I),I=1,4)
00015         CALL ASSIGN(ITREN,FNAME)
00016         WRITE(IOUT,14)
00017 14      FORMAT(2X,'TREN TITLE (60 CHARS) : ')
00018         READ(IIN,20) (TITLE(I),I=1,15)
00019 20      FORMAT(15A4)
00020         WRITE(IOUT,25)
00021 25      FORMAT(2X,'IM DIAGONAL ELEMENTS (NZ-OF-THEM) : ')
00022         READ(IIN,30) (XIM(I),I=1,15)
00023 30      FORMAT(15E)
      C           READ THE FOLLOWING FROM THE PROBLEM FILE:
      C             DIMENSIONS  IDEN=2
      C             N MNEMONICS IDEN=4
      C            TS MNEMONICS IDEN=5
      C             V MNEMONICS IDEN=6
      C           TOL MNEMONICS IDEN=7
      C            YO MNEMONICS IDEN=9
      C             Z MNEMONICS IDEN=10
      C            XC MNEMONICS IDEN=12
      C             G/C MATRIX  IDEN=16
      C             CC MATRIX   IDEN=17
      C             CC MATRIX   IDEN=18
      C             FC MATRIX   IDEN=19
      C             IM MATRIX   IDEN=20
      C             FM MATRIX   IDEN=21
      C             FD MATRIX   IDEN=22
      C             E1 MATRIX   IDEN=29
      C            XT MNEMONICS IDEN=34
      C             DIMENSIONS  IDEN=54
      C             AF MATRIX   IDEN=116
```

```
      C          BF MATRIX   IDEN=117
      C          CF MATRIX   IDEN=118
      C          DF MATRIX   IDEN=119
00024          DO 40 K=1,200
00025          READ(IPFIN,END=45) IDEN
00026          IF(IDEN.EQ.1) READ(IPFIN) IDEN,NXS,NXC,NU,NV,NZ,NYS,NYO
00028          IF(IDEN.EQ.3) READ(IPFIN) IDEN,NX,(MNN(I),I=1,NX)
00030          IF(IDEN.EQ.4) READ(IPFIN) IDEN,NX,(MNS(I),I=1,NX)
00032          IF(IDEN.EQ.5) READ(IPFIN) IDEN,NX,(MNV(I),I=1,NX)
00034          IF(IDEN.EQ.6) READ(IPFIN) IDEN,NX,(MNPOL(I),I=1,NX)
00036          IF(IDEN.EQ.8) READ(IPFIN) IDEN,NX,(MNYO(I),I=1,NX)
00038          IF(IDEN.EQ.9) READ(IPFIN) IDEN,NX,(MNZ(I),I=1,NX)
00040          IF(IDEN.EQ.11) READ(IPFIN) IDEN,NX,(MNXC(I),I=1,NX)
00042          IF(IDEN.EQ.15) READ(IPFIN) IDEN,NX,NY,((GMC(I,J),J=1,NY),I=1,NX)
00044          IF(IDEN.EQ.16) READ(IPFIN) IDEN,NX,NY,((GC(I,J),J=1,NY),I=1,NX)
00046          IF(IDEN.EQ.17) READ(IPFIN) IDEN,NX,NY,((CC(I,J),J=1,NY),I=1,NX)
00048          IF(IDEN.EQ.18) READ(IPFIN) IDEN,NX,NY,((FC(I,J),J=1,NY),I=1,NX)
00050          IF(IDEN.EQ.19) READ(IPFIN) IDEN,NX,NY,((HM(I,J),J=1,NY),I=1,NX)
00052          IF(IDEN.EQ.20) READ(IPFIN) IDEN,NX,NY,((FM(I,J),J=1,NY),I=1,NX)
00054          IF(IDEN.EQ.21) READ(IPFIN) IDEN,NX,NY,((HO(I,J),J=1,NY),I=1,NX)
00056          IF(IDEN.EQ.22) READ(IPFIN) IDEN,NX,NY,((EI(I,J),J=1,NY),I=1,NX)
00058          IF(IDEN.EQ.33) READ(IPFIN) IDEN,NX,(MNXT(I),I=1,NX)
00060          IF(IDEN.EQ.53) READ(IPFIN) IDEN,M1,M1P,M2,M2P
00062          IF(IDEN.EQ.115) READ(IPFIN) IDEN,NX,NY,((AF(I,J),J=1,NY),I=1,NX)
00064          IF(IDEN.EQ.116) READ(IPFIN) IDEN,NX,NY,((BF(I,J),J=1,NY),I=1,NX)
00066          IF(IDEN.EQ.117) READ(IPFIN) IDEN,NX,NY,((CF(I,J),J=1,NY),I=1,NX)
00068          IF(IDEN.EQ.118) READ(IPFIN) IDEN,NX,NY,((DF(I,J),J=1,NY),I=1,NX)
00070 40     CONTINUE
00071 45     NN=NX-NXC
00072          CALL CLOSE(IPFIN)
      C                      (SI-ACC    -ACF     0)
      C        WRITE AS = ( -AFC   SI-AFF     0)
      C                      ( -HTF      0       I)
00073          IF(NXC.EQ.0) GO TO 230
00075          DO 100 I=1,NXC
00076          DO 100 J=1,NXC
00077          DUMMY(I,J)=-FC(I,J)
00078          IF(M2.EQ.0.OR.NU.EQ.0) GO TO 90
00080          DO 60 K=1,M1+M2
00081          DUM=0.
00082          DO 50 L=1,NU
00083 50       DUM=DUM+GC(I,L)*DF(L,K)
00084 60       DUMMY2(I,K)=DUM
00085          DUM=0.
00086          DO 70 K=1,M1+M2
00087 70       DUM=DUM+DUMMY2(I,K)*HM(K,J)*XIM(K)
00088          DUMMY(I,J)=DUMMY(I,J)-DUM
00089 80       A1=0.
00090          IF(I.EQ.J) A1=1.0
00092          IF(A1.EQ.0..AND.DUMMY(I,J).EQ.0.) GO TO 100
00094          WRITE(ITREN,90) I,J,A2,A1,DUMMY(I,J),K0
00095 90       FORMAT(2I,3E,I)
00096 100    CONTINUE
00097          IF(M1P.EQ.0) GO TO 175
```

```
00099          A1=0.
00100          DO 170 I=1,M1P
00101          II=I+NXC
00102          DO 170 J=1,NXC
00103          DUM=0.
00104          DO 150 K=1,M1+M2
00105 150      DUM=DUM-BF(I,K)*TM(K,J)*XIM(K)
00106          IF(DUM.EQ.0.) GO TO 170
00107          WRITE(ITRFN,90) II,J,A2,A1,DUM,K0
00109 170      CONTINUE
00110 175      DO 190 I=1,NZ
00111          II=I+NXC+M1P
00112          DO 180 J=1,NXC
00113          DUM=-TM(I,J)
00114          IF(DUM.EQ.0.) GO TO 180
00116          WRITE(ITRFN,90) II,J,A2,A1,DUM,K0
00117 180      CONTINUE
00118          IF(NYO.EQ.0) GO TO 200
00120          DO 190 I=1,NYO
00121          II=I+NXC+M1P+NZ
00122          DO 190 J=1,NXC
00123          DUM=-HO(I,J)
00124          IF(DUM.EQ.0.) GO TO 190
00126          WRITE(ITRFN,90) II,J,A2,A1,DUM,K0
00127 190      CONTINUE
00128 200      CONTINUE
00129          IF(M1P.EQ.0) GO TO 245
00131          IF(NU.EQ.0) GO TO 230
00133          DO 220 I=1,NXC
00134          DO 220 J=1,M1P
00135          JJ=J+NXC
00136          DUM=0.
00137          DO 210 K=1,NU
00138 210      DUM=DUM-GC(I,K)*CF(K,J)
00139          IF(DUM.EQ.0.) GO TO 220
00141          WRITE(ITRFN,90) I,JJ,A2,A1,DUM,K0
00142 220      CONTINUE
00143 230      DO 240 I=1,M1P
00144          II=I+NXC
00145          DO 240 J=1,M1P
00146          JJ=J+NXC
00147          A1=0.
00148          IF(I.EQ.J) A1=1.0
00150          DUM=-AI(I,J)
00151          IF(A1.EQ.0..AND.DUM.EQ.0.) GO TO 240
00153          WRITE(ITRFN,90) II,JJ,A2,A1,DUM,K0
00154 240      CONTINUE
00155 245      A1=0.
00156          DO 250 I=1,NZ+NYO
00157          II=I+NXC+M1P
00159 250      WRITE(ITRFN,90) II,II,A2,A1,XI,K0
00159          WRITE(ITRFN,90) K0
               C                    (BCS   BCV   BCL   G4C)
               C       WRITE BS = (BFS   BFV   BFL    0 )
```

```
      C                                (E1     0     0     0 )
00160         IF(NXC.EQ.0) GO TO 310
00162         IF(NYS.EQ.0) GO TO 350
00164         DO 300 I=1,NXC
00165         DO 300 J=1,NYS
00166         IF(M2.EQ.0.OR.NU.EQ.0) GO TO 290
00168         DO 270 K=1,M1+M2
00169         DUM=0.
00170         DO 260 L=1,NU
00171  260    DUM=DUM+GC(I,L)*DF(L,K)
00172  270    DUMMY2(I,K)=DUM
00173         DUM=0.
00174         DO 280 K=1,M1+M2
00175  280    DUM=DUM+DUMMY2(I,K)*EM(K,J)*XIM(K)
00176  290    DUMMY(I,J)=CC(I,J)+DUM
00177         IF(DUMMY(I,J).EQ.0.) GO TO 300
00179         WRITE(ITRFN,90) I,J,A2,A1,DUMMY(I,J),K0
00180  300    CONTINUE
00181  310    CONTINUE
00182         IF(M1P.EQ.0) GO TO 335
00184         DO 330 I=1,M1P
00185         II=I+NXC
00186         DO 330 J=1,NYS
00187         DUM=0.
00188         DO 320 K=1,M1+M2
00189  320    DUM=DUM+BF(I,K)*EM(K,J)*XIM(K)
00190         IF(DUM.EQ.0.) GO TO 330
00192         WRITE(ITRFN,90) II,J,A2,A1,DUM,K0
00193  330    CONTINUE
00194  335    DO 340 I=1,NZ+NYO
00195         II=I+NXC+M1P
00196         DO 340 J=1,NYS
00197         IF(E1(I,J).EQ.0.) GO TO 340
00199         WRITE(ITRFN,90) II,J,A2,A1,E1(I,J),K0
00200  340    CONTINUE
00201  350    CONTINUE
00202         IF(NXC.EQ.0.OR.NU.EQ.0.OR.M2.EQ.0) GO TO 390
00204         DO 370 I=1,NXC
00205         DO 370 J=1,M1+M2
00206         JJ=J+NYS
00207         DUM=0.
00208         DO 360 K=1,NU
00209  360    DUM=DUM+GC(I,K)*DF(K,J)
00210         IF(DUM.EQ.0.) GO TO 370
00212         WRITE(ITRFN,90) I,JJ,A2,A1,DUM,K0
00213  370    CONTINUE
00214         IF(M1P.EQ.0) GO TO 395
00216  390    DO 390 I=1,M1P
00217         II=I+NXC
00218         DO 390 J=1,M1+M2
00219         JJ=J+NYS
00220         IF(BF(I,J).EQ.0.) GO TO 390
00222         WRITE(ITRFN,90) II,JJ,A2,A1,BF(I,J),K0
00223  390    CONTINUE
```

```
00224 395   CONTINUE
00225       IF(NYS.EQ.0) GO TO 470
00227       IF(NXC.EQ.0.OR.NU.EQ.0.OR.M2.EQ.0) GO TO 440
00229       DO 410 I=1,NXC
00230       DO 410 J=1,M1+M2
00231       DUM=0.
00232       DO 400 K=1,NU
00233 400   DUM=DUM+GC(I,K)*DF(K,J)
00234 410   DUMMY(I,J)=DUM
00235       DO 430 I=1,NXC
00236       DO 430 J=1,NYS
00237       JJ=J+NYS+M1+M2
00238       DUM=0.
00239       DO 420 K=1,M1+M2
00240 420   DUM=DUM+DUMMY(I,K)*FM(K,J)
00241       IF(DUM.EQ.0.) GO TO 430
00243       WRITE(ITRFN,90) I,JJ,A2,A1,DUM,K0
00244 430   CONTINUE
00245       IF(M1P.EQ.0) GO TO 470
00247 440   DO 460 I=1,M1P
00248       II=I+NXC
00249       DO 450 J=1,NYS
00250       JJ=J+NYS+M1+M2
00251       DUM=0.
00252       DO 450 K=1,M1+M2
00253 450   DUM=DUM+BF(I,K)*FM(K,J)
00254       IF(DUM.EQ.0.) GO TO 460
00256       WRITE(ITRFN,90) II,JJ,A2,A1,DUM,K0
00257 460   CONTINUE
00258 470   CONTINUE
00259       IF(NU.EQ.0.OR.NXC.EQ.0) GO TO 490
00261       DO 480 I=1,NXC
00262       DO 480 J=1,NU
00263       JJ=I+2*NYS+M1+M2
00264       IF(GAC(I,J).EQ.0.) GO TO 480
00265       WRITE(ITRFN,90) I,JJ,A2,A1,GAC(I,J),K0
00267 480   CONTINUE
00268 490   WRITE(ITRFN,90) K0
      C       WRITE TITLE, DIMENSIONS AND COLUMN CODES
00269       WRITE(ITRFN,20) (TITLE(I),I=1,15)
00270       NX=NXC+M1P+NZ+NYO
00271       NY=2*NYS+M1+M2+NU
00272       WRITE(ITRFN,500) NX,NY,K0
00273 500   FORMAT(3I)
00274       IF(NXC.EQ.0) GO TO 520
00276       DO 510 I=1,NXC
00277 510   ANN(I)=ANXC(I)
00278 520   CONTINUE
00279       IF(M1P.EQ.0) GO TO 540
00281       DO 530 I=1,M1P
00282       II=I+NXC
00283 530   ANN(II)=ANXT(I)
00284 540   CONTINUE
00285       DO 550 I=1,NZ
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
00286          II=I+NXC+M1P
00287 550      MNN(II)=MNZ(I)
00288          IF(NYO.EQ.0) GO TO 570
00290          DO 560 I=1,NYO
00291          II=I+NXC+M1P+NZ
00292 560      MNN(II)=MNYO(I)
00293 570      CONTINUE
00294          IF(NYS.EQ.0) GO TO 590
00296          DO 580 I=1,NYS
00297 580      MNM(I)=MNYS(I)
00298 590      DO 600 I=1,NZ
00299          II=I+NYS
00300 600      MNM(II)=MNV(I)
00301          IF(NYS.EQ.0) GO TO 620
00303          DO 610 I=1,NYS
00304          II=I+NYS+NZ
00305 610      MNM(II)=MNTOL(I)
00306 620      CONTINUE
00307          IF(NN.EQ.0) GO TO 640
00309          DO 630 I=1,NN
00310          II=I+2*NYS+NZ
00311 630      MNM(II)=MNN(I)
00312 640      CONTINUE
00313          IF(NX.LE.18) GO TO 670
00315          WRITE(ITRFN,650) (MNN(I),I=1,18),IP
00316 650      FORMAT(17A4,A3,A1)
00317          IF(NX.LE.36) GO TO 660
00319          WRITE(ITRFN,650) (MNN(I),I=19,36),IP
00320          ISTART=37
00321          GO TO 680
00322 660      ISTART=19
00323          GO TO 680
00324 670      ISTART=1
00325 680      WRITE(ITRFN,650) (MNN(I),I=ISTART,NX)
00326          IF(NY.LE.18) GO TO 700
00328          WRITE(ITRFN,650) (MNM(I),I=1,18),IP
00329          IF(NY.LE.36) GO TO 690
00331          WRITE(ITRFN,650) (MNM(I),I=19,36),IP
00332          ISTART=37
00333          GO TO 710
00334 690      ISTART=19
00335          GO TO 710
00336 700      ISTART=1
00337 710      WRITE(ITRFN,650) (MNM(I),I=ISTART,NY)
       C          THE END
00338          STOP
00339          END
```

```
        00045RD 00047RD 00049RD 00051RD 00053RD 00055RD 00057RD 00059RD
        00063RD 00065RD 00067RD 00069RD 00270=  00272WR 00313IF 00317IF
        00325WR
NXC     00027RD 00071   00073IF 00075DO 00076DO 00101   00102DO 00111
        00112DO 00121   00122DO 00133DO 00135   00144   00146   00157
        00160IF 00164DO 00165   00195   00202IF 00204DO 00217   00227IF
        00229DO 00235DO 00248   00259IF 00261DO 00270   00274IF 00276DO
        00282   00286   00291
NXS     00027RD
NY      00043RD 00045RD 00047RD 00049RD 00051RD 00053RD 00055RD 00057RD
        00063RD 00065RD 00067RD 00069RD 00271=  00272WR 00326IF 00329IF
        00337WR
NYO     00027RD 00118IF 00120DO 00156DO 00194DO 00270   00288IF 00290DO
NYS     00027RD 00162IF 00165DO 00186DO 00196DO 00206   00219   00225IF
        00236DO 00237   00249DO 00250   00263   00271   00294IF 00296DO
        00299   00301IF 00303DO 00304   00310
NZ      00027RD 00110DO 00121   00156DO 00194DO 00270   00235DO 00291
        00298DO 00304   00310
TITLE   00001DI 00018RD 00269WR
XIM     00003DI 00022RD 00087   00105   00175   00189
XI      00007DA 00158WR
10      00008WR 00009*
100     00075DO 00076DO 00093GT 00096*
12      00012WR 00013*
14      00018WR 00017*
160     00104DO 00105*
170     00100DO 00102DO 00107GT 00109*
175     00099GT 00110*
180     00110DO 00112DO 00115GT 00117*
190     00120DO 00122DO 00125GT 00127*
20      00010RD 00014RD 00018RD 00019*  00269WR
200     00119GT 00120*
210     00137DO 00138*
220     00133DO 00134DO 00140GT 00142*
230     00074GT 00132GT 00143*
240     00143DO 00145DO 00152GT 00154*
245     00130GT 00155*
25      00022WR 00021*
250     00156DO 00158*
260     00170DO 00171*
270     00168DO 00172*
280     00174DO 00175*
290     00167GT 00176*
30      00022RD 00023*
300     00164DO 00165DO 00173GT 00180*
310     00161GT 00181*
320     00188DO 00189*
330     00184DO 00186DO 00191GT 00193*
335     00183GT 00194*
340     00194DO 00196DO 00199GT 00200*
350     00163GT 00201*
360     00209DO 00209*
370     00204DO 00205DO 00211GT 00213*
380     00200GT 00216*
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
390     00216DO 00218DO 00221GT 00223*
395     00215GT 00224*
40      00024DO 00070*
400     00232DO 00233*
410     00229DO 00230DO 00234*
420     00239DO 00240*
430     00235DO 00236DO 00242GT 00244*
440     00220GT 00247*
45      00025RD 00071*
450     00252DO 00253*
460     00247DO 00249DO 00255GT 00257*
470     00226GT 00246GT 00258*
480     00261DO 00262DO 00265GT 00267*
490     00266GT 00268*
50      00082DO 00083*
500     00272WR 00273*
510     00276DO 00277*
520     00275GT 00278*
530     00281DO 00283*
540     00286GT 00284*
550     00285DO 00287*
560     00290DO 00292*
570     00289GT 00293*
580     00296DO 00297*
590     00295GT 00298*
60      00084DO 00084*
600     00299DO 00300*
610     00303DO 00305*
620     00302GT 00306*
630     00309DO 00311*
640     00308GT 00312*
650     00315WR 00316*  00319WR 00325WR 00326WR 00331WR 00337WR
660     00318GT 00322*
670     00314GT 00324*
680     00321GT 00323* 00325*
690     00330GT 00334*
70      00086DO 00087*
700     00327GT 00335*
710     00333GT 00335GT 00337*
80      00079GT 00089*
90      00094WR 00095*  00109WR 00113WR 00120WR 00141WR 00153WR 00158WR
        00159WR 00179WR 00192WR 00199WR 00212WR 00222WR 00243WR 00253WR
        00263WR 00268WR
```

```
      C       BLOCK 10 - CLOSED LOOP SYSTEM PERFORMANCE ANALYSIS
00001         DIMENSION FNAME(4),TTLNEW(5)
      C         INPUT MATRIX ARRAYS
00002         DIMENSION GM(15,15),QD(5),A(30,30),TZ(15,15),RD(15),
     +                  BF(15,15),HR(30,15),DF(5,15),H(15,15),CF(5,15)
      C         OUTPUT MATRIX ARRAYS
00003         DIMENSION CV(30,30),CM(30,30),COVYU(35,35)
      C         WORKING SPACE
00004         DIMENSION DUMMY(35,35),ARRAY(900),IARRAY(900),QM(30,30),
     +                  QV(30,30),E(30,30),XF(30,30),TR(35),YU(35,30),
     +                  SCALE(30)
      C         EQUIVALENCES
00005         EQUIVALENCE (YU,A,ARRAY,IARRAY),(COVYU,XF),(COVYU(26,26),TZ),
     +                  (COVYU(1,32),DF),(QV,QM),(DUMMY,E,HR),(H,BF),
     +                  (TZ,CF),(GM,XF),(SCALE,RD)
00006         COMMON/NONAME/ NDIM
      C         DATA
00007         DATA IIN,IOUT,IPFIN,IPFOUT /5,7,20,21/
00008         DATA LASTID,TTLNEW,II,RP,I900 /120,' "" ',4*' ',1,2.,900/
      C         READ FILE NAMES
00009         WRITE(IOUT,10)
00010 10      FORMAT(2X,'STOCHASTIC PERFORMANCE ANALYSIS - BLOCK 10',
     +       +     /,2X,'INPUT PROBLEM FILE NAME ? ',$)
00011         READ(IIN,20) (FNAME(I),I=1,4)
00012         CALL ASSIGN(IPFIN,FNAME)
00013         WRITE(IOUT,12)
00014 12      FORMAT(2X,'OUTPUT PROBLEM FILE NAME ? ',$)
00015         READ(IIN,20) (FNAME(I),I=1,4)
00016         CALL ASSIGN(IPFOUT,FNAME)
00017 20      FORMAT(5A4)
      C         READ AND WRITE PROBLEM FILE
00018         CALL RENR(IPFIN,IPFOUT,TTLNEW,I900,LASTID,ARRAY,IARRAY)
00019         REWIND IPFIN
      C         READ THE FOLLOWING FROM THE PROBLEM FILE:
      C           DIMENSIONS   IDEN=2
      C           GM MATRIX    IDEN=25
      C           TZ MATRIX    IDEN=35
      C           QD VECTOR    IDEN=50
      C           RD VECTOR    IDEN=51
      C           DIMENSIONS   IDEN=54
      C           BF MATRIX    IDEN=117
      C           A MATRIX     IDEN=120
00020         DO 30 K=1,200
00021         READ(IPFIN,END=40) IDEN
00022         IF(IDEN.EQ.1) READ(IPFIN) IDEN,NXS,NXC,NU,NM,NZ,NYS,NYO
00024         IF(IDEN.EQ.25) READ(IPFIN) IDEN,NX,NY,((GM(I,J),J=1,NY),I=1,NX)
00026         IF(IDEN.EQ.34) READ(IPFIN) IDEN,NX,NY,((TZ(I,J),J=1,NY),I=1,NX)
00028         IF(IDEN.EQ.49) READ(IPFIN) IDEN,NX,(QD(I),I=1,NX)
00030         IF(IDEN.EQ.50) READ(IPFIN) IDEN,NX,(RD(I),I=1,NX)
00032         IF(IDEN.EQ.53) READ(IPFIN) IDEN,M1,M1P,M2,M2P
00034         IF(IDEN.EQ.116) READ(IPFIN) IDEN,NX,NY,((BF(I,J),J=1,NY),I=1,NX)
00036         IF(IDEN.EQ.119) READ(IPFIN) IDEN,NX,NY,((A(I,J),J=1,NY),I=1,NX)
00038 30      CONTINUE
00039 40      REWIND IPFIN
```

```
00040          NXT=NXS+NXC
00041          NW=NW-NXC
00042          NTOT=NXT+M1P
00043          IDEN=LASTID
               C                          (GM)
               C     COMPOSE QW = (  ) QD (GMT  0), SOLVE FOR CW AND WRITE
               C                          (0 )
00044          DO 670 I=1,NTOT
00045          DO 670 J=1,NTOT
00046 670      QW(I,J)=0.
00047          IF(NW.EQ.0) GO TO 700
00049          DO 690 I=1,NXT
00050          DO 690 J=1,NXT
00051          DUM=0.
00052          DO 680 K=1,NW
00053 680      DUM=DUM+GM(I,K)*GM(J,K)*QD(K)
00054 690      QW(I,J)=DUM
00055          NDIM=30
00056          CALL LINEQ1(NDIM,NTOT,A,QW,CW,E,XF,TR,SCALE)
00057 700      IDEN=IDEN+1
00058          IF(NW.EQ.0) GO TO 705
00060          WRITE(IPFOUT) IDEN,NTOT,NTOT,((CW(I,J),J=1,NTOT),I=1,NTOT)
00061          DO 704 I=1,NTOT
00062 704      TR(I)=SQRT(CW(I,I))
00063          IDEN=IDEN+1
00064          WRITE(IPFOUT) IDEN,NTOT,(TR(I),I=1,NTOT)
00065          GO TO 708
00066 705      WRITE(IPFOUT) IDEN,I1,I1,R2
00067          IDEN=IDEN+1
00068          WRITE(IPFOUT) IDEN,I1,R0
               C                          (0 )
               C       COMPOSE QV =  (  ) TZI*RD*TZIT (J  BFT), SOLVE FOR CV AND WRITE
               C                          (BF)
00069 708      DO 710 I=1,NTOT
00070          DO 710 J=1,NTOT
00071 710      QV(I,J)=0.
00072          IF(M1.EQ.0) GO TO 800
00074          DO 720 I=1,NZ
00075          DO 720 J=1,NZ
00076 720      DUMMY(I,J)=TZ(I,J)
00077          NDIM=35
00078          CALL GMINV(NZ,NZ,DUMMY,TZ,WR,J)
00079          DO 740 I=1,NZ
00080          DO 740 J=1,NZ
00081          DUM=0.
00082          DO 730 K=1,NZ
00083 730      DUM=DUM+TZ(I,K)*TZ(J,K)*RD(K)
00084 740      DUMMY(I,J)=DUM
00085          DO 750 I=1,NZ
00086          DO 750 J=1,NZ
00087 750      TZ(I,J)=DUMMY(I,J)
00088          DO 770 I=1,M1P
00089          DO 770 J=1,NZ
00090          DUM=0.
```

