FINAL REPORT

NASA GRANT: NSG-1414, Suppl. 2

THE DYNAMICS AND CONTROL OF LARGE FLEXIBLE SPACE STRUCTURES-III

PART A: SHAPE AND ORIENTATION CONTROL OF A PLATFORM IN ORBIT USING POINT ACTUATORS

HOWARD UNIVERSITY
SCHOOL OF ENGINEERING
DEPARTMENT OF MECHANICAL ENGINEERING
WASHINGTON, D.C. 20059

FINAL REPORT

NASA GRANT: NSG-1414, Suppl. 2

THE DYNAMICS AND CONTROL OF LARGE FLEXIBLE SPACE STRUCTURES-III

PART A: SHAPE AND ORIENTATION CONTROL OF A PLATFORM IN ORBIT USING POINT ACTUATORS

by

Peter M. Bainum
Professor of Aerospace Engineering
Principal Investigator

and

A.S.S.R. Reddy
R. Krishna
Paul K. James
Graduate Research Assistants

June 1980

ABSTRACT

The dynamics and attitude and shape control of a large thin flexible square platform in orbit are studied. Attitude and shape control is assumed to result from actuators placed perpendicular to the main surface and one edge and their effect on the rigid body and elastic modes is modelled to first order. The equations of motion are linearized about three different nominal orientations: (1) the platform following the local vertical with its major surface perpendicular to the orbital plane; (2) the platform following the local horizontal with its major surface normal to the local vertical; and (3) the platform following the local vertical with its major surface perpendicular to the orbit normal. The stability of the uncontrolled system is investigated analytically. Once controllability is established for a set of actuator locations, control law development is based on decoupling, pole placement, and linear optimal control theory. Frequencies and elastic modal shape functions are obtained using a finite element computer algorithm and two different approximate analytical methods and the results of the three methods compared.

Table of Contents

List of Tables by Chapters

List of Figures by Chapters

I. INTRODUCTION

The present grant represents a continuation of the effort attempted in the previous grant years (May 1977 - May 1979) and reported in Refs. 1 - 4*. Attitude control techniques for the pointing and stabilization of very large, inherently flexible spacecraft systems are being investigated in this research. First the attitude dynamics and control of a long, homogeneous flexible beam whose center of mass is assumed to follow a circular orbit have been treated [1,2]. In the initial phase, first-order effects of gravity-gradient were included, whereas external perturbations and related orbital station keeping maneuvers were ignored. Three mathematical models describing the system's rotations and deflections within the orbital plane have been developed--one model, which treats the beam as a number of discretized mass particles connected by massless links[1], and two continuum-type models.[2,3] The natural (uncontrolled) dynamics of this system have been simulated. The concept of distributed modal control[1], which provides a means for controlling a particular system mode independently of all other modes, has been examined, along with other types of control laws including an application of optimal control theory and the use of decoupling techniques.[3] The effect of varying the number of modes in our model as well as the number and location of control devices has been examined, analytically, where possible, and numerically for general cases.[3]

---

*For references cited in this report please see list of references ater each chapter.

Towards the end of the second grant year the three dimensional model of a free-free plate in orbit was developed and a limited number of computer simulations of the uncontrolled dynamics in response to initial perturbations about a specific equilibrium orientation were performed.[4] Frequency values associated with the basic structural modes of a square plate were obtained from energy considerations based on approximate expressions developed by Warburton.[5] It was suggested at the final oral grant presentation that a comparison with results obtained using finite element methods and/or other analytical approaches should be examined to guarantee accuracy, particularly for higher order modes.

With this background and in accordance with our proposal to NASA dated January 25, 1979[6], a plan of study was developed and has been extended to include the current grant year as outlined in Table I. The items indicated by a check mark have been completed by the end of the third grant year while those indicated by "IP" are currently in progress.

In this part of the 1979-80 final report (Part A) the control of an orbiting square shaped platform based on the continuum model of Ref. 2 with point actuators taken at selected locations on the platform surfaces is examined . A paper to be presented at the following conference forms the basis of Chapter II:

1980 AIAA/AAS Astrodynamics Conference, Danvers, Mass., Aug. 11-13, 1980 (only the contributions by A.S.S.R. Reddy, P.M. Bainum, and R. Krishna are included here).

In Chapter III the results of two approximate analytical methods for predicting modal frequencies and modal shape functions are compared with the results obtained using a finite element computer algorithm using the homogeneous plate as an example.

TABLE I - STUDY PLAN 1977-1980

1. MODEL DEVELOPMENT

✓ A. Development of General Form of 3-Dimensional Equations for A Flexible Structure - Given the Modal Shape Functions

✓ B. Development of 3-Dimensional Equations of a Thin Homogenous Free-Free Beam

✓ (1). The Case of No Longitudinal Vibrations: i.e. $\phi_x^{(n)} = 0$

✓ (2). The Case of No Yaw: i.e. $\Psi = 0$

C. Determination of Modal Shape Functions and Frequencies for Different Structural Models

✓ (1). Circular Homogenous Membrane

✓ (2). Rectangular Homogenous Membrane

✓ (3). Rectangular Homogenous Plate (and Square Plate)

✓ (4). Circular Homogenous Plate

✓ (5). Shallow Spherical Shell Structure

D. Implementation of One or More of the Structural Models for Digital Simulation

✓ (1). Rectangular Homogenous Plate

✓ (2). Thin-Homogenous Beam with Stabilizing Dumbbell (Local Horizontal Orientation)

(3). Square Plate with Stabilizing Dumbbell (Local Horizontal Orientation)

✓ (4). Shallow Spherical Shell Structure with Stabilizing Dumbbell

✓ (5). Circular Homogenous Plate with Stabilizing Dumbbell

IP E. Provide Equations in a Form Suitable for Control Implementation

✓ - Items completed

IP - Items in progress

2. CONTROL CONCEPTS - LARGE FLEXIBLE SPACE STRUCTURES

A. Model Development

✓ (1). Concentrated on continuum model of large flexible beam in orbit (Santini and Howard University Formulation)

✓ (2). Modelled control devices as point actuators at specific locations along the beam

(3). Modelling of control devices as point or distributed actuators for other large flexible systems

✓ (a) Rectangular Homogenous Plate

(b) Circular Homogenous Plate

(c) Shallow Spherical Shell Structure

B. Control Concepts:

✓ (1). Modal Control - considered with discretized beam model during 1977-78

For independent control of all modes (N) retained in the model, the number of actuators (P) must be equal to N(P=N)

✓ (2). Establish relationship between P and N according to controllability requirements (applications of theorems developed by Balas) P can be less than N. (Applied to continuum beam model 1978-79).

✓ (3). Selection of control system gains - considers both position and rate feedback. (Applied to continuum beam model 1978-79)

✓ a. Develop criteria for complete decoupling of linearized controlled equations using the fundamental theorem of a system of N linear equations and P unknowns

For unique solution of gains, P=N consistent with modal control; for non unique solution P<N

✓ b. Application of linear regulator problem to the original linearized and/or transformed equations

IP (4) Application of control concepts to more complex structures

IP C. Modelling of Sensors-the Problem of Observability

D. Treatment of Observation and Control Spillover

✓ Items completed

IP Items in progress

References are given separately for each chapter; symbols used in Chapter II are defined either in the text or in Appendix A of Chapter II, while symbols used in Chapter III are defined in the text where used.

Chapter IV describes general conclusions together with recommendations for future work.

Part B of this report, under separate cover, concentrates on the mathematical modelling and analysis of more complex structures such as beams and plates with connected gimballed dumbbells to provide gravitational stability about the local horizontal orientation, and also the analysis of the dynamics of a shallow shell-type structure in orbit.

I.1 References - Introduction

1. Bainum P.M. and Sellappan, R., "The Dynamics and Control of Large Flexible Space Structures," Final Report NASA Grant: NSG-1414, Part A: Discrete Model and Modal Control, Howard University, May 1978.

2. Bainum Peter M., Kumar, V.K., and James, Paul K., "The Dynamics and Control of Large Flexible Space Structures," Final Report, NASA Grant: NSG-1414. Part B: Development of Continuum Model and Computer Simulation, Howard University, May 1978.

3. Bainum, P.M. and Reddy, A.S.S.R., "The Dynamics and Control of Large Space Structures II," Final Report, NASA Grant NSG-1414, Suppl. 1, Part A: Shape and Orientation Control Using Point Actuators, Howard University, June 1979.

4. Bainum, P.M., James P.K., Krishna, R., and Kumar, V.K., "The Dynamics and Control of Large Flexible Space Structures II" Final Report, NASA Grant NSG-1414, Suppl. 1., Part B: Model Development and Computer Simulation, Howard University, June 1979.

5. Warburton, G.B., "The Vibration of Rectangular Plates," Proc. Institute of Mechanical Engineers, Vol. 168, No. 12, 1954, pp. 371-394.

6. Bainum, P.M., "Proposal for Research Grant on: The Dynamics and Control of Large Flexible Space Structures-III," Howard University, (Submitted to NASA), Jan. 25, 1979.

II. CONTROL OF A LARGE FLEXIBLE PLATE IN ORBIT

Abstract

The dynamics and attitude and shape control of a large thin flexible platform in orbit are studied. Attitude and shape control is assumed to result from actuators placed perpendicular to the main surface and one edge and their effect on the rigid body and elastic modes is modelled to first order. The equations of motion are linearized about nominal orientations where the undeformed plate follows either the local vertical or local horizontal. The stability of the uncontrolled system is investigated analytically. Once controllability is established for a set of actuator locations, control law development is based on pole placement, decoupling, and linear optimal control theory.

1. Introduction

Large, flexible spacecraft systems have been proposed for future applications in widespread communications, electronic orbitally based mail systems, and as possible collectors of solar energy for transmittal to earth-based receiving stations.[1,2] For such missions the size of the orbiting system may be several times larger than that of the earth-based receiving station(s), and both orientation and shape control of the orbiting system will be required.

In order to gain insight into the dynamics of such a large flexible system the equations of motion of a long, flexible free-free beam in orbit were developed[3] using a slightly modified version of the general formulation of the dynamics of a general flexible orbiting body formulated by Santini.[4] This specific example considered only the inplane rotations and deformations of the uncontrolled beam and demonstrated the possibility of instability for very small values of the ratio of the fundamental flexural frequency to the orbit angular velocity. Two related papers treated the modelling of point actuators located at specific points along the beam with the associated criteria for controllability[5] and also the problem of selecting control law feedback gains based on decoupling techniques and application of the linear regulator problem.[6] Also included were numerical results showing the effects of control spillover on the uncontrolled modes when the number of controllers is less than the number of modes in the model, and the effects of inaccurate knowledge of the control influence coefficients which lead to errors in the calculated feedback gains.[6]

In the present paper the two dimensional model considered in Refs. 3,5, and 6 is extended to three dimensions by developing the equations of motion for a large flexible rectangular plate (platform) in orbit. These equations include three rigid body equations plus the generic mode elastic equations.

2. Model Development

In the present paper three different nominal orientations of the platform in orbit are assumed about which attitude and shape control are to be achieved. These are:

- Case (i) the platform following the local vertical with its larger surface perpendicular to the plane of the orbit (Fig. 1a);
- Case (ii) the platform following the local horizontal with its larger surface area normal to the local vertical (Fig. 1b);
- Case (iii) the platform following the local vertical with its larger surface perpendicular to the orbit normal (Fig. 1c).

From the general formulation of Refs. 3 and 4, the equations of motion of the structure are obtained:

A. Rotational Equations of Motion:

$$\dot{\omega}_x = \frac{I_y - I_z}{I_x}\omega_y\omega_z + \frac{G_{R_x}}{I_x} + \frac{T_x}{I_x} + \frac{C_x}{I_x}$$

$$\dot{\omega}_y = \frac{I_z - I_x}{I_z}\omega_x\omega_z + \frac{G_{R_y}}{I_y} + \frac{T_y}{I_y} + \frac{C_y}{I_y} \qquad (1)$$

$$\dot{\omega}_z = \frac{I_x - I_y}{I_z}\omega_x\omega_y + \frac{G_{R_z}}{I_z} + \frac{T_z}{I_z} + \frac{C_z}{I_z}$$

Using Euler angles to represent rigid body orientations relative to the local vertical (horizontal) system, the transformation from Euler angular rates to body rates is given by:

$$\omega_x = \dot{\psi} + (\dot{\theta}+\omega_c)\sin\phi$$

$$\omega_y = (\dot{\theta}+\omega_c)\cos\phi\cos\psi + \dot{\phi}\sin\psi \qquad (2)$$

$$\omega_z = \dot{\phi}\cos\psi - (\dot{\theta}+\omega_c)\sin\psi\cos\phi$$

(Note: Symbols used are defined in Appendix A.)

G_{R_x}, G_{R_y}, G_{R_z} represent the gravity-gradient torques about the principal undeformed body axes and can be evaluated as:

$$
\begin{aligned}
G_{R_x} &= 3\omega_c^2(I_z-I_y)(-c\psi s\phi c\theta+s\psi c\theta)(s\psi s\phi c\theta+c\psi s\theta)\\
G_{R_y} &= 3\omega_c^2(I_x-I_z)(s\psi s\phi c\theta+c\psi s\theta)c\theta c\phi\\
G_{R_z} &= 3\omega_c^2(I_y-I_x)(-c\psi s\phi c\theta+s\psi s\theta)c\theta c\phi
\end{aligned}
\qquad (3)
$$

where $s(\ ) = \sin(\ )$ and $c(\ ) = \cos(\ )$.