```
00091          DO 760 K=1,NZ
00092  760     DUM=DUM+BF(I,K)*TZ(K,J)
00093  770     DUMMY(I,J)=DUM
00094          DO 790 I=1,M1P
00095          II=I+NXT
00096          DO 790 J=1,M1P
00097          JJ=J+NXT
00098          DUM=0.
00099          DO 780 K=1,NZ
00100  780     DUM=DUM+DUMMY(I,K)*BF(J,K)
00101  790     QV(II,JJ)=DUM
00102  800     NDIM=30
00103          CALL LINEQ1(NDIM,NTOT,A,QV,CV,E,XF,TR,SCALE)
00104          IDEN=IDEN+1
00105          WRITE(IPFOUT) IDEN,NTOT,NTOT,((CV(I,J),J=1,NTOT),I=1,NTOT)
00106          DO 804 I=1,NTOT
00107  804     TR(I)=SQRT(CV(I,I))
00108          IDEN=IDEN+1
00109          WRITE(IPFOUT) IDEN,NTOT,(TR(I),I=1,NTOT)
       C           WRITE COV=CV+CV
00110          IDEN=IDEN+1
00111          DO 810 I=1,NTOT
00112          DO 810 J=1,NTOT
00113          IF(NV.EQ.0.AND.M1P.NE.0) CV(I,J)=CV(I,J)
00115          IF(NV.NE.0.AND.M1P.EQ.0) CV(I,J)=CV(I,J)
00117          IF(NV.EQ.0.AND.M1P.EQ.0) GO TO 815
00119          IF(NV.NE.0.AND.M1P.NE.0) CV(I,J)=CV(I,J)+CV(I,J)
00121  810     CONTINUE
00122          WRITE(IPFOUT) IDEN,NTOT,NTOT,((CV(I,J),J=1,NTOT),I=1,NTOT)
00123          DO 812 I=1,NTOT
00124  812     TR(I)=SQRT(CV(I,I))
00125          IDEN=IDEN+1
00126          WRITE(IPFOUT) IDEN,NTOT,(TR(I),I=1,NTOT)
00127          GO TO 818
00128  815     WRITE(IPFOUT) IDEN,I1,I1,R0
00129          IDEN=IDEN+1
00130          WRITE(IPFOUT) IDEN,I1,R0
00131  818     CONTINUE
00132          LASTID=IDEN
       C           READ THE FOLLOWING FROM THE PROBLEM FILE:
       C             H MATRIX   IDEN=27
       C             HR MATRIX  IDEN=28
       C             CF MATRIX  IDEN=117
       C             DF MATRIX  IDEN=119
00133          DO 819 K=1,200
00134          READ(IPFIN,END=820) IDEN
00135          IF(IDEN.EQ.26) READ(IPFIN) IDEN,NX,NY,((H(I,J),J=1,NY),I=1,NX)
00137          IF(IDEN.EQ.27) READ(IPFIN) IDEN,NX,NY,((HR(I,J),J=1,NY),I=1,NX)
00139          IF(IDEN.EQ.117) READ(IPFIN) IDEN,NX,NY,((CF(I,J),J=1,NY),I=1,NX)
00141          IF(IDEN.EQ.118) READ(IPFIN) IDEN,NX,NY,((DF(I,J),J=1,NY),I=1,NX)
00143  819     CONTINUE
00144  820     IDEN=LASTID
       C                        ( HR    0)
       C           FORM YU = (          )
```

```
      C                    (DF*H  CF)
00145          DO 825 I=1,NZ+NYO+NU
00146          DO 825 J=1,NXT+M1P
00147  825     YU(I,J)=0.
00148          DO 830 I=1,NZ+NYO
00149          DO 830 J=1,NXT
00150          YU(I,J)=HR(I,J)
00151  830     IF(NU.EQ.0) GO TO 870
00153          DO 850 I=1,NU
00154          II=I+NZ+NYO
00155          DO 850 J=1,NXT
00156          DUM=0.
00157          DO 840 K=1,NZ
00158  840     DUM=DUM+DF(I,K)*H(K,J)
00159  850     YU(II,J)=DUM
00160          IF(M1P.EQ.0) GO TO 870
00162          DO 860 I=1,NU
00163          II=I+NZ+NYO
00164          DO 860 J=1,M1P
00165          JJ=J+NXT
00166  860     YU(II,JJ)=CF(I,J)
      C          COMPUTE COVYU = YU*(CV+CV)*YUT AND WRITE
00167  870     DO 890 I=1,NZ+NYO+NU
00168          DO 890 J=1,NXT+M1P
00169          DUM=0.
00170          DO 880 K=1,NXT+M1P
00171  880     DUM=DUM+YU(I,K)*CV(K,J)
00172  890     DUMMY(I,J)=DUM
00173          DO 910 I=1,NZ+NYO+NU
00174          DO 910 J=1,NZ+NYO+NU
00175          DUM=0.
00176          DO 900 K=1,NXT+M1P
00177  900     DUM=DUM+DUMMY(I,K)*YU(J,K)
00178  910     COVYU(I,J)=DUM
00179          IDEN=IDEN+1
00180          NTOT=NZ+NYO+NU
00181          WRITE(IPFOUT) IDEN,NTOT,NTOT,((COVYU(I,J),J=1,NTOT),I=1,NTOT)
00182          DO 920 I=1,NTOT
00183          TR(I)=0.
00184          IF(COVYU(I,I).GT.0.) TR(I)=SQRT(COVYU(I,I))
00185  920     CONTINUE
00187          IDEN=IDEN+1
00188          WRITE(IPFOUT) IDEN,NTOT,(TR(I),I=1,NTOT)
00189          DO 940 I=1,NTOT
00190          DO 940 J=1,NTOT
00191          IF(TR(I).EQ.0.) GO TO 930
00193          COVYU(I,J)=COVYU(I,J)/(TR(I)*TR(J))
00194          GO TO 940
00195  930     COVYU(I,J)=0.
00196          COVYU(J,I)=0.
00197          COVYU(I,I)=1.
00198  940     CONTINUE
00199          IDEN=IDEN+1
00200          WRITE(IPFOUT) IDEN,NTOT,NTOT,((COVYU(I,J),J=1,NTOT),I=1,NTOT)
```

```
000001          DO 950 I=IDEN+1,130
000002 950      WRITE(IPFOUT) IDEN,I1,I1,R0
                C       THE END
000003          STOP
000004          END
```

```
A       000202DI 000305EQ 000337RD 000356AG 000103AG
ARRAY   000304DI 000305EQ 000018AG
ASSIGN  000212CL 000216CL
BF      000302DI 000305EQ 000235RD 000092   000100
CF      000302DI 000305EQ 000140RD 000166
COMMON  000303DI 000305EQ 000170=  000181#R 000184IF 000185   000193=  000195=
        000196=  000197=  000203#R
CV      000303DI 000103AG 000109#R 000107   000114=  000116=  000120=  000122#R
        000124   000171
CW      000303DI 000356AG 000367#R 000362   000116   000120
DF      000302DI 000305EQ 000142RD 000158
DUM     000351=  000353=  000354   000381=  000293=  000284   000093=  000092=
        000093   000096=  000100=  000101   000156=  000158=  000159   000159=
        000171=  000172   000175=  000177=  000178
DUMMY   000304DI 000305EQ 000075=  000370AG 000304=  000307   000093=  000100
        000172=  000177
E       000304DI 000305EQ 000356AG 000103AG
FNAME   000211DI 000211RD 000212AG 000215RD 000216AG
G4      000302DI 000305EQ 000225RD 000353
G4INV   000170CL
H       000002DI 000305EQ 000136RD 000158
HR      000302DI 000305EQ 000136RD 000158
I       000211RD 000215RD 000225RD 000227RD 000229RD 000231RD 000235RD 000237RD
        000244DO 000246   000249DO 000253   000254   000256#R 000261DO 000262
        000266#R 000269DO 000271   000274DO 000276   000279DO 000283   000284
        000285DO 000287   000289DO 000292   000293   000294DO 000295   000100
        000138#R 000176DO 000127   000130#R 000111DO 000114   000116   000120
        000122#R 000123DO 000124   000129#R 000136RD 000138RD 000140RD 000142RD
        000145DO 000147   000148DO 000150   000153DO 000154   000158   000162DO
        000163   000166   000167DO 000171   000172   000173DO 000177   000178
        000181#R 000182DO 000183   000184IF 000185   000188#R 000189DO 000191IF
        000194   000195   000196   000197   000202#R 000201DO
IARRAY  000304DI 000305EQ 000018AG
ICON    000221RD 000222IF 000223RD 000224IF 000225RD 000227RD 000228IF
        000229RD 000230IF 000231RD 000232IF 000233RD 000234IF 000235RD 000236IF
        000237RD 000243=  000257=  000260#R 000263=  000256#R 000258#R 000257=
        000259#R 000174=  000175#R 000103=  000139#R 000110=  000122#R 000125=
        000128#R 000129#R 000120=  000130#R 000132   000134RD 000135IF 000136RD
        000137IF 000138RD 000139IF 000140RD 000141IF 000142RD 000144=  000179=
        000182#R 000187=  000188#R 000199=  000200#R 000201DO 000202#R
II      000095=  000101   000154=  000159   000163=  000166
IIN     000207DA 000211RD 000215RD
IOUT    000207DA 000209#R 000213#R
IPFIN   000207DA 000212AG 000218AG 000219RA 000221RD 000223RD 000225RD 000227RD
        000229RD 000231RD 000232RD 000235RD 000237RD 000239R# 000134RD 000136RD
        000138RD 000140RD 000142RD
IPFOUT  000207DA 000212AG 000218AG 000256#R 000256#R 000259#R 000259#R 000184#R
        000189#R 000122#R 000126#R 000120#R 000133#R 000181#R 000183#R 000202#R
        000202#R
I1      000203DA 000256#R 000259#R 000128#R 000133#R 000202#R
I90J    000203DA 000219AG
J       000225RD 000227RD 000235RD 000237RD 000245DO 000246   000256DO 000253
        000254   000256#R 000270DO 000271   000275DO 000276   000280DO 000283
        000284   000289DO 000287   000290DO 000292   000293   000296DO 000297
```

```
        00100   00105NR 00112DO 00114   00116   00120   00122NR 00136RD
        00138RD 00140RD 00142RD 00146DO 00147   00149DO 00150   00155DO
        00158   00159   00164DO 00165   00166   00168DO 00171   00172
        00174DO 00177   00178   00181NR 00190DO 00193   00195   00196
        00200NR
JJ      00097=  00101   00165=  00166
K       00020DO 00052DO 00053   00092DO 00083   00091DO 00092   00099DO
        00100   00133DO 00157DO 00158   00173DO 00171   00176DO 00177
LASTID  00088DA 00310AG 00043   00132=  00144
LINEQI  00056CL 00103CL
MR      00070AG
M1      00033RD 00072IF
M1P     00033RD 00042   00088DC 00094DO 00096DO 00113IF 00115IF 00117IF
        00119IF 00146DO 00160IF 00164DO 00168DO 00170DO 00176DO
M2      00033RD
M2P     00033RD
NDIM    00056CM 00055=  00055AG 00077=  00102=  00103AG
NONAME  00056CN
NTOT    00042=  00044DO 00045DO 00056AG 00057NR 00061DO 00064NR 00069DO
        00070DO 00103AG 00105NR 00106DO 00109NR 00111DO 00112DO 00122NR
        00123DO 00123NR 00180=  00181NR 00182DO 00188NR 00189DO 00190DO
        00200NR
NU      00023RD 00145DO 00151IF 00153DO 00162DO 00167DO 00173DO 00174DO
        00180
NN      00023RD 00041=  00047IF 00052DO 00058IF 00113IF 00115IF 00117IF
        00119IF
NX      00025RD 00027RD 00029RD 00031RD 00035RD 00037RD 00136RD 00138RD
        00140RD 00142RD
NXC     00023RD 00040   00041
NXS     00023RD 00040
NXT     00040=  00042   00046DO 00050DO 00095   00097   00146DO 00149DO
        00155DO 00165   00168DO 00170DO 00176DO
NY      00025RD 00027RD 00035RD 00037RD 00136RD 00138RD 00142RD 00142RD
NYO     00023RD 00145DO 00149DO 00154   00163   00167DO 00173DO 00174DO
        00180
NYS     00023RD
NZ      00023RD 00074DO 00075DO 00078AG 00079DO 00082DO 00082DO 00085DO
        00088DO 00089DO 00091DO 00099DO 00145DO 00149DO 00154   00157DO
        00163   00167DO 00173DO 00174DO 00180
QD      00052DI 00029RD 00053
QV      00054DI 00055EQ 00071=  00101=  00103AG
QN      00054DI 00055EQ 00046=  00054=  00056AG
RD      00052DI 00055EQ 00031RD 00093
RENR    00011CL
RO      00008DA 00053NR 00058NR 00128NR 00136NR 00202NR
SCALE   00054DI 00055EQ 00056AG 00103AG
SQRT    00052   00107   00124   00105
TR      00024DI 00056AG 00052=  00054NR 00103AG 00107=  00109NR 00124=
        00128NR 00193=  00185=  00183NR 00191IF 00193
TTLNEW  00001DI 00008DA 00218AG
TZ      00002DI 00055EQ 00027RD 00076   00078AG 00083   00087=  00092
XF      00004DI 00055EQ 00056AG 00103AG
YU      00004DI 00055EQ 00147=  00150=  00159=  00166=  00171   00177
10      00009NR 00010*
```

```
12      00013NR 00014*
20      00011RD 00015RD 00017*
30      00020DO 00038*
40      00021RD 00039*
670     00044DO 00045DO 00046*
680     00052DO 00053*
690     00049DO 00056DO 00054*
700     00040GT 00057*
704     00061DO 00062*
705     00059GT 00066*
708     00065GT 00069*
710     00069DO 00073DO 00071*
720     00074DO 00075DO 00076*
730     00082DO 00083*
740     00079DO 00088DO 00084*
750     00085DO 00086DO 00087*
760     00091DO 00092*
770     00088DO 00089DO 00093*
780     00099DO 00103*
790     00094DO 00096DO 00101*
800     00073GT 00102*
804     00106DO 00107*
810     00111DO 00112DO 00121*
812     00123DO 00124*
815     00118GT 00128*
818     00127GT 00131*
819     00133DO 00143*
820     00134RD 00144*
825     00145DO 00146DO 00147*
830     00148DO 00149DO 00151*
840     00157DO 00158*
850     00153DO 00155DO 00159*
860     00162DO 00164DO 00166*
870     00152GT 00160GT 00167*
880     00170DO 00171*
890     00156DO 00169DO 00172*
900     00176DO 00177*
910     00173DO 00174DO 00178*
920     00182DO 00186*
930     00192GT 00195*
940     00169DO 00190DO 00194GT 00193*
950     00201DO 00202*
```

```
      C        ROUTINE TO READ PROBLEM FILE AND FORMAT
      C        THE DATA FOR HARD COPY
00001         DIMENSION RARRAY(35,35),CARRAY(35,15),CLABEL(35),RLABEL(35)
00002         COMPLEX CMPRAY(15,15),BARRAY(15)
00003         REAL*8 DBLANK,XDATE(2),XTIME
00004         COMMON/IO/ IIN,IKB
00005         DATA BLANK /' '/,DBLANK /' '/
00006         DATA IIN,IOUT,IFIN,IFOUT /5,7,20,21/
      C        GET FILENAME FOR PROBLEM FILE
00007         NDIM=35
00008         IKB=24
00009         CALL ASSIGN(IKB,'KB:/C')
      C       CALL DATE(XDATE)
      C       CALL TIME(XTIME)
      C       WRITE(IKB,5) (XDATE(I),I=1,2),XTIME
      C5      FORMAT(20X,A8,A1,5X,A8,/)
      C       REWIND IKB
00010         WRITE(IOUT,10)
00011 10      FORMAT(5X,'PROBLEM FILE NAME ? ',$)
00012         READ(IIN,20) (CLABEL(I),I=1,4)
00013 20      FORMAT(4A4)
00014         CALL ASSIGN(IFIN,CLABEL)
00015         DO 25 K=1,200
00016         READ(IFIN,END=26) IDEN
00017         IF(IDEN.EQ.1) READ(IFIN) IDEN,NXS,NXC,NU,NW,NZ,NYS,NYO
00019         IF(IDEN.EQ.53) READ(IFIN) IDEN,M1,M1P,M2,M2P
00021 25      CONTINUE
00022 26      REWIND IFIN
00023         NXT=NXS+NXC
00024         NZT=NZ+NYO
      C        GET INDICES OF ELEMENTS TO BE LISTED
00025         WRITE(IOUT,30)
00026 30      FORMAT(5X,'INPUT INDEX OF EACH ELEMENT TO BE LISTED',/,
     +    5X,'ONE PER LINE - LAST INDEX SHOULD BE 0')
00027 40      READ(IIN,50) IX
00028 50      FORMAT(I3)
00029         IF(IX.EQ.0) GO TO 3000
      C        FORMAT THE LISTING
00031         REWIND IFIN
00032         GO TO (65,70,90,90,90,90,90,90,90,90,90,90,100,120,140,160,
     +  180,200,220,240,260,280,300,320,340,360,380,400,420,440,460,
     +  90,90,90,480,500,500,540,560,580,600,620,640,660,680,700,
     +  720,740,760,780,800,820,840,80,900,900,920,940,960,980,
     +  1000,1020,1040,1060,1080,1100,1100,1140,1160,1180,1200,
     +  1220,1240,1200,1220,1240,1200,1220,1240,1200,1220,1240,1200,
     +  1220,1240,1200,1220,1240,1200,1220,1240,1200,1220,1240,1200,
     +  1220,1240,1200,1220,1240,1200,1220,1240,1200,1220,1240,1200,
     +  1220,1240,1200,1220,1240,1200,1220,1240,1260,1280,1300,1320,
     +  1340,1340,1360,1340,1360,1340,1360,1380,1400,1380),IX
      C        TITLE
00033 65      READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00034         WRITE(IKB,68) (CLABEL(I),I=1,NX)
00035 68      FORMAT(5X,'***** ',5A4,10(/,5X,5A4)/)
00036         GO TO 40
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      C        DIMENSIONS
00037 70      WRITE(IKB,75) NXS,NXC,NU,NW,NZ,NYS,NYO
00038 75      FORMAT(5X,'VECTOR DIMENSIONS : '/,
             +   7X,'NXS =',I3,' NXC =',I3,' NU =',I3,' NW =',I3,
             +       ' NZ =',I3,' NYS =',I3,' NYO =',I3/)
00039         GO TO 40
00040 80      WRITE(IKB,85) M1,M2,M1P
00041 85      FORMAT(5X,'MEASUREMENT PARTITION : M1 = ',I3,' M2 = ',I3,/,
             +       5X,'NUMBER OF FILTER STATES : M1P = ',I3,/)
00042         GO TO 40
      C        MNEMONICS
00043 90      DO 98 L=1,IX
00044         READ(IFIN) IDEN
00045         IF(IDEN.NE.IX-1) GO TO 98
00047         READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00048         WRITE(IKB,95) (CLABEL(I),I=1,NX)
00049 95      FORMAT(5X,15A4/)
00050 98      CONTINUE
00051         GO TO 40
      C        MATRICES
00052 100     CONTINUE
00053         IF(NXS.EQ.0.OR.NW.EQ.0) GO TO 1950
00055         DO 110 L=1,IX
00056         READ(IFIN) IDEN
00057         IF(IDEN.EQ.3) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00059         IF(IDEN.EQ.10) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00061         IF(IDEN.EQ.12) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00063 110     CONTINUE
00064         CALL MATLIS(13,'GAMMAS MATRIX',NDIM,NXS,NW,RLABEL,CLABEL,RARRAY,
             +           CARRAY,0)
00065         GO TO 40
00066 120     CONTINUE
00067         IF(NXS.EQ.0) GO TO 1950
00069         DO 130 L=1,IX
00070         READ(IFIN) IDEN
00071         IF(IDEN.EQ.10) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00073         IF(IDEN.EQ.13) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00075 130     CONTINUE
00076         CALL MATLIS(9,'PS MATRIX',NDIM,NXS,NXS,CLABEL,CLABEL,RARRAY,
             +           CARRAY,P)
00077         GO TO 40
00078 140     CONTINUE
00079         IF(NYS.EQ.0.OR.NXS.EQ.0) GO TO 1950
00081         DO 150 L=1,IX
00082         READ(IFIN) IDEN
00083         IF(IDEN.EQ.4) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00085         IF(IDEN.EQ.10) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00087         IF(IDEN.EQ.14) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00089 150     CONTINUE
00090         CALL MATLIS(9,'HS MATRIX',NDIM,NYS,NXS,RLABEL,CLABEL,RARRAY,
             +           CARRAY,0)
00091         GO TO 40
00092 160     CONTINUE
00093         IF(NXC.EQ.0.OR.NW.EQ.0) GO TO 1950
```

```
00095          DO 170 L=1,IX
00096          READ(IFIN) IDEN
00097          IF(IDEN.EQ.3) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00099          IF(IDEN.EQ.11) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00101          IF(IDEN.EQ.15) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00103 170      CONTINUE
00104          CALL MATLIS(13,'GAMMAC MATRIX',NDIM,NXC,NN,RLABEL,CLABEL,RARRAY,
              +                CARRAY,0)
00105          GO TO 40
00106 180      CONTINUE
00107          IF(NXC.EQ.0.OR.NU.EQ.0) GO TO 1950
00109          DO 190 L=1,IX
00110          READ(IFIN) IDEN
00111          IF(IDEN.EQ.2) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00113          IF(IDEN.EQ.11) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00115          IF(IDEN.EQ.16) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00117 190      CONTINUE
00118          CALL MATLIS(9,'GC MATRIX',NDIM,NXC,NU,RLABEL,CLABEL,RARRAY,
              +                CARRAY,0)
00119          GO TO 40
00120 200      CONTINUE
00121          IF(NXC.EQ.0.OR.NYS.EQ.0) GO TO 1950
00123          DO 210 L=1,IX
00124          READ(IFIN) IDEN
00125          IF(IDEN.EQ.4) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00127          IF(IDEN.EQ.11) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00129          IF(IDEN.EQ.17) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00131 210      CONTINUE
00132          CALL MATLIS(9,'CC MATRIX',NDIM,NXC,NYS,RLABEL,CLABEL,RARRAY,
              +                CARRAY,0)
00133          GO TO 40
00134 220      CONTINUE
00135          IF(NXC.EQ.0) GO TO 1950
00137          DO 230 L=1,IX
00138          READ(IFIN) IDEN
00139          IF(IDEN.EQ.11) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00141          IF(IDEN.EQ.18) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00143 230      CONTINUE
00144          CALL MATLIS(9,'FC MATRIX',NDIM,NXC,NXC,RLABEL,RLABEL,RARRAY,
              +                CARRAY,0)
00145          GO TO 40
00146 240      CONTINUE
00147          IF(NZ.EQ.0.OR.NXC.EQ.0) GO TO 1950
00149          DO 250 L=1,IX
00150          READ(IFIN) IDEN
00151          IF(IDEN.EQ.9) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00153          IF(IDEN.EQ.11) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00155          IF(IDEN.EQ.19) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00157 250      CONTINUE
00158          CALL MATLIS(9,'HM MATRIX',NDIM,NZ,NXC,RLABEL,CLABEL,RARRAY,
              +                CARRAY,0)
00159          GO TO 40
00160 260      CONTINUE
00161          IF(NZ.EQ.0.OR.NYS.EQ.0) GO TO 1950
```

```
00163          DO 270 L=1,IX
00164          READ(IFIN) IDEN
00165          IF(IDEN.EQ.4) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00167          IF(IDEN.EQ.9) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00169          IF(IDEN.EQ.20) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00171 270      CONTINUE
00172          CALL MATLIS(9,'FM MATRIX',NDIM,NZ,NYS,RLABEL,CLABEL,RARRAY,
              +            CARRAY,0)
00173          GO TO 40
00174 280      CONTINUE
00175          IF(NYO.EQ.0.OR.NXC.EQ.0) GO TO 1950
00177          DO 290 L=1,IX
00178          READ(IFIN) IDEN
00179          IF(IDEN.EQ.8) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00181          IF(IDEN.EQ.11) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00183          IF(IDEN.EQ.21) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00185 290      CONTINUE
00186          CALL MATLIS(9,'HO MATRIX',NDIM,NYO,NXC,RLABEL,CLABEL,RARRAY,
              +            CARRAY,0)
00187          GO TO 40
00188 300      CONTINUE
00189          IF(NYO.EQ.0.OR.NYS.EQ.0) GO TO 1950
00191          DO 310 L=1,IX
00192          READ(IFIN) IDEN
00193          IF(IDEN.EQ.4) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00195          IF(IDEN.EQ.8) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00197          IF(IDEN.EQ.22) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00199 310      CONTINUE
00200          CALL MATLIS(9,'FO MATRIX',NDIM,NX,NYS,RLABEL,CLABEL,RARRAY,
              +            CARRAY,0)
00201          GO TO 40
```

```
00202 320   CONTINUE
00203       IF(NXT.EQ.0) GO TO 1950
00205       DO 330 L=1,IX
00206       READ(IFIN) IDEN
00207       IF(IDEN.EQ.10) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00209       IF(IDEN.EQ.11) READ(IFIN) IDEN,NY,(RLABEL(I),I=1+NXS,NXS+NY)
00211       IF(IDEN.EQ.23) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00213 330   CONTINUE
00214       CALL MATLIS(8,'F MATRIX',NDIM,NXT,NXT,RLABEL,RLABEL,RARRAY,
           +            CARRAY,0)
00215       GO TO 40
00216 340   CONTINUE
00217       IF(NXT.EQ.0.OR.NU.EQ.0) GO TO 1950
00219       DO 350 L=1,IX
00220       READ(IFIN) IDEN
00221       IF(IDEN.EQ.2) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00223       IF(IDEN.EQ.10) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00225       IF(IDEN.EQ.11) READ(IFIN) IDEN,NY,(RLABEL(I),I=1+NXS,NXS+NY)
00227       IF(IDEN.EQ.24) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00229 350   CONTINUE
00230       CALL MATLIS(8,'G MATRIX',NDIM,NXT,NY,RLABEL,CLABEL,RARRAY,
           +            CARRAY,0)
00231       GO TO 40
00232 360   CONTINUE
00233       IF(NXT.EQ.0.OR.NV.EQ.0) GO TO 1950
00235       DO 370 L=1,IX
00236       READ(IFIN) IDEN
00237       IF(IDEN.EQ.3) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00239       IF(IDEN.EQ.10) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00241       IF(IDEN.EQ.11) READ(IFIN) IDEN,NY,(RLABEL(I),I=1+NXS,NXS+NY)
00243       IF(IDEN.EQ.25) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00245 370   CONTINUE
00246       CALL MATLIS(12,'GAMMA MATRIX',NDIM,NXT,NV,RLABEL,CLABEL,RARRAY,
           +            CARRAY,0)
00247       GO TO 40
00248 380   CONTINUE
00249       IF(NZ.EQ.0.OR.NXT.EQ.0) GO TO 1950
00251       DO 390 L=1,IX
00252       READ(IFIN) IDEN
00253       IF(IDEN.EQ.9) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00255       IF(IDEN.EQ.10) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00257       IF(IDEN.EQ.11) READ(IFIN) IDEN,NY,(CLABEL(I),I=1+NXS,NXS+NY)
00259       IF(IDEN.EQ.26) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00261 390   CONTINUE
00262       CALL MATLIS(8,'H MATRIX',NDIM,NZ,NXT,RLABEL,CLABEL,RARRAY,
           +            CARRAY,0)
00263       GO TO 40
00264 400   CONTINUE
00265       IF(NZT.EQ.0.OR.NXT.EQ.0) GO TO 1950
00267       DO 410 L=1,IX
00268       READ(IFIN) IDEN
00269       IF(IDEN.EQ.7) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00271       IF(IDEN.EQ.8) READ(IFIN) IDEN,NY,(RLABEL(I),I=1+NZ,NZ+NY)
00273       IF(IDEN.EQ.10) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
```