B. Generic Mode Equations

The generic modal equations may be obtained for each of the three nominal orientations considered in terms of the modal amplitude (A_r):[3,4]

For Case (i)

$$\ddot{A}_r + [\omega_r^2-(\omega_x^2+\omega_y^2)-M_{zz}]\,A_r = E_r/M_r \qquad (4a)$$

where

$$M_{zz} = \omega_c^2\,[3(s^2\psi s^2\phi c^2\theta+c^2\psi s^2\theta+2s\psi s\phi c\theta c\psi s\theta)-1]$$

For Case (ii)

$$\ddot{A}_r + [\omega_r^2-(\omega_y^2+\omega_z^2)-M_{xx}]\,A_r = E_r/M_r \qquad (4b)$$

where

$$M_{xx} = \omega_c^2\,[3c^2\theta c^2\phi-1]$$

For Case (iii)

$$\ddot{A}_r + [\omega_r^2-(\omega_x^2+\omega_z^2)-M_{yy}]\,A_r = \frac{E_r}{M_r} \qquad (4c)$$

where

$$M_{yy} = \omega_c^2\,[3(c^2\psi s^2\phi c^2\theta+s^2\psi s^2\theta-2c\psi s\phi s\psi s\theta)-1]$$

C. Linearization

With the assumption of small amplitudes, the rotational equations of motion given by Eq. (1) become:

$$
\begin{aligned}
\ddot{\psi} &= \omega_c\dot{\phi}\,\left[\frac{I_y-I_z}{I_x}-1\right]-\omega_c^2\,\left(\frac{I_y-I_z}{I_x}\right)\psi+\frac{T_x}{I_x}+\frac{C_x}{I_x}\\
\ddot{\phi} &= \omega_c\dot{\psi}\,\left[\frac{I_x-I_y}{I_z}+1\right]+4\omega_c^2\,\left(\frac{I_x-I_y}{I_z}\right)\phi+\frac{T_z}{I_z}+\frac{C_z}{I_z}\\
\ddot{\theta} &= 3\omega_c^2\,\left(\frac{I_x-I_z}{I_y}\right)\theta+\frac{T_y}{I_y}+\frac{C_y}{I_y}
\end{aligned}
\qquad (5)
$$

For the present analysis, the platform is assumed to be square, thin and homogeneous, such that the following relationships among the principal moments of inertia are valid:

$$
\begin{aligned}
&\text{Case (i):} && I_x = I_y \text{ and } I_z = 2I_x = 2I_y\\
&\text{Case (ii):} && I_y = I_z \text{ and } I_x = 2I_y = 2I_z\\
&\text{Case (iii):} && I_x = I_z \text{ and } I_y = 2I_x = 2I_z
\end{aligned}
\qquad (6)
$$

For small amplitude angles the generic mode equations become:

$$
\begin{aligned}
&\text{Case (i):} && \ddot{A}_r+\omega_r^2A_r = E_r/M_r\\
&\text{Case (ii):} && \ddot{A}_r+(\omega_r^2-3\omega_c^2)A_r = E_r/M_r\\
&\text{Case (iii):} && \ddot{A}_r+(\omega_r^2-\omega_c^2)A_r = E_r/M_r
\end{aligned}
\qquad (7)
$$

D. Modelling of Point Actuators

For an actuator which can generate a force of the type

$$\bar{f} = f_x\hat{i}+f_y\hat{j}+f_z\hat{k} \qquad (8)$$

and placed at a location (x,y,z), the resultant control torque is given by

$$\bar{T} = \bar{R}\times\bar{f} \qquad (9)$$

where $\bar{R} = x\hat{i}+y\hat{j}+z\hat{k}$ describes the position of the actuator on the surface (or edge) of the plate. Actuators can be placed perpendicular to the XY, YZ or XZ planes of the plate, so for an actuator whose force axis is perpendicular to the XY plane the torque is given by (since $f_x = f_y = 0$)

$$\bar{T} = yf_z\hat{i}-xf_z\hat{j} \qquad (10)$$

For an actuator whose force axis is perpendicular to the YZ plane, the torque is given by (since $f_y = f_z = 0$)

$$\bar{T} = zf_x\hat{j}-yf_x\hat{k} \qquad (11)$$

For an actuator perpendicular to the ZX plane, the torque is given by (since $f_z = f_x = 0$)

$$\bar{T} = -zf_y\hat{i}+xf_y\hat{k} \qquad (12)$$

The generic force due to one i^{th} actuator on the r^{th} mode is given by[3,4]

$$
\begin{aligned}
E_r &= \iint W_r(x,y)\hat{k}\cdot\delta(x-x_i,y-y_i)f_i(t)\hat{k}\,dx\,dy\\
&= W_r(x_i,y_i)f_i(t)
\end{aligned}
\qquad (13)
$$

where $W_r(x,y)$ is the r^{th} modal (spatial) function of the deformed plate with vibrations assumed to occur along the Z direction, whose amplitudes are assumed to be much smaller than a characteristic plate length.

For n actuators placed on the XY plane of the plate with force axes normal to that deformed surface, the generic force on r^{th} mode is given by

$$E_r = \sum_{i=1}^{n} W_r(x_i,y_i)f_i \qquad (14)$$

where x_i, y_i are the coordinates of the i^{th} actuator. An actuator placed normal to the X,Y plane won't produce a torque about the Z-axis; in order to obtain a direct torque about the Z-axis, actuators may have to be located on the other surfaces (edges) of the plate.

E. Modelling of Distributed Actuators

If the force is distributed along the surfaces of the plate, the force can be represented by

$$
\begin{aligned}
\bar{f} &= f_x(x,y,z,t)\,\hat{i}+f_y(x,y,z,t)\hat{j}\\
&\quad+f_z(x,y,z,t)\hat{k}
\end{aligned}
\qquad (15)
$$

where the force components are now both spatially and time dependent.

The torque due to such an actuator is given by

$$\bar{T} = \bar{R}\times\bar{f} \tag{16}$$

The total torque is given by

$$\bar{T} = \int_{\forall}(\bar{R}\times\bar{f})\,dxdydz \tag{17}$$

Using series expansions and separation of variables between spatially and time dependent functions, one can very accurately represent (e.g. for the x component),

$$f_x(x,y,z,t) = \sum_{R=1}^{L} f_{x_R}(x,y,z)g_{x_R}(t) \tag{18}$$

The integral for the torque is then given by

$$T = \int[(\sum_{i=1}^{N} yf_{z_i}(x,y,z)g_{z_i}(t)-\sum_{m=1}^{M} f_{y_m}(x,y,z)g_{y_m}(t))\hat{i} + (\sum_{R=1}^{L} zf_{x_R}(x,y,z)g_{x_R}(t)-\sum_{i=1}^{N} xf_{z_i}(x,y,z)g_{z_i}(t))\hat{j} + (\sum_{m=1}^{N} xf_{y_m}(x,y,z)g_{y_m}(t)-\sum_{R=1}^{L} f_{x_R}(x,y,z)g_{x_R}(t))\hat{k}]\times dxdydz \tag{19}$$

The resulting generic force is then obtained in the same manner as in Eq. (13) with the result ,

$$E_r = \int[W_r(x,y)][\sum_{i=1}^{I} f_{z_i}(x,y,z)g_{z_i}(t)]dxdydz \tag{20}$$

3. Uncontrolled Motion-Numerical Example

The platform is assumed to have the following physical properties:

$a = 100$ m (side of square plate)

$M = 276800$ kg

Minimum Moment of Inertia = 2.354×10^{7} kg-m^2

Maximum Moment of Inertia = 4.7088×10^{7} kg-m^2

For an assumed orbital altitude of 250 n.mi. (circular)

$\omega_c = 1.25\times10^{-3}$ rad/sec.

The modal frequencies of the elastic modes have been obtained using a finite element computer algorithm.[7] For the first three flexible modes:

$\omega_1 = 2.0931947\times10^{-2}$ rad/sec

$\omega_2 = 3.040474\times10^{-2}$ rad/sec

$\omega_3 = 3.9088122\times10^{-2}$ rad/sec

The uncontrolled motion of the linear system through small amplitude deviations with respect to each of the three nominal orientations will now be considered.

Case(i): $I_x=I_y$, $I_z = 2I_x = 2I_y$

The rotational equations of motion and the generic modal equations are non-dimensionalized by the orbital period and the length variable ($\tau = \omega_c t$, $Z_r = A_r/a$, $\phi' = d\phi/d\tau$, etc)

$$\psi'' = [(I_y-I_z-I_x)/\omega_c I_x]\phi'-[(I_y-I_z)/I_x]\psi \tag{21}$$

$$\phi'' = [(I_x-I_y+I_z)/\omega_c I_z]\psi'+4[(I_x-I_y)/I_z]\phi \tag{22}$$

$$\theta'' = 3\theta \tag{23}$$

The generic mode equations become:

$$Z_r'' = -(\omega_r/\omega_c)^2 Z_r \tag{24}$$

The pitch and the generic mode equations are decoupled from roll and yaw. The pitch and generic modes exhibit simple harmonic motions. After substituting inertia values into the roll and yaw equations,

$$\psi'' = -(2/\omega_c)\phi'+\psi \tag{25}$$

$$\phi'' = \psi'/\omega_c \tag{26}$$

The characteristic equation for the system (25) and (26) is, $s^2(s^2-1+2/\omega_c^2) = 0$

It can be seen that the roll and yaw motion has a double pole at the origin and thus the uncontrolled roll/yaw motion is unstable. The analytical solution is obtained using Laplace transform techniques. A typical response for initial perturbations in both roll and yaw rate(s) is shown in Fig. 2.

Case (ii): $I_y = I_z$ and $I_x = 2I_y$

The rotational equations of motion are

$$\psi'' = -(1/\omega_c)\phi' \tag{27}$$

$$\phi'' = (2/\omega_c)\psi'+4\phi \tag{28}$$

$$\theta'' = 3\theta \tag{29}$$

The generic mode equations can be represented by,

$$Z_r'' = -[(\omega_r/\omega_c)^2-3]Z_r \tag{30}$$

From Eq. (29) the pitch amplitude increases exponentially in response to an initial displacement, whereas from Eq. (30), for $\omega_r/\omega_c > \sqrt{3}$ the generic modal amplitudes exhibit simple harmonic motion.

The characteristic equation for the combined roll/yaw motion is:

$$s^2(s^2-4+2/\omega_c^2) = 0 \tag{31}$$

The roll/yaw motion is characterized by a double pole at the origin and is thus unstable.

Case (iii): $I_x = I_z$ and $I_y = 2I_x = 2I_z$

The rotational equations of motion are

$$\psi'' = -\psi;\ \phi'' = -4\phi;\ \theta'' = 0 \tag{32}$$

while the generic mode equations can be expressed by,

$$Z_r'' = -[(\omega_r/\omega_c)^2-1]Z_r \tag{33}$$

In this case, roll, yaw, pitch and the generic modes are decoupled from each other. The generic modes, roll and yaw exhibit simple harmonic motion, while the pitch amplitude increases linearly with time for a given initial pitch rate.

4. Controlled Motion

The rotational equations of motion are combined with the generic modal equations using the nondimensional orbital time and length variables and then recast into conventional state space form :

$$X' = AX + BU \tag{34}$$

where the state vector, X, is defined as

$$X = (x_1, x_2, x_3, \ldots x_{i+3}, \ldots, x_{n+6+1}, \ldots, x_{2n+6})^T$$

and

$$x_1 = \phi;\ x_2 = \psi;\ x_3 = \theta;\ x_{i+3} = z_i = A_i/a;$$

$i = 1, 2, \ldots, n$ generic modes

$$x_{n+4} = \phi';\ x_{n+5} = \psi';\ x_{n+6} = \theta'$$

$$x_{n+6+i} = z_i' = x_{i+3}',\ i = 1,2,\ldots n$$

For the examples to be considered in this paper it is assumed that the system can be modelled by three rigid body rotational modes and the first three generic (flexible) modes.

The general A matrix

$$[A] = \left[\begin{array}{c|c} \underset{6\times6}{O} & \underset{6\times6}{[I]} \\ \hline \text{diagonal} & \begin{array}{cccc} 0 & A_{78} & \cdots & 0 \\ A_{78} & 0 & \cdots & 0 \\ & & & \\ 0 & 0 & \cdots & 0 \end{array} \end{array}\right] \tag{35}$$

$A_{6\times6}$

The non-zero and non-unity elements appearing in A are;

$$A_{7,1} = 4(I_x - I_y)/I_z\ ;\ A_{8,2} = -(I_y - I_z)/I_x\ ;$$

$$A_{9,3} = 3(I_x - I_z)/I_y\ ;\ A_{10,4} = -(\omega_1/\omega_c)^2;$$

$$A_{11,5} = (\omega_2/\omega_c)^2\ ;\ A_{12,6} = -(\omega_3/\omega_c)^2;$$

$$A_{8,7} = (I_y - I_z - I_x)/\omega_c I_x;\ A_{7,8} = (I_x - I_y + I_z)/\omega_c I_z$$

The general B matrix:

$$B = \left[\frac{O_{6\times6}}{B_{6\times6}}\right]$$

where the lower part of the B matrix depends on actuator locations.

Control Law Selection

Control laws are developed using 3 different techniques. They are: (a) decoupling of the original state equations using state variable feedback; (b) stabilizing the system by clustering the poles on a line parallel to the imaginary axis and in the negative s-plane using the control law of the type $U = -KX$; (c) applying the linear regular theory to the original system equations.

(a) Decoupling of Original State Equations Using State Variable Feedback

The equations of motion of the platform can be written as

$$\ddot{X} = A\dot{X} + CX + BU \tag{36}$$

where $X = (x_1, x_2 \ldots x_{n+3})$

After selecting $U = K_r\dot{X} + K_p X$ we can rewrite the controlled motion equations as

$$\ddot{X} = (A + BK_r)\dot{X} + (C + BK_p)X \tag{37}$$

K_r and K_p are evaluated such that $(A+BK_r)$ and $(C+BK_p)$ are diagonalized and thus yield required damping and frequency of the controlled modes. The number of modes must be equal to the number of actuators to avoid the use of pseudo-inverse matrices.

Two sets of actuator locations have been assumed for each of the three nominal orientations previously described. For all orientations, (i)-(iii), it is assumed that five actuators are located on the larger surface (with force axis normal to it) and a sixth actuator along an edge. The body coordinates of the six actuators are taken as

Case (i)

First Location $a = 100m$

$f_1(-a/6, -a/6, 0)$; $f_2(a/6, -a/6, 0)$; $f_3(-a/6, 0, 0)$

$f_4(a/6, 0, 0)$; $f_5(-a/6, a/6, 0)$; $f_6(a/2, a/6, 0)$

Second Location $a = 100m$

$f_1(-a/2, -a/2, 0)$; $f_2(a/2, -a/2, 0)$; $f_3(-a/2, a/2, 0)$

$f_4(a/2, a/2, 0)$; $f_5(-a/2, 0, 0,)$; $f_6(a/2, a/2, 0)$

Case (ii)

First Location $a = 100m$

$f_1(0, -a/6, -a/6)$; $f_2(0, -a/6, a/6)$; $f_3(0, 0, -a/6)$

$f_4(0, 0, a/6)$; $f_5(0, a/6, -a/6)$; $f_6(0, a/6, a/2)$

Second Location $a = 100m$

$f_1(0, -a/2, -a/2)$; $f_2(0, -a/2, a/2)$; $f_3(0, a/2, -a/2)$

$f_4(0, a/2, a/2)$; $f_5(0, 0, -a/2)$; $f_6(0, a/2, a/2)$

Case (iii)

First Location $a = 100m$

$f_1(-a/6, 0, -a/6$; $f_2(-a/6, 0, a/6)$, $f_3(0, 0, -a/6)$

$f_4(0, 0, a/6)$; $f_5(a/6, 0, -a/6)$; $f_6(a/6, 0, a/2)$

Second Location $a = 100m$

$f_1(-a/2, 0, -a/2)$; $f_2(-a/2, 0, a/2)$; $f_3(a/2, 0, -a/2)$

$f_4(a/2, 0, a/2)$; $f_5(0, 0, -a/2)$; $f_6(a/2, 0, a/2)$

Actuator positions for the two different sets of locations are illustrated in Fig. 3. The system A and B matrices corresponding to different combinations of the three platform orientations and the two sets of actuator location are listedas follows.

Case (i) Platform Following Local Vertical With Major Surface Normal to the Orbit Plane.

The non zero elements of the A matrix are:

$A_{i,i+6}=1$ for $i = 1,\ldots,6$; $A_{7,8} = 800$;

$A_{8,7} = -1600$; $A_{8,2} = 1$; $A_{9,3} = -3$;

$A_{10,4} = -277.414$, $A_{11,5} = -588.647$;

$A_{12,6} = -976.844$

Location I (Lower Part of B matrix)

0.0	0.0	0.0	0.0	0.0	-0.22645
-0.4529	-0.4529	0.0	0.0	0.4529	0.0
0.4529	-0.4529	0.4529	-0.4529	0.4529	0.0
+0.003126	-0.003126	0.0	0.0	-0.003126	0.0
0.0	0.0	-0.003084	-0.003084	0.0	0.0
-0.008786	-0.008786	-0.0115	-0.0115	-0.008786	0.0

Location II (Lower Part of B matrix)

0	0	0	0	0	-0.6796
-1.3592	-1.3592	1.3592	1.3592	0	0.0
1.3592	-1.3592	1.3592	-1.3592	1.3592	0.0
0.023	-0.023	-0.023	0.023	0.0	0.0
0.0	0.0	0.0	0.0	-0.023	0.0
0.023	0.023	0.023	0.023	0.023	0.0

Case (ii) Platform Along Local Horizontal

The elements of the A matrix that are different from Case (ii) are:

$A_{7,1} = 4$; $A_{7,8} = 1600$; $A_{8,2} = 0$; $A_{8,7} = -800$;

$A_{9,3} = 3$.

Location I (Lower Part of B matrix)

0.4592	0.4529	0.0	0.0	-0.4529	0.0
0.0	0.0	0.0	0.0	0.0	0.22645
-0.4529	0.4529	-0.4529	0.4529	-0.4529	0.0
0.003126	-0.003126	0.0	0.0	-0.003126	0.0
0.0	0.0	-0.003084	-0.003084	0.0	0.0
-0.008786	-0.008786	-0.0115	-0.0115	-0.008786	0.0

Location II (Lower Part of B matrix)

1.3592	1.3592	-1.3592	-1.3592	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.6796
-1.3592	1.3592	-1.3592	1.3592	-1.3592	0.0
0.023	-0.023	-0.023	0.023	0.0	0.0
0	0.0	0.0	0.0	-0.023	0.0
0.023	0.023	0.023	0.023	0.023	0.0

Case (iii) Major Surface in Orbit Plane

The elements of the A matrix that are different from Case (ii) are:

$A_{7,1} = -4$; $A_{7,8} = 0$; $A_{8,2} = -1$; $A_{8,7} = 0$;

$A_{9,3} = 0$.

Location I (Lower Part of B matrix)

-0.4529	-0.4529	0.0	0.0	0.4529	0.0
0.4529	-0.4529	-0.4529	-0.4529	0.4529	0.0
0.0	0.0	0.0	0.0	0.0	-0.22645
0.003126	-0.003126	0.0	0.0	-0.003126	0.0
0.0	0.0	-0.0030844	-0.0030844	0.0	0.0
-0.008786	-0.008786	-0.0115	-0.0115	-0.008786	0.0

Location II (Lower Part of B matrix)

-1.3592	-1.3592	1.3592	1.3592	0.0	0.0
1.3592	-1.3592	1.3592	-1.3592	1.3592	0.0
0.0	0.0	0.0	0.0	0.0	-0.6796
0.023	-0.023	-0.023	0.023	0.0	0.0
0.0	0.0	0.0	0.0	-0.023	0.0
0.023	0.023	0.023	0.023	0.023	0.0

For all combinations considered above the gains are selected so as to produce 20% of critical damping in each of the rigid body modes and the first generic mode, and 10% of critical damping in the second and third generic modes. In order to provide a better transient response in the lower frequency fundamental elastic mode, the percentage of critical damping is selected to be twice that in the remaining flexible modes. The time response of the rigid body modes and the generic modal amplitudes for all combinations considered and for equal initial position displacements in all components of the state is illustrated in Fig. 4a.

As an example of the time history of the required control forces, Fig. 4b. shows such a time response for the exterior (II nd) location of the actuators with the platform nominally following the local vertical and the major surface area of the platform in the orbital plane. A complete summary of the maximum force amplitudes required for all combinations of actuator locations and platform orientations is given in Table I. In interpreting the results of Table I, it should be pointed out that, in the process of achieving both orientation and shape control, the maximum force(s) required of any actuator will vary with both the moment arm about the principal body axes and the value of the modal shape function at the particular actuator location for all modes contained within the mathematical model.

(b) Stabilizing the System by Pole Clustering

The equations of motion of the platform when recast in state space format can be written as

$\dot{X} = AX+BU$ (38), $X \rightarrow 2(n+3)x1$

Table 1 Maximum Force Amplitudes (Newtons) for Different Combinations of Cases with Actuator Locations

Force	Location I (interior) Case i	Case ii	Case iii	Location II (exterior) Case i	Case ii	Case iii
f_1	-807.8	270.63	270.6	57.6	62.56	47.4
f_2	270.0	136.2	-133.3	28.19	26.20	-17.2
f_3	1141.0	387.3	-425.0	-65.89	-49.5	30.0
f_4	527.9	195.6	163.7	17.10	-30.90	25.08
f_5	489.09	-253.3	153.0	-44.6	-44.67	-44.67
f_6	-264.58	-272.0	4.4	88.16	-88.06	1.47

——— Indicates maximum level of control force required for each case.

The control, $U = -KX$ is selected by using a digital computer algorithm[5] such that $(A-BK)$ has the required identical negative real part in each of its eigenvalues. Although the number of actuators can be less than the number of modes (one half of the dimensionality of the state vector), a limitation of this algorithm is that the gains are selected such that all of the closed-loop poles lie on a line parallel to the imaginary axis. However this algorithm is useful when it is important that each mode in the system satisfy some minimum damping characteristics.

As an example of this technique we consider the system with four actuators and six modes where control about the first orientation (Case i) is desired. Three of the actuators are assumed to provide forces perpendicular to the major surface with the remaining actuator thrusting normal to an edge. The actuator coordinates in the body system (Fig. 1a) are: $f_1(-a/6, -a/6, 0)$; $f_2(a/6, -a/6, 0)$; $f_3(-a/6, 0, 0)$; and $f_4(a/2, a/6, 0)$ where $a = 100m$. It is assumed that the minimum damping requirement on the system has a time constant of (13.33 min or $(1/2\pi)$ dimensional orbital time). The control influence matrix is then calculated based on the assumed coordinates of the four actuators. The control $U = -KX$ can be calculated by the ORACLS pole clustering algorithm. Based on these gains time histories of the required control forces are then obtained.

The control influence matrix (lower part), closed loop poles, and maximum force amplitudes required are summarized as follows:

B matrix (Lower Part)

$$\begin{bmatrix} 0.0 & 0.0 & 0.0 & -0.22645 \\ -0.4529 & -0.4529 & 0.0 & 0.0 \\ 0.4529 & -0.4529 & 0.0 & 0.0 \\ 0.003126 & -0.003126 & 0.0 & 0.0 \\ 0.0 & 0.0 & -0.0030844 & 0.0 \\ -0.008786 & -0.008786 & -0.0115 & 0.0 \end{bmatrix}$$

Closed Loop Poles (Nondimensionalised)

The real part is -1.0 and the imaginary parts are ± 0.000485, ± 0.993, ± 16.82, ± 24.26, ± 31.33 and ± 1131.37. The Maximum force amplitudes (Newtons) are calculated as

$|f_1| = 78.5$, $|f_2| = 36.4$, $|f_3| = 169.5$, and

$|f_4| = 35.3$.

An interesting comparison can now be made between this result and that shown in Table I for case (i) and the first (I) location of the six actuators considered there. It can be seen that by using fewer actuators, appropriately placed, that better transient response characteristics can be obtained with smaller maximum force amplitudes. However a disadvantage of this method is that some of the controlled frequencies may be orders of magnitude greater than the highest frequency of the uncontrolled system (for this example compare 1131 with $\sqrt{976} = 31.24$). Depending on the nature of the expected disturbance forces this result could be very undesirable.

(c) Application of the Linear Regulator Theory

The control law, $U = -KX$, is selected such that the following performance index is minimized

$$J = \int_0^\infty (X^T Q X + U^T R U) dt \qquad (39)$$

where Q and R are positive definite penalty matrices. The steady state solution of the matrix Riccati equation of dimension equal to the state has to be solved in order to evaluate the gain matrix, K.

A computer algorithm within the ORACLS[6] software package is used to obtain the gain matrices K for different combinations of the Q and R penalty matrices. This algorithm utilizes the Newton Raphson method of solving the Ricatti equation. In the examples considered here four actuators are assumed with the system represented by three rigid body and three flexible modes. The locations of the four actuators are taken to be the same as in Section (b), and control about the first nominal orientation (i) is considered.

The weighting matrix, Q, is selected based on the following considerations. For the example considered here it can be seen from Eq. (34), (35), and the B matrix that the uncontrolled system dynamics is either described by sets of uncoupled harmonic oscillators, or (in the case of roll/yaw motion) by a coupled two dimensional harmonic oscillator. The latter motion can be represented by

$$\begin{bmatrix} \omega_z' \\ \omega_x' \end{bmatrix} = \begin{bmatrix} 0 & a \\ -b & 0 \end{bmatrix} \begin{bmatrix} \omega_z \\ \omega_x \end{bmatrix} \qquad (40)$$

where the system oscillates at the frequency $\Omega = \sqrt{ab}$. It is desired that the control remove a maximum "transverse" angular rate, $\omega_{T_{max}} = \max\sqrt{\omega_z^2 + \omega_x^2} = \sqrt{\omega_z^2(0) + \omega_x^2(0)}$

so that a strategy for selecting the elements of Q could be[9,10]

$$Q = \begin{bmatrix} f & 0 \\ 0 & f \end{bmatrix} \quad \text{where } f = \Omega^2/\omega^2_{T_{max}}, \tag{41}$$

when the control penalty matrix is fixed. The remaining equations for any of the uncoupled oscillators can also be expressed by

$$\begin{bmatrix} z_i' \\ z_i'' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -(\omega_i/\omega_c)^2 & 0 \end{bmatrix} \begin{bmatrix} z_i \\ z_i' \end{bmatrix} \tag{42}$$

in the same format as Eq. (39), and thus the weights can be obtained in a similar manner.

The Q matrix for the case considered here (control about nominal orientation (i) with actuator locations as given in Section (b)) is obtained using the relations given by Eq. (41) and is a diagonal matrix, Q_o, with the following elements:

$$Q_{1,1} = 4.324\text{x}10^9,\ Q_{2,2} = 8.539\text{x}10^9,$$

$$Q_{3,3} = Q_{9,9} = 3.0\text{x}10^4,\ Q_{4,4} = Q_{10,10} = 2.77414\text{x}10^6,\ Q_{5,5} = Q_{11,11} = 5.88647\text{x}10^6,$$

$$Q_{6,6} = Q_{12,12} = 9.74844\text{x}10^6,\ Q_{7,7} = Q_{8,8} = 2.222\text{x}10^3.$$

The R_o matrix is chosen as an identity matrix. A parametric study is done using various multiples of the Q_o ($Q=\alpha Q_o$) and R_o matrices obtained above which are plotted against the negative real part of the least damped mode of the controlled system in Fig. 5. All the loci of the negative real part of the least damped mode approach unity and no significant improvement is observed by increasing the state penalty, $Q=\alpha Q_o$, any further. Thus one wishes to operate on the horizontal line between the points (1) and (2). The maximum amplitude of the forces for R = I and R = 1000 I are calculated and plotted in Fig. 6. The closed loop poles of the controlled system at points (1) and (2) are virtually the same and are given as follows (nondimensionalized):

$-1.0043,\ -1.8\pm i16.64,\ -2.16\pm i24.20,\ -17.18\pm i19.79,$

$-26.23,\ -36.22,\ -137.64$ and $-38.66\pm i1132.11$

The maximum force amplitudes as shown in Fig. 6 are less than those corresponding to Case (i) - Location I of Table 1 for comparable transient responses, whereas these are high as compared to the forces obtained using the pole clustering technique (Section (b)). This is due to the large negative real parts of the other modes in the linear regulator case when compared to the pole clustering technique where all the poles have an equal negative real part (-1.0). Both the linear regulator and pole clustering technique have the draw back that the controlled frequencies can be quite high compared to the uncontrolled frequencies. On the otherhand, these techniques have the advantage that they can be applied to situations where the number of actuators is less than the number of modes in the mathematical model, in contrast to the decoupling technique of Section (a).