```
00275          IF(IDEN.EQ.11) READ(IFIN) IDEN,NY,(CLABEL(I),I=1+NXS,NXS+NY)
00277          IF(IDEN.EQ.27) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00279 410    CONTINUE
00280          CALL MATLIS(9,'HR MATRIX',NDIM,NZT,NXT,RLABEL,CLABEL,RARRAY,
              +                CARRAY,0)
00281          GO TO 40
00282 420    CONTINUE
00283          IF(NZT.EQ.0.OR.NYS.EQ.0) GO TO 1950
00285          DO 430 L=1,IX
00286          READ(IFIN) IDEN
00287          IF(IDEN.EQ.4) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00289          IF(IDEN.EQ.7) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00291          IF(IDEN.EQ.8) READ(IFIN) IDEN,NY,(RLABEL(I),I=1+NZ,NZ+NY)
00293          IF(IDEN.EQ.28) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00295 430    CONTINUE
00296          CALL MATLIS(9,'E1 MATRIX',NDIM,NZT,NYS,RLABEL,CLABEL,RARRAY,
              +                CARRAY,0)
00297          GO TO 40
00298 440    CONTINUE
00299          IF(NZ.EQ.0.OR.NYS.EQ.0) GO TO 1950
00301          DO 450 L=1,IX
00302          READ(IFIN) IDEN
00303          IF(IDEN.EQ.4) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00305          IF(IDEN.EQ.9) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00307          IF(IDEN.EQ.29) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00309 450    CONTINUE
00310          CALL MATLIS(9,'E2 MATRIX',NDIM,NZ,NYS,RLABEL,CLABEL,RARRAY,
              +                CARRAY,0)
00311          GO TO 40
00312 460    CONTINUE
00313          IF(NXT.EQ.0.OR.NYS.EQ.0) GO TO 1950
00315          DO 470 L=1,IX
00316          READ(IFIN) IDEN
00317          IF(IDEN.EQ.4) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00319          IF(IDEN.EQ.10) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00321          IF(IDEN.EQ.11) READ(IFIN) IDEN,NY,(RLABEL(I),I=1+NXS,NXS+NY)
00323          IF(IDEN.EQ.30) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00325 470    CONTINUE
00326          CALL MATLIS(9,'E3 MATRIX',NDIM,NXT,NYS,RLABEL,CLABEL,RARRAY,
              +                CARRAY,0)
00327          GO TO 40
00328 480    CONTINUE
00329          IF(NZ.EQ.0) GO TO 1950
00331          DO 490 L=1,IX
00332          READ(IFIN) IDEN
00333          IF(IDEN.EQ.32) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00335          IF(IDEN.EQ.34) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00337 490    CONTINUE
00338          CALL MATLIS(9,'TZ MATRIX',
              +      NDIM,NZ,NZ,RLABEL,RLABEL,RARRAY,CARRAY,0)
00339          GO TO 40
00340 500    CONTINUE
00341          IF(NXT.EQ.0) GO TO 1950
00343          DO 510 L=1,IX
```

```
00344          READ(IFIN) IDEN
00345          IF(IDEN.EQ.10) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00347          IF(IDEN.EQ.11) READ(IFIN) IDEN,NY,(CLABEL(I),I=1+NXS,NXS+NY)
00349          IF(IDEN.EQ.33) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00351          IF(IDEN.EQ.IX-1) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),
              +       I=1,NX)
00353 510      CONTINUE
00354          IF(IX.EQ.36) CALL MATLIS(8,'T MATRIX'
              +    ,NDIM,NXT,NXT,RLABEL,CLABEL,RARRAY,CARRAY,0)
00356          IF(IX.EQ.37) CALL MATLIS(9,'TI MATRIX'
              +    ,NDIM,NXT,NXT,CLABEL,RLABEL,RARRAY,CARRAY,0)
00358          GO TO 40
00359 540      CONTINUE
00360          IF(M1P.EQ.0) GO TO 1950
00362          DO 550 L=1,IX
00363          READ(IFIN) IDEN
00364          IF(IDEN.EQ.33) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00366          IF(IDEN.EQ.37) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00368 550      CONTINUE
00369          CALL MATLIS(10,'F11 MATRIX',NDIM,M1P,M1P,RLABEL,RLABEL,RARRAY,
              +    CARRAY,0)
00370          GO TO 40
00371 560      CONTINUE
00372          IF(M1P.EQ.0.OR.M2.EQ.0) GO TO 1950
00374          DO 570 L=1,IX
00375          READ(IFIN) IDEN
00376          IF(IDEN.EQ.33) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,M1P),
              +     (CLABEL(I),I=1,M2)
00378          IF(IDEN.EQ.38) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00380 570      CONTINUE
00381          CALL MATLIS(10,'F12 MATRIX',NDIM,M1P,M2,RLABEL,CLABEL,RARRAY,
              +    CARRAY,0)
00382          GO TO 40
00383 580      CONTINUE
00384          IF(M2.EQ.0.OR.M1P.EQ.0) GO TO 1950
00386          DO 590 L=1,IX
00387          READ(IFIN) IDEN
00388          IF(IDEN.EQ.33) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,M1P),
              +       (RLABEL(I),I=1,M2)
00390          IF(IDEN.EQ.39) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00392 590      CONTINUE
00393          CALL MATLIS(10,'F21 MATRIX',NDIM,M2,M1P,RLABEL,CLABEL,RARRAY,
              +    CARRAY,0)
00394          GO TO 40
```

```
00395 600   CONTINUE
00396       IF(M2.EQ.0) GO TO 1950
00398       DO 610 L=1,IX
00399       READ(IFIN) IDEN
00400       IF(IDEN.EQ.33.AND.M1P.NE.0) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,M1P)
           +      ,(RLABEL(I),I=1,M2)
00402       IF(IDEN.EQ.33.AND.M1P.EQ.0) READ(IFIN) IDEN,NX,
           +       (RLABEL(I),I=1,M2)
00404       IF(IDEN.EQ.40) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00405 610   CONTINUE
00407       CALL MATLIS(10,'F22 MATRIX',
           +     NDIM,M2,M2,RLABEL,RLABEL,RARRAY,CARRAY,0)
00408       GO TO 40
00409 620   CONTINUE
00410       IF(M1P.EQ.0.OR.NU.EQ.0) GO TO 1950
00412       DO 630 L=1,IX
00413       READ(IFIN) IDEN
00414       IF(IDEN.EQ.2) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00416       IF(IDEN.EQ.33) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00418       IF(IDEN.EQ.41) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00420 630   CONTINUE
00421       CALL MATLIS(9,'G1 MATRIX',
           +     NDIM,M1P,NU,RLABEL,CLABEL,RARRAY,CARRAY,0)
00422       GO TO 40
00423 640   CONTINUE
00424       IF(M2.EQ.0.OR.NU.EQ.0) GO TO 1950
00426       DO 650 L=1,IX
00427       READ(IFIN) IDEN
00428       IF(IDEN.EQ.2) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00430       IF(IDEN.EQ.33.AND.M1P.NE.0) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,M1P)
           +      ,(RLABEL(I),I=1,M2)
00432       IF(IDEN.EQ.33.AND.M1P.EQ.0) READ(IFIN) IDEN,NX,
           +       (RLABEL(I),I=1,NX)
00434       IF(IDEN.EQ.42) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00435 650   CONTINUE
00437       CALL MATLIS(9,'G2 MATRIX',
           +     NDIM,M2,NU,RLABEL,CLABEL,RARRAY,CARRAY,0)
00438       GO TO 40
00439 660   CONTINUE
00440       IF(M1P.EQ.0.OR.NW.EQ.0) GO TO 1950
00442       DO 670 L=1,IX
00443       READ(IFIN) IDEN
00444       IF(IDEN.EQ.3) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00446       IF(IDEN.EQ.33) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00448       IF(IDEN.EQ.43) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00450 670   CONTINUE
00451       CALL MATLIS(13,'GAMMA1 MATRIX'
           +     ,NDIM,M1P,NW,RLABEL,CLABEL,RARRAY,CARRAY,0)
00452       GO TO 40
00453 680   CONTINUE
00454       IF(M2.EQ.0.OR.NW.EQ.0) GO TO 1950
00455       DO 690 L=1,IX
00457       READ(IFIN) IDEN
00458       IF(IDEN.EQ.3) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
```

```
00460       IF(IDEN.EQ.33.AND.M1P.NE.0) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,M1P)
           +     ,(RLABEL(I),I=1,M2)
00462       IF(IDEN.EQ.33.AND.M1P.EQ.0) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00464       IF(IDEN.EQ.44) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00465 690   CONTINUE
00467       CALL MATLIS(13,'GAMMA2 MATRIX'
           +     ,NDIM,M2,NN,RLABEL,CLABEL,RARRAY,CARRAY,0)
00468       GO TO 40
00469 700   CONTINUE
00470       IF(NZ.EQ.0.OR.NXT.EQ.0) GO TO 1950
00472       DO 710 L=1,IX
00473       READ(IFIN) IDEN
00474       IF(IDEN.EQ.10) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00476       IF(IDEN.EQ.11) READ(IFIN) IDEN,NY,(CLABEL(I),I=1+NXS,NXS+NY)
00478       IF(IDEN.EQ.32) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00480       IF(IDEN.EQ.45) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00482 710   CONTINUE
00483       CALL MATLIS(18,'REORDERED H MATRIX'
           +     ,NDIM,NZ,NXT,RLABEL,CLABEL,RARRAY,CARRAY,0)
00484       GO TO 40
00485 720   CONTINUE
00486       IF(M1.EQ.0.OR.M1P.EQ.0) GO TO 1950
00488       DO 730 L=1,IX
00489       READ(IFIN) IDEN
00490       IF(IDEN.EQ.32) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00492       IF(IDEN.EQ.33) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00494       IF(IDEN.EQ.46) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00496 730   CONTINUE
00497       CALL MATLIS(10,'H11 MATRIX'
           +     ,NDIM,M1,M1P,RLABEL,CLABEL,RARRAY,CARRAY,0)
00498       GO TO 40
00499 740   CONTINUE
00500       IF(M1.EQ.0.OR.M2.EQ.0) GO TO 1950
00502       DO 750 L=1,IX
00503       READ(IFIN) IDEN
00504       IF(IDEN.EQ.33.AND.M1P.NE.0) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,M1P)
           +     ,(CLABEL(I),I=1,M2)
00506       IF(IDEN.EQ.33.AND.M1P.EQ.0) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00508       IF(IDEN.EQ.33) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00510       IF(IDEN.EQ.47) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00512 750   CONTINUE
00513       CALL MATLIS(10,'H12 MATRIX'
           +     ,NDIM,M1,M2,RLABEL,CLABEL,RARRAY,CARRAY,0)
00514       GO TO 40
00515 760   CONTINUE
00516       IF(M2.EQ.0) GO TO 1950
00518       DO 770 L=1,IX
00519       READ(IFIN) IDEN
00520       IF(IDEN.EQ.32.AND.M1.NE.0) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,M1),
           +     (RLABEL(I),I=1,M2)
00522       IF(IDEN.EQ.32.AND.M1.EQ.0) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00524       IF(IDEN.EQ.33.AND.M1P.NE.0) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,M1P)
           +     ,(CLABEL(I),I=1,M2)
00526       IF(IDEN.EQ.33.AND.M1P.EQ.0) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
```

```
00528         IF(IDEN.EQ.48) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00530 770     CONTINUE
00531         CALL MATLIS(10,'H22 MATRIX'
             +     ,NDIM,M2,M2,RLABEL,CLABEL,RARRAY,CARRAY,0)
00532         GO TO 40
00533 780     CONTINUE
00534         IF(NW.EQ.0) GO TO 1950
00536         DO 790 L=1,IX
00537         READ(IFIN) IDEN
00538         IF(IDEN.EQ.3) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00540         IF(IDEN.EQ.49) READ(IFIN) IDEN,NX,(RARRAY(I,1),I=1,NX)
00542 790     CONTINUE
00543         NC=1
00544         CLABEL(1)=BLANK
00545         CALL MATLIS(25,'Q MATRIX DIAGONAL, FILTER'
             +     ,NDIM,NW,NC,RLABEL,CLABEL,RARRAY,CARRAY,0)
00546         GO TO 40
00547 800     CONTINUE
00548         IF(NZ.EQ.0) GO TO 1950
00550         DO 810 L=1,IX
00551         READ(IFIN) IDEN
00552         IF(IDEN.EQ.32) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00554         IF(IDEN.EQ.50) READ(IFIN) IDEN,NX,(RARRAY(I,1),I=1,NX)
00556 810     CONTINUE
00557         NC=1
00558         CLABEL(1)=BLANK
00559         CALL MATLIS(25,'R MATRIX DIAGONAL, FILTER'
             +     ,NDIM,NZ,NC,RLABEL,CLABEL,RARRAY,CARRAY,0)
00560         GO TO 40
00561 820     CONTINUE
00562         IF(M1P.EQ.0) GO TO 1950
00564         DO 830 L=1,IX
00565         READ(IFIN) IDEN
00566         IF(IDEN.EQ.33) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00568         IF(IDEN.EQ.51) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00570 830     CONTINUE
00571         DO 835 I=1,M1P
00572 835     RLABEL(I+M1P)=RLABEL(I)
00573         NR=2*M1P
00574         CALL MATLIS(33,'EULER LAGRANGE SYS MATRIX, FILTER'
             +     ,NDIM,NR,NR,RLABEL,RLABEL,RARRAY,CARRAY,0)
00575         GO TO 40
00576 840     CONTINUE
00577         IF(M2.EQ.0) GO TO 1950
00579         DO 850 L=1,IX
00580         READ(IFIN) IDEN
00581         IF(IDEN.EQ.33.AND.M1P.EQ.0) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00583         IF(IDEN.EQ.33.AND.M1P.NE.0) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,M1P)
             +     ,(RLABEL(I),I=1,M2)
00585         IF(IDEN.EQ.52) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00587 850     CONTINUE
00588         CALL MATLIS(36,'AUGMENTED GM2*QD*GM2T INVERSE MATRIX'
             +     ,NDIM,M2,M2,RLABEL,RLABEL,RARRAY,CARRAY,0)
00589         GO TO 40
```

```
00590 930   CONTINUE
00591       IF(M1P.EQ.0) GO TO 1950
00593       DO 910 L=1,IX
00594       READ(IFIN) IDEN
00595       IF(IDEN.EQ.33) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00597       IF(IDEN.EQ.IX-1) READ(IFIN) IDEN,NX,(BARRAY(I),I=1,NX)
00599 910   CONTINUE
00600       DO 915 I=1,M1P
00601       RARRAY(I,1)=CABS(BARRAY(I))
00602 915   CARRAY(I,1)=57.29578*ATAN2(AIMAG(BARRAY(I)),REAL(BARRAY(I)))
00603       NC=1
00604       CLABEL(1)=BLANK
00605       IF(IX.EQ.55) CALL MATLIS(29,'OPEN LOOP EIGENVALUES, FILTER'
          +       ,NDIM,M1P,NC,RLABEL,CLABEL,RARRAY,CARRAY,1)
00607       IF(IX.EQ.56) CALL MATLIS(31,'CLOSED LOOP EIGENVALUES, FILTER'
          +       ,NDIM,M1P,NC,RLABEL,CLABEL,RARRAY,CARRAY,1)
00609       GO TO 40
00610 920   CONTINUE
00611       IF(M1P.EQ.0) GO TO 1950
00613       DO 930 L=1,IX
00614       READ(IFIN) IDEN
00615       IF(IDEN.EQ.33) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00617       IF(IDEN.EQ.56) READ(IFIN) IDEN,NX,NY,((C1PRAY(I,J),J=1,NY),I=1,NX)
00619 930   CONTINUE
00620       DO 935 I=1,M1P
00621       DO 935 J=1,M1P
00622       RARRAY(I,J)=REAL(C1PRAY(I,J))
00623 935   CARRAY(I,J)=AIMAG(C1PRAY(I,J))
00624       CALL MATLIS(32,'CLOSED LOOP EIGENVECTORS, FILTER'
          +       ,NDIM,M1P,M1P,RLABEL,RLABEL,RARRAY,CARRAY,1)
00625       GO TO 40
00626 940   CONTINUE
00627       IF(M1P.EQ.0) GO TO 1950
00629       DO 950 L=1,IX
00630       READ(IFIN) IDEN
00631       IF(IDEN.EQ.33) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00633       IF(IDEN.EQ.57) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00635 950   CONTINUE
00636       CALL MATLIS(32,'RICCATI MATRIX SOLUTIONS, FILTER'
          +       ,NDIM,M1P,M1P,RLABEL,RLABEL,RARRAY,CARRAY,0)
00637       GO TO 40
00638 960   CONTINUE
00639       IF(M1P.EQ.0.OR.M1.EQ.0) GO TO 1950
00641       DO 970 L=1,IX
00642       READ(IFIN) IDEN
00643       IF(IDEN.EQ.32) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00645       IF(IDEN.EQ.33) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00647       IF(IDEN.EQ.58) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00649 970   CONTINUE
00650       CALL MATLIS(23,'K11 GAIN MATRIX, FILTER'
          +       ,NDIM,M1P,M1,RLABEL,CLABEL,RARRAY,CARRAY,0)
00651       GO TO 40
00652 980   CONTINUE
00653       IF(M1P.EQ.0.OR.M2.EQ.0) GO TO 1950
```

```
00655       DO 990 L=1,IX
00656       READ(IFIN) IDEN
00657       IF(IDEN.EQ.32.AND.M1.NE.0) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,M1)
           +      ,(CLABEL(I),I=1,M2)
00659       IF(IDEN.EQ.32.AND.M1.EQ.0) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00661       IF(IDEN.EQ.33) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00663       IF(IDEN.EQ.59) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00665 990   CONTINUE
00666       CALL MATLIS(23,'K12 GAIN MATRIX, FILTER'
           +     ,NDIM,M1P,M2,RLABEL,CLABEL,RARRAY,CARRAY,0)
00667       GO TO 40
00668 1000  CONTINUE
00669       IF(NXT.EQ.0) GO TO 1950
00671       DO 1010 L=1,IX
00672       READ(IFIN) IDEN
00673       IF(IDEN.EQ.10) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00675       IF(IDEN.EQ.11) READ(IFIN) IDEN,NY,(RLABEL(I),I=1+NXS,NXS+NY)
00677       IF(IDEN.EQ.60) READ(IFIN) IDEN,NX,(RARRAY(I,1),I=1,NX)
00679 1010  CONTINUE
00680       NC=1
00681       CLABEL(1)=BLANK
00682       CALL MATLIS(27,'R4S STATE EST ERROR, FILTER'
           +     ,NDIM,NXT,NC,RLABEL,CLABEL,RARRAY,CARRAY,0)
00683       GO TO 40
00684 1020  CONTINUE
00685       IF(NXT.EQ.0) GO TO 1950
00687       DO 1030 L=1,IX
00688       READ(IFIN) IDEN
00689       IF(IDEN.EQ.10) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00691       IF(IDEN.EQ.11) READ(IFIN) IDEN,NY,(RLABEL(I),I=1+NXS,NXS+NY)
00693       IF(IDEN.EQ.61) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00695 1030  CONTINUE
00696       CALL MATLIS(34,'STATE EST ERROR COV MATRIX, FILTER'
           +     ,NDIM,NXT,NXT,RLABEL,RLABEL,RARRAY,CARRAY,0)
00697       GO TO 40
00698 1040  CONTINUE
00699       IF(NZT.EQ.0) GO TO 1950
00701       DO 1050 L=1,IX
00702       READ(IFIN) IDEN
00703       IF(IDEN.EQ.7) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00705       IF(IDEN.EQ.8) READ(IFIN) IDEN,NY,(RLABEL(I),I=1+NZ,NZ+NY)
00707       IF(IDEN.EQ.62) READ(IFIN) IDEN,NX,(RARRAY(I,1),I=1,NX)
00709 1050  CONTINUE
00710       NC=1
00711       CLABEL(1)=BLANK
00712       CALL MATLIS(26,'Q MATRIX DIAGONAL, REGULATOR'
           +     ,NDIM,NZT,NC,RLABEL,CLABEL,RARRAY,CARRAY,0)
00713       GO TO 40
00714 1060  CONTINUE
00715       IF(NU.EQ.0) GO TO 1950
00717       DO 1070 L=1,IX
00718       READ(IFIN) IDEN
00719       IF(IDEN.EQ.2) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00721       IF(IDEN.EQ.63) READ(IFIN) IDEN,NX,(RARRAY(I,1),I=1,NX)
```

```
00723 1070  CONTINUE
00724       NC=1
00725       CLABEL(1)=BLANK
00726       CALL MATLIS(28,'R MATRIX DIAGONAL, REGULATOR'
           +     ,NDIM,NU,NC,RLABEL,CLABEL,RARRAY,CARRAY,0)
00727       GO TO 40
00728 1080  CONTINUE
00729       IF(NXT.EQ.0) GO TO 1950
00731       DO 1090 L=1,IX
00732       READ(IFIN) IDEN
00733       IF(IDEN.EQ.10) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00735       IF(IDEN.EQ.11) READ(IFIN) IDEN,NY,(RLABEL(I),I=1+NXS,NXS+NY)
00737       IF(IDEN.EQ.64) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00739 1090  CONTINUE
00740       DO 1095 I=1,NXT
00741 1095  RLABEL(I+NXT)=RLABEL(I)
00742       NR=2*NXT
00743       CALL MATLIS(36,'EULER LAGRANGE SYS MATRIX, REGULATOR'
           +     ,NDIM,NR,NR,RLABEL,RLABEL,RARRAY,CARRAY,0)
00744       GO TO 40
00745 1100  CONTINUE
00746       IF(NXT.EQ.0) GO TO 1950
00748       DO 1110 L=1,IX
00749       READ(IFIN) IDEN
00750       IF(IDEN.EQ.10) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00752       IF(IDEN.EQ.11) READ(IFIN) IDEN,NY,(RLABEL(I),I=1+NXS,NXS+NY)
00754       IF(IDEN.EQ.IX-1) READ(IFIN) IDEN,NX,(RARRAY(I),I=1,NX)
00756 1110  CONTINUE
00757       DO 1115 I=1,NXT
00758       RARRAY(I,1)=CABS(RARRAY(I))
00759 1115  CARRAY(I,1)=57.29578*ATAN2(AIMAG(RARRAY(I)),REAL(RARRAY(I)))
00760       NC=1
00761       CLABEL(1)=BLANK
00762       IF(IX.EQ.66) CALL MATLIS(32,'OPEN LOOP EIGENVALUES, REGULATOR'
           +     ,NDIM,NXT,NC,RLABEL,CLABEL,RARRAY,CARRAY,1)
00764       IF(IX.EQ.67) CALL MATLIS(34,'CLOSED LOOP EIGENVALUES, REGULATOR'
           +     ,NDIM,NXT,NC,RLABEL,CLABEL,RARRAY,CARRAY,1)
00766       GO TO 40
00767 1140  CONTINUE
00768       IF(NXT.EQ.0) GO TO 1950
00770       DO 1150 L=1,IX
00771       READ(IFIN) IDEN
00772       IF(IDEN.EQ.10) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00774       IF(IDEN.EQ.11) READ(IFIN) IDEN,NY,(RLABEL(I),I=1+NXS,NXS+NY)
00776       IF(IDEN.EQ.67) READ(IFIN) IDEN,NX,NY,((CMPRAY(I,J),J=1,NY),I=1,NX)
00778 1150  CONTINUE
00779       DO 1155 I=1,NXT
00780       DO 1155 J=1,NXT
00781       RARRAY(I,J)=REAL(CMPRAY(I,J))
00782 1155  CARRAY(I,J)=AIMAG(CMPRAY(I,J))
00783       CALL MATLIS(35,'CLOSED LOOP EIGENVECTORS, REGULATOR'
           +     ,NDIM,NXT,NXT,RLABEL,RLABEL,RARRAY,CARRAY,1)
00784       GO TO 40
00785 1160  CONTINUE
```

```
00786         IF(NXT.EQ.0) GO TO 1950
00788         DO 1170 L=1,IX
00789         READ(IFIN) IDEN
00790         IF(IDEN.EQ.10) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00792         IF(IDEN.EQ.11) READ(IFIN) IDEN,NY,(RLABEL(I),I=1+NXS,NXS+NY)
00794         IF(IDEN.EQ.68) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00795 1170  CONTINUE
00797         CALL MATLIS(35,'RICCATI MATRIX SOLUTIONS, REGULATOR'
             +     ,NDIM,NXT,NXT,RLABEL,RLABEL,RARRAY,CARRAY,0)
00798         GO TO 40
00799 1180  CONTINUE
00800         IF(NU.EQ.0.OR.NXT.EQ.0) GO TO 1950
00802         DO 1190 L=1,IX
00803         READ(IFIN) IDEN
00804         IF(IDEN.EQ.2) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00806         IF(IDEN.EQ.10) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00808         IF(IDEN.EQ.11) READ(IFIN) IDEN,NY,(CLABEL(I),I=1+NXS,NXS+NY)
00810         IF(IDEN.EQ.69) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),I=1,NX)
00812 1190  CONTINUE
00813         CALL MATLIS(21,'REGULATOR GAIN MATRIX'
             +     ,NDIM,NU,NXT,RLABEL,CLABEL,RARRAY,CARRAY,0)
00814         GO TO 40
```