5. Conclusions

In this paper the dynamics, stability, and control of an orbiting homogeneous, flexible square platform are considered. Three different nominal orientations of the platform are examined. When the platform is nominally following the local vertical with its larger surface perpendicular to the orbital plane and also when the platform follows the local horizontal with its larger surface normal to the local vertical, it is seen that the uncontrolled roll/yaw motion is unstable. For the case where the platform follows the local vertical with its large surface perpendicular to the orbit normal, the uncontrolled pitch motion is found to be unstable.

Three different control techniques are considered for the selection of the control laws:

a) The decoupling of the original state equations using state variable feedback eliminates the need of a transformation from the original coordinates to the modal coordinates and provides a method of specifying directly the amount of damping and frequency of the individual components of the state vector. However, with this technique the number of actuators must be equal to the number of coordinates (modes) in the model.

b) The pole placement algorithm (ORACLS) guarantees the over-all required damping of the system and does not restrict the number of actuators to be equal to the number of modes in the model. However, it is seen that the closed-loop frequencies may be greatly increased when compared to the open-loop values which may cause problems with externally induced periodic excitations.

c) The linear regulator theory can provide acceptable performance once the state and penalty matrices are properly selected, and the number of actuators can be less than the number of modes in the model. Computer capacity and accuracy limit the number of modes that can be considered. Here, too, an undesirable increase in the closed-loop frequencies may result in order to provide satisfactory responses with maximum allowable force amplitudes.

References

1. "Outlook for Space," NASA Report SP-386, January 1976.

2. "Industry Workshop on Large Space Structures," NASA Contractor Report CR-2709, Contract No. NAS-1-12436, for NASA Langley Research Center, May 1976.

3. Kumar, V.K. and Bainum, P.M., "Dynamics of a Flexible Body in Orbit," AIAA/AAS Astrodynamics Conference, Palo Alto, Calif., August 7-9, 1978, Paper No. 78-1418; also, Journal of Guidance and Control, Vol. 3, No. 1, January-February 1980, pp. 90-92.

4. Santini, Paolo, "Stability of Flexible Spacecrafts," Acta Astronautica, Vol. 3, 1977, pp. 685-713.

5. Bainum, P.M. and Reddy, A.S.S.R., "On the Controllability of a Long Flexible Beam in Orbit," Proceedings of the Second AIAA Symposium on Dynamics and Control of Large Flexible Spacecraft, June 21-23, 1979, VPI & SU Press, (Ed. L. Mairovitch), 1980, pp. 145-159.

6. Reddy, A.S.S.R., Bainum, P.M., and Hamer, H.A., "Decoupling Control of a Long Flexible Beam in Orbit," Astrodynamics 1979, Advances in the Astronautical Sciences, Vol. 40, Part II, AAS Publication, 1980, pp. 649-673.

7. Bainum, P.M., Reddy, A.S.S.R., Krishna, R., and James, P., "The Dynamics and Control of Large Space Structures III," Final Report, NASA Grant NSG-1414, Supp. 2, Part A: Shape and Orientation Control of a Platform in Orbit using Point Actuators, Howard University, June 1980.

8. Armstrong, E.S., "ORACLS - A System for Linear-Quadratic-Gaussian Control Law Design," NASA Technical Paper 1106, April 1978.

9. Bainum, P.M. and Sellappan, R., "Optimal Control of Spin-Stabilized Spacecraft with Telescoping Appendages," The Journal of the Astronautical Sciences, Vol. XXIV, No. 4, October-December 1976, pp. 329-346.

10. Amieux, J.C. and Liegeois, A., "Design and Ground Test of a Pendulum - Type Active Nutation Damper," Journal of Spacecraft and Rockets, Vol. 11, No. 11, Nov. 1974, pp. 790-792.

Appendix A - Nomenclature

A_r	r^{th} modal amplitude function
B	Control influence matrix
C_x, C_y, C_z	Disturbance torques about the principal undeformed body axes
E_r	Generic force on r^{th} mode
f	Force due to an actuator
f_x, f_y, f_z	Force components due to an actuator
$G_{R_x}, G_{R_y}, G_{R_z}$	Gravity gradient torques about the principal undeformed body axes
I_x, I_y, I_z	Moments of inertia about the principal axes
K	Gain matrix
K_r, K_p	Rate and position feedback gain matrices
M_r	r^{th} modal mass
T	Torque due to an actuator
T_x, T_y, T_z	Torque components
$W_r(x,y)$	r^{th} modal shape function
Z_r	Nondimensionalized r^{th} modal amplitude function
ω_c	Orbital frequency
$\omega_x, \omega_y, \omega_z$	Angular body rates
ϕ, ψ, θ	Roll, yaw, pitch, respectively
$\omega_1, \omega_2, \omega_3$	First three modal frequencies of the plate

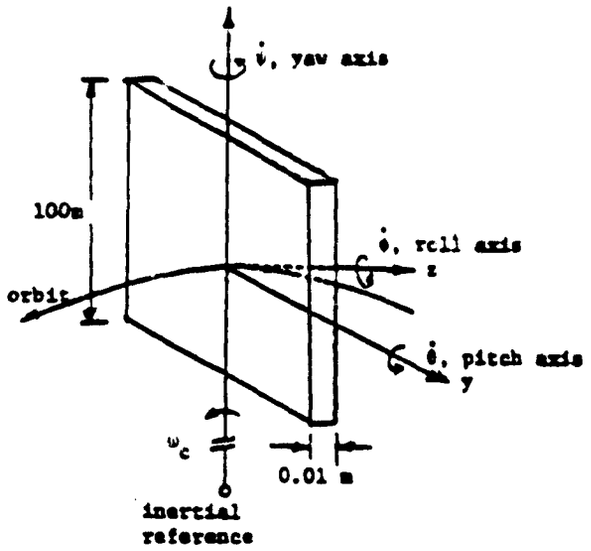


Fig. 1a. Platform following local vertical with major surface normal to the orbit plane- Case (i)

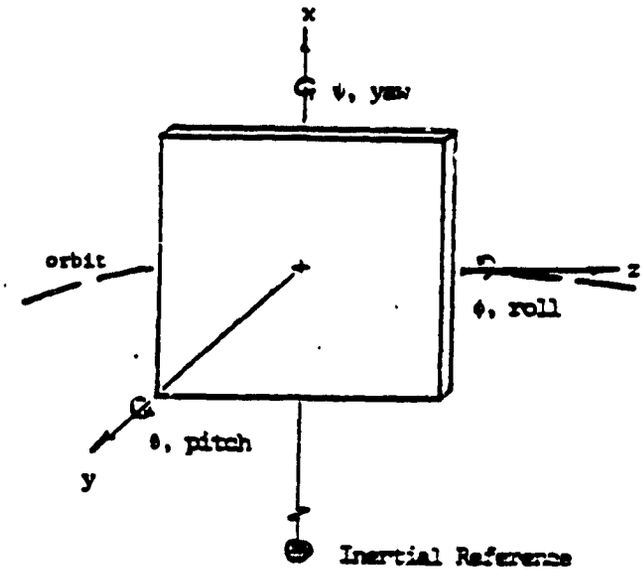


Fig. 1c. Platform following local vertical with major surface in the orbit plane - Case (iii)

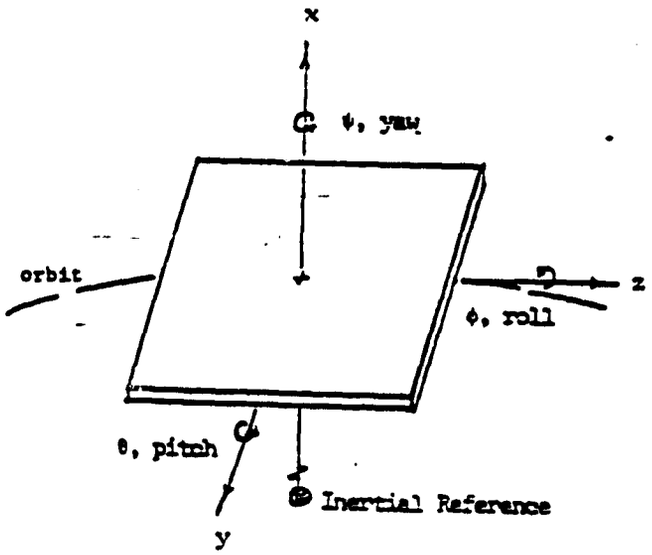


Fig. 1b. Platform along local horizontal - Case (ii)

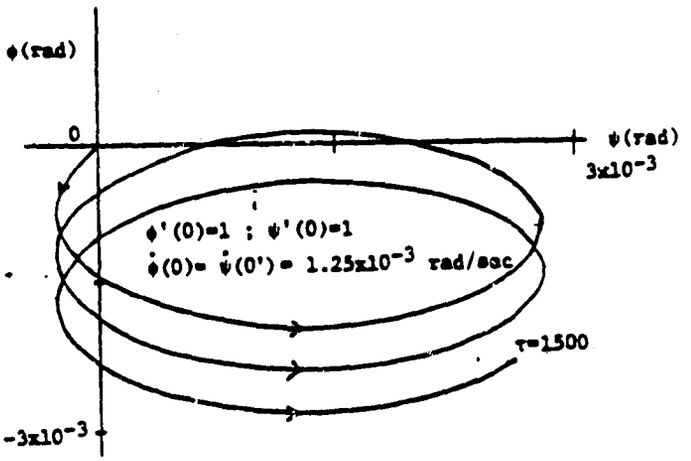


Fig. 2. Roll/yaw motion (uncontrolled) - Case (i)

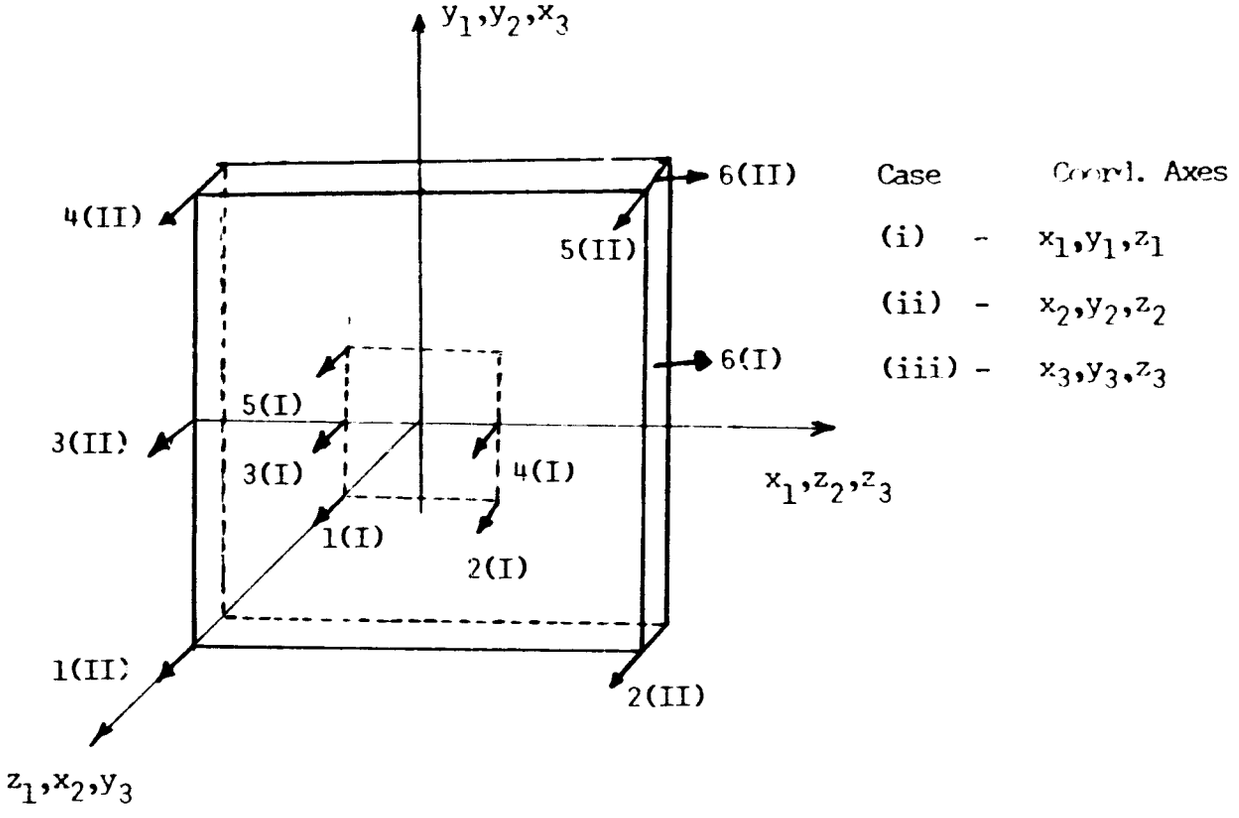


Fig 3. Location of two sets of actuators (I & II)