```
00815 1200  CONTINUE
00816       IF(NXT.EQ.0) GO TO 1950
00818       DO 1210 L=1,IX
00819       READ(IFIN) IDEN
00820       IF(IDEN.EQ.10) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00822       IF(IDEN.EQ.11) READ(IFIN) IDEN,NY,(RLABEL(I),I=1+NXS,NXS+NY)
00824       IF(IDEN.EQ.66) READ(IFIN) IDEN,NX,(BARRAY(I),I=1,NX)
00826       IF(IDEN.EQ.IX-1) READ(IFIN) IDEN,NX,NY,((CMPRAY(I,J),J=1,NY),
           +       I=1,NX)
00828 1210  CONTINUE
00829       K=0
00830       DO 1212 I=1,NXT
00831       IF(AIMAG(BARRAY(I)).LT.0.) GO TO 1212
00833       K=K+1
00834       BARRAY(K)=BARRAY(I)
00835 1212  CONTINUE
00836       L=(IDEN-68)/3
00837       WRITE(IWO,1213) BARRAY(L)
00838 1213  FORMAT(5X,'NORMALIZED F MATRIX SENSITIVITY TO EIGENVALUE : ',
           +       2G11.3/)
00839       DO 1215 I=1,NXT
00840       DO 1215 J=1,NXT
00841       RARRAY(I,J)=CABS(CMPRAY(I,J))
00842 1215  CARRAY(I,J)=57.29578*ATAN2(AIMAG(CMPRAY(I,J)),REAL(CMPRAY(I,J)))
00843       CALL MATLIS(8,DBLANK,NDIM,NXT,NXT,RLABEL,RLABEL,RARRAY,CARRAY,1)
00844       GO TO 40
00845 1220  CONTINUE
00846       IF(NXT.EQ.0.OR.NU.EQ.0)  GO TO 1950
00848       DO 1230 L=1,IX
00849       READ(IFIN) IDEN
00850       IF(IDEN.EQ.2) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00852       IF(IDEN.EQ.10) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00854       IF(IDEN.EQ.11) READ(IFIN) IDEN,NY,(RLABEL(I),I=1+NXS,NXS+NY)
00856       IF(IDEN.EQ.66) READ(IFIN) IDEN,NX,(BARRAY(I),I=1,NX)
00858       IF(IDEN.EQ.IX-1) READ(IFIN) IDEN,NX,NY,((CMPRAY(I,J),J=1,NY),
           +       I=1,NX)
00860 1230  CONTINUE
00861       K=0
00862       DO 1232 I=1,NXT
00863       IF(AIMAG(BARRAY(I)).LT.0.) GO TO 1232
00865       K=K+1
00866       BARRAY(K)=BARRAY(I)
00867 1232  CONTINUE
00868       L=(IDEN-69)/3
00869       WRITE(IWO,1233) BARRAY(L)
00870 1233  FORMAT(5X,'NORMALIZED G MATRIX SENSITIVITY TO EIGENVALUE : ',
           +       2G11.3)
00871       DO 1235 I=1,NXT
00872       DO 1235 J=1,NU
00873       RARRAY(I,J)=CABS(CMPRAY(I,J))
00874 1235  CARRAY(I,J)=57.29578*ATAN2(AIMAG(CMPRAY(I,J)),REAL(CMPRAY(I,J)))
00875       CALL MATLIS(8,DBLANK,NDIM,NXT,NU,RLABEL,CLABEL,RARRAY,CARRAY,1)
00876       GO TO 40
00877 1240  CONTINUE
```

```
00870         IF(NU.EQ.0.OR.NXT.EQ.0) GO TO 1950
00890         DO 1250 L=1,IX
00891         READ(IFIN) IDEN
00892         IF(IDEN.EQ.2) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00894         IF(IDEN.EQ.10) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00896         IF(IDEN.EQ.11) READ(IFIN) IDEN,NY,(CLABEL(I),I=1+NXS,NXS+NY)
00898         IF(IDEN.EQ.66) READ(IFIN) IDEN,NX,(BARRAY(I),I=1,NX)
00890         IF(IDEN.EQ.IX-1) READ(IFIN) IDEN,NX,NY,((CMPRAY(I,J),J=1,NY),
             +      I=1,NX)
00892 1250    CONTINUE
00893         K=0
00894         DO 1252 I=1,NXT
00895         IF(AIMAG(BARRAY(I)).LT.0.) GO TO 1252
00897         K=K+1
00898         BARRAY(K)=BARRAY(I)
00899 1252    CONTINUE
00900         L=(IDEN-70)/3
00901         WRITE(IKB,1253) BARRAY(L)
00902 1253    FORMAT(5X,'NORMALIZED C MATRIX SENSITIVITY TO EIGENVALUE : ',
             +     2G11.3)
00903         DO 1255 I=1,NU
00904         DO 1255 J=1,NXT
00905         BARRAY(I,J)=CABS(CMPRAY(I,J))
00906 1255    CARRAY(I,J)=57.29578*ATAN2(AIMAG(CMPRAY(I,J)),REAL(CMPRAY(I,J)))
00907         CALL MATLIS(8,DBLANK,NDIM,NU,NXT,RLABEL,CLABEL,BARRAY,CARRAY,1)
00908         GO TO 40
00909 1260    CONTINUE
00910         IF(M1P.EQ.0) GO TO 1950
00912         DO 1270 L=1,IX
00913         READ(IFIN) IDEN
00914         IF(IDEN.EQ.33) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00916         IF(IDEN.EQ.115) READ(IFIN) IDEN,NX,NY,((BARRAY(I,J),J=1,NY),
             +      I=1,NX)
00918 1270    CONTINUE
00919         CALL MATLIS(9,'AF MATRIX'
             +     ,NDIM,M1P,M1P,RLABEL,RLABEL,BARRAY,CARRAY,0)
00920         GO TO 40
00921 1280    CONTINUE
00922         IF(M1P.EQ.0.OR.NZ.EQ.0) GO TO 1950
00924         DO 1290 L=1,IX
00925         READ(IFIN) IDEN
00926         IF(IDEN.EQ.9) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00928         IF(IDEN.EQ.33) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00930         IF(IDEN.EQ.116) READ(IFIN) IDEN,NX,NY,((BARRAY(I,J),J=1,NY),
             +     I=1,NX)
00932 1290    CONTINUE
00933         CALL MATLIS(9,'BF MATRIX'
             +     ,NDIM,M1P,NZ,RLABEL,CLABEL,BARRAY,CARRAY,0)
00934         GO TO 40
00935 1300    CONTINUE
00936         IF(NU.EQ.0.OR.M1P.EQ.0) GO TO 1950
00938         DO 1310 L=1,IX
00939         READ(IFIN) IDEN
00940         IF(IDEN.EQ.2) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
```

```
00942         IF(IDEN.EQ.33) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00944         IF(IDEN.EQ.117) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),
             +      I=1,NX)
00946 1310    CONTINUE
00947         CALL MATLIS(9,'CF MATRIX'
             +      ,NDIM,NU,NZ,RLABEL,CLABEL,RARRAY,CARRAY,0)
00948         GO TO 40
00949 1320    CONTINUE
00950         IF(NU.EQ.0.OR.NZ.EQ.0) GO TO 1950
00952         DO 1330 L=1,IX
00953         READ(IFIN) IDEN
00954         IF(IDEN.EQ.2) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00955         IF(IDEN.EQ.9) READ(IFIN) IDEN,NX,(CLABEL(I),I=1,NX)
00958         IF(IDEN.EQ.118) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),
             +      I=1,NX)
00960 1330    CONTINUE
00961         CALL MATLIS(9,'DF MATRIX'
             +      ,NDIM,NU,NZ,RLABEL,CLABEL,RARRAY,CARRAY,0)
00962         GO TO 40
00963 1340    CONTINUE
00964         IF(NXT+M1P.EQ.0) GO TO 1950
00966         DO 1350 L=1,IX
00967         READ(IFIN) IDEN
00968         IF(IDEN.EQ.10) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00970         IF(IDEN.EQ.11) READ(IFIN) IDEN,NY,(RLABEL(I),I=1+NXS,NXS+NY)
00972         IF(IDEN.EQ.33) READ(IFIN) IDEN,NY,(RLABEL(I),I=1+NXT,NXT+NY)
00974         IF(IDEN.EQ.IX-1) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),
             +      I=1,NX)
00976 1350    CONTINUE
00977         NR=NXT+M1P
00978         IF(IX.EQ.120) CALL MATLIS(29,'CLOSED LOOP SYS SYSTEM MATRIX'
             +      ,NDIM,NR,NR,RLABEL,RLABEL,RARRAY,CARRAY,0)
00980         IF(IX.EQ.121) CALL MATLIS(25,'COV MATRIX, PROCESS NOISE'
             +      ,NDIM,NR,NR,RLABEL,RLABEL,RARRAY,CARRAY,0)
00982         IF(IX.EQ.123) CALL MATLIS(29,'COV MATRIX, MEASUREMENT NOISE'
             +      ,NDIM,NR,NR,RLABEL,RLABEL,RARRAY,CARRAY,0)
00984         IF(IX.EQ.125) CALL MATLIS(17,'COV MATRIX, TOTAL'
             +      ,NDIM,NR,NR,RLABEL,RLABEL,RARRAY,CARRAY,0)
00986         GO TO 40
00987 1360    CONTINUE
00988         IF(NXT+M1P.EQ.0) GO TO 1950
00990         DO 1370 L=1,IX
00991         READ(IFIN) IDEN
00992         IF(IDEN.EQ.10) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
00994         IF(IDEN.EQ.11) READ(IFIN) IDEN,NY,(RLABEL(I),I=1+NXS,NXS+NY)
00996         IF(IDEN.EQ.33) READ(IFIN) IDEN,NY,(RLABEL(I),I=1+NXT,NXT+NY)
00998         IF(IDEN.EQ.IX-1) READ(IFIN) IDEN,NX,(RARRAY(I,1),I=1,NX)
01000 1370    CONTINUE
01001         NR=NXT+M1P
01002         CLABEL(1)=BLANK
01003         NC=1
01004         IF(IX.EQ.122) CALL MATLIS(18,'RMS, PROCESS NOISE'
             +      ,NDIM,NR,NC,RLABEL,CLABEL,RARRAY,CARRAY,0)
01006         IF(IX.EQ.124) CALL MATLIS(21,'RMS, MEASUREMENT NOISE'
```

```
     +      ,NDIM,NR,NC,RLABEL,CLABEL,RARRAY,CARRAY,Ø)
Ø1ØØ8        IF(IX.EQ.126) CALL MATLIS(1Ø,'R4S, TOTAL'
     +      ,NDIM,NR,NC,RLABEL,CLABEL,RARRAY,CARRAY,Ø)
Ø1Ø1Ø         GO TO 4Ø
Ø1Ø11 1390   CONTINUE
Ø1Ø12        IF(NZ+NYO+NU.EQ.Ø) GO TO 1950
Ø1Ø14        DO 1390 L=1,IX
Ø1Ø15        READ(IFIN,END=1392) IDEN
Ø1Ø16        IF(IDEN.EQ.2.AND.NU.NE.Ø) READ(IFIN) IDEN,NX,(RLABEL(I),
     +          I=1+NZT,NZT+NX)
Ø1Ø18        IF(IDEN.EQ.8.AND.NYO.NE.Ø) READ(IFIN) IDEN,NX,(RLABEL(I),
     +          I=1+NZ,NZ+NX)
Ø1Ø2Ø        IF(IDEN.EQ.9) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
Ø1Ø22        IF(IDEN.EQ.IX-1) READ(IFIN) IDEN,NX,NY,((RARRAY(I,J),J=1,NY),
     +      I=1,NX)
Ø1Ø24 1390   CONTINUE
Ø1Ø25 1392   NR=NZ+NYO+NU
Ø1Ø26        IF(IX.EQ.127) CALL MATLIS(27,'COV MATRIX, OUTPUT+CONTROLS'
     +      ,NDIM,NR,NR,RLABEL,RLABEL,RARRAY,CARRAY,Ø)
Ø1Ø28        IF(IX.EQ.129) CALL MATLIS(27,'COR MATRIX, OUTPUT+CONTROLS'
     +      ,NDIM,NR,NR,RLABEL,RLABEL,RARRAY,CARRAY,Ø)
Ø1Ø3Ø         GO TO 4Ø
Ø1Ø31 1400   CONTINUE
Ø1Ø32        IF(NZ+NYO+NU.EQ.Ø) GO TO 1950
Ø1Ø34        DO 1410 L=1,IX
Ø1Ø35        READ(IFIN,END=1412) IDEN
Ø1Ø36        IF(IDEN.EQ.2.AND.NU.NE.Ø) READ(IFIN) IDEN,NX,(RLABEL(I),
     +       I=1+NZT,NZT+NX)
Ø1Ø38        IF(IDEN.EQ.8.AND.NYO.NE.Ø) READ(IFIN) IDEN,NX,(RLABEL(I),
     +       I=1+NZ,NZ+NX)
Ø1Ø4Ø         IF(IDEN.EQ.9) READ(IFIN) IDEN,NX,(RLABEL(I),I=1,NX)
Ø1Ø42        IF(IDEN.EQ.127) READ(IFIN) IDEN,NX,(RARRAY(I,1),I=1,NX)
Ø1Ø44 1410   CONTINUE
Ø1Ø45 1412   NR=NZ+NYO+NU
Ø1Ø46        NC=1
Ø1Ø47        CLABEL(1)=BLANK
Ø1Ø48        CALL MATLIS(2Ø,'R4S, OUTPUT+CONTROLS'
     +      ,NDIM,NR,NC,RLABEL,CLABEL,RARRAY,CARRAY,Ø)
Ø1Ø49         GO TO 4Ø
Ø1Ø5Ø 1950   WRITE(1Ø3,196Ø) IX
Ø1Ø51 196Ø   FORMAT(1X,'PROBLEM FILE ELEMENT ',I3,' HAS ZERO DIMENSIONS'/)
Ø1Ø52         GO TO 4Ø
Ø1Ø53 3ØØØ   STOP
Ø1Ø54        END
```

```
        00646RD 00548RD 00658RD 00660RD 00662RD 00664RD 00674RD 00676RD
        00678RD 00690RD 00692RD 00694RD 00704RD 00706RD 00708RD 00720RD
        00722RD 00734RD 00736RD 00738RD 00740DO 00741   00751RD 00753RD
        00755RD 00757DO 00758   00759   00773RD 00775RD 00777RD 00779DO
        00781   00782   00791RD 00793RD 00795RD 00805RD 00807RD 00809RD
        00811RD 00821RD 00823RD 00825RD 00827RD 00830DO 00931IF 00834
        00839DO 00841   00842   00851RD 00853RD 00855RD 00857RD 00859RD
        00862DO 00863IF 00866   00871DO 00873   00874   00883RD 00885RD
        00887RD 00889RD 00891RD 00894DO 00895IF 00898   00903DO 00905
        00906   00915RD 00917RD 00927RD 00929RD 00931RD 00941RD 00943RD
        00945RD 00955RD 00957RD 00959RD 00969RD 00971RD 00973RD 00975RD
        00993RD 00995RD 00997RD 00999RD 01017RD 01019RD 01021RD 01023RD
        01037RD 01039RD 01041RD 01043RD
IDEN    00016RD 00017IF 00018RD 00019IF 00020RD 00033RD 00044RD 00045IF
        00047RD 00056RD 00057IF 00058RD 00059IF 00060RD 00061IF 00062RD
        00070RD 00071IF 00072RD 00073IF 00074RD 00082RD 00083IF 00084RD
        00085IF 00086RD 00087IF 00088RD 00096RD 00097IF 00098RD 00099IF
        00100RD 00101IF 00102RD 00110RD 00111IF 00112RD 00113IF 00114RD
        00115IF 00116RD 00124RD 00125IF 00126RD 00127IF 00128RD 00129IF
        00130RD 00138RD 00139IF 00140RD 00141IF 00142RD 00150RD 00151IF
        00152RD 00153IF 00154RD 00155IF 00156RD 00164RD 00165IF 00166RD
        00167IF 00168RD 00169IF 00170RD 00178RD 00179IF 00180RD 00181IF
        00182RD 00183IF 00184RD 00192RD 00193IF 00194RD 00195IF 00196RD
        00197IF 00198RD 00206RD 00207IF 00208RD 00209IF 00210RD 00211IF
        00212RD 00220RD 00221IF 00222RD 00223IF 00224RD 00225IF 00226RD
        00227IF 00228RD 00236RD 00237IF 00238RD 00239IF 00240RD 00241IF
        00242RD 00243IF 00244RD 00252RD 00253IF 00254RD 00255IF 00256RD
        00257IF 00258RD 00259IF 00260RD 00268RD 00269IF 00270RD 00271IF
        00272RD 00273IF 00274RD 00275IF 00276RD 00277IF 00278RD 00286RD
        00287IF 00288RD 00289IF 00290RD 00291IF 00292RD 00293IF 00294RD
        00302RD 00303IF 00304RD 00305IF 00306RD 00307IF 00308RD 00316RD
        00317IF 00318RD 00319IF 00320RD 00321IF 00322RD 00323IF 00324RD
        00332RD 00333IF 00334RD 00335IF 00336RD 00344RD 00345IF 00346RD
        00347IF 00348RD 00349IF 00350RD 00351IF 00352RD 00363RD 00364IF
        00365RD 00366IF 00367RD 00375RD 00376IF 00377RD 00378IF 00379RD
        00387RD 00388IF 00389RD 00390IF 00391RD 00399RD 00400IF 00401RD
        00402IF 00403RD 00404IF 00405RD 00413RD 00414IF 00415RD 00416IF
        00417RD 00418IF 00419RD 00427RD 00428IF 00429RD 00430IF 00431RD
        00432IF 00433RD 00434IF 00435RD 00443RD 00444IF 00445RD 00446IF
        00447RD 00448IF 00449RD 00457RD 00458IF 00459RD 00460IF 00461RD
        00462IF 00463RD 00464IF 00465RD 00473RD 00474IF 00475RD 00476IF
        00477RD 00478IF 00479RD 00480IF 00481RD 00489RD 00490IF 00491RD
        00492IF 00493RD 00494IF 00495RD 00503RD 00504IF 00505RD 00506IF
        00507RD 00508IF 00509RD 00510IF 00511RD 00519RD 00520IF 00521RD
        00522IF 00523RD 00524IF 00525RD 00526IF 00527RD 00528IF 00529RD
        00537RD 00538IF 00539RD 00540IF 00541RD 00551RD 00552IF 00553RD
        00554IF 00555RD 00565RD 00566IF 00567RD 00568IF 00569RD 00580RD
        00581IF 00592RD 00593IF 00594RD 00595IF 00596RD 00604RD 00605IF
        00606RD 00607IF 00608RD 00614RD 00615IF 00616RD 00617IF 00618RD
        00630RD 00631IF 00632RD 00633IF 00634RD 00642RD 00643IF 00644RD
        00645IF 00646RD 00647IF 00648RD 00656RD 00657IF 00658RD 00659IF
        00660RD 00661IF 00662RD 00663IF 00664RD 00672RD 00673IF 00674RD
        00675IF 00676RD 00677IF 00678RD 00688RD 00689IF 00690RD 00691IF
        00692RD 00693IF 00694RD 00702RD 00703IF 00704RD 00705IF 00706RD
```

RT-11 INDEX V03 CROSS REFERENCE LISTING 10-APR-79 12:00:08 PAGE 00022

```
I       00004CM 00008=  00009AG 00034WR 00037WR 00040WR 00048WR 00937WR
        00869WR 00901WR 01050WR
IO      00004CN
IOUT    00006DA 00010WR 00025WR
IX      00027RD 00029IF 00032GT 00043DO 00045IF 00055DO 00059DO 00081DO
        00095DO 00109DO 00123DO 00137DO 00149DO 00163DO 00177DO 00191DO
        00205DO 00219DO 00235DO 00251DO 00267DO 00285DO 00301DO 00315DO
        00331DO 00343DO 00351IF 00354IF 00356IF 00362DO 00374DO 00386DO
        00398DO 00412DO 00426DO 00442DO 00456DO 00472DO 00488DO 00502DO
        00518DO 00536DO 00550DO 00564DO 00579DO 00593DO 00597IF 00605IF
        00607IF 00613DO 00629DO 00641DO 00655DO 00671DO 00687DO 00701DO
        00717DO 00731DO 00748DO 00754IF 00762IF 00764IF 00770DO 00788DO
        00802DO 00818DO 00826IF 00848DO 00858IF 00880DO 00890IF 00912DO
        00924DO 00938DO 00952DO 00966DO 00974IF 00978IF 00930IF 00982IF
        00984IF 00990DO 00998IF 01004IF 01006IF 01008IF 01014DO 01022IF
        01026IF 01028IF 01034DO 01050WR
J       00052RD 00074RD 00080RD 00102RD 00116RD 00130RD 00142RD 00156RD
        00170RD 00184RD 00198RD 00212RD 00228RD 00244RD 00258RD 00276RD
        00294RD 00308RD 00324RD 00336RD 00352RD 00367RD 00379RD 00391RD
        00405RD 00419RD 00435RD 00449RD 00465RD 00481RD 00495RD 00511RD
        00529RD 00569RD 00586RD 00618RD 00621DO 00622   00623   00634RD
        00648RD 00664RD 00694RD 00738RD 00777RD 00780DO 00781   00782
        00795RD 00811RD 00827RD 00840DO 00841   00842   00859RD 00872DO
        00873   00874   00891RD 00904DO 00905   00906   00917RD 00931RD
        00945RD 00959RD 00975RD 01023RD
K       00015DO 00829=  00833=  00834   00861=  00865=  00866   00893=
        00897=  00898
L       00043DO 00055DO 00069DO 00081DO 00095DO 00109DO 00123DO 00137DO
        00149DO 00163DO 00177DO 00191DO 00205DO 00219DO 00235DO 00251DO
        00267DO 00285DO 00301DO 00315DO 00331DO 00343DO 00362DO 00374DO
        00386DO 00398DO 00412DO 00426DO 00442DO 00456DO 00472DO 00488DO
        00502DO 00518DO 00536DO 00550DO 00564DO 00579DO 00593DO 00613DO
        00629DO 00641DO 00655DO 00671DO 00687DO 00701DO 00717DO 00731DO
        00748DO 00770DO 00788DO 00802DO 00818DO 00836=  00837WR 00848DO
        00868=  00869WR 00880DO 00900=  00901WR 00912DO 00924DO 00938DO
        00952DO 00966DO 00990DO 01014DO 01034DO
MATLIS  00064CL 00076CL 00090CL 00104CL 00118CL 00132CL 00144CL 00158CL
        00172CL 00186CL 00200CL 00214CL 00230CL 00246CL 00262CL 00280CL
        00296CL 00310CL 00326CL 00338CL 00355CL 00357CL 00369CL 00381CL
        00393CL 00407CL 00421CL 00437CL 00451CL 00467CL 00483CL 00497CL
        00513CL 00531CL 00545CL 00559CL 00574CL 00590CL 00606CL 00608CL
        00624CL 00636CL 00650CL 00666CL 00682CL 00696CL 00712CL 00726CL
        00743CL 00763CL 00765CL 00783CL 00797CL 00813CL 00843CL 00875CL
        00907CL 00919CL 00933CL 00947CL 00961CL 00979CL 00981CL 00983CL
        00985CL 01005CL 01007CL 01009CL 01027CL 01029CL 01048CL
M1      00020RD 00040WR 00486IF 00497AG 00500IF 00513AG 00520IF 00521RD
        00522IF 00639IF 00650AG 00657IF 00658RD 00659IF
M1P     00020RD 00040WR 00360IF 00369AG 00372IF 00377RD 00381AG 00384IF
        00389RD 00393AG 00400IF 00401RD 00402IF 00410IF 00421AG 00430IF
        00431RD 00432IF 00440IF 00451AG 00460IF 00461RD 00462IF 00480IF
        00497AG 00504IF 00505RD 00506IF 00524IF 00525RD 00526IF 00562IF
        00571DO 00572   00573   00581IF 00583IF 00584RD 00591IF 00600DO
        00606AG 00608AG 00611IF 00620DO 00621DO 00624AG 00627IF 00636AG
        00639IF 00650AG 00653IF 00656AG 00910IF 00919AG 00922IF 00933AG
```

```
            00355RD 00369AG 00377RD 00381AG 00389RD 00393AG 00401RD 00403RD
            00407AG 00417RD 00421AG 00431RD 00433RD 00437AG 00447RD 00451AG
            00461RD 00463RD 00467AG 00479RD 00483AG 00491RD 00497AG 00509RD
            00513AG 00521RD 00523RD 00531AG 00539RD 00545AG 00553RD 00559AG
            00567RD 00572=  00574AG 00582RD 00584RD 00588AG 00596RD 00600AG
            00608AG 00616RD 00624AG 00632RD 00636AG 00646RD 00650AG 00662RD
            00666AG 00674RD 00676RD 00682AG 00690RD 00692RD 00696AG 00704RD
            00706RD 00712AG 00720RD 00726AG 00734RD 00736RD 00741=  00743AG
            00751RD 00753RD 00763AG 00765AG 00773RD 00775RD 00783AG 00791RD
            00793RD 00797AG 00805RD 00813AG 00821RD 00823RD 00843AG 00853RD
            00855RD 00875AG 00883RD 00907AG 00915RD 00919AG 00929RD 00933AG
            00941RD 00947AG 00955RD 00961AG 00969RD 00971RD 00973RD 00979AG
            00981AG 00983AG 00985AG 00993RD 00995RD 00997RD 01005AG 01007AG
            01009AG 01017RD 01019RD 01021RD 01027AG 01029AG 01037RD 01039RD
            01041RD 01048AG
XDATE       000003DP
XTIME       000003DP
10          000107NR 00011*
100         000032GT 00052*
1000        000032GT 00568*
1010        000671DO 00679*
1020        000032GT 00684*
1030        000687DO 00695*
1040        000032GT 00698*
1050        000701DO 00709*
1060        000032GT 00714*
1070        000717DO 00723*
1080        000032GT 00728*
1090        000731DO 00739*
1095        000740DO 00741*
110         000055DO 00063*
1100        000032GT 00745*
1110        000748DO 00756*
1115        000757DO 00759*
1140        000032GT 00767*
1150        000770DO 00778*
1155        000779DO 000780DO 00782*
1160        000032GT 00785*
1170        000788DO 00796*
1180        000032GT 00799*
1190        000802DO 00812*
120         000032GT 00065*
1200        000032GT 00815*
1210        000818DO 00828*
1212        000830DO 000832GT 00835*
1213        000837NR 00838*
1215        000839DO 000840DO 00842*
1220        000032GT 00845*
1230        000848DO 00850*
1232        000962DO 000964GT 00867*
1233        000969NR 00870*
1235        000971DO 000972DO 00874*
1240        000032GT 00877*
1250        000980DO 00892*
```


C-3

```
1252    000894DO 000896GT 000899*
1253    000911NR 000912*
1255    000933DO 000940DO 000945*
1260    000832GT 000909*
1270    000912DO 000918*
1280    000832GT 000921*
1290    000924DO 000932*
130     000969DO 000975*
1300    000832GT 000935*
1310    000938DO 000946*
1320    000832GT 000949*
1330    000952DO 000959*
1340    000832GT 000963*
1350    000966DO 000976*
1360    000832GT 000987*
1370    000990DO 001000*
1380    000832GT 001011*
1390    001014DO 001024*
1392    001015RD 001025*
140     000832GT 000978*
1400    000832GT 001031*
1410    001034DO 001044*
1412    001035RD 001045*
150     000981DO 000989*
160     000832GT 000992*
170     000995DO 001003*
180     000832GT 001006*
190     001009DO 001017*
1950    000056GT 000068GT 000284GT 000394GT 000148GT 000122GT 000136GT 000148GT
        000162GT 000176GT 000190GT 000204GT 000218GT 000234GT 000250GT 000266GT
        000282GT 000300GT 000314GT 000330GT 000342GT 000361GT 000373GT 000385GT
        000397GT 000411GT 000425GT 000441GT 000455GT 000471GT 000487GT 000501GT
        000517GT 000535GT 000549GT 000563GT 000578GT 000592GT 000612GT 000620GT
        000644GT 000654GT 000670GT 000686GT 000704GT 000716GT 000730GT 000747GT
        000769GT 000787GT 000801GT 000817GT 000847GT 000879GT 000911GT 000923GT
        000937GT 000951GT 000965GT 000989GT 001013GT 001033GT 001050*
1960    001058NR 001051*
20      000812RD 000813*
200     000832GT 000120*
210     000120DO 000131*
220     000832GT 000134*
230     000137DO 000143*
240     000832GT 000146*
25      000150DO 000121*
250     000140DO 000157*
26      000116RD 000122*
260     000832GT 000160*
270     000163DO 000171*
280     000832GT 000174*
290     000177DO 000185*
30      000125NR 000126*
300     000832GT 000188*
3000    000834GT 001053*
310     000191DO 000199*
```