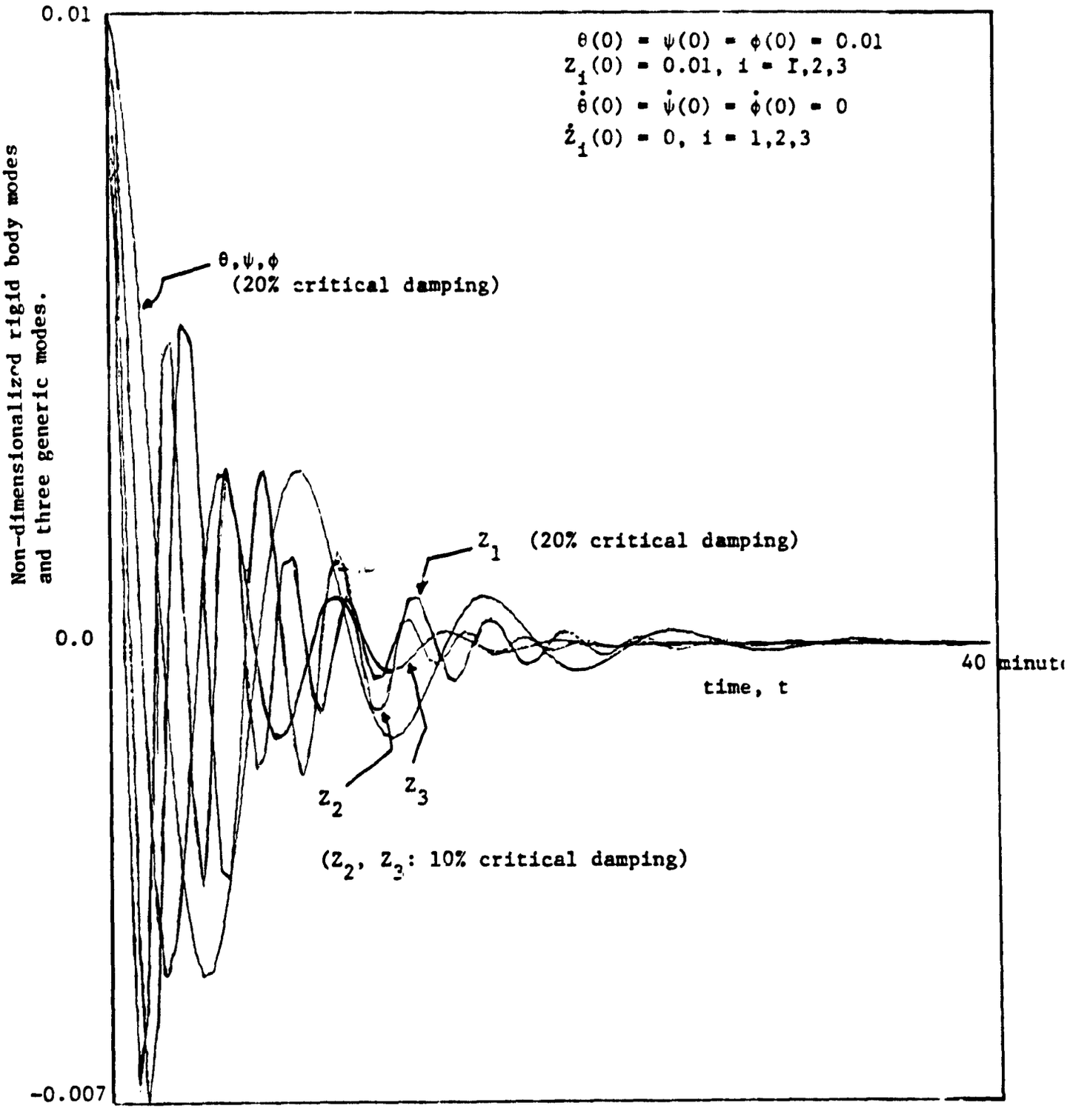


Fig 4a. Controlled state response for all combinations of orientations and actuator locations.

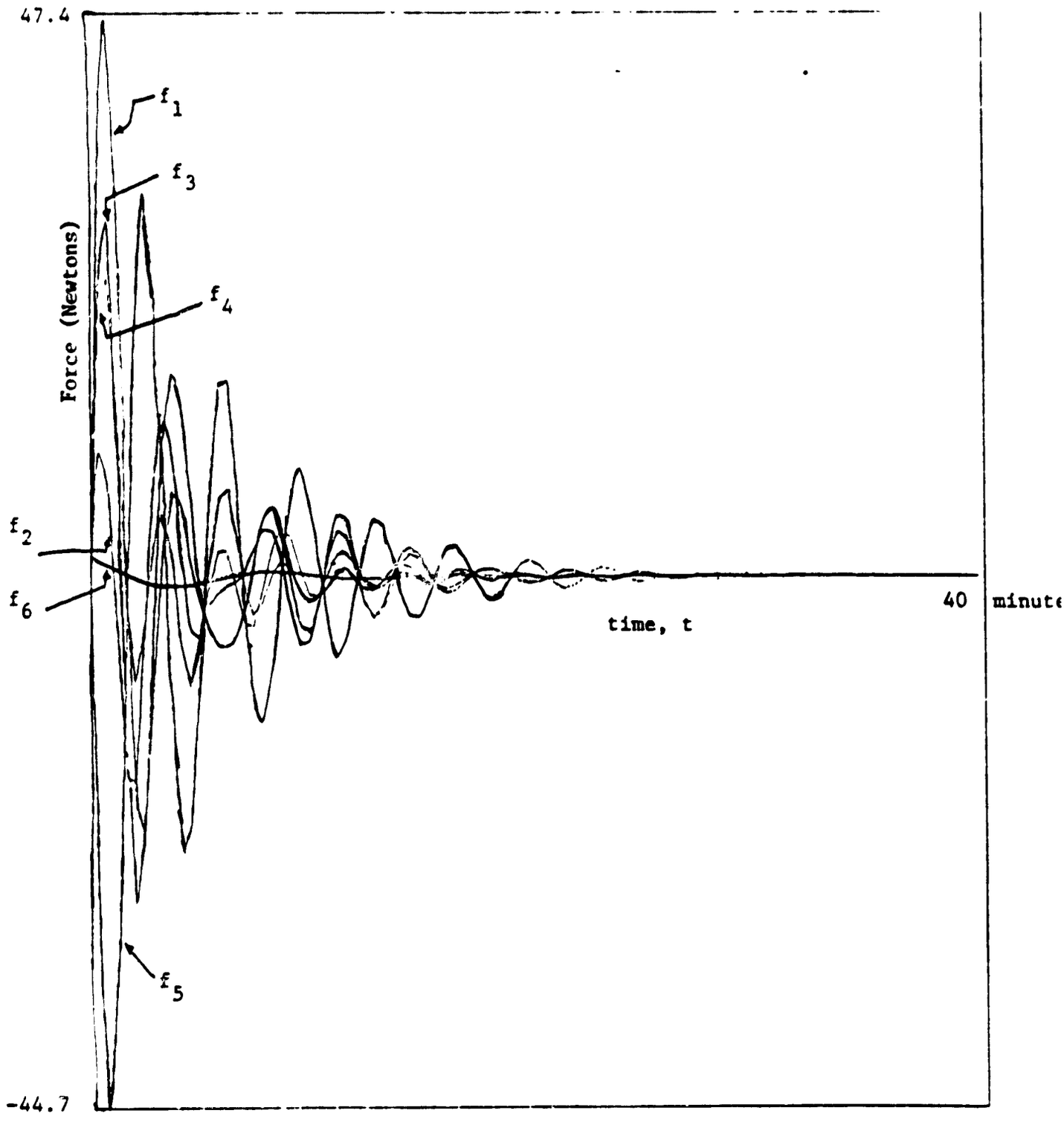


Fig 4b. Control force time history for Case (iii) - II

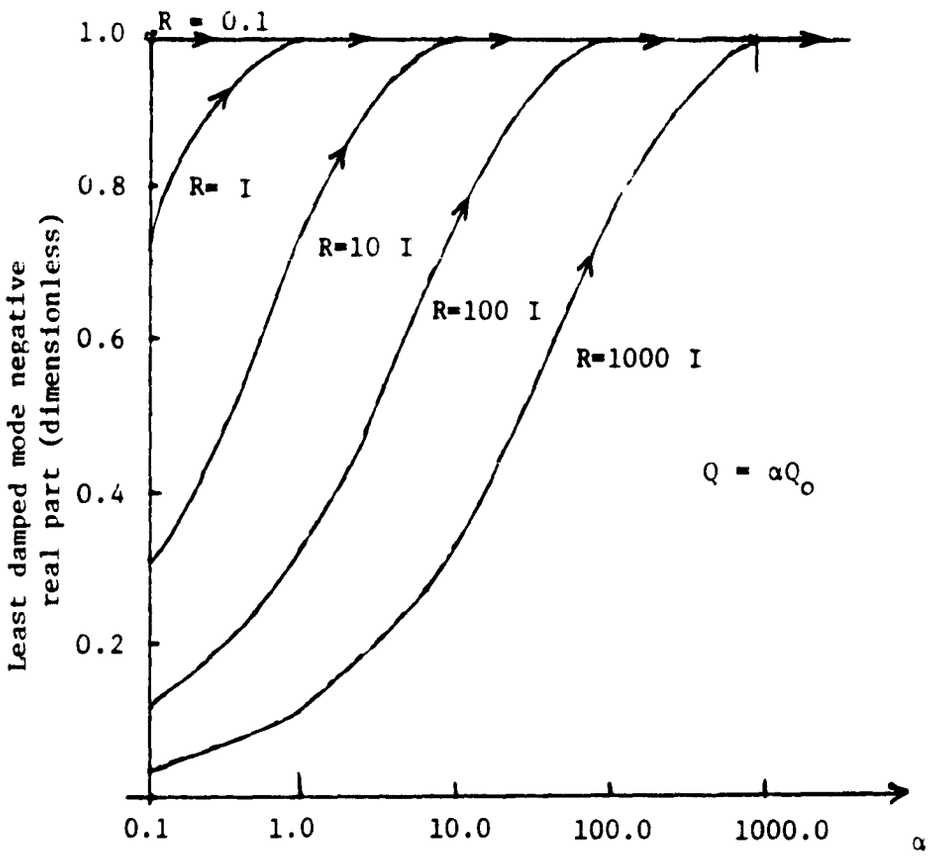


Fig. 5. Variation of least damped mode negative real part with α and R.

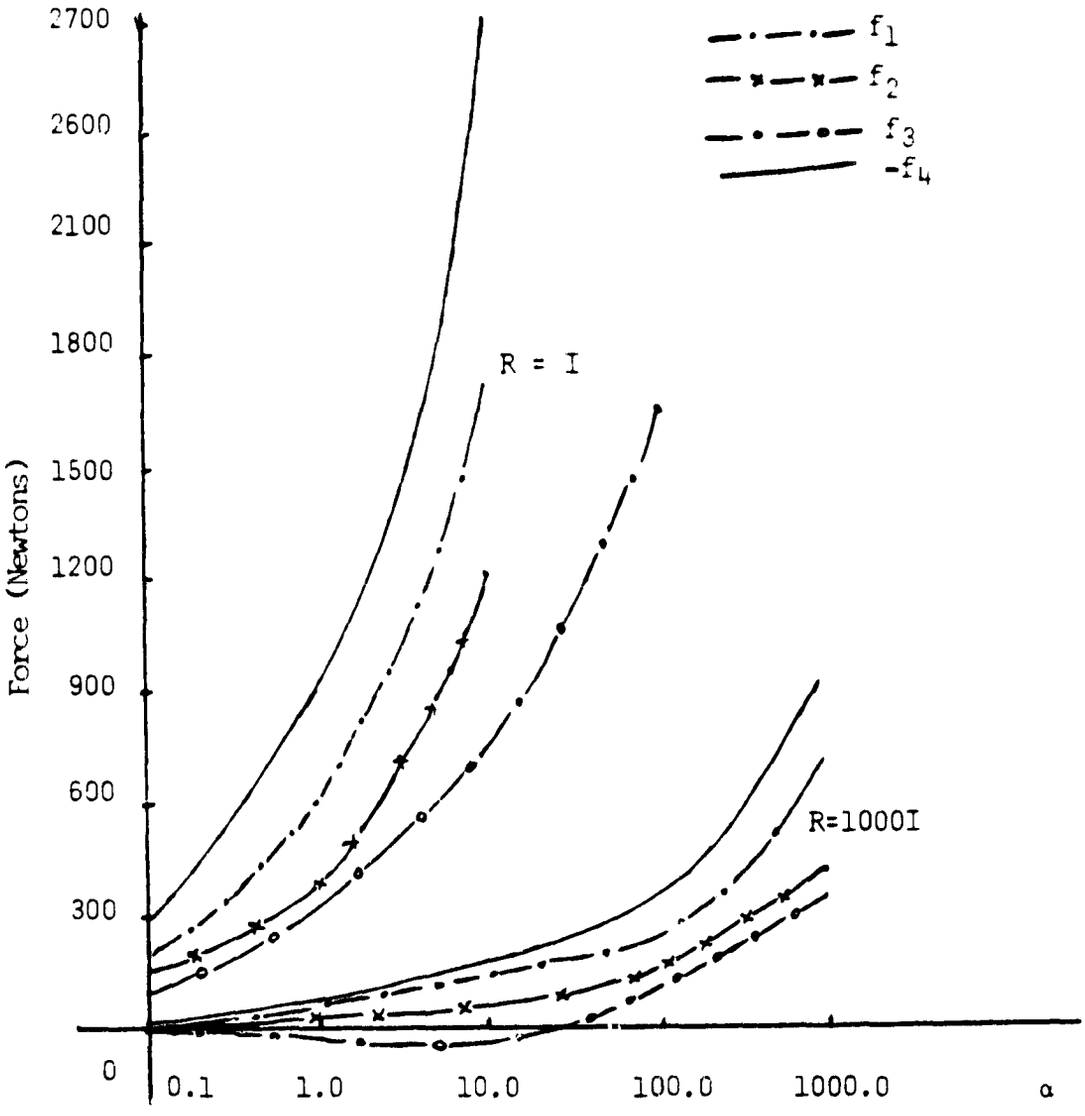


Fig 6. Maximum force amplitudes as a function of α and R for all actuators - (application of linear regulator theory).