```
320     000332GT 00202*
330     000350D  00213*
340     000332GT 00216*
350     000219D  00229*
360     000332GT 00232*
370     000235D  00245*
380     000332GT 00248*
390     000251D  00261*
40      000027*  000336GT 000396GT 000426GT 000516GT 000556GT 000776GT 000916GT
        001056GT 001196GT 001336GT 001456GT 001596GT 001736GT 001876GT 002016GT
        002156GT 002316GT 002476GT 002636GT 002816GT 002976GT 003116GT 003276GT
        003396GT 003586GT 003746GT 003926GT 004076GT 004246GT 004386GT 004546GT
        004526GT 004686GT 004846GT 004996GT 005146GT 005326GT 005466GT 005606GT
        005756GT 005896GT 006056GT 006256GT 006376GT 006516GT 006576GT 006936GT
        006976GT 007136GT 007276GT 007446GT 007666GT 007846GT 007986GT 008146GT
        008446GT 008766GT 009086GT 009206GT 009346GT 009486GT 009526GT 009856GT
        010106GT 010306GT 010496GT 010526GT
400     000332GT 00264*
410     000267D  00279*
420     000332GT 00282*
430     000295D  00295*
440     000332GT 00298*
450     000301D  00309*
460     000332GT 00312*
470     000315D  00325*
480     000332GT 00328*
490     000331D  00337*
50      000027RD 00028*
500     000332GT 00340*
510     000343D  00353*
540     000332GT 00359*
550     000362D  00368*
560     000332GT 00371*
570     000374D  00382*
580     000332GT 00393*
590     000386D  00392*
600     000332GT 00395*
610     000399D  00406*
620     000332GT 00409*
630     000412D  00420*
640     000332GT 00423*
65      000332GT 00033*
650     000426D  00436*
660     000332GT 00439*
670     000442D  00450*
68      000334WR 00035*
680     000332GT 00453*
690     000456D  00466*
70      000332GT 00037*
700     000332GT 00469*
710     000472D  00482*
720     000332GT 00485*
730     000488D  00496*
740     000332GT 00499*
```

```
75      00037#R 00038*
750     00502D0 00512*
760     00032GT 00515*
770     00518D0 00530*
780     00032GT 00533*
790     00536D0 00542*
80      00032GT 00040*
800     00032GT 00547*
810     00550D0 00556*
820     00032GT 00561*
830     00564D0 00570*
835     00571D0 00572*
840     00032GT 00576*
85      00040#R 00041*
850     00579D0 00587*
90      00032GT 00043*
900     00032GT 00590*
910     00593D0 00599*
915     00600D0 00602*
920     00032GT 00610*
930     00613D0 00619*
935     00620D0 00521D0 00623*
940     00032GT 00626*
95      00048#R 00049*
950     00629D0 00635*
960     00032GT 00638*
970     00641D0 00649*
98      00043D0 00046GT 00050*
980     00032GT 00652*
990     00655D0 00665*
```

```
00001        SUBROUTINE MATLIS(MT,TITLE,NDIM,MR,MC,RLABEL,CLABEL,RARRAY,
            +                  CARRAY,IX)
      C         ROUTINE TO OUTPUT MATRICES IN READABLE FORM.
00002        DIMENSION RLABEL(MR),CLABEL(MC),RARRAY(NDIM,MC),CARRAY(NDIM,MC),
            +          NUM(7)
00003        LOGICAL*1 TITLE(MT)
00004        COMMON/IO/ IIN,IOUT
00005        DATA IDASHL,IDOTL,IDASHR,IDOTR,IEX,IB /' -','. ','- ',' .','!',
            +     ' '/
00006        DO 500 IPAGE=1,MC,6
00007        IF(IPAGE.EQ.1) WRITE(IOUT,10) (TITLE(I),I=1,MT)
00009  10    FORMAT(5X,40A1,/)
00010        NPAGE=IPAGE+5
00011        IF(NPAGE.GT.MC) NPAGE=MC
00013        DO 20 I=IPAGE,NPAGE
00014        II=I-IPAGE+1
00015  20    NUM(II)=I
00016        WRITE(IOUT,30) (NUM(I),I=1,NPAGE-IPAGE+1)
00017  30    FORMAT(2X,I8,5I11)
00018        WRITE(IOUT,40) (CLABEL(I),I=IPAGE,NPAGE)
00019  40    FORMAT(2X,A8,5A11)
00020        DO 45 I=1,7
00021  45    NUM(I)=IB
00022        IF(IPAGE.EQ.1) NUM(1)=IDASHL
00024        IF(IPAGE.NE.1) NUM(1)=IDOTL
00025        IF(NPAGE.EQ.MC) NUM(NPAGE-IPAGE+2)=IDASHR
00026        IF(NPAGE.NE.MC) NUM(7)=IDOTR
00030        WRITE(IOUT,50) (NUM(I),I=1,7)
00031  50    FORMAT(3X,A2,6(9X,A2))
00032        DO 400 I=1,MR
00033        IF(NPAGE.EQ.MC) GO TO 150
00035        IF(IX.EQ.1) GO TO 100
00037        IF(IPAGE.EQ.1) WRITE(IOUT,60) IEX,IEX,(RARRAY(I,J),J=IPAGE,NPAGE),
            +                              IEX
00039        IF(IPAGE.NE.1) WRITE(IOUT,60) IB,IB,(RARRAY(I,J),J=IPAGE,NPAGE),IB
00041  60    FORMAT(3X,A1,/,3X,A1,6G11.3,/,3X,A1)
00042        GO TO 400
00043  100   CONTINUE
00044        IF(IPAGE.EQ.1) WRITE(IOUT,110) IEX,IEX,(RARRAY(I,J),J=IPAGE,NPAGE)
            +                              ,IEX,(CARRAY(I,J),J=IPAGE,NPAGE),IEX
00045        IF(IPAGE.NE.1) WRITE(IOUT,110) IB,IB,(RARRAY(I,J),J=IPAGE,NPAGE),
            +                              IB,(CARRAY(I,J),J=IPAGE,NPAGE),IB
00048  110   FORMAT(3X,A1,/,3X,A1,6G11.3,/,3X,A1,6G11.3,/,3X,A1)
00049        GO TO 400
00050  150   CONTINUE
00051        GO TO (160,200,240,280,320,360),NPAGE-IPAGE+1
00052  160   CONTINUE
00053        IF(IX.EQ.1) GO TO 180
00055        IF(IPAGE.EQ.1) WRITE(IOUT,170) IEX,IEX,I,IEX,(RARRAY(I,J),J=IPAGE,
            +     NPAGE),IEX,RLABEL(I),IEX,IEX
00057        IF(IPAGE.NE.1) WRITE(IOUT,170) IB,IEX,I,IB,(RARRAY(I,J),
            +     J=IPAGE,NPAGE),
            +     IEX,RLABEL(I),IB,IEX
00059  170   FORMAT(3X,A1,11X,A1,3X,I2,/,3X,A1,G11.3,A1,1X,A3,/,3X,A1,11X,A1)
```

```
00060          GO TO 400
00061 190      CONTINUE
00062          IF(IPAGE.EQ.1) WRITE(IOUT,193) IEX,IEX,IEX,(RARRAY(I,J),J=IPAGE,
     +             NPAGE),IEX,I,IEX,(CARRAY(I,J),J=IPAGE,NPAGE),IEX,
     +             RLABEL(I),IEX,IEX
00064          IF(IPAGE.NE.1) WRITE(IOUT,193) IB,IEX,IB,(RARRAY(I,J),
     +             J=IPAGE,NPAGE),
     +             IB,I,IB,(CARRAY(I,J),J=IPAGE,NPAGE),IEX,RLABEL(I),IB,IEX
00066 193      FORMAT(3X,A1,11X,A1,/,3X,A1,G11.3,A1,2X,I2,/,3X,A1,G11.3,A1,
     +             1X,A3,/,3X,A1,11X,A1)
00067          GO TO 400
00068 200      CONTINUE
00069          IF(IX.EQ.1) GO TO 220
00071          IF(IPAGE.EQ.1) WRITE(IOUT,213) IEX,IEX,I,IEX,(RARRAY(I,J),J=IPAGE,
     +             NPAGE),IEX,RLABEL(I),IEX,IEX
00073          IF(IPAGE.NE.1) WRITE(IOUT,213) IB,IEX,I,IB,(RARRAY(I,J),
     +             J=IPAGE,NPAGE),
     +             IEX,RLABEL(I),IB,IEX
00075 213      FORMAT(3X,A1,22X,A1,/,3X,A1,2G11.3,A1,1X,A3,/,3X,A1,22X,A1)
00076          GO TO 400
00077 220      CONTINUE
00078          IF(IPAGE.EQ.1) WRITE(IOUT,233) IEX,IEX,IEX,(RARRAY(I,J),J=IPAGE,
     +             NPAGE),IEX,I,IEX,(CARRAY(I,J),J=IPAGE,NPAGE),IEX,
     +             RLABEL(I),IEX,IEX
00080          IF(IPAGE.NE.1) WRITE(IOUT,233) IB,IEX,IB,(RARRAY(I,J),
     +             I=IPAGE,NPAGE),
     +             IB,I,IB,(CARRAY(I,J),J=IPAGE,NPAGE),IEX,RLABEL(I),IB,IEX
00082 233      FORMAT(3X,A1,22X,A1,/,3X,A1,2G11.3,A1,2X,I2,/,3X,A1,2G11.3,A1,
     +             1X,A3,/,3X,A1,22X,A1)
00083          GO TO 400
00084 240      CONTINUE
00085          IF(IX.EQ.1) GO TO 260
00087          IF(IPAGE.EQ.1) WRITE(IOUT,253) IEX,IEX,I,IEX,(RARRAY(I,J),J=IPAGE,
     +             NPAGE),IEX,RLABEL(I),IEX,IEX
00089          IF(IPAGE.NE.1) WRITE(IOUT,253) IB,IEX,I,IB,(RARRAY(I,J),
     +             J=IPAGE,NPAGE),
     +             IEX,RLABEL(I),IB,IEX
00091 253      FORMAT(3X,A1,33X,A1,2X,I2,/,3X,A1,2G11.3,A1,1X,A3,/,3X,A1,33X,A1)
00092          GO TO 400
00093 260      CONTINUE
00094          IF(IPAGE.EQ.1) WRITE(IOUT,273) IEX,IEX,IEX,(RARRAY(I,J),J=IPAGE,
     +             NPAGE),IEX,I,IEX,(CARRAY(I,J),J=IPAGE,NPAGE),IEX,
     +             RLABEL(I),IEX,IEX
00096          IF(IPAGE.NE.1) WRITE(IOUT,273) IB,IEX,IB,(RARRAY(I,J),
     +             I=IPAGE,NPAGE),
     +             IB,I,IB,(CARRAY(I,J),J=IPAGE,NPAGE),IEX,RLABEL(I),IB,IEX
00098 273      FORMAT(3X,A1,33X,A1,/,3X,A1,2G11.3,A1,2X,I2,/,3X,A1,2G11.3,A1,
     +             1X,A3,/,3X,A1,33X,A1)
00099          GO TO 400
00100 280      CONTINUE
00101          IF(IX.EQ.1) GO TO 300
00103          IF(IPAGE.EQ.1) WRITE(IOUT,293) IEX,IEX,I,IEX,(RARRAY(I,J),J=IPAGE,
     +             NPAGE),IEX,RLABEL(I),IEX,IEX
00105          IF(IPAGE.NE.1) WRITE(IOUT,293) IB,IEX,I,IB,(RARRAY(I,J),
```

```
     +      J=IPAGE,NPAGE),
     +      IEX,RLABEL(I),IB,IEX
00107 390   FORMAT(3X,A1,44X,A1,2X,I2,/,3X,A1,4G11.3,A1,1X,A3,/,3X,A1,44X,A1)
00108       GO TO 400
00109 340   CONTINUE
00110       IF(IPAGE.EQ.1) WRITE(IOUT,310) IEX,IEX,IEX,(RARRAY(I,J),J=IPAGE,
     +      NPAGE),IEX,I,IEX,(CARRAY(I,J),J=IPAGE,NPAGE),IEX,
     +      RLABEL(I),IEX,IEX
00112       IF(IPAGE.NE.1) WRITE(IOUT,310) IB,IEX,IB,(RARRAY(I,J),
     +      J=IPAGE,NPAGE),
     +      IB,I,IB,(CARRAY(I,J),J=IPAGE,NPAGE),IEX,RLABEL(I),IB,IEX
00114 310   FORMAT(3X,A1,44X,A1,/,3X,A1,4G11.3,A1,2X,I2,/,3X,A1,4G11.3,A1,
     +      1X,A3,/,3X,A1,44X,A1)
00115       GO TO 400
00116 320   CONTINUE
00117       IF(IX.EQ.1) GO TO 340
00119       IF(IPAGE.EQ.1) WRITE(IOUT,330) IEX,IEX,I,IEX,(RARRAY(I,J),J=IPAGE,
     +      NPAGE),IEX,RLABEL(I),IEX,IEX
00121       IF(IPAGE.NE.1) WRITE(IOUT,330) IB,IEX,I,IB,(RARRAY(I,J),
     +      J=IPAGE,NPAGE),
     +      IEX,RLABEL(I),IB,IEX
00123 330   FORMAT(3X,A1,55X,A1,2X,I2,/,3X,A1,5G11.3,A1,1X,A3,/,3X,A1,55X,A1)
00124       GO TO 400
00125 340   CONTINUE
00126       IF(IPAGE.EQ.1) WRITE(IOUT,350) IEX,IEX,IEX,(RARRAY(I,J),J=IPAGE,
     +      NPAGE),IEX,I,IEX,(CARRAY(I,J),J=IPAGE,NPAGE),IEX,
     +      RLABEL(I),IEX,IEX
00128       IF(IPAGE.NE.1) WRITE(IOUT,350) IB,IEX,IB,(RARRAY(I,J),
     +      J=IPAGE,NPAGE),
     +      IB,I,IB,(CARRAY(I,J),J=IPAGE,NPAGE),IEX,RLABEL(I),IB,IEX
00130 350   FORMAT(3X,A1,55X,A1,/,3X,A1,5G11.3,A1,2X,I2,/,3X,A1,5G11.3,A1,
     +      1X,A3,/,3X,A1,55X,A1)
00131       GO TO 400
00132 360   CONTINUE
00133       IF(IX.EQ.1) GO TO 390
00135       IF(IPAGE.EQ.1) WRITE(IOUT,370) IEX,IEX,I,IEX,(RARRAY(I,J),J=IPAGE,
     +      NPAGE),IEX,RLABEL(I),IEX,IEX
00137       IF(IPAGE.NE.1) WRITE(IOUT,370) IB,IEX,I,IB,(RARRAY(I,J),
     +      J=IPAGE,NPAGE),
     +      IEX,RLABEL(I),IB,IEX
00139 370   FORMAT(3X,A1,66X,A1,2X,I2,/,2X,A1,6G11.3,A1,1X,A3,/,3X,A1,66X,A1)
00140       GO TO 400
00141 380   CONTINUE
00142       IF(IPAGE.EQ.1) WRITE(IOUT,390) IEX,IEX,IEX,(RARRAY(I,J),J=IPAGE,
     +      NPAGE),IEX,I,IEX,(CARRAY(I,J),J=IPAGE,NPAGE),IEX,
     +      RLABEL(I),IEX,IEX
00144       IF(IPAGE.NE.1) WRITE(IOUT,390) IB,IEX,IB,(RARRAY(I,J),
     +      J=IPAGE,NPAGE),
     +      IB,I,IB,(CARRAY(I,J),J=IPAGE,NPAGE),IEX,RLABEL(I),IB,IEX
00146 390   FORMAT(3X,A1,66X,A1,/,2X,A1,6G11.3,A1,2X,I2,/,3X,A1,6G11.3,A1,
     +      1X,A3,/,3X,A1,66X,A1)
00147 400   CONTINUE
00148       WRITE(IOUT,50) (SUM(I),I=1,7)
00149       WRITE(IOUT,450)
```

```
00150 450   FORMAT(/////)
00151 500   CONTINUE
00152       REWIND IOUT
00153       RETURN
00154       END
```

```
CARRAY  00001AG 00002DI 00045#R 00047#R 00053#R 00055#R 00079#R 00081#R
        00095#R 00097#R 00111#R 00113#R 00127#R 00129#R 00143#R 00145#R
CLABEL  00001AG 00002DI 00010#R
I       00008#R 00013DO 00014   00015   00016#R 00018#R 00020DO 00021
        00032#R 00032DO 00038#R 00044#R 00045#R 00047#R 00056#R 00058#R
        00063#R 00065#R 00072#R 00074#R 00079#R 00081#R 00088#R 00090#R
        00095#R 00097#R 00104#R 00106#R 00111#R 00113#R 00120#R 00122#R
        00127#R 00129#R 00136#R 00138#R 00143#R 00145#R 00148#R
IB      00005DA 00021   00044#R 00047#R 00058#R 00065#R 00074#R 00081#R
        00095#R 00097#R 00106#R 00113#R 00122#R 00129#R 00138#R 00145#R
IDASHL  00005DA 00023
IDASHR  00005DA 00027
IDOTL   00005DA 00025
IDOTR   00005DA 00029
IEX     00005DA 00038#R 00045#R 00058#R 00058#R 00063#R 00065#R 00072#R
        00074#R 00079#R 00081#R 00088#R 00090#R 00095#R 00097#R 00104#R
        00106#R 00111#R 00113#R 00120#R 00122#R 00127#R 00129#R 00136#R
        00138#R 00143#R 00145#R
II      00014=  00015
IIN     00004CN
IO      00004CN
IOUT    00004C4 00008#R 00016#R 00018#R 00030#R 00038#R 00040#R 00045#R
        00047#R 00056#R 00058#R 00063#R 00065#R 00072#R 00074#R 00079#R
        00081#R 00088#R 00090#R 00095#R 00097#R 00104#R 00106#R 00111#R
        00113#R 00120#R 00122#R 00127#R 00129#R 00136#R 00138#R 00143#R
        00148#R 00148#R 00149#R 00152#R
IPAGE   00006DO 00007IF 00010   00013DO 00014   00016#R 00018#R 00022IF
        00024IF 00027   00037IF 00038#R 00039IF 00040#R 00044IF 00045#R
        00046IF 00047#R 00050GT 00055IF 00058#R 00057IF 00058#R 00062IF
        00063#R 00064IF 00065#R 00071IF 00072#R 00073IF 00074#R 00078IF
        00079#R 00080IF 00081#R 00087IF 00088#R 00089IF 00090#R 00094IF
        00095#R 00096IF 00097#R 00103IF 00104#R 00105IF 00106#R 00110IF
        00111#R 00112IF 00113#R 00119IF 00120#R 00121IF 00122#R 00126IF
        00127#R 00128IF 00129#R 00135IF 00136#R 00137IF 00138#R 00142IF
        00143#R 00144IF 00145#R
IX      00001AG 00035IF 00053IF 00058IF 00085IF 00101IF 00117IF 00133IF
J       00038#R 00040#R 00045#R 00047#R 00058#R 00058#R 00063#R 00065#R
        00072#R 00074#R 00079#R 00081#R 00088#R 00090#R 00095#R 00097#R
        00104#R 00106#R 00111#R 00113#R 00120#R 00122#R 00127#R 00129#R
        00138#R 00138#R 00143#R 00145#R
MATLIS  00001SU
NC      00001AG 00002DI 00006DO 00011IF 00012   00026IF 00028IF 00033IF
NDIM    00001AG 00002DI
NPAGE   00010=  00011IF 00012=  00013DO 00016#R 00018#R 00026IF 00027
        00028IF 00033IF 00038#R 00040#R 00045#R 00047#R 00051GT 00058#R
        00058#R 00063#R 00065#R 00072#R 00074#R 00079#R 00081#R 00088#R
        00090#R 00095#R 00097#R 00104#R 00106#R 00111#R 00113#R 00120#R
        00122#R 00127#R 00129#R 00136#R 00138#R 00143#R 00145#R
NR      00001AG 00002DI 00032DO
NT      00001AG 00003LG 00038#R
NUM     00002DI 00015=  00016#R 00021=  00023=  00025=  00027=  00029=
        00038#R 00148#R
RARRAY  00001AG 00002DI 00038#R 00040#R 00045#R 00047#R 00058#R 00058#R
        00063#R 00065#R 00072#R 00074#R 00079#R 00081#R 00088#R 00090#R
```

```
RT-11 INDEX V03  CROSS REFERENCE LISTING 10-APR-79 04:30:10 PAGE 00006

        00099#R 00097#R 00134#R 00100#R 00112#R 00113#R 00120#R 00122#R
        00127#R 00129#R 00130#R 00130#R 00143#R 00145#R
RLABEL  00001AG 00002DI 00059#R 00057#R 00053#R 00055#R 00072#R 00074#R
        00079#R 00081#R 00088#R 00097#R 00095#R 00097#R 00104#R 00103#R
        00111#R 00113#R 00120#R 00122#R 00127#R 00129#R 00130#R 00138#R
        00143#R 00145#R
TITLE   00001AG 00003LG 00008#R
10      00008#R 00009*
100     00036GT 00043*
110     00045#R 00047#R 00040*
150     00034GT 00050*
160     00051GT 00052*
170     00053#R 00058#R 00059*
180     00054GT 00061*
190     00063#R 00065#R 00066*
20      00013DO 00015*
200     00051GT 00069*
210     00077#R 00074#R 00075*
220     00070GT 00077*
230     00079#R 00081#R 00082*
240     00051GT 00084*
250     00088#R 00090#R 00091*
260     00086GT 00093*
270     00095#R 00097#R 00098*
280     00051GT 00100*
290     00104#R 00106#R 00107*
30      00010#R 00017*
300     00102GT 00109*
310     00112#R 00113#R 00114*
320     00051GT 00116*
330     00120#R 00122#R 00123*
340     00118GT 00125*
350     00127#R 00129#R 00130*
360     00051GT 00132*
370     00136#R 00138#R 00139*
380     00134GT 00141*
390     00143#R 00145#R 00146*
40      00010#R 00019*
400     00032DO 00042GT 00049GT 00050GT 00057GT 00074GT 00083GT 00092GT
        00099GT 00108GT 00115GT 00124GT 00131GT 00140GT 00147*
45      00002DO 00021*
450     00149#R 00150*
50      00023#R 00021*  00145#R
500     00004DO 00151*
60      00039#R 00040#R 00041*
```

```
00001        SUBROUTINE BALANC(NM,N,A,LOW,IGH,SCALE)
00002        INTEGER I,J,K,L,M,N,JJ,NM,IGH,LOW,IEXC
00003        REAL A(NM,N),SCALE(N)
00004        REAL C,F,G,R,S,B2,RADIX
00005        REAL ABS
00006        LOGICAL NOCONV
             C
             C
             C
00007        RADIX=2
00008        B2=RADIX*RADIX
00009        K=1
00010        L=N
00011        GO TO 100
             C
             C
00012 20     SCALE(M)=J
00013        IF(J.EQ.M) GO TO 50
00015        DO 30 I=1,L
00016        F=A(I,J)
00017        A(I,J)=A(I,M)
00018        A(I,M)=F
00019 30     CONTINUE
00020        DO 40 I=K,N
00021        F=A(J,I)
00022        A(J,I)=A(M,I)
00023        A(M,I)=F
00024 40     CONTINUE
00025 50     GO TO (80,130),IEXC
             C
             C
00026 80     IF(L.EQ.1) GO TO 280
00028        L=L-1
             C
00029 100    DO 120 JJ=1,L
00030        J=L+1-JJ
00031        DO 110 I=1,L
00032        IF(I.EQ.J) GO TO 110
00034        IF(A(J,I).NE.0.0) GO TO 120
00036 110    CONTINUE
00037        M=L
00038        IEXC=1
00039        GO TO 20
00040 120    CONTINUE
00041        GO TO 140
             C
             C
00042 130    K=K+1
00043 140    DO 170 J=K,L
00044        DO 150 I=K,L
00045        IF(I.EQ.J) GO TO 150
00047        IF(A(I,J).NE.0.0) GO TO 170
00049 150    CONTINUE
00050        M=K
```

```
00051          IEXC=2
00052          GO TO 20
00053 170      CONTINUE
       C       ********** NOW BALANC THE SUBMATRIX IN ROWS K TO L **********
00054          DO 180 I=K,L
00055 180      SCALE(I)=1.0
       C       ********** ITERATIVE LOOP FOR NORM REDUCTION **********
00056 190      NOCONV=.FALSE.
00057          DO 270 I=K,L
00058          C=0.0
00059          R=0.0
00060          DO 200 J=K,L
00061          IF(J.EQ.I) GO TO 200
00063          C=C+ABS(A(J,I))
00064          R=R+ABS(A(I,J))
00065 200      CONTINUE
       C       ********** GUARD AGAINST ZERO C OR R DUE TO UNDERFLOW **********
00066          IF(C.EQ.0.0.OR.R.EQ.0.0) GO TO 270
00068          G=R/RADIX
00069          F=1.0
00070          S=C+R
00071 210      IF(C.GE.G) GO TO 220
00073          F=F*RADIX
00074          C=C*B2
00075          GO TO 210
00076 220      G=R*RADIX
00077 230      IF(C.LT.G) GO TO 240
00079          F=F/RADIX
00080          C=C/B2
00081          GO TO 230
       C       ********** NOW BALANC **********
00082 240      IF((C+R)/F.GE.0.95*S) GO TO 270
00084          G=1.0/F
00085          SCALE(I)=SCALE(I)*F
00086          NOCONV=.TRUE.
00087          DO 250 J=K,N
00088 250      A(I,J)=A(I,J)*G
00089          DO 260 J=1,L
00090 260      A(J,I)=A(J,I)*F
00091 270      CONTINUE
00092          IF(NOCONV) GO TO 190
00094 280      LOW=K
00095          IGH=L
00096          RETURN
00097          END
```

```
A       000001AG 000003RL 000016  000017=  000018=  000021  000022=  000023=
        000034IF 000047IF 000063  000064  000088=  000094=
ABS     000005RL 000063  000064
BALANC  000001SU
B2      000004RL 000008=  000074  000084
C       000004RL 000056=  000063=  000066IF 000070  000071IF 000074=  000077IF
        000080=  000082IF
F       000004RL 000016=  000018  000021=  000023  000069=  000073=  000079=
        000082IF 000084  000085  000090
G       000004RL 000068=  000071IF 000076=  000077IF 000084=  000086
I       000002IN 000015DO 000016  000017  000018  000020DO 000021  000022
        000023  000031DO 000032IF 000034IF 000044DO 000045IF 000047IF 000054DO
        000055  000057DO 000061IF 000063  000064  000085  000088  000090
IEXC    000002IN 000025GT 000030=  000051=
IGN     000001AG 000002IN 000095=
J       000002IN 000012  000013IF 000016  000017  000021  000022  000037=
        000032IF 000034IF 000043DO 000045IF 000047IF 000060DO 000061IF 000063
        000064  000067DO 000088  000089DO 000093
JJ      000002IN 000029DO 000030
K       000002IN 000039=  000020DO 000042=  000043DO 000044DO 000058  000054DO
        000057DO 000060DO 000087DO 000094
L       000002IN 000013=  000015DO 000026IF 000028=  000029DO 000030  000031DO
        000037  000043DO 000044DO 000054DO 000057DO 000060DO 000089DO 000095
LOW     000001AG 000002IN 000094=
M       000002IN 000012  000013IF 000017  000018  000022  000023  000037=
        000050=
N       000001AG 000002IN 000003RL 000010  000020DO 000087DO
N4      000001AG 000002IN 000003RL
NOCONV  000005LG 000056=  000096=  000092IF
R       000004RL 000059=  000064=  000066IF 000068  000070  000076  000082IF
RADIX   000006RL 000037=  000038  000068  000073  000076  000079
S       000004RL 000070=  000002IF
SCALE   000001AG 000003RL 000012=  000055=  000085=
100     000011GT 000029*
110     000031DO 000036GT 000036*
120     000029DO 000035GT 000040*
130     000025GT 000042*
140     000041GT 000043*
150     000044DO 000046GT 000046*
170     000043DO 000048GT 000053*
180     000054DO 000055*
190     000050*  000093GT
20      000012*  000039GT 000052GT
200     000050DO 000052GT 000045*
210     000071*  000075GT
220     000072GT 000076*
230     000077*  000081GT
240     000078GT 000082*
250     000087DO 000088*
260     000089DO 000090*
270     000057DO 000062GT 000093GT 000091*
280     000022GT 000094*
30      000015DO 000019*
40      000020DO 000024*
```

RT-11 INDEX V03 CROSS REFERENCE LISTING 13-APR-79 03:33:59 PAGE 0004

50 [illegible] 00025*
30 [illegible] 00026*

```
00001          SUBROUTINE BALBAK(NM,N,LOW,IGH,SCALE,M,Z)
00002          INTEGER I,J,K,M,N,II,NM,IGH,LOW
00003          REAL SCALE(N),Z(NM,M)
00004          REAL S
00005          IF(M.EQ.0) GO TO 200
00006          IF(IGH.EQ.LOW) GO TO 120
00007          DO 110 I=LOW,IGH
00008          S=SCALE(I)
               C     ********** LEFT HAND EIGENVECTORS ARE BACK TRANSFORMED
               C                IF THE FOREGOING STATEMENT IS REPLACED BY
               C                S=1.0/SCALE(I). **********
00009          DO 100 J=1,M
00010    100   Z(I,J)=Z(I,J)*S
00011    110   CONTINUE
               C     ********** FOR I=LOW-1 STEP -1 UNTIL 1,
               C                IGH+1 STEP 1 UNTIL N DO -- **********
00012    120   DO 140 II=1,N
00013          I=II
00014          IF(I.GE.LOW.AND.I.LE.IGH) GO TO 140
00015          IF(I.LT.LOW) I=LOW-II
00016          K=SCALE(I)
00017          IF(K.EQ.I) GO TO 140
00018          DO 130 J=1,M
00019          S=Z(I,J)
00020          Z(I,J)=Z(K,J)
00021          Z(K,J)=S
00022    130   CONTINUE
00023    140   CONTINUE
00024    200   RETURN
00025          END
```

```
BALANK  000150
I       000121N 000900  00010   00012   00015* 00016IF 00018IF 00019*
        00020   000021F 00024   00025
ISH     000024G 000021N 000037F 000900  00016IF
II      000021N 000140  00015   00019
J       000021N 000120  00012   000230  00024   00025   00026
K       000021N 00020*  000021F 00025   00026
LOW     000024G 000021N 000037F 000900  00016IF 00018IF 00019
M       000024G 000021N 000038L 000051F 000120  000230
N       000024G 000021N 000038L 000140
NE      000024G 000021N 000038L
S       000028L 00010*  00012   00024*  00026
SMALL   000024G 000038L 00010   00024
Z       000024G 000028L 00012*  00024   00025*  00026*
100     000120  00012*
110     000130  00013*
120     000230  00016*
130     000700  00020*
140     000900  000120  000230  00023*
200     000140  00020*
```

```
00001       SUBROUTINE DECOMP(NDIM,N,A,COND,IPVT,WORK)
00002       INTEGER NDIM,N
00003       COMPLEX A(NDIM,N),T
00004       INTEGER IPVT(N)
            C     DECOMPOSES A COMPLEX MATRIX BY GAUSSIAN ELIMINATION
            C     AND ESTIMATES THE CONDITION OF THE MATRIX.
            C     USE SOLVE TO COMPUTE SOLUTIONS TO LINEAR SYSTEMS.
            C     INPUT..
            C       NDIM = DECLARED ROW DIMENSION OF THE ARRAY CONTAINING A.
            C       N = ORDER OF THE MATRIX.
            C       A = MATRIX TO BE TRIANGULARIZED.
            C     OUTPUT..
            C       A CONTAINS AN UPPER TRIANGULAR MATRIX U AND A PERMUTED
            C        VERSION OF A LOWER TRIANGULAR MATRIX I-L SO THAT
            C        (PERMUTATION MATRIX)*A = L*U
            C       COND = AN ESTIMATE OF THE CONDITION OF A.
            C         FOR THE LINEAR SYSTEM A*X = B, CHANGES IN A AND B
            C         MAY CAUSE CHANGES COND TIMES AS LARGE IN X.
            C         IF COND+1.0 .EQ. COND, A IS SINGULAR TO WORKING
            C         PRECISION. COND IS SET TO 1.0E+32 IF EXACT
            C         SINGULARITY IS DETECTED.
            C       IPVT = THE PIVOT VECTOR.
            C         IPVT(K) = THE INDEX OF THE K-TH PIVOT ROW
            C         IPVT(N) = (-1)**(NUMBER OF INTERCHANGES)
            C     WORK SPACE.. THE VECTOR WORK MUST BE DECLARED AND INCLUDED
            C                  IN THE CALL. ITS INPUT CONTENTS ARE IGNORED.
            C                  ITS OUTPUT CONTENTS ARE USUALLY UNIMPORTANT.
            C     THE DETERMINANT OF A CAN BE OBTAINED ON OUTPUT BY
            C       DET(A) = IPVT(N)*A(1,1)*A(2,2)*...*A(N,N).
00005       INTEGER NM1,I,J,K,KP1,KB,KM1,M
00006       COND=1.0
00007       IPVT(N)=1
00008       IF(N.EQ.1) GO TO 80
00010       NM1=N-1
            C     GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING
00011       DO 35 K=1,NM1
00012       KP1=K+1
            C     FIND PIVOT
00013       M=K
00014       DO 15 I=KP1,N
00015       IF(CABS(A(I,K)).GT.CABS(A(M,K))) M=I
00017 15    CONTINUE
00018       IPVT(K)=M
00019       IF(M.NE.K) IPVT(N)=-IPVT(N)
00021       T=A(M,K)
00022       A(M,K)=A(K,K)
00023       A(K,K)=T
            C     SKIP STEP IF PIVOT IS ZERO
00024       IF(T.EQ.CMPLX(0.0,0.0)) GO TO 35
            C     COMPUTE MULTIPLIERS
00026       DO 20 I=KP1,N
00027       A(I,K)=-A(I,K)/T
00028 20    CONTINUE
            C     INTERCHANGE AND ELIMINATE BY COLUMNS
```

```
00029          DO 30 J=KP1,N
00030          T=A(M,J)
00031          A(M,J)=A(K,J)
00032          A(K,J)=T
00033          IF(T.EQ.CMPLX(0.0,0.0)) GO TO 30
00035          DO 25 I=KP1,N
00036          A(I,J)=A(I,J)+A(I,K)*T
00037    25    CONTINUE
00038    30    CONTINUE
00039    35    CONTINUE
00040          DO 40 I=1,N
00041    40    IF(A(I,I).EQ.CMPLX(0.,0.)) GO TO 90
00043          RETURN
         C     1 BY 1
00044    90    COND=1.0
00045          IF(A(1,1).NE.CMPLX(0.0,0.0)) RETURN
         C     EXACT SINGULARITY
00047    95    COND=1.0E+32
00048          RETURN
00049          END
```

```
RT-11 INDEX V03  CROSS REFERENCE LISTING 10-APR-79 03:50:53 PAGE 00003

A       000001AG 000003CX 000015IF 00021   00022=  00023=  00027=  00033
        00031=  00032=  00035=  000041IF 000045IF
CABS    000015IF
CMPLX   000024IF 000033IF 000041IF 000045IF
COND    000001AG 00005=  00044=  00047=
DECOMP  000001GU
I       000005IN 000140DO 000151F 00016   000229DO 00027   000350DO 00036
        000400NO 000411F
IPVT    000001AG 000004IN 00007=  00010=  00027=
J       000005IN 000229DO 00030   00031   00032   00036
K       000005IN 000110DO 00012   00013   000151F 00018   000191F 00021
        00022   00023   00027   00031   00032   00036
KB      000005IN
KM1     000005IN
KP1     000005IN 00012=  000140DO 000229DO 000350DO
M       000005IN 00013=  000151F 00016=  00018   000191F 00021   00022
        00030   00031
N       000001AG 000002IN 000003CX 000004IN 00007   000008IF 00010   000140DO
        00020   000229DO 000230DO 000350DO 000400DO
NDIM    000001AG 000002IN 000003CX
NM1     000005IN 00010=  000110DO
T       000003CX 00021=  00023   000024IF 00027   00033=  00032   000331F
        00036
WORK    000001AG
15      000140NO 00017*
20      000229DO 00020*
25      000350DO 00037*
30      000229DO 000340GT 00038*
35      000110DO 000350GT 00039*
40      000140NO 00041*
80      000400GT 00044*
90      000411GT 00047*
```

ORIGINAL PAGE IS
POOR QUALITY

```
00001       SUBROUTINE EIGRF(A,N,IA,IJOB,W,Z,INT,SCALE,WR,WI,WK,IER)
            C  FUNCTION - TO CALCULATE EIGENVALUES AND (OPTIONALLY)
            C             EIGENVECTORS OF A REAL GENERAL MATRIX.
            C  ARGUMENTS  A   - INPUT REAL GENERAL MATRIX WITH ROW
            C                   DIMENSION IA AND COLUMN DIMENSION
            C                   AT LEAST N.
            C             N   - ORDER OF MATRIX A.
            C             IA  - ROW DIMENSION OF MATRIX A; IA MUST
            C                   BE GREATER THAN OR EQUAL TO N.
            C             IJOB- OPTION PARAMETER:
            C                   IJOB=0, COMPUTE EIGENVALUES ONLY
            C                   IJOB=1, COMPUTE EIGENVALUES AND
            C                           EIGENVECTORS.
            C             W   - OUTPUT COMPLEX VECTOR OF LENGTH N
            C                   CONTAINING THE EIGENVALUES OF A.
            C             Z   - OUTPUT N BY N COMPLEX MATRIX
            C                   CONTAINING THE EIGENVECTORS OF A;
            C                   THE EIGENVECTOR IN COLUMN J OF Z
            C                   CORRESPONDS TO THE EIGENVALUE W(J).
            C             INT,SCALE,WR,WI,WK - WORKING SPACE
            C             IER - IER=128+J INDICATES THAT MORE THAN
            C                   30 ITERATIONS WERE REQUIRED TO
            C                   COMPUTE THE JTH EIGENVALUE. EIGENVALUES
            C                   J+1,J+2,...,N HAVE BEEN CORRECTLY
            C                   COMPUTED, WHILE EIGENVALUES 1,...,J
            C                   ARE SET TO ZERO. NO EIGENVECTORS ARE
            C                   COMPUTED.
00002       INTEGER I,IA,IER,IERR,IGH,IJOB,J,LOW,N
00003       INTEGER INT(IA)
00004       DIMENSION SCALE(IA),WR(IA),WI(IA)
00005       DIMENSION A(IA,N),WK(IA,N)
00006       COMPLEX W(N),Z(IA,N)
00007       IER=0
00008       IF(IJOB.NE.0.AND.IJOB.NE.1) IJOB=1
00009       CALL BALANC(IA,N,A,LOW,IGH,SCALE)
00011       CALL ELMHES(IA,N,LOW,IGH,A,INT)
            C     EIGENVALUES ONLY
00012       IF(IJOB.NE.0) GO TO 10
00014       CALL HQR(IA,N,LOW,IGH,A,WR,WI,IERR)
00015       IF(IERR.NE.0) GO TO 20
00017       GO TO 30
            C     EIGENVALUES AND EIGENVECTORS
00018 10    CALL ELTRAN(IA,N,LOW,IGH,A,INT,WK)
00019       CALL HQR2(IA,N,LOW,IGH,A,WR,WI,WK,IERR)
00021       IF(IERR.NE.0) GO TO 20
00022       CALL BABAK(IA,N,LOW,IGH,SCALE,N,WK)
00023       GO TO 30
            C     ERROR HANDLING
00024 20    IER=128+IERR
            C     ENCODE EIGENVALUES AND EIGENVECTORS
            C     AS COMPLEX NUMBERS
00025 30    DO 40 I=1,N
00026 40    W(I)=CMPLX(WR(I),WI(I))
00027       IF(IJOB.EQ.0.OR.IER.NE.0) RETURN
```

```
00029          DO 50 J=1,N
00030          DO 50 I=1,N
00031          IF(WI(J).EQ.0.) Z(I,J)=CMPLX(WR(I,J),0.)
00033          IF(WI(J).GT.0.) Z(I,J)=CMPLX(WR(I,J),WR(I,J+1))
00035          IF(WI(J).LT.0.) Z(I,J)=CMPLX(WR(I,J-1),-WR(I,J))
00037 50       CONTINUE
00038          RETURN
00039          END
```

RT-11 INDEX V03 CROSS REFERENCE LISTING 13-APR-79 34:23:41 PAGE 0003

```
A       0001AG 0005DI 0010AG 0011AG 0014AG 0018AG 0019AG
BALANC  0013CL
BALBAK  0022CL
CMPLX   0026   0032   0034   0035
EIGRF   0001SU
ELMHES  0011CL
ELTRAN  0010CL
HQR     0014CL
HQR2    0019CL
I       0002IN 0025DO 0026   0030DO 0032   0034   0035
IA      0001AG 0002IN 0003IN 0004DI 0005DI 0006CX 0010AG 0011AG
        0014AG 0018AG 0019AG 0022AG
IER     0001AG 0002IN 0007=  0024=  0027IF
IERR    0002IN 0004AG 0015IF 0019AG 0032IF 0024
IGH     0002IN 0010AG 0011AG 0014AG 0018AG 0019AG 0022AG
IJOB    0001AG 0002IN 0003IF 0009=  0012IF 0027IF
INT     0001AG 0003IN 0011AG 0018AG
J       0002IN 0025DO 0031IF 0032   0033IF 0034   0035IF 0035
LOW     0002IN 0010AG 0011AG 0014AG 0018AG 0019AG 0022AG
N       0001AG 0002IN 0005DI 0006CX 0010AG 0011AG 0014AG 0018AG
        0019AG 0022AG 0025DO 0029DO 0030DO
SCALE   0001AG 0004DI 0010AG 0022AG
W       0001AG 0006CX 0026=
WI      0001AG 0004DI 0014AG 0019AG 0025   0031IF 0033IF 0035IF
WK      0001AG 0005DI 0018AG 0019AG 0022AG 0032   0034   0035
WR      0001AG 0004DI 0014AG 0019AG 0025
Z       0001AG 0006CX 0032=  0034=  0035=
10      0013GT 0018*
20      0014GT 0021GT 0024*
30      0017GT 0023GT 0025*
40      0025DO 0026*
50      0029DO 0030DO 0037*
```

```
00001       SUBROUTINE ELMHES(NM,N,LOW,IGH,A,INT)
00002       INTEGER I,J,M,N,LA,NM,IGH,KP1,LOW,MM1,MP1
00003       REAL A(NM,N)
00004       REAL X,Y
00005       REAL ABS
00006       INTEGER INT(IGH)
00007       LA=IGH-1
00008       KP1=LOW+1
00009       IF(LA.LT.KP1) GO TO 200
00011       DO 180 M=KP1,LA
00012       MM1=M-1
00013       X=0.0
00014       I=M
00015       DO 100 J=M,IGH
00016       IF(ABS(A(J,MM1)).LE.ABS(X)) GO TO 100
00018       X=A(J,MM1)
00019       I=J
00020 100   CONTINUE
00021       INT(M)=I
00022       IF(I.EQ.M) GO TO 130
      C     ********** INTERCHANGE ROWS AND COLUMNS OF A **********
00024       DO 110 J=MM1,N
00025       Y=A(I,J)
00026       A(I,J)=A(M,J)
00027       A(M,J)=Y
00028 110   CONTINUE
00029       DO 120 J=1,IGH
00030       Y=A(J,I)
00031       A(J,I)=A(J,M)
00032       A(J,M)=Y
00033 120   CONTINUE
      C     ********** END INTERCHANGE **********
00034 130   IF(X.EQ.0.0) GO TO 180
00035       MP1=M+1
00037       DO 160 I=MP1,IGH
00038       Y=A(I,MM1)
00039       IF(Y.EQ.0.0) GO TO 160
00041       Y=Y/X
00042       A(I,MM1)=Y
00043       DO 140 J=M,N
00044 140   A(I,J)=A(I,J)-Y*A(M,J)
00045       DO 150 J=1,IGH
00046 150   A(J,M)=A(J,M)+Y*A(J,I)
00047 160   CONTINUE
00048 180   CONTINUE
00049 200   RETURN
00050       END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
A       000001AG 000003RL 000016IF 000019   000025   000026=  000027=  000030
        000031=  000032=  000038   000042=  000044=  000046=
ABS     000000RL 000016IF
ELMNTS  000001SU
I       000002IN 000014=  000019=  000021   000022IF 000025   000026   000030
        000031   000037DO 000038   000042   000044   000046
IGH     000001AG 000002IN 000006IN 000007   000015DO 000029DO 000037DO 000045DO
INT     000001AG 000006IN 000021=
J       000002IN 000015DO 000016IF 000018   000019   000024DO 000025   000026
        000027   000029DO 000030   000031   000032   000043DO 000044   000045DO
        000046
KPI     000002IN 000008=  000009IF 000011DO
LA      000002IN 000007=  000009IF 000011DO
LON     000001AG 000002IN 000008
M       000002IN 000011DO 000012   000014   000015DO 000021   000022IF 000026
        000027   000031   000032   000036   000043DO 000044   000046
MM1     000002IN 000012=  000016IF 000018   000024DO 000038   000042
MPI     000002IN 000036=  000037DO
N       000001AG 000002IN 000003RL 000024DO 000043DO
NM      000001AG 000002IN 000003RL
X       000004RL 000013=  000016IF 000018=  000034IF 000041
Y       000004RL 000025=  000027   000030=  000032   000039=  000039IF 000041=
        000042   000044   000046
100     000015DO 000017GT 000020*
110     000024DO 000028*
120     000029DO 000033*
130     000023GT 000034*
140     000043DO 000044*
150     000045DO 000046*
160     000037DO 000040GT 000047*
180     000011DO 000035GT 000048*
200     000010GT 000049*
```

```
00001        SUBROUTINE ELTRAN(NM,N,LOW,IGH,A,INT,Z)
00002        INTEGER I,J,N,KL,MM,MP,NM,IGH,LOW,MP1
00003        REAL A(NM,IGH),Z(NM,N)
00004        INTEGER INT(IGH)
     C       ********** INITIALIZE Z TO IDENTITY MATRIX **********
00005        DO 80 I=1,N
00006        DO 60 J=1,N
00007 60     Z(I,J)=0.0
00008        Z(I,I)=1.0
00009 80     CONTINUE
00010        KL=IGH-LOW-1
00011        IF(KL.LT.1) GO TO 200
     C       ********** FOR MP=IGH-1 STEP -1 UNTIL LOW+1 DO -- **********
00013        DO 140 MM=1,KL
00014        MP=IGH-MM
00015        MP1=MP+1
00016        DO 100 I=MP1,IGH
00017 100    Z(I,MP)=A(I,MP-1)
00018        I=INT(MP)
00019        IF(I.EQ.MP) GO TO 140
00021        DO 130 J=MP,IGH
00022        Z(MP,J)=Z(I,J)
00023        Z(I,J)=0.0
00024 130    CONTINUE
00025        Z(I,MP)=1.0
00026 140    CONTINUE
00027 200    RETURN
00028        END
```

```
 MT-11 INDEX V03  CROSS REFERENCE LISTING 13-APR-79 04:21:14 PAGE 00002
A       00021AG 00003RL 00017
ELTRAN  00041GU
I       00002IN 00005DO 00007   00008   00016DO 00017   00018=  00019IF
        00022   00023   00025
IGH     00001AG 00002IN 00003RL 00004IN 00010   00014   00016DO 00021DO
INT     00001AG 00004IN 00018
J       00002IN 00006DO 00007   00021DO 00022   00023
KL      00002IN 00010=  00011IF 00213DO
LOW     00001AG 00002IN 00010
M4      00002IN 00013DO 00014
MP      00002IN 00014=  00015   00017   00018   00019IF 00021DO 00022
        00025
MP1     00002IN 00015=  00016DO
N       00001AG 00002IN 00003RL 00005DO 00006DO
N4      00001AG 00002IN 00003RL
Z       00001AG 00003RL 00007=  00008=  00017=  00022=  00023=  00025=
16      00016DO 00017*
13      00021DO 00024*
14      00012DO 00020GT 00026*
20      00012GT 00027*
5M      00005DO 00007*
8J      00005DO 00009*
```

```
00001        SUBROUTINE GMINV(NR,NC,A,U,MR,MT)
      C        GENERAL MATRIX INVERSE
      C        INPUTS - NR = NUMBER OF ROWS IN A
      C                 NC = NUMBER OF COLUMNS IN A
      C                  A = MATRIX TO BE INVERTED
      C                 MT = NOT USED
      C        OUTPUTS-  U = INVERTED MATRIX
      C                 MR = RANK OF INVERTED MATRIX
00002        DIMENSION A(1),U(1),S(30)
00003        COMMON/NONAME/ NDIM
00004        NDIM1=NDIM+1
00005        TOL=1.E-14
00006        ADV=1.E-24
00007        MR=NC
00008        NRM1=NR-1
00009        TOL1=0.
00010        JJ=1
00011        DO 10 J=1,NC
00012        S(J)=DOT(NR,A(JJ),A(JJ))
00013        IF(S(J).GT.TOL1) TOL1=S(J)
00015 10     JJ=JJ+NDIM
00016        TOL1=ADV*TOL1
00017        ADV=TOL1
00018        JJ=1
00019        DO 100 J=1,NC
00020        FAC=S(J)
00021        JM1=J-1
00022        JRM=JJ+NRM1
00023        JCM=JJ+JM1
00024        DO 20 I=JJ,JCM
00025 20     U(I)=0.
00026        U(JCM)=1.0
00027        IF(J.EQ.1) GO TO 54
00029        KK=1
00030        DO 30 K=1,JM1
00031        IF(S(K).EQ.1.0) GO TO 30
00033        TEMP=-DOT(NR,A(JJ),A(KK))
00034        CALL VADD(K,TEMP,U(JJ),U(KK))
00035 30     KK=KK+NDIM
00036        DO 50 L=1,2
00037        KK=1
00038        DO 50 K=1,JM1
00039        IF(S(K).EQ.0.) GO TO 50
00041        TEMP=-DOT(NR,A(JJ),A(KK))
00042        CALL VADD(NR,TEMP,A(JJ),A(KK))
00043        CALL VADD(K,TEMP,U(JJ),U(KK))
00044 50     KK=KK+NDIM
00045        TOL1=TOL*FAC+ADV
00046        FAC=DOT(NR,A(JJ),A(JJ))
00047 54     IF(FAC.GT.TOL1) GO TO 70
00049        DO 55 I=JJ,JRM
00050 55     A(I)=0.
00051        S(J)=0.
00052        KK=1
```

```
00353          DO 65 K=1,JM1
00354          IF(S(K).EQ.0.) GO TO 65
00356          TEMP=-DOT(K,U(KK),U(JJ))
00357          CALL VADD(NR,TEMP,A(JJ),A(KK))
00358    65    KK=KK+NDIM
00359          FAC=DOT(J,U(JJ),U(JJ))
00360          NR=NR-1
00361          GO TO 75
00362    70    S(J)=1.0
00363          KK=1
00364          DO 72 K=1,JM1
00365          IF(S(K).EQ.1.) GO TO 72
00367          TEMP=-DOT(NR,A(JJ),A(KK))
00368          CALL VADD(K,TEMP,U(JJ),U(KK))
00369    72    KK=KK+NDIM
00370    75    FAC=1./SQRT(FAC)
00371          DO 80 I=JJ,JCM
00372    80    A(I)=A(I)*FAC
00373          DO 85 I=JJ,JCM
00374    85    U(I)=U(I)*FAC
00375   100    JJ=JJ+NDIM
00376          NEND=NC*NDIM
00377          JJ=1
00378          DO 135 J=1,NC
00379          DO 125 I=1,NR
00380          II=I-J
00381          S(I)=0.
00382          DO 125 KK=JJ,NEND,NDIM
00383   125    S(I)=S(I)+A(II+KK)*U(KK)
00384          II=J
00385          DO 133 I=1,NR
00386          U(II)=S(I)
00387   133    II=II+NDIM
00388   135    JJ=JJ+NDIM1
00389          RETURN
00390          END
```

```
RT-11 MACRO V03  CROSS REFERENCE LISTING 10-APR-79 04:26:18 PAGE 00003

A       000014G 000020I 000012  000033  000041  000042G 000046  000050*
        000057G 000067  000072* 000083
ADV     000006* 000016  000017* 000045
DOT     000012  000033  000041  000046  000056  000059  000067
FAC     000020* 000045  000045* 000047I 000059* 000070* 000072  000074
GETINV  000150U
I       000020D 000026  000049D 000050  000071D 000072  000073D 000074
        000079D 000080  000081  000083  000085D 000086
II      000080* 000083  000084* 000086  000087*
J       000120D 000012  000131I 000014  000119D 000020  000021  000127I
        000051  000059  000062  000073D 000080  000084
JCM     000023* 000024D 000026  000073D
JJ      000010* 000012  000015* 000018* 000022  000023  000024D 000033
        000034G 000041  000042G 000043G 000045  000049D 000050  000057G
        000059  000067  000068G 000071D 000073D 000075* 000077* 000082D
        000088*
JMI     000021* 000023  000030D 000039D 000050D 000064D
JRN     000022* 000049D 000071D
K       000030D 000031I 000034G 000039D 000039I 000043G 000050D 000054I
        000056  000064D 000065I 000068G
OK      000020* 000033  000034G 000035* 000037* 000041  000042G 000043G
        000044* 000052* 000056  000057G 000058* 000062* 000067  000068G
        000069* 000082D 000083
L       000030D
NR      000014G 000007* 000050*
NT      000031G
NC      000014G 000007  000012D 000019D 000076  000077D
NDIM    000030I 000004  000015  000035  000044  000058  000059  000075
        000076  000082D 000087
NDIM1   000004* 000028
NEXT    000076* 000082D
NONAME  000030N
NR      000014G 000028  000012  000033  000041  000042G 000046  000057G
        000067  000079D 000085D
NRM1    000008* 000022
S       000020I 000012* 000131I 000014  000020  000031I 000039I 000050*
        000054I 000062* 000065I 000081* 000083* 000086
START   000070
TEMP    000033* 000034G 000041* 000042G 000043G 000056* 000057G 000067*
        000068G
TOL     000005* 000045
TOL1    000009* 000013I 000014* 000016* 000017  000045* 000047I
U       000014G 000020I 000025* 000026* 000034G 000043G 000056  000059
        000068G 000074* 000083  000085*
VALUE   000034CL 000042CL 000043CL 000057CL 000068CL
11      000112D 000015*
100     000100D 000075*
125     000079D 000080D 000083*
130     000085D 000087*
135     000078D 000089*
20      000020D 000025*
30      000030D 000032I 000035*
50      000036D 000039D 000040I 000044*
54      000040I 000047*
```