III. FREQUENCIES AND MODE SHAPES FOR RECTANGULAR PLATES

The ability to determine accurately the frequencies and mode shapes is essential for the analysis and control of large structures in orbit. A thin rectangular plate, an important basic structure for several space applications, is considered for vibrational analysis. In the following sections the plate is assumed to be large, thin, and homogeneous, and all the edges are assumed to be free to vibrate. First, the approximate frequencies and mode shapes of a rectangular plate obtained by Warburton[1] is discussed. This analysis also includes the special case of a square plate. Next, the analytical results for a square plate using the method of Lemke[2] is considered. For a specific example of a square plate both analytical results are applied to determine the frequencies and mode shapes. An available finite element computer program[3] is also used to obtain the frequencies and mode shapes of this plate. The results of both analytical methods and the computer routine are compared and discussed.

1. Formulation by Warburton

The approximate frequency formula is derived by applying the Raleigh method. The details of this method are given in the earlier contract report.[4] The basic equation used was the plate vibrational equation in the cartesian co-ordinate system (x,y), with the length and width of the plate taken along the x and y directions, respectively, and is given as

$$\frac{\partial^4 W}{\partial x^4} + 2\frac{\partial^4 W}{\partial x^2 \partial y^2} + \frac{\partial^4 W}{\partial y^4} + \frac{12\rho(1-\sigma^2)}{Egh^2}\frac{\partial^2 W}{\partial t^2} = 0 \qquad \text{(III-1)}$$

where ρ, σ and E are the density, Poisson's ratio and Young's modulus of the plate material, respectively, h is the plate thickness, and g is the acceleration due to gravity. The displacement, w, at any point (x,y) at time t is given by

$$w = W \sin \omega t = A\, \theta(x)\, \phi(y) \sin \omega t \qquad \text{(III-2)}$$

$\theta(x)$ and $\phi(y)$ can be taken as the beam functions orthogonal to each other and can be used to approximate plate behavior. After taking the appropriate free-free beam functions for $\theta(x)$ and $\phi(y)$, the frequency expressions for a rectangular plate was derived as[1]

$$\lambda^2 = \frac{\rho a^4 (2\pi f)^2\, 12(1-\sigma^2)}{\pi^4\, Eh^2 g} \qquad \text{(III-3)}$$

and $$\lambda^2 = G_x{}^4 + G_y{}^4 \frac{a^4}{b^4} + 2\,\frac{a^2}{b^2}\,[\sigma\, H_x H_y + (1-\sigma)\, J_x J_y] \qquad \text{(III-4)}$$

where λ is a non-dimensional frequency factor, a and b are the length and width of the plate, and G_x, H_x, J_x, G_y, H_y, and J_y are functions associated with the number of nodal lines, m and n, parallel to x and y, respectively, for the beam functions $\theta(x)$ and $\phi(y)$, and are given in Table III-1. From Eq. (III-3), the frequency is obtained as

$$f = \frac{\lambda h \pi}{a^2}\left[\frac{Eg}{48\,\rho\,(1-\sigma^2)}\right]^{\frac{1}{2}} \quad \text{(cps)} \qquad \text{(III-5)}$$

Eq. (III-5) is valid for thin rectangular plates. However, for square plates, (m,n) $\pm$ (n,m) types of modes exist, and for these cases λ in Eq. (III-5) must be modified. These cases are discussed in detail in Ref. 1 and a few relevant results are given here.

$$\underline{\text{Modes (m,o) + (o,m), m is even}}$$

$$\lambda^2 = (m-\tfrac{1}{2})^4 \pm 2\,\sigma(m-\tfrac{1}{2})^2\,\frac{8}{\pi^2}$$

$$\underline{\text{Modes (m,1) } \pm \text{ (1,m), for m = 3,5,7,....}} \qquad \text{(III-6)}$$

$$\lambda^2 = (m-\tfrac{1}{2})^4 + 2(1-\sigma)\,(m-\tfrac{1}{2})^2\left[1 + \frac{6}{(m-\tfrac{1}{2})\pi}\right]\frac{12}{\pi^2}$$

$$\pm 2\,\sigma(m-\tfrac{1}{2})^2\,\frac{24}{\pi^2}\left[1 - \frac{2}{(m-\tfrac{1}{2})\pi}\right]^2 \pm 2(1-\sigma)\,\frac{192}{\pi^4}$$

For any mode of vibration the nodal pattern is defined by m and n, the number of nodal lines in the x and y directions, respectively. The mode shapes are obtained by using the corresponding modal frequencies in the beam functions and then evaluating the product, $\theta(x)\cdot\phi(y)$, numerically.

2. Formulation by Lemke[2]

The frequencies and mode shapes were computed for a square plate using the Raleigh-Ritz method. The results are readily available only for six of the modes obtained by Warburton's method. Lemke uses displacement functions of the type,

$$W = \Sigma\, A_{m,n}\ \theta_m(x)\ \phi_n(y) \qquad \text{(III-7)}$$

where $\theta_m(x)$ and $\phi_n(y)$ are the free beam functions given as

$$\theta_m(x) = \frac{\cosh k_m \cos k_m x + \cos k_m \cosh k_m \bar{x}}{\sqrt{\cosh^2 k_m + \cos^2 k_m}} \quad (m \text{ even}) \qquad \text{(III-8)}$$

$$= \frac{\sinh k_m \sin k_m x + \sin k_m \sinh k_m x}{\sqrt{\sinh^2 k_m - \sin^2 k_m}} \quad (m \text{ odd})$$

$\phi_n(y)$ is obtained from Eq. (III-8) by replacing x by y and m by n.

The values, k_m, are the roots of the equations

$$\tan k_m + \tanh k_m = 0 \qquad \text{(m even)}$$

$$\tan k_m - \tanh k_m = 0 \qquad \text{(m odd)}$$

which result from the spatial boundary conditions. Further, it was shown by an energy principle that [2,4]

$$\omega^2 = \frac{U_{max}}{\frac{\rho h}{2g} \int_0^a \int_0^b W^2 \, dxdy} \qquad \text{(III-9)}$$

where U_{max} is the maximum potential energy due to bending. The co-efficients, A_{mn}, in Eq. (III-7) are determined to make ω^2 in Eq. (III-9) a minimum. Lemke obtained the coefficients, A_{mn}, by taking six or more terms in the series (III-7) and using four different values of Poisson's ratio. Expressions for six mode shapes and frequencies along with the co-efficents, A_{mn}, are tabulated in Ref. 2. As an example the expression for the first mode is given here.

$$W(x,y) = x_1y_1 + 0.0325\,(x_1y_3 + x_3y_1) - .005\,x_3y_3 - .00257\,(x_1y_5 + x_5y_1) + .00121\,(x_3y_5 + x_5y_3) - .000365\,x_5y_5 + \ldots.$$

and $$\omega = \frac{13.086}{a^2}\sqrt{\frac{E\,h^3}{12\,\rho\,(1-\sigma^2)}} \quad \text{for } \sigma = .343$$

3. Finite Element Computer Program

The computer program used is the Structural Design Language (STRUDL) which uses the finite element method to determine the mode shapes and the frequencies of vibration. The input to the computer routine is given by specifying the type of structure and supplying other physical properties and dimensions of the structure. For a rectangular plate, the finite elements can be specified as rectangular elements and the number of elements into which the plate should be divided depends upon the accuracy required. STRUDL gives deflections at each corner of the elements for all the modes from which the mode shapes can be determined. Further, a set of frequencies corresponding to the modes generated is obtained. In general, the accuracies of the frequencies and mode shapes will improve if the plate is modelled with a higher number of elements. However, computational errors due to truncation and round-off errors may predominate as the order of the elements increases beyond a limit. Further, the limitations of the computers will restrict the number of elements into which the plate can be divided to obtain more accurate results.

4. Discussion of Numerical Results

A square palte of sides 100 meters each and thickness 0.01 meters is considered to obtain the numerical results. The material of the plate is assumed to be aluminium with the following properties.

density = 2768.0 kg/m^3

Young's modulus = 0.7441×10^{10} kg/m^2

Poisson's ratio = 0.33

Using Warburton's results, Eq. (III-4), Eq. (III-5), Table (III-1), and expressions for $\theta(x)$ and $\phi(y)$, frequencies and mode shapes are calculated for different combinations of the number of nodal lines, m and n, starting with combinations of m=0 and n=1, through m=3 and n=3. The first three combinations of nodal line numbers, (0,0), (1,0) and (0,1), represent rigid body motion. The first fundamental flexural frequency is seen to be due to a combination of m=1 and n=1. The corresponding mode shape for the plate is obtained by multiplying the beam functions, $\theta(x)$ and $\phi(y)$, for (beam) mode numbers 1 and 1, respectively (Fig. 1). Since the plate is approximated by sets of orthogonal beams in the x and y directions, the nodal pattern is also obtained by plotting the nodal points of these beams for their first modes. The next two higher frequencies are obtained by combinations of m=0 and n=2, but the nodal patterns (Figs. 2,4) can not be visualized as before. This is because these frequencies are of a special type resulting from a combination of the (2,0) and (0,2) plate modes. It can be seen that when the mode corresponding to (2,0) (Fig. 3(a)) is superimposed on the mode - (0,2) (Fig. 3(b)) the mode shape depicted in Fig. 2 results. Similarly by superimposing the (2,0) and (0,2) modes the third mode shape (Fig. 4) is obtained. The two combinations of nodal patterns m=1 and n=2, give identical frequencies for the fourth and fifth mode and the corresponding shapes (Fig. 5) are as expected. The next two higher frequencies are also identical and result from combinations of the (3,0) and (0,3) modes. The eighth frequency is obtained from m=2 and n=2 and the mode shape obtained is shown in Fig. 7.

However, the ninth and tenth mode shapes obtained by the (3,1) and (1,3) combinations, are once again of a special type. The ninth mode shape is obtained by superimposing the (1,3) and (3,1) patterns (Fig. 8) and the tenth mode shape is obtained by superimposing (1,3) and (3,1) nodal patterns (Fig. 9). The next higher frequencies are obtained from combinations of the (3,2), (2,3) and (3,3) modes, respectively. The frequencies and nodal patterns obtained for all these modes are shown in Table 2.

Frequencies and mode shapes are also obtained by using the expressions for the six modes given by Lemke.[2] The first three frequencies and mode shapes obtained agree with the frequencies and mode shapes computed from Warburton's formulas (Table 1). However, the next three frequencies obtained by Lemeke's method correspond to higher frequencies and mode shapes obtained by Warburton's method. Also the nodal patterns obtained by Lemke's method compare approximately with the nodal patterns obtained by Warburton's method although the frequencies do not correspond in all cases. The results obtained by Lemke do not show the four intermediate frequencies corresponding to the fourth, fifth, sixth and seventh modes obtained by Warburton's method. The frequencies and nodal pattern obtained by Lemke's method are shown in Table 2.

For implementation of the computer program STRUDL, first the plate is divided into four elements. The first six modes (in terms of increasing frequencies) as predicted by STRUDL are also apparent from Warburton's results. The plate is assumed to be divided into 9, 16, 36 and 64 elements, respectively. The results of STRUDL are tabulated in Table 2. It can be seen that STRUDL frequencies approach the frequencies obtained by Warburton's method as the number of plate elements is increased.

However, in the cases of 36 and 64 elements some of the frequencies show a tendency to oscillate about an average value. This probably is due to the computational round off errors which begin to dominate with the increasing computations associated with larger number of elements. Thus, the advantage of taking a large number of elements may not be fully realized due to numerical accuracy limitations. Computation with more elements requires more computation time and a larger computer memory. For the 64 elements case, it was not possible to obtain the mode shapes due to memory limitations. It was also observed that the convergence of the frequencies, with an increase in the numer of elements, is faster than the convergence of the mode shapes. It can be seen from Table 2, that the numerical results of STRUDL using 36 elements correlate with the results of Warburton both in frequency and mode shapes.

Table 3 and Table 4 compare non-dimensionalized deflections at the nodes (corners of elements) obtained by the three methods for the second mode (Fig. 2). For locations where deflections exist and do not correspond to maximum amplitude ($\pm$ 1.0) in all cases the results predicted by STRUDL lie in between the results obtained by the analytical methods of Lemke and Warburton.

The results of this comparative study give an indication of the types of modelling errors that would be expected in the estimation of the frequencies and mode shpaes of the fundamental and lower order flexural modes of a large platform type structure in orbit. As an extension to this study the use of more powerful (and accurate) finite element computer algorithms, not currently available at Howard University, is recommended.

5. References

1. Warburton, G.B., "The Vibration of Rectangular Plates," Proc. Inst. Mech. Engrs., Vol. 168, No. 12, 1954, pp. 371-394.

2. Leissa, A.W., "Vibration of Plates", NASA SP-160, NASA, Washington, D.C., 1969, pp. 87-110.

3. ICES-STRUDL-II, Engineering Users Manual, Volume II, 1972.

4. Bainum, P.M. and James, P.K., "The Dynamics and Control of Large Flexible Space Structures - II. PART B: Model Development and Computer Simulation," June 1979.

TABLE III-1. Evaluation of Parameters in Frequency Expression (Warburton)

m	G_x	H_x	J_x	n	G_y	H_y	J_y
0	0	0	0	0	0	0	0
1	0	0	$\frac{12}{\pi^2}$	1	0	0	$\frac{12}{\pi^2}$
2	1.506	1.248	5.017	2	1.506	1.248	5.017
3 4 5 ⋮	$m-\frac{1}{2}$	$(m-\frac{1}{2})^2 P$	$(m-\frac{1}{2})^2 Q$	3 4 5 ⋮	$(n-\frac{1}{2})$	$(n-\frac{1}{2})^2 P$	$(n-\frac{1}{2})^2 Q$

$P = [1-2/(m-0.5)\pi]$ $\qquad$ $Q = [1+6/(m-0.5)\pi]$

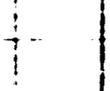
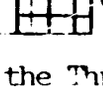
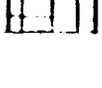
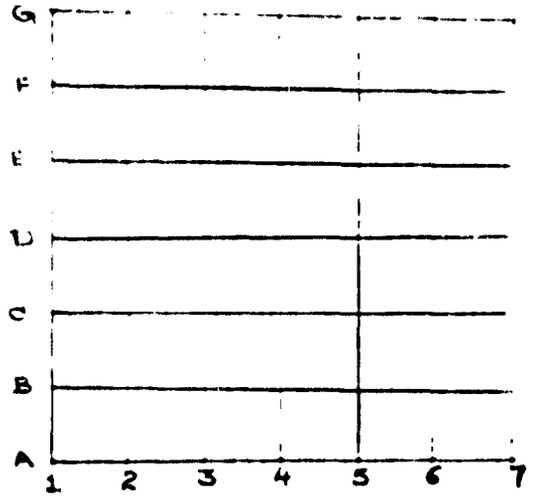
	STRUDL					WARBURTON			LEMKE	
	NUMBER OF ELEMENTS					Nodal Pattern	Freq. (cps)	(m,n)	Nodal Pattern	Freq. (cps)	Nodal Pattern
	4	9	16	36	64						
1	.003165	.003284	.003315	.003314	.003358		.003471	(1,1)		.003287	
2	.004291	.004664	.004776	.004839	.004856		.004846	(2,0)-(0,2)		.004831	
3	.005549	.006133	.006221	.006221	.006207		.006175	(2,0)+(0,2)		.006175	
4	.007619	.008453	.008632	.008690	.008692		.00900	(2,1)		.015744	
5	.007619	.008453	.008632	.008690	.008692		.00900	(1,2)		.017169	
6	.01323	.01512	.01600	.01596	.01579		.01542	(3,0)		.01944	
7		.01512	.01600	.01596	.01579		.01542	(0,3)			
8		.01584	.01618	.01615	.01607		.01656	(2,2)			
9		.01630	.01727	.01750	.01746		.01715	(3,1)-(1,3)			
10		.01843	.02022	.02031	.02005		.0208	(1,3)+(3,1)			
11		.02465	.02693	.02693	.02700		.02726	(3,2)			
12		.02465	.02693	.02693	.02700		.02726	(2,3)			
13		.03362	.03029	.03162	.03104		0.03941	(3,3)			

TABLE-III-2. Frequencies and Nodal Patterns Obtained by the Three Methods.

Case 1: 4 Elements

LOCATION	STRUDL	LEMKE	WARBURTON
A1	0	0	0
A2	1.0	1.0	1.0
B1	-1.0	-1.0	-1.0
B2	0	0	0



grid used for 36 elements

Case 2: 9 Elements

Case 3: 16 Elements

LOCATION	STRUDL	LEMKE	WARBURTON	STRUDL	LEMKE	WARBURTON
A1	0	0	0	0	0	0
A2	1.0	1.0	1.0	.7014	.7327	.6837
A3	1.0	1.0	1.0	1.0	1.0	1.0
B1	-1.0	-1.0	-1.0	-.7014	-.7327	-.6837
B2	0	0	0	0	0	0
B3	0	0	0	.2862	.2673	.3163
C1	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
C2	0	0	0	-.2862	-.2673	-.3163
C3	0	0	0	0	0	0

TABLE-III-3. Normalized Deflections at the Nodal Points Obtained by the Three Methods - Second Mode.

Case 4: 36 Elements

LOCATION	STRUDL	WARBURTON	LEMKE	LOCATION	STRUDL	WARBURTON	LEMKE
A1	0	0	0	C_1	-.8590	-.8524	-.9793
A_2	.4874	.4733	.5346	C_2	-.3595	-.3792	-.3447
A_3	.8590	.8524	.9793	C_3	0	0	0
A4	1.0	1.0	1.0	C_4	.1334	.1475	.1207
B1	-.4873	-.4733	-.5346	D_1	-1.0	-1.0	-1.0
B2	0	0	0	D_2	-.4938	-.5267	-.4654
B3	.3594	.3792	.3447	D_3	-.1334	-.1475	-.1207
B4	.4938	.5267	.4654	D_4	0	0	0

TABLE III-4. Normalized Deflections at the Nodal Points Obtained by the Three Methods - Second Mode.

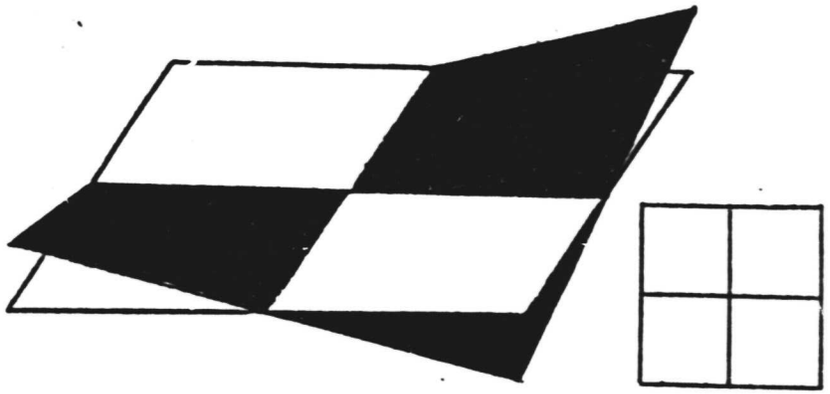


Fig. 1 First Mode

(1,1)

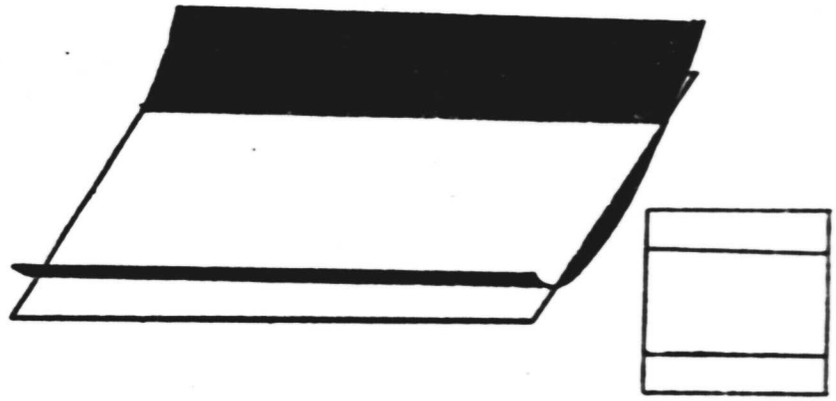


Fig. 3(a) (2,0) Mode

(2,0)

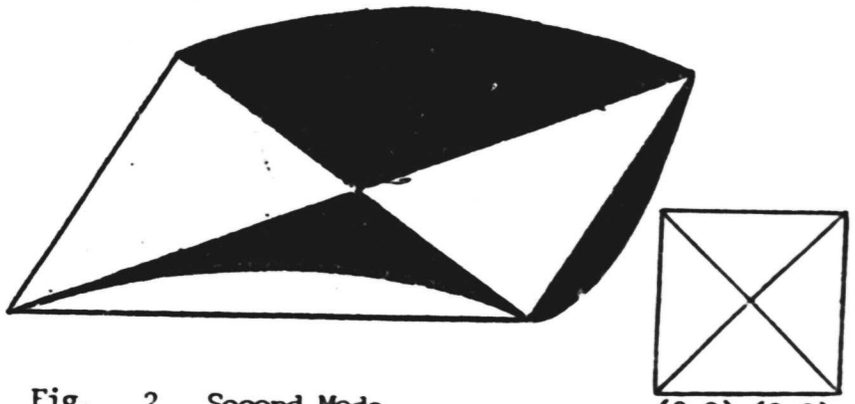


Fig. 2 Second Mode

(2,0)-(0,2)

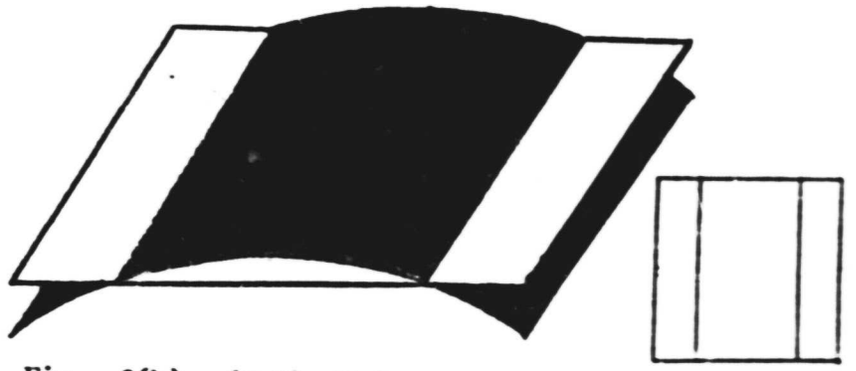


Fig 3(b) (0,2) Mode

(0,2)

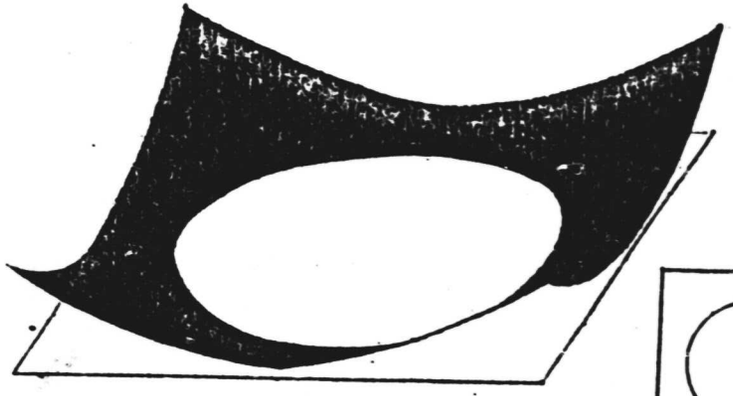
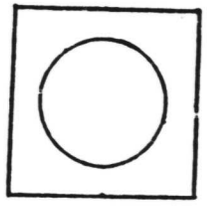


Fig. 4: Third Mode



(2,0) + (0,2)

Fig. 6: Sixth Mode

(3,0)

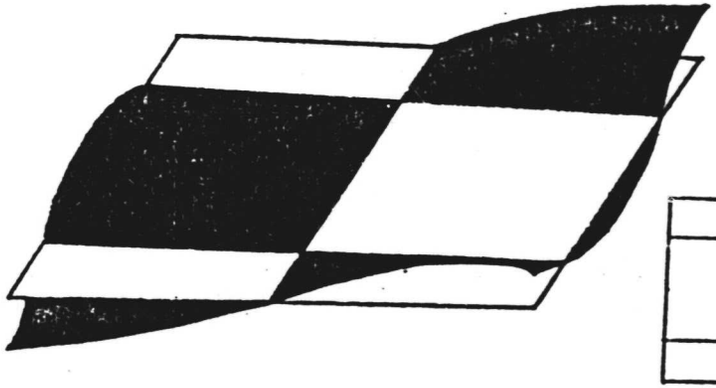
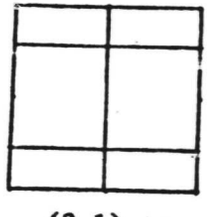


Fig. 5: Fourth Mode



(2,1)

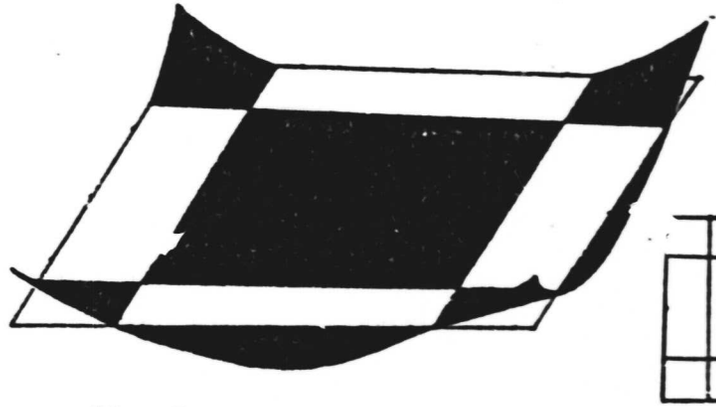
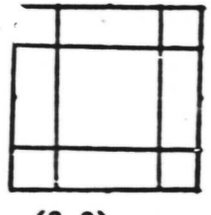


Fig. 7: Eighth Mode



(2,2)

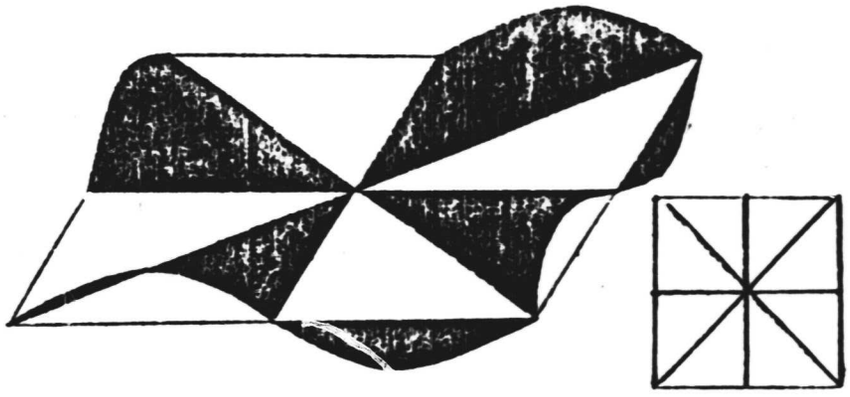


Fig. 8: Ninth Mode

(1,3)-(3,1)

Fig. 9: Tenth Mode

(1,3)+(3,1)

IV. GENERAL CONCLUSIONS AND RECOMMENDATIONS

A model is developed for predicting the dynamics of a large flexible free-free thin platform in orbit under the influence of control devices which are considered to be placed at specific locations on the major surface and one of the edges. Control about three different nominal orientations is considered. In the absence of control, for the case of a completely homogeneous platform instability in at least some of the modes is indicated for small amplitude motion about each of the three orientations. Once controllability is established, for a set of actuator locations, three different techniques are employed for the selection of actuator control laws:

(1) the decoupling of the original state equations using state variable feedback;

(2) a pole placement algorithm; and

(3) an application of the linear regulator theory

It is seen that each of the three techniques have certain distinct advantages and also specific limitations), which are discussed in detail in Chapter II. For systems involving multi-degrees of freedom (such as in this application), the implementation of these techniques requires the extensive usage of computer algorithms.