```
55      000490  0035*
65      000500  0055#  0053*
70      000522  0052*
72      000600  0035#  0059*
75      000620  0077*
80      000720  0072*
85      000750  0076*
```

```
      C       FUNCTION TO COMPUTE THE DOT PRODUCT
00001         FUNCTION DOT(NR,A,B)
00002         DIMENSION A(1),B(1)
00003         DOT=0.
00004         DO 1 I=1,NR
00005       1 DOT=DOT+A(I)*B(I)
00006         RETURN
00007         END
```

RT-11 INDEX V03 CROSS REFERENCE LISTING 10-APR-79 04:26:19 PAGE 00002

```
A       000016G 000042D  00005
B       000018G 000042D  00005
DOT     000010U 00003=   00005=
I       000040D 00005
NR      000014G 000040D
1       000040D 00005*
```

```
      C        SUBROUTINE TO COMPUTE THE VECTOR SUM
0001          SUBROUTINE VADD(N,C1,A,B)
0002          DIMENSION A(1),B(1)
0003          DO 1 I=1,N
0004        1 A(I)=A(I)+C1*B(I)
0005          RETURN
0006          END
```

```
RT-11 INDEX V03  CROSS REFERENCE LISTING 10-APR-79 04:26:18 PAGE 0002

A       0000AG  0002DI  0004=
B       0000AG  0002DI  0004
C1      0000AG  0004
I       0003DO  0004
N       0000AG  0003DO
VALID   0001SU
X       0003DO  0004*
```

```
00001          SUBROUTINE HQR(NM,N,LOW,IGH,H,WR,WI,IERR)
00002          INTEGER I,J,K,L,M,N,EN,LL,MM,NA,NM,IGH,ITS,LOW,MP2,ENM2,IERR
00003          REAL H(NM,N),WR(N),WI(N)
00004          REAL P,Q,R,S,T,W,X,Y,ZZ,NORM,MACHEP
00005          REAL SQRT,ABS,SIGN
00006          INTEGER MIN0
00007          LOGICAL NOTLAS
               C      ********** MACHEP IS A MACHINE DEPENDENT PARAMETER SPECIFYING
               C                 THE RELATIVE PRECISION OF FLOATING POINT ARITHMETIC.
               C                 **********
00008          MACHEP=2.**(-26)
00009          IERR=0
00010          NORM=0.0
00011          K=1
               C      ********** STORE ROOTS ISOLATED BY BALANC
               C                 AND COMPUTE MATRIX NORM **********
00012          DO 50 I=1,N
00013          DO 40 J=K,N
00014     40   NORM=NORM+ABS(H(I,J))
00015          K=I
00016          IF(I.GE.LOW.AND.I.LE.IGH) GO TO 50
00018          WR(I)=H(I,I)
00019          WI(I)=0.0
00020     50   CONTINUE
00021          EN=IGH
00022          T=0.0
               C      ********** SEARCH FOR NEXT EIGENVALUES **********
00023     60   IF(EN.LT.LOW) GO TO 1001
00025          ITS=0.0
00026          NA=EN-1
00027          ENM2=NA-1
               C      ********** LOOK FOR SINGLE SMALL SUB-DIAGONAL ELEMENT
               C                 FOR L=EN STEP -1 UNTIL LOW DO -- **********
00028     70   DO 80 LL=LOW,EN
00029          L=EN+LOW-LL
00030          IF(L.EQ.LOW) GO TO 100
00032          S=ABS(H(L-1,L-1))+ABS(H(L,L))
00033          IF(S.EQ.0.0) S=NORM
00035          IF(ABS(H(L,L-1)).LE.MACHEP*S) GO TO 100
00037     80   CONTINUE
               C      ********** FORM SHIFT **********
00038    100   X=H(EN,EN)
00039          IF(L.EQ.EN) GO TO 270
00041          Y=H(NA,NA)
00042          W=H(EN,NA)*H(NA,EN)
00043          IF(L.EQ.NA) GO TO 280
00045          IF(ITS.EQ.30) GO TO 1000
00047          IF(ITS.NE.10.AND.ITS.NE.20) GO TO 130
               C      ********** FORM EXCEPTIONAL SHIFT **********
00049          T=T+X
00050          DO 120 I=LOW,EN
00051    120   H(I,I)=H(I,I)-X
00052          S=ABS(H(EN,NA))+ABS(H(NA,ENM2))
00053          X=0.75*S
```

```
00054          Y=X
00055          W=-0.4375*S*S
00056 130      ITS=ITS+1
               C     ********** LOOK FOR TWO CONSECUTIVE SMALL
               C                SUB-DIAGONAL ELEMENTS.
               C                FOR M=EN-2 STEP -1 UNTIL L DO -- **********
00057          DO 140 MM=L,ENM2
00058          M=ENM2+L-MM
00059          ZZ=H(M,M)
00060          R=X-ZZ
00061          S=Y-ZZ
00062          P=(R*S-W)/H(M+1,M)+H(M,M+1)
00063          Q=H(M+1,M+1)-ZZ-R-S
00064          R=H(M+2,M+1)
00065          S=ABS(P)+ABS(Q)+ABS(R)
00066          P=P/S
00067          Q=Q/S
00068          R=R/S
00069          IF(M.EQ.L) GO TO 150
00071          IF(ABS(H(M,M-1))*(ABS(Q)+ABS(R)).LE.MACHEP*ABS(P)
              +    *(ABS(H(M-1,M-1))+ABS(ZZ)+ABS(H(M+1,M+1)))) GO TO 150
00073 140      CONTINUE
00074 150      MP2=M+2
00075          DO 160 I=MP2,EN
00076          H(I,I-2)=0.0
00077          IF(I.EQ.MP2) GO TO 160
00079          H(I,I-3)=0.0
00080 160      CONTINUE
               C     ********** DOUBLE QR STEP INVOLVING ROWS L TO EN AND
               C                COLUMNS M TO EN **********
00081          DO 260 K=M,NA
00082          NOTLAS=K.NE.NA
00083          IF(K.EQ.M) GO TO 170
00085          P=H(K,K-1)
00086          Q=H(K+1,K-1)
00087          R=0.0
00088          IF(NOTLAS) R=H(K+2,K-1)
00090          X=ABS(P)+ABS(Q)+ABS(R)
00091          IF(X.EQ.0.0) GO TO 260
00093          P=P/X
00094          Q=Q/X
00095          R=R/X
00096 170      S=SIGN(SQRT(P*P+Q*Q+R*R),P)
00097          IF(K.EQ.M) GO TO 180
00099          H(K,K-1)=-S*X
00100          GO TO 190
00101 180      IF(L.NE.M) H(K,K-1)=-H(K,K-1)
00103 190      P=P+S
00104          X=P/S
00105          Y=Q/S
00106          ZZ=R/S
00107          Q=Q/P
00109          R=R/P
               C     ********** ROW MODIFICATION **********
```

```
00109         DO 210 J=K,EN
00110         P=H(K,J)+Q*H(K+1,J)
00111         IF(.NOT.NOTLAS) GO TO 200
00113         P=P+R*H(K+2,J)
00114         H(K+2,J)=H(K+2,J)-P*ZZ
00115 200     H(K+1,J)=H(K+1,J)-P*Y
00116         H(K,J)=H(K,J)-P*X
00117 210     CONTINUE
00118         J=MIN0(EN,K+3)
      C       ********** COLUMN MODIFICATION **********
00119         DO 230 I=L,J
00120         P=X*H(I,K)+Y*H(I,K+1)
00121         IF(.NOT.NOTLAS) GO TO 220
00123         P=P+ZZ*H(I,K+2)
00124         H(I,K+2)=H(I,K+2)-P*R
00125 220     H(I,K+1)=H(I,K+1)-P*Q
00126         H(I,K)=H(I,K)-P
00127 230     CONTINUE
00128 260     CONTINUE
00129         GO TO 70
      C       ********** ONE ROOT FOUND **********
00130 270     WR(EN)=X+T
00131         WI(EN)=0.0
00132         EN=NA
00133         GO TO 60
      C       ********** TWO ROOTS FOUND **********
00134 280     P=(Y-X)/2.0
00135         Q=P*P+W
00136         ZZ=SQRT(ABS(Q))
00137         X=X+T
00138         IF(Q.LT.0.0) GO TO 320
      C       ********** REAL PAIR **********
00140         ZZ=P+SIGN(ZZ,P)
00141         WR(NA)=X+ZZ
00142         WR(EN)=WR(NA)
00143         IF(ZZ.NE.0.0) WR(EN)=X-W/ZZ
00145         WI(NA)=0.0
00146         WI(EN)=0.0
00147         GO TO 330
      C       ********** COMPLEX PAIR **********
00148 320     WR(NA)=X+P
00149         WR(EN)=X+P
00150         WI(NA)=ZZ
00151         WI(EN)=-ZZ
00152 330     EN=ENM2
00153         GO TO 60
      C       ********** SET ERROR -- NO CONVERGENCE TO AN
      C                   EIGENVALUE AFTER 30 ITERATIONS **********
00154 1000    IERR=EN
00155 1001    RETURN
00156         END
```

```
ABS     00005RL 00014   00032   00035IF 00052   00055   00071IF 00090
        00136
EN      00002IN 00021=  00023IF 00026   00028DO 00029   00038   00039IF
        00042   00050DO 00052   00075DO 00109DO 00118   00130   00131
        00132=  00142   00144   00145   00149   00151   00152=  00154
EN42    00002IN 00027=  00052   00057DO 00058   00152
H       00001AG 00003RL 00014   00018   00032   00035IF 00038   00041
        00042   00051=  00052   00059   00062   00063   00064   00071IF
        00076=  00079=  00085   00086   00089   00099=  00102=  00110
        00113   00114=  00115=  00116=  00120   00123   00124=  00125=
        00126=
HDR     00001SU
I       00002IN 00012DO 00014   00015   00016IF 00018   00019   00050DO
        00051   00075DO 00076   00077IF 00079   00119DO 00120   00123
        00124   00125   00126
IERR    00001AG 00002IN 00009=  00154=
IGH     00001AG 00002IN 00016IF 00021
ITS     00002IN 00025=  00045IF 00047IF 00056=
J       00002IN 00013DO 00014   00109DO 00110   00113   00114   00115
        00116   00118=  00119DO
K       00002IN 00011=  00013DO 00015=  00081DO 00082   00083IF 00085
        00086   00089   00097IF 00099   00102   00109DO 00110   00113
        00114   00115   00116   00118   00120   00123   00124   00125
        00126
L       00002IN 00029=  00030IF 00032   00035IF 00039IF 00043IF 00057DO
        00058   00059IF 00101IF 00119DO
LL      00002IN 00028DO 00029
LOW     00001AG 00002IN 00016IF 00023IF 00028DO 00029   00038IF 00050DO
M       00002IN 00058=  00059   00062   00063   00064   00069IF 00071IF
        00074   00081DO 00083IF 00097IF 00101IF
MACHEP  00004RL 00008=  00035IF 00071IF
MING    00006IN 00118
MM      00002IN 00057DO 00058
MP2     00002IN 00074=  00075DO 00077IF
N       00001AG 00002IN 00003RL 00012DO 00013DO
NA      00002IN 00026=  00027   00041   00042   00043IF 00052   00081DO
        00082   00132   00141   00142   00145   00148   00150
NM      00001AG 00002IN 00003RL
NORM    00004RL 00010=  00014=  00034
NOTLAS  00007LG 00082=  00088IF 00111IF 00121IF
P       00004RL 00052=  00055   00056=  00071IF 00085=  00090   00093=
        00095   00103=  00104   00107   00108   00110=  00113=  00114
        00115   00116   00120=  00123=  00124   00125   00126   00134=
        00135   00140   00148   00149
Q       00004RL 00053=  00055   00057=  00071IF 00086=  00090   00094=
        00095   00105   00107=  00110   00125   00135=  00136   00139IF
R       00004RL 00050=  00052   00053   00054=  00055   00058=  00071IF
        00087=  00089=  00090   00095=  00095   00105   00108=  00113
        00124
S       00004RL 00032=  00033IF 00034=  00035IF 00052=  00053   00055
        00061=  00062   00063   00065=  00066   00067   00068   00096=
        00099   00103   00104   00105   00106
SIGN    00005RL 00096   00140
SQRT    00005RL 00096   00135
```

```
T       000004RL 000022=  000049=  000130   000137
W       000004RL 000042=  000055=  000062   000135   000144
WI      000001AG 000003RL 000019=  000131=  000145=  000146=  000150=  000151=
WR      000001AG 000003RL 000018=  000130=  000141=  000142=  000144=  000148=
        000149=
X       000004RL 000038=  000049   000051   000053=  000054   000060   000090=
        000091IF 000093   000094   000095   000099   000104=  000116   000120
        000130   000134   000137=  000141   000144   000148   000149
Y       000004RL 000041=  000054=  000061   000105=  000115   000120   000134
ZZ      000004RL 000059=  000060   000061   000063   000071IF 000106=  000114
        000123   000136=  000140=  000141   000143IF 000144   000150   000151
100     000031GT 000036GT 000038*
1000    000040GT 000154*
1001    000024GT 000155*
120     000050DO 000051*
130     000048GT 000056*
140     000057DO 000073*
150     000070GT 000072GT 000074*
160     000075DO 000078GT 000080*
170     000084GT 000096*
180     000090GT 000101*
190     000100GT 000103*
200     000112GT 000115*
210     000109DO 000117*
220     000122GT 000125*
230     000119DO 000127*
260     000081DO 000092GT 000128*
270     000066GT 000130*
280     000044GT 000134*
320     000139GT 000148*
330     000147GT 000152*
40      000013DO 000014*
50      000012DO 000017GT 000020*
60      000023*  000130GT 000153GT
70      000026*  000129GT
80      000028DO 000037*
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
00001       SUBROUTINE HQR2(NM,N,LOW,IGH,H,WR,WI,Z,IERR)
00002       INTEGER I,J,K,L,M,N,EN,II,JJ,LL,MM,NA,NM,NN,
           +        IGH,ITS,LOW,MP2,ENM2,IERR
00003       REAL H(NM,N),WR(N),WI(N),Z(NM,N)
00004       REAL P,Q,R,S,T,W,X,Y,RA,SA,VI,VR,ZZ,NORM,MACHEP
00005       REAL SQRT,ABS,SIGN
00006       INTEGER MIN0
00007       LOGICAL NOTLAS
00008       COMPLEX Z3
00009       COMPLEX CMPLX
00010       REAL REAL,AIMAG
      C     ********** MACHEP IS A MACHINE DEPENDENT PARAMETER SPECIFYING
      C                THE RELATIVE PRECISION OF FLOATING POINT ARITHMETIC.
      C                **********
00011       MACHEP=2.**(-26)
00012       IERR=0
00013       NORM=0.0
00014       K=1
      C     ********** STORE ROOTS ISOLATED BY BALANC
      C                AND COMPUTE MATRIX NORM **********
00015       DO 50 I=1,N
00016       DO 40 J=K,N
00017  40   NORM=NORM+ABS(H(I,J))
00018       K=I
00019       IF(I.GE.LOW.AND.I.LE.IGH) GO TO 50
00021       WR(I)=H(I,I)
00022       WI(I)=0.0
00023  50   CONTINUE
00024       EN=IGH
00025       T=0.0
      C     ********** SEARCH FOR NEXT EIGENVALUES **********
00026  60   IF(EN.LT.LOW) GO TO 340
00028       ITS=0
00029       NA=EN-1
00030       ENM2=NA-1
      C     ********** LOOK FOR SINGLE SMALL SUB-DIAGONAL ELEMENT
      C                FOR L=EN STEP -1 UNTIL LOW DO -- **********
00031  70   DO 80 LL=LOW,EN
00032       L=EN+LOW-LL
00033       IF(L.EQ.LOW) GO TO 100
00035       S=ABS(H(L-1,L-1))+ABS(H(L,L))
00036       IF(S.EQ.0.0) S=NORM
00038       IF(ABS(H(L,L-1)).LE.MACHEP*S) GO TO 100
00040  80   CONTINUE
      C     ********** FORM SHIFT **********
00041  100  X=H(EN,EN)
00042       IF(L.EQ.EN) GO TO 270
00044       Y=H(NA,NA)
00045       W=H(EN,NA)*H(NA,EN)
00046       IF(L.EQ.NA) GO TO 280
00048       IF(ITS.EQ.30) GO TO 1000
00050       IF(ITS.NE.10.AND.ITS.NE.20) GO TO 130
      C     ********** FORM EXCEPTIONAL SHIFT **********
00052       T=T+X
```

```
00053        DO 120 I=LOW,EN
00054 120    H(I,I)=H(I,I)-X
00055        S=ABS(H(EN,NA))+ABS(H(NA,ENM2))
00056        X=0.75*S
00057        Y=X
00058        W=-0.4375*S*S
00059 130    ITS=ITS+1
     C       ********** LOOK FOR TWO CONSECUTIVE SMALL
     C                   SUB-DIAGONAL ELEMENTS.
     C                   FOR M=EN-2 STEP -1 UNTIL L DO -- **********
00060        DO 140 MM=L,ENM2
00061        M=ENM2+L-MM
00062        ZZ=H(M,M)
00063        R=X-ZZ
00064        S=Y-ZZ
00065        P=(R*S-W)/H(M+1,M)+H(M,M+1)
00066        Q=H(M+1,M+1)-ZZ-R-S
00067        R=H(M+2,M+1)
00068        S=ABS(P)+ABS(Q)+ABS(R)
00069        P=P/S
00070        Q=Q/S
00071        R=R/S
00072        IF(M.EQ.L) GO TO 150
00074        IF(ABS(H(M,M-1))*(ABS(Q)+ABS(R)).LE.MACHEP*ABS(P)
            +    *(ABS(H(M-1,M-1))+ABS(ZZ)+ABS(H(M+1,M+1)))) GO TO 150
00075 140    CONTINUE
00077 150    MP2=M+2
00078        DO 160 I=MP2,EN
00079        H(I,I-2)=0.0
00080        IF(I.EQ.MP2) GO TO 160
00082        H(I,I-3)=0.0
00083 160    CONTINUE
     C       ********** DOUBLE QR STEP INVOLVING ROWS L TO EN AND
     C                   COLUMNS M TO EN **********
00084        DO 260 K=M,NA
00085        NOTLAS=K.NE.NA
00086        IF(K.EQ.M) GO TO 170
00088        P=H(K,K-1)
00089        Q=H(K+1,K-1)
00090        R=0.0
00091        IF(NOTLAS) R=H(K+2,K-1)
00093        X=ABS(P)+ABS(Q)+ABS(R)
00094        IF(X.EQ.0.0) GO TO 260
00095        P=P/X
00097        Q=Q/X
00098        R=R/X
00099 170    S=SIGN(SQRT(P*P+Q*Q+R*R),P)
00100        IF(K.EQ.M) GO TO 180
00102        H(K,K-1)=-S*X
00103        GO TO 190
00104 180    IF(L.NE.M) H(K,K-1)=-H(K,K-1)
00105 190    P=P+S
00107        X=P/S
00108        Y=Q/S
```

```
00109          ZZ=R/S
00110          Q=Q/P
00111          R=R/P
               C     ********** ROW MODIFICATION **********
00112          DO 210 J=K,N
00113          P=H(K,J)+Q*H(K+1,J)
00114          IF(.NOT.NOTLAS) GO TO 200
00115          P=P+R*H(K+2,J)
00117          H(K+2,J)=H(K+2,J)-P*ZZ
00118 200      H(K+1,J)=H(K+1,J)-P*Y
00119          H(K,J)=H(K,J)-P*X
00120 210      CONTINUE
00121          J=MINO(EN,K+3)
               C     ********** COLUMN MODIFICATION **********
00122          DO 230 I=1,J
00123          P=X*H(I,K)+Y*H(I,K+1)
00124          IF(.NOT.NOTLAS) GO TO 220
00126          P=P+ZZ*H(I,K+2)
00127          H(I,K+2)=H(I,K+2)-P*R
00128 220      H(I,K+1)=H(I,K+1)-P*Q
00129          H(I,K)=H(I,K)-P
00130 230      CONTINUE
               C     ********** ACCUMULATE TRANSFORMATIONS **********
00131          DO 250 I=LOW,IGH
00132          P=X*Z(I,K)+Y*Z(I,K+1)
00133          IF(.NOT.NOTLAS) GO TO 240
00135          P=P+ZZ*Z(I,K+2)
00136          Z(I,K+2)=Z(I,K+2)-P*R
00137 240      Z(I,K+1)=Z(I,K+1)-P*Q
00138          Z(I,K)=Z(I,K)-P
00139 250      CONTINUE
00140 260      CONTINUE
00141          GO TO 70
               C     ********** ONE ROOT FOUND **********
00142 270      H(EN,EN)=X+T
00143          WR(EN)=H(EN,EN)
00144          WI(EN)=0.0
00145          EN=NA
00146          GO TO 60
               C     ********** TWO ROOTS FOUND **********
00147 280      P=(Y-X)/2.0
00148          Q=P*P+W
00149          ZZ=SQRT(ABS(Q))
00150          H(EN,EN)=X+T
00151          X=H(EN,EN)
00152          H(NA,NA)=Y+T
00153          IF(Q.LT.0.0) GO TO 320
               C     ********** REAL PAIR **********
00155          ZZ=P+SIGN(ZZ,P)
00156          WR(NA)=X+ZZ
00157          WR(EN)=WR(NA)
00159          IF(ZZ.NE.0.0) WR(EN)=X-W/ZZ
00160          WI(NA)=0.0
00161          WI(EN)=0.0
```

```
00162          X=H(EN,NA)
00163          S=ABS(X)+ABS(ZZ)
00164          P=X/S
00165          Q=ZZ/S
00166          R=SQRT(P*P+Q*Q)
00167          P=P/R
00168          Q=Q/R
               C     ********** ROW MODIFICATION **********
00169          DO 290 J=NA,N
00170          ZZ=H(NA,J)
00171          H(NA,J)=Q*ZZ+P*H(EN,J)
00172          H(EN,J)=Q*H(EN,J)-P*ZZ
00173 290      CONTINUE
               C     ********** COLUMN MODIFICATION **********
00174          DO 300 I=1,EN
00175          ZZ=H(I,NA)
00176          H(I,NA)=Q*ZZ+P*H(I,EN)
00177          H(I,EN)=Q*H(I,EN)-P*ZZ
00178 300      CONTINUE
               C     ********** ACCUMULATE TRANSFORMATIONS **********
00179          DO 310 I=LOW,IGH
00180          ZZ=Z(I,NA)
00181          Z(I,NA)=Q*ZZ+P*Z(I,EN)
00182          Z(I,EN)=Q*Z(I,EN)-P*ZZ
00183 310      CONTINUE
00184          GO TO 330
               C     ********** COMPLEX PAIR **********
00185 320      WR(NA)=X+P
00186          WR(EN)=X+P
00187          WI(NA)=ZZ
00188          WI(EN)=-ZZ
00189 330      EN=EN-2
00190          GO TO 60
               C     ********** ALL ROOTS FOUND. BACKSUBSTITUTE TO FIND
               C                VECTORS OF UPPER TRIANGULAR FORM **********
00191 340      IF(NORM.EQ.0.0) GO TO 1001
               C     ********** FOR EN=N STEP -1 UNTIL 1 DO -- **********
00193          DO 800 NN=1,N
00194          EN=N+1-NN
00195          P=WR(EN)
00196          Q=WI(EN)
00197          NA=EN-1
00198          IF(Q) 710,600,800
               C     ********** REAL VECTOR **********
00199 600      M=EN
00200          H(EN,EN)=1.0
00201          IF(NA.EQ.0) GO TO 800
               C     ********** FOR I=EN-1 STEP -1 UNTIL 1 DO -- **********
00203          DO 700 II=1,NA
00204          I=EN-II
00205          W=H(I,I)-P
00206          R=H(I,EN)
00207          IF(M.GT.NA) GO TO 620
00209          DO 610 J=M,NA
```

ORIGINAL PAGE IS OF POOR QUALITY

```
00210 510     R=R+H(I,J)*H(J,IN)
00211 520     IF(WI(I).GE.0.0) GO TO 530
00213         ZZ=W
00214         S=R
00215         GO TO 700
00216 530     W=I
00217         IF(WI(I).NE.0.0) GO TO 540
00219         T=W
00220         IF(W.EQ.0.0) T=MACHEP*NORM
00222         H(I,IN)=-R/T
00223         GO TO 700
              C     ********** SOLVE REAL EQUATIONS **********
00224 540     X=H(I,I+1)
00225         Y=H(I+1,I)
00226         Q=(WR(I)-P)*(WR(I)-P)+WI(I)*WI(I)
00227         T=(X*S-ZZ*R)/Q
00228         H(I,IN)=T
00229         IF(ABS(X).LE.ABS(ZZ)) GO TO 550
00231         H(I+1,IN)=(-R-Y*T)/X
00232         GO TO 700
00233 550     H(I+1,IN)=(-S-Y*T)/ZZ
00234 700     CONTINUE
              C     ********** END REAL VECTOR **********
00235         GO TO 800
              C     ********** COMPLEX VECTOR **********
00236 710     M=NA
              C     ********** LAST VECTOR COMPONENT CHOSEN IMAGINARY SO THAT
              C                EIGENVECTOR MATRIX IS TRIANGULAR **********
00237         IF(ABS(H(EN,NA)).LE.ABS(H(NA,EN))) GO TO 720
00239         H(NA,NA)=Q/H(EN,NA)
00240         H(NA,EN)=-(H(EN,EN)-P)/H(EN,NA)
00241         GO TO 730
00242 720     ZZ=CMPLX(0.0,-H(NA,EN))/CMPLX(H(NA,NA)-P,Q)
00243         H(NA,NA)=REAL(ZZ)
00244         H(NA,EN)=AIMAG(ZZ)
00245 730     H(EN,NA)=0.0
00246         H(EN,EN)=1.0
00247         EN2=NA-1
00248         IF(EN2.EQ.0) GO TO 800
              C     ********** FOR I=EN-2 STEP -1 UNTIL 1 DO -- **********
00250         DO 790 II=1,EN2
00251         I=NA-II
00252         W=H(I,I)-P
00253         RA=0.0
00254         SA=H(I,IN)
00255         DO 760 J=M,NA
00256         RA=RA+H(I,J)*H(J,NA)
00257         SA=SA+H(I,J)*H(J,IN)
00258 760     CONTINUE
00259         IF(WI(I).GE.0.0) GO TO 770
00261         ZZ=W
00262         R=RA
00263         S=SA
00264         GO TO 790
```

```
00265 770    M=I
00266        IF(WI(I).NE.0.0) GO TO 780
00268        ZZ=CMPLX(-RA,-SA)/CMPLX(W,Q)
00269        H(I,NA)=REAL(ZZ)
00270        H(I,EN)=AIMAG(ZZ)
00271        GO TO 790
             C     ********** SOLVE COMPLEX EQUATIONS **********
00272 780    X=H(I,I+1)
00273        Y=H(I+1,I)
00274        VR=(WR(I)-P)*(WR(I)-P)+WI(I)*WI(I)-Q*Q
00275        VI=(WR(I)-P)*2.0*Q
00276        IF(VR.EQ.0.0.AND.VI.EQ.0.0) VR=MACHEP*NORM
            +    *(ABS(W)+ABS(Q)+ABS(X)+ABS(Y)+ABS(ZZ))
00278        ZZ=CMPLX(X*R-ZZ*RA+Q*SA,X*S-ZZ*SA-Q*RA)/CMPLX(VR,VI)
00279        H(I,NA)=REAL(ZZ)
00280        H(I,EN)=AIMAG(ZZ)
00281        IF(ABS(X).LE.ABS(ZZ)+ABS(Q)) GO TO 785
00283        H(I+1,NA)=(-RA-W*H(I,NA)+Q*H(I,EN))/X
00284        H(I+1,EN)=(-SA-W*H(I,EN)-Q*H(I,NA))/X
00285        GO TO 790
00286 785    CC=CMPLX(-R-Y*H(I,NA),-S-Y*H(I,EN))/CMPLX(ZZ,Q)
00287        H(I+1,NA)=REAL(ZZ)
00288        H(I+1,EN)=AIMAG(ZZ)
00289 790    CONTINUE
             C     ********** END COMPLEX VECTOR **********
00290 800    CONTINUE
             C     ********** END BACK SUBSTITUTION.
             C                VECTORS OF ISOLATED ROOTS **********
00291        DO 840 I=1,N
00292        IF(I.GE.LOW.AND.I.LE.IGH) GO TO 840
00294        DO 820 J=I,N
00295 820    Z(I,J)=H(I,J)
00296 840    CONTINUE
             C     ********** MULTIPLY BY TRANSFORMATION MATRIX TO GIVE
             C                VECTORS OF ORIGINAL FULL MATRIX.
             C                FOR J=N STEP -1 UNTIL LOW DO -- **********
00297        DO 880 JJ=LOW,N
00298        J=N+LOW-JJ
00299        M=MIN0(J,IGH)
00300        DO 880 I=LOW,IGH
00301        ZZ=0.0
00302        DO 860 K=LOW,M
00303 860    ZZ=ZZ+Z(I,K)*H(K,J)
00304        Z(I,J)=ZZ
00305 880    CONTINUE
00306        GO TO 1001
             C     ********** SET ERROR -- NO CONVERGENCE TO AN
             C                EIGENVALUE AFTER 30 ITERATIONS **********
00307 1000   IERR=EN
00308 1001   RETURN
00309        END
```

```
WR      000001AG 00003RL 00021= 00143= 00156= 00157= 00159= 00165=
        00196= 00195  00226  00274  00275
X       000004RL 00041= 00052  00054  00056= 00057  00063  00093=
        000094IF 00096  00097  00098  00102  00107= 00119  00123
        00132  00142  00147  00150  00151= 00155  00159  00162=
        00163  00164  00165  00186  00224= 00227  00229IF 00231
        00272= 00277  00278  00281IF 00283  00284
Y       000004RL 00044= 00057= 00064  00108= 00118  00123  00132
        00147  00152  00225= 00233  00273= 00277  00285
Z       000001AG 00003RL 00132  00135  00136= 00137= 00138= 00180
        00181= 00182= 00295= 00303  00304=
ZZ      000004RL 00052= 00063  00064  00065  00074IF 00109= 00117
        00126  00135  00149= 00155= 00156  00158IF 00159  00163
        00165  00170= 00171  00172  00175= 00176  00177  00180=
        00181  00182  00187  00188  00213= 00227  00229IF 00233
        00251= 00277  00278  00281IF 00285  00301= 00303= 00304
Z3      000003CX 00242= 00243  00244  00259= 00259  00270  00278=
        00279  00280  00286= 00287  00288
100     000346T 00359T 00341*
1000    000496T 00307*
1001    000926T 00306T 00308*
120     000560D 00054*
130     000516T 00059*
140     000600D 00076*
150     000736T 00756T 00777*
160     000796D 001016T 00083*
170     000876T 00099*
180     001016T 00104*
190     001036T 00106*
200     001156T 00118*
210     001126D 00120*
220     001256T 00128*
230     001226D 00130*
240     001346T 00137*
250     001316D 00139*
260     003460D 003956T 00144*
270     003476T 00142*
280     003476T 00147*
290     003596D 00173*
300     003746D 00178*
310     003796D 00183*
320     004056T 00185*
330     004076T 00189*
340     004276T 00191*
40      000160D 00017*
50      000196D 00026T 00023*
60      00025*  00146T 00190T
600     001931F 00199*
610     002096D 00210*
620     002096T 00211*
630     002176T 00216*
640     002176T 00224*
650     002336T 00233*
70      00031*  00141T
```

```
700     00207DO 00215GT 00223GT 00232GT 00234*
710     00199IF 00236*
720     00238GT 00242*
730     00241GT 00245*
750     00255DO 00258*
770     00260GT 00265*
780     00267GT 00272*
785     00282GT 00286*
790     00254DO 00254GT 00271GT 00285GT 00289*
80      00031DO 00040*
900     00193DO 00199IF 00202GT 00239GT 00249GT 00290*
820     00294DO 00295*
840     00291DO 00293GT 00296*
860     00307DO 00333*
880     00297DO 00300DO 00305*
```

```
00001        SUBROUTINE LEQT2C(A,N,IA,B,M,IB,IPVT,C,WK,IWK,IER)
             C  FUNCTION - MATRIX DECOMPOSITION, LINEAR EQUATION SOLUTION
             C             FOR COMPLEX MATRICES.
             C  ARGUMENTS  A - INPUT COMPLEX MATRIX, DIMENSIONED N BY N.
             C             N - ORDER OF A.
             C            IA - ROW DIMENSION OF A.
             C             B - INPUT COMPLEX MATRIX DIMENSIONED N BY M.
             C                 ON OUTPUT, SOLUTION MATRIX X REPLACES B.
             C             M - NUMBER OF RIGHT HAND SIDES.
             C            IB - ROW DIMENSION OF B.
             C          IPVT - PIVOT VECTOR
             C          C,WK - WORKING SPACE
             C           IWK - DIMENSION OF WK
             C           IER - TERMINAL ERROR 128+N:
             C                 N = 2 INDICATES A IS SINGULAR.
00002        DIMENSION IPVT(IA)
00003        COMPLEX A(IA,N),B(IB,M),C(IA),WK(IWK)
00004        IER=0
00005        NDIM=IA
00006        II=0
00007        DO 10 J=1,N
00008        DO 10 I=1,IA
00009        II=II+1
00010        WK(II)=A(I,J)
00011 10     CONTINUE
00012        CALL DECOMP(NDIM,N,A,COND,IPVT,WORK)
00013        IF(COND.EQ.COND+1) GO TO 50
00015        DO 40 J=1,M
00016        DO 20 I=1,N
00017 20     C(I)=B(I,J)
00018        CALL SOLVE(NDIM,N,A,C,IPVT)
00019        DO 30 I=1,N
00020 30     B(I,J)=C(I)
00021 40     CONTINUE
00022        II=0
00023        DO 45 J=1,N
00024        DO 45 I=1,IA
00025        II=II+1
00026        A(I,J)=WK(II)
00027 45     CONTINUE
00028        RETURN
00029 50     IER=129
00030        RETURN
00031        END
```

```
A       00001AG 00003CX 00010   00012AG 00018AG 00026=
B       00001AG 00003CX 00017   00020=
C       00001AG 00003CX 00017=  00018AG 00020
COND    00012AG 00013EF
DECOMP  00012CL
I       00007DO 00010   00015DO 00017   00019DO 00020   00024DO 00026
IA      00001AG 00002DI 00003CX 00005   00008DO 00024DO
IB      00001AG 00003CX
IER     00001AG 00004=  00029=
II      00006=  00009=  00010   00022=  00025=  00026
IPVT    00001AG 00002DI 00012AG 00018AG
IWK     00001AG 00003CX
J       00007DO 00010   00015DO 00017   00020   00023DO 00026
LEQT2C  00001SU
M       00001AG 00003CX 00015DO
N       00001AG 00003CX 00007DO 00012AG 00019DO 00018AG 00019DO 00023DO
NDIM    00005=  00012AG 00018AG
SOLVE   00018CL
WK      00001AG 00003CX 00010=  00026
WORK    00012AG
10      00007DO 00008DO 00011*
20      00015DO 00017*
30      00019DO 00020*
40      00019DO 00021*
45      00023DO 00024DO 00027*
50      00014GT 00029*
```

```
00001        SUBROUTINE LINEQ1(NDIM,N,A,C,X,E,F,TR,SCALE)
     C          SOLVES  AX + XA' + C = 0
00002        COMMON/NONAME/ NDIM1
00003        DIMENSION A(NDIM,N),C(NDIM,N),X(NDIM,N),E(NDIM,N),F(NDIM,N),TR(N)
00004        DIMENSION SCALE(N)
     C          BALANCE THE A AND C MATRICES
00005        CALL BALANC(NDIM,N,A,LOW,IGH,SCALE)
00006        DO 10 I=1,N
00007        DO 10 J=1,N
00008        E(I,J)=0.
00009 10     E(I,I)=1.
00010        DO 50 J=1,N
00011        IF(J.GE.LOW.AND.J.LE.IGH) GO TO 40
00013        K=SCALE(J)
00014        DO 20 I=1,N
00015        DUM=C(I,J)
00016        C(I,J)=C(I,K)
00017 20     C(I,K)=DUM
00018        DO 30 I=1,N
00019        DUM=C(J,I)
00020        C(J,I)=C(K,I)
00021 30     C(K,I)=DUM
00022        GO TO 50
00023 40     E(J,J)=SCALE(J)
00024 50     CONTINUE
00025        NDIM1=NDIM
00026        CALL G1INV(N,N,E,X,MT,0)
00027        CALL MMUL(NDIM,X,C,N,N,N,F)
00028        CALL MMUL(NDIM,F,X,N,N,N,C)
     C          SOLVE AX + XA' + C = 0
00029        TOL=.0001
00030        DT=.5
00031        DT1=0.
00032        DO 70 I=1,N
00033 70     DT1=DT1+A(I,I)
00034        DT1=-DT1/N
00035        IF(DT1-4.0) 90,90,80
00036 80     DT=DT*4.0/DT1
00037 90     DO 110 I=1,N
00038        DO 100 J=1,N
00039 100    X(I,J)=DT*A(J,I)
00040 110    X(I,I)=X(I,I)-.5
00041        NDIM1=NDIM
00042        CALL G1INV(N,N,X,F,MR,0)
00043        CALL MMUL(NDIM,C,F,N,N,N,E)
00044        DO 130 I=1,N
00045        DO 120 J=1,N
00046 120    C(I,J)=F(J,I)*DT
00047 130    F(I,I)=F(I,I)+1.
00048        CALL MMUL(NDIM,C,E,N,N,N,X)
00049        DO 180 IT=1,20
00050        ME2=0
00051        CALL MMUL(NDIM,X,F,N,N,N,C)
00052        DO 170 I=1,N
```

```
00053          TR(I)=X(I,I)
00054          DO 150 J=I,N
00055          E(I,J)=F(I,J)
00056          E(J,I)=F(J,I)
00057          DO 140 K=1,N
00058  140     X(I,J)=X(I,J)+F(K,I)*C(K,J)
00059  150     X(J,I)=X(I,J)
00060          IF(ABS(X(I,I)-TR(I))-(.0000001+TOL*ABS(X(I,I)))) 160,160,170
00061  160     NEZ=NEZ+1
00062  170     CONTINUE
00063          ITT=IT
00064          IF(NEZ-N) 180,190,190
00065  180     CALL MMUL(NDIM,E,E,N,N,N,F)
00066  190     CONTINUE
00067          TOL=ITT
       C            BALANCE BACK THE MATRIX
00068          DO 230 I=1,N
00069          DO 230 J=1,N
00070          E(I,J)=0.
00071          E(I,I)=1.
00072          IF(I.GE.LOW.AND.I.LE.IGH) E(I,I)=SCALE(I)
00074  230     CONTINUE
00075          CALL MMUL(NDIM,E,X,N,N,N,F)
00076          CALL MMUL(NDIM,F,E,N,N,N,X)
00077          DO 240 J=1,N
00078          IF(J.GE.LOW.AND.J.LE.IGH) GO TO 240
00080          K=SCALE(J)
00081          DO 220 I=1,N
00082          DUM=X(I,J)
00083          X(I,J)=X(I,K)
00084  220     X(I,K)=DUM
00085          DO 230 I=1,N
00086          DUM=X(J,I)
00087          X(J,I)=X(K,I)
00088  230     X(K,I)=DUM
00089  240     CONTINUE
00090          CALL MMUL(NDIM,E,A,N,N,N,F)
00091          DO 250 I=1,N
00092  250     E(I,I)=1./E(I,I)
00093          CALL MMUL(NDIM,F,E,N,N,N,A)
00094          DO 280 J=1,N
00095          IF(J.GE.LOW.AND.J.LE.IGH) GO TO 280
00097          K=SCALE(J)
00098          DO 260 I=1,N
00099          DUM=A(I,J)
00100          A(I,J)=A(I,K)
00101  260     A(I,K)=DUM
00102          DO 270 I=1,N
00103          DUM=A(J,I)
00104          A(J,I)=A(K,I)
00105  270     A(K,I)=DUM
00106  280     CONTINUE
       C            THE END
00107          RETURN
```

00108 END

```
A       000001AG 000003DI 000005AG 000033   000039   000090AG 000093AG 000099
        000100=  000101=  000103   000104=  000105=
ABS     000000
BALANC  000035CL
C       000021AG 000023DI 000015   000016=  000017=  000019   000020=  000021=
        000027AG 000028AG 000043AG 000046=  000043AG 000051AG 000058
DT      000030=  000036=  000039   000046
DT1     000031=  000033=  000034=  000035IF 000036
DUM     000015=  000017   000019=  000021   000032=  000034   000036=  000088
        000099=  000101   000103=  000105
E       000021AG 000023DI 000008=  000009=  000023=  000026AG 000043AG 000048AG
        000055=  000056=  000065AG 000070=  000071=  000073=  000075AG 000076AG
        000090AG 000092=  000093AG
F       000021AG 000023DI 000027AG 000028AG 000042AG 000043AG 000046   000047=
        000051AG 000055   000056   000058   000065AG 000075AG 000076AG 000093AG
        000093AG
G41NV   000020CL 000042CL
I       000000DO 000008   000009   000014DO 000015   000016   000017   000018DO
        000019   000020   000021   000032DO 000033   000037DO 000039   000040
        000044DO 000046   000047   000052DO 000053   000054DO 000055   000056
        000058   000059   000065IF 000069DO 000070   000071   000072IF 000073
        000080DO 000082   000083   000084   000085DO 000086   000087   000088
        000091DO 000092   000098DO 000099   000100   000101   000102DO 000103
        000104   000105
IGN     000045AG 000011IF 000072IF 000078IF 000095IF
IT      000049DO 000063
ITT     000063=  000067
J       000007DO 000008   000010DO 000011IF 000013   000015   000016   000019
        000020   000023   000038DO 000039   000045DO 000046   000054DO 000055
        000056   000058   000059   000069DO 000070   000077DO 000078IF 000080
        000082   000083   000086   000087   000094DO 000095IF 000097   000099
        000100   000103   000104
K       000013=  000016   000017   000020   000021   000057DO 000058   000084=
        000083   000084   000087   000088   000097=  000100   000101   000104
        000105
LINBUF  000000GU
LOW     000045AG 000011IF 000072IF 000078IF 000095IF
*MUL    000027CL 000028CL 000043CL 000048CL 000051CL 000055CL 000075CL 000076CL
        000090CL 000093CL
MR      000042AG
MT      000026AG
N       000021AG 000023DI 000024DI 000025AG 000026DO 000027DO 000010DO 000014DO
        000018DO 000026AG 000027AG 000028AG 000032DO 000034   000037DO 000038DO
        000042AG 000043AG 000044DO 000045DO 000048AG 000051AG 000052DO 000054DO
        000057DO 000061IF 000065AG 000068DO 000069DO 000075AG 000076AG 000077DO
        000081DO 000085DO 000090AG 000091DO 000093AG 000094DO 000095DO 000102DO
NDIM    000021AG 000023DI 000025AG 000025   000027AG 000029AG 000041   000043AG
        000048AG 000051AG 000065AG 000075AG 000076AG 000090AG 000093AG
NDIM1   000022CM 000025=  000041=
NEZ     000054=  000061=  000064IF
NONAME  000002GN
SCALE   000021AG 000024DI 000025AG 000013   000023   000073   000088   000097
TOL     000029=  000061IF 000067=
TR      000021AG 000023DI 000053=  000060IF
```

```
X       00031AG 00003DI 00026AG 00027AG 00028AG 00039=  00040=  00042AG
        00048AG 00051AG 00053   00058=  00059=  00060IF 00075AG 00076AG
        00082   00083=  00084=  00086   00087=  00088=
10      00006DO 00007DO 00009*
100     00039DO 00039*
110     00037DO 00040*
120     00045DO 00046*
130     00044DO 00047*
140     00057DO 00058*
150     00054DO 00059*
160     00060IF 00061*
170     00052DO 00060IF 00062*
180     00049DO 00064IF 00065*
190     00054IF 00066*
20      00014DO 00017*
200     00059DO 00059DO 00074*
210     00[illegible]0 00084*
220     00085DO 00088*
230     00077DO 00079GT 00089*
240     00091DO 00092*
250     00098DO 00101*
260     00102DO 00105*
270     00094DO 00096GT 00106*
30      00018DO 00021*
40      00012GT 00023*
50      00016DO 00022GT 00024*
70      00032DO 00033*
80      00035IF 00036*
90      00035IF 00037*
```

```
                C       MATRIX MULTIPLY
00001                   SUBROUTINE MMUL(NDIM,A,B,N,L,M,C)
00002                   DIMENSION A(NDIM,L),B(NDIM,M),C(NDIM,M)
00003                   DO 1 I=1,N
00004                   DO 1 J=1,M
00005         1         C(I,J)=0.
00006                   DO 2 I=1,N
00007                   DO 2 J=1,M
00008                   DO 2 K=1,L
00009         2         C(I,J)=C(I,J)+A(I,K)*B(K,J)
00010                   RETURN
00011                   END
```

```
A       00001AG 00002DI 00009
B       00001AG 00002DI 00009
C       00001AG 00002DI 00005=  00009=
I       00003DO 00005   00006DO 00009
J       00004DO 00005   00007DO 00009
K       00008DO 00009
L       00001AG 00002DI 00008DO
M       00001AG 00002DI 00004DO 00007DO
MMUL    00001GU
N       00001AG 00003DO 00005DO
NDIM    00001AG 00002DI
1       00003DO 00004DO 00005*
2       00006DO 00007DO 00008DO 00009*
```

```
00001          SUBROUTINE RENR(IIN,IOUT,TTLNEW,IS,LASTID,ARRAY,IARRAY)
               C     SUBROUTINE READS DATA FROM IIN, COMPOSES NEW TITLE
               C     AND WRITES DATA TO IOUT
00002          DIMENSION TTLNEW(5),ARRAY(IS),IARRAY(IS)
00003          DO 100 I=1,LASTID
00004          IF(I.EQ.1) GO TO 10
00006          IF(I.EQ.2) GO TO 20
00008          IF(I.EQ.54) GO TO 25
00010          IF(I.GE.3.AND.I.LE.12) GO TO 30
00012          IF(I.GE.32.AND.I.LE.34) GO TO 30
00014          IF(I.GE.13.AND.I.LE.31) GO TO 40
00016          IF(I.GE.35.AND.I.LE.49) GO TO 40
00018          IF(I.GE.58.AND.I.LE.60) GO TO 40
00020          IF(I.EQ.52.OR.I.EQ.53) GO TO 40
00022          IF(I.EQ.62.OR.I.EQ.65) GO TO 50
00024          IF(I.EQ.69.OR.I.EQ.70) GO TO 60
00026          IF(I.GE.116.AND.I.LE.123) GO TO 60
00028          IF(I.EQ.57.OR.I.EQ.68) GO TO 70
00030          IF(I.GE.71.AND.I.LE.115) GO TO 70
00032          IF(I.EQ.50.OR.I.EQ.51) GO TO 30
00034          IF(I.EQ.61) GO TO 30
00036          IF(I.EQ.63.OR.I.EQ.64) GO TO 30
00038          IF(I.EQ.55.OR.I.EQ.56) GO TO 40
00040          IF(I.EQ.66.OR.I.EQ.67) GO TO 40
00042          GO TO 100
00043    10    READ(IIN) IDEN,LEN,(ARRAY(J),J=1,LEN)
00044          NENLEN=LEN+5
00045          WRITE(IOUT) IDEN,NENLEN,(ARRAY(J),J=1,LEN),(TTLNEW(J),J=1,5)
00046          GO TO 100
00047    20    READ(IIN) IDEN,(IARRAY(J),J=1,7)
00048          WRITE(IOUT) IDEN,(IARRAY(J),J=1,7)
00049          GO TO 100
00050    25    READ(IIN) IDEN,(IARRAY(J),J=1,4)
00051          WRITE(IOUT) IDEN,(IARRAY(J),J=1,4)
00052          GO TO 100
00053    30    READ(IIN) IDEN,NX,(ARRAY(J),J=1,NX)
00054          NXT=NX
00055          GO TO 80
00056    40    READ(IIN) IDEN,NX,(ARRAY(J),J=1,2*NX)
00057          NXT=NX*2
00058    50    WRITE(IOUT) IDEN,NX,(ARRAY(J),J=1,NXT)
00059          GO TO 100
00060    60    READ(IIN) IDEN,NX,NY,(ARRAY(J),J=1,NX*NY)
00061          NTOT=NX*NY
00062          GO TO 90
00063    70    READ(IIN) IDEN,NX,NY,(ARRAY(J),J=1,2*NX*NY)
00064          NTOT=2*NX*NY
00065    90    WRITE(IOUT) IDEN,NX,NY,(ARRAY(J),J=1,NTOT)
00066   100    CONTINUE
00067          RETURN
00068          END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
ARRAY   000001AG 000002DI 000043RD 000045WR 000053RD 000056RD 000058WR 000060RD
        000063RD 000065WR
I       000030O  000041IF 000051F  000081F  000101F  000121F  000141F  000161F
        000201F  000221F  000241F  000261F  000281F  000301F  000321F  000341F
        000361F  000381F  000401F
IARRAY  000001AG 000002DI 000047RD 000049WR 000050RD 000051WR
IDEN    000043RD 000045WR 000047RD 000048WR 000050RD 000051WR 000053RD 000056RD
        000058WR 000060RD 000063RD 000065WR
IIN     000001AG 000043RD 000047RD 000050RD 000053RD 000056RD 000060RD 000063RD
IOUT    000001AG 000045WR 000048WR 000051WR 000058WR 000065WR
IS      000001AG 000002DI
J       000043RD 000045WR 000047RD 000048WR 000050RD 000051WR 000053RD 000056RD
        000058WR 000060RD 000063RD 000065WR
LASTID  000001AG 000030O
LEN     000043RD 000044   000045WR
NEWLEN  000044=  000045WR
NTOT    000051=  000054=  000065WR
NX      000053RD 000054   000056RD 000057   000058WR 000060RD 000061   000063RD
        000064   000065WR
NXT     000054=  000057=  000058WR
NY      000060RD 000061   000063RD 000064   000065WR
RESR    000016U
TTLNEW  000001AG 000002DI 000045WR
10      000056T  000043*
100     000030O  000042GT 000043GT 000049GT 000052GT 000059GT 000065*
20      000037GT 000047*
25      000040GT 000050*
30      000011GT 000013GT 000033GT 000035GT 000037GT 000053*
40      000039GT 000041GT 000056*
50      000055GT 000063*
60      000015GT 000017GT 000019GT 000021GT 000023GT 000025GT 000027GT 000060*
70      000029GT 000031GT 000062*
80      000062GT 000065*
```

```
00001          SUBROUTINE SOLVE(NDIM,N,A,B,IPVT)
00002          INTEGER NDIM,N,IPVT(N)
00003          COMPLEX A(NDIM,N),B(N),T
               C
               C  SOLUTION OF LINEAR SYSTEM A*X=C
               C  DO NOT USE OF DECOMP HAS DETECTED SINGULARITY
               C  INPUT..
               C    NDIM = DECLARED ROW DIMENSION OF ARRAY CONTAINING A
               C    N = ORDER OF MATRIX.
               C    A = TRIANGULARIZED MATRIX OBTAINED FROM DECOMP.
               C    B = RIGHT HAND SIDE VECTOR.
               C    IPVT = PIVOT VECTOR OBTAINED FROM DECOMP
               C  OUTPUT..
               C    B = SOLUTION VECTOR, X.
00004          INTEGER KB,KM1,NM1,KP1,I,K,M
               C  FORWARD ELIMINATION
00005          IF(N.EQ.1) GO TO 50
00007          NM1=N-1
00008          DO 20 K=1,NM1
00009          KP1=K+1
00010          M=IPVT(K)
00011          T=B(M)
00012          B(M)=B(K)
00013          B(K)=T
00014          DO 10 I=KP1,N
00015          B(I)=B(I)+A(I,K)*T
00016    10    CONTINUE
00017    20    CONTINUE
               C  BACK SUBSTITUTION
00018          DO 40 KB=1,NM1
00019          KM1=N-KB
00020          K=KM1+1
00021          B(K)=B(K)/A(K,K)
00022          T=-B(K)
00023          DO 30 I=1,KM1
00024          B(I)=B(I)+A(I,K)*T
00025    30    CONTINUE
00026    40    CONTINUE
00027    50    B(1)=B(1)/A(1,1)
00028          RETURN
00029          END
```

RT-11 INDEX V03 CROSS REFERENCE LISTING 10-APR-79 04:22:21 PAGE 00002

```
A       000001AG 00003CX 00015   00021   00024   00027
B       000011AG 00003CX 00011   00012=  00013=  00015=  00021=  00022
        00024=  00027=
I       00004IN 00014DO 00015   00023DO 00024
IPVT    00001AG 00002IN 00010
K       00004IN 00038DO 00009   00010   00012   00013   00015   00020=
        00021   00022   00024
KB      00004IN 00018DO 00019
K1      00004IN 00019=  00020   00023DO
KP1     00004IN 00009=  00014DO
M       00004IN 00010=  00011   00012
N       00001AG 00002IN 00003CX 00005IF 00007   00014DO 00019
NDIM    00001AG 00002IN 00003CX
R1      00004IN 00007=  00038DO 00018DO
SOLVE   00001SU
T       00003CX 00011=  00013   00015   00022=  00024
10      00014DO 00016*
20      00038DO 00017*
30      00023DO 00025*
40      00018DO 00026*
50      00005GT 00027*
```

END

DATE

FILMED

JAN 30 1981