As a logical extension to the present study which assumes perfect instantaneous knowledge of the state, the modelling of the sensor dynamics and related problem of observability should be considered, once specific information on the types of sensors required for monitoring the performance of large flexible systems is available.

The problems caused by both observation and control spillover could also be treated, perhaps by beginning with the simpler model of the control of a long, flexible beam in orbit and then extending this analysis to the three dimensional model of the platform.

A model of the uncontrolled dynamics of a large flexible shallow spherical shell (representative of an antenna dish or large radiometer) in orbit has been developed during the present grant year (see Part B this report). It is suggested that the effect of control devices be included in this model and that control laws could then be developed using different algorithms already in existence.

APPENDIX

Modifications to ORACLS Software Package

The ORACLS[8] Software Package that was developed at Langley which operates on the Control Data Cyber Computer System was modified to suit the IBM 370/165 Computer System that is available at Howard University. The major modifications that were done are described below:

(1) As the single precision accuracy on the CDC is approximately equal to the double precision on the IBM/370 System, the entire package was converted into double precision.

(2) Some of the machine dependent constants were changed accordingly.

(3) As the IBM System accepts only six letters for a subroutine/function name all the names that exceeded six letters were changed and the list of those subroutines is given below:

	Old Name	New Name
(1)	TESTSTA	TESTSA
(2)	VARANCE	VARANC
(3)	TRANSIT	TRNSIT
(4)	DISCREG	DISREG
(5)	CNTNREG	CNTREG
(6)	RICTNWT	RICNWT
(7)	ASYMREG	ASMREG
(8)	ASYMFIL	ASMFIL
(9)	EXPMDFL	EXPMDF
(10)	IMPMDFL	IMPMDF

(4) Some of the additional supporting subroutines/functions required were added and the names of these subroutines are given here:

(1) PNCH

(2) DIMAG

(3) DREAL

(4) BLOCK DATA

(5) None of the arguments of the subroutines were changed

The listing of the modified ORACLS package is given in the following pages. These routines have to be used in conjunction with Ref. (8). The numbers that appear in front of the FORTRAN statements are line numbers and have to be ommitted.

```
      SUBROUTINE RDTITL                                                 RDT00010
      IMPLICIT REAL*8 (A-H,O-Z)                                         RDT00020
      COMMON/LINES/TITLE(10),TIL(3),NLP,LIN                             RDT00030
      COMMON/FORM/FMT1(2),FMT2(2),NEPR                                  RDT00040
      COMMON/TOL/EPSAM,EPSBM,IACM                                       RDT00050
      COMMON/CONV/SUMCV,RICTCV,SERCV,MAXSUM                             RDT00060
C     NLP = NO. LINES/PAGE VARIES WITH THE INSTALLATION                 RDT00070
      READ(5,100,END=90,ERR=91) TITLE                                   RDT00080
  100 FORMAT(10A8)                                                      RDT00090
      CALL LNCNT(100)                                                   RDT00100
      RETURN                                                            RDT00110
   90 CONTINUE                                                          RDT00120
      STOP 1                                                            RDT00130
   91 CONTINUE                                                          RDT00140
      STOP 2                                                            RDT00150
      END                                                               RDT00160
```

```
      SUBROUTINE LNCNT  (N)                                              LNC00010
      IMPLICIT REAL*8 (A-H,O-Z)                                          LNC00020
      COMMON/LINES/TITLE(10),TIL(3),NLP,LIN                              LNC00030
      LIN=LIN+N                                                          LNC00040
      IF  (LIN.LE.NLP)  GO TO 20                                         LNC00050
      WRITE(6,1010) TITLE,TIL                                            LNC00060
 1010 FORMAT(1H1,10A8,3A8/)                                              LNC00070
      LIN=2+N                                                            LNC00080
      IF  (N.GT.NLP)  LIN=2                                              LNC00090
   20 RETURN                                                             LNC00100
      END                                                                LNC00110
```

```
      SUBROUTINE READ(I,A,NA,B,NB,C,NC,D,ND,E,NE)                       REA00010
      IMPLICIT REAL*8 (A-H,O-Z)                                         REA00020
      DIMENSION A(1),B(1),C(1),D(1),E(1)                                REA00030
      DIMENSION NA(2),NB(2),NC(2),ND(2),NE(2),NZ(2)                     REA00040
      READ(5,100) LAB,              NZ(1), NZ(2)                        REA00050
      CALL READ1(A, NA,NZ,  LAB)                                        REA00060
      IF(I .EQ. 1) GO TO 999                                            REA00070
      READ(5,100) LAB,              NZ(1), NZ(2)                        REA00080
      CALL READ1(B, NB,NZ,  LAB)                                        REA00090
      IF(I .EQ. 2) GO TO 999                                            REA00100
      READ(5,100) LAB,              NZ(1), NZ(2)                        REA00110
      CALL READ1(C,NC,NZ,LAB)                                           REA00120
      IF(I .EQ. 3) GO TO 999                                            REA00130
      READ(5,100) LAB,              NZ(1), NZ(2)                        REA00140
      CALL READ1(D, ND,NZ,  LAB)                                        REA00150
      IF(I .EQ. 4) GO TO 999                                            REA00160
      READ(5,100) LAB,              NZ(1), NZ(2)                        REA00170
      CALL READ1(E, NE,NZ,  LAB)                                        REA00180
  100 FORMAT(A4,4X,2I4)                                                 REA00190
  999 RETURN                                                            REA00200
      END                                                               REA00210
```

```
      SUBROUTINE  PRNT(A,NA,NAM,IOP)                                    PRN00010
      IMPLICIT REAL*8 (A-H,O-Z)                                         PRN00020
      DIMENSION A(1),NA(2)                                              PRN00030
      COMMON  /FORM/FMT1(2),FMT2(2),NEPR                                PRN00040
      COMMON/LINES/TITLE(10),TIL(3),NLP,LIN                             PRN00050
C- NOTE NLP NO. LINES/PAGE VARIES WITH THE INSTALLATION.                PRN00060
      DATA KZ,KW,KB /1H0,1H1,1H /                                       PRN00070
      NAME = NAM                                                        PRN00080
      II = IOP                                                          PRN00090
      NR = NA(1)                                                        PRN00100
      NC = NA(2)                                                        PRN00110
      NLST = NR * NC                                                    PRN00120
      IF( NLST .LT. 1 .OR. NR .LT. 1 ) GO TO 16                         PRN00130
      IF(NAME  .EQ. 0) NAME = KB                                        PRN00140
C- SKIP HEADLINE IF REQUESTED.                                          PRN00150
      GO TO (11,10,132,12), II                                          PRN00160
   10 CALL LNCNT(100)                                                   PRN00170
   11 CALL LNCNT(2)                                                     PRN00180
    3 WRITE(6,177) KZ,NAME,NR,NC                                        PRN00190
177   FORMAT(A1,5X,A4,8H  MATRIX,5X,I3,5H ROWS,5X,I3,8H COLUMNS)        PRN00200
      GO TO 13                                                          PRN00210
   12 CALL LNCNT(100)                                                   PRN00220
      GO TO 13                                                          PRN00230
  132 CALL LNCNT(2)                                                     PRN00240
      WRITE (6,891)                                                     PRN00250
  891 FORMAT (1H0)                                                      PRN00260
C- BELOW COMPUTE NR OF LINES/ ROW --DECIDE IF 1 EXTRA BLANK LINE        PRN00270
   13 J=(NC-1)/NEPR+1                                                   PRN00280
C- WHY ALWAYS ADD 1 LINE- BECAUSE IF MULTIPLE, USE 1 BLK LINE EXTRA.    PRN00290
      NLPW=J
      JST=1
C     COMPUTE LAST ROW POSITION -1 BELOW
      NLST = NLST -NR                                                   PRN00330
      MN=NC                                                             PRN00340
      IF  (NC.GT.NEPR)   MN=NEPR                                        PRN00350
      KLST=NR*(MN-1)                                                    PRN00360
91    CONTINUE                                                          PRN00370
      DO 912 J = JST, NR                                                PRN00380
      CALL LNCNT(NLPW)                                                  PRN00390
      KLST = KLST +1                                                    PRN00400
      WRITE (6,FMT1) (A(N), N=J,KLST,NR)                                PRN00410
      IF  (NC.LE.NEPR)   GO TO 912                                      PRN00420
      NLST = NLST +1                                                    PRN00430
      KNR=KLST+NR                                                       PRN00440
      WRITE (6,FMT2) (A(N), N=KNR,NLST,NR)                              PRN00450
912   CONTINUE                                                          PRN00460
      RETURN                                                            PRN00470
16    CALL  LNCNT(1)
      WRITE (6,916) NAM,NA                                              PRN00490
  916 FORMAT  (' ERROR IN PRNT  MATRIX ',A4,' HAS NA=',2I6)             PRN00500
      RETURN                                                            PRN00510
      END                                                               PRN00520
```

```
      SUBROUTINE EQUATE(A,NA,B,NB)                                      EQU00010
      IMPLICIT REAL*8 (A-H,O-Z)                                         EQU00020
      DIMENSION A(1),B(1),NA(2),NB(2)                                   EQU00030
      NB(1) = NA(1)                                                     EQU00040
      NB(2) =NA(2)                                                      EQU00050
      L=NA(1)*NA(2)                                                     EQU00060
      IF( NA(1) .LT. 1 .OR. L .LT. 1 ) GO TO 999                        EQU00070
      DO 300 I=1,L                                                      EQU00080
  300 B(I)=A(I)                                                         EQU00090
 1000 RETURN                                                            EQU00100
  999 CALL LNCNT (1)                                                    EQU00110
      WRITE  (6,50)  NA                                                 EQU00120
   50 FORMAT  (' DIMENSION ERROR IN EQUATE  NA=',2I6)                   EQU00130
      RETURN                                                            EQU00140
      END                                                               EQU00150
```

```
      SUBROUTINE TRANP(A,NA,B,NB)                                       TRA00010
      IMPLICIT REAL*8 (A-H,O-Z)                                         TRA00020
      DIMENSION A(1),B(1),NA(2),NB(2)                                   TRA00030
      NB(1)=NA(2)                                                       TRA00040
      NB(2)=NA(1)                                                       TRA00050
      NR=NA(1)                                                          TRA00060
      NC=NA(2)                                                          TRA00070
      L=NR*NC                                                           TRA00080
      IF( NR .LT. 1 .OR. L .LT. 1  )  GO TO 999                         TRA00090
      IR=0                                                              TRA00100
      DO 300 I=1,NR                                                     TRA00110
      IJ=I-NR                                                           TRA00120
      DO 300 J=1,NC                                                     TRA00130
      IJ=IJ+NR                                                          TRA00140
      IR=IR+1                                                           TRA00150
  300 B(IR)=A(IJ)                                                       TRA00160
      RETURN                                                            TRA00170
  999 CALL LNCNT(1)                                                     TRA00180
      WRITE  (6,50)  NA                                                 TRA00190
   50 FORMAT  (' DIMENSION ERROR IN TRANP   NA=',2I6)                   TRA00200
      RETURN                                                            TRA00210
      END                                                               TRA00220
```

```
      SUBROUTINE SCALE (A, NA, B, NB, S)                                SCA00010
      IMPLICIT REAL*8 (A-H,O-Z)                                         SCA00020
      DIMENSION A(1),B(1),NA(2),NB(2)                                   SCA00030
      NB(1) = NA(1)                                                     SCA00040
      NB(2) =NA(2)                                                      SCA00050
      L = NA(1)*NA(2)                                                   SCA00060
      IF( NA(1) .LT. 1 .OR. L .LT. 1 ) GO TO 999                        SCA00070
      DO 300 I=1,L                                                      SCA00080
  300 B(I)=A(I)*S                                                       SCA00090
 1000 RETURN                                                            SCA00100
  999 CALL LNCNT(1)                                                     SCA00110
      WRITE  (6,50) NA                                                  SCA00120
   50 FORMAT  (' DIMENSION ERROR IN SCALE   NA=',2I6)                   SCA00130
      RETURN                                                            SCA00140
      END                                                               SCA00150
```

```
      SUBROUTINE UNITY(A,NA)                                            UNI00010
      IMPLICIT REAL*8 (A-H,O-Z)                                         UNI00020
      DIMENSION A(1),NA(2)                                              UNI00030
      IF(NA(1).NE.NA(2)) GO TO 999                                      UNI00040
      L=NA(1)*NA(2)                                                     UNI00050
      DO 100 IT=1,L                                                     UNI00060
  100 A(IT)=0.0                                                         UNI00070
      J = - NA(1)                                                       UNI00080
      NAX = NA(1)                                                       UNI00090
      DO 300 I=1,NAX                                                    UNI00100
      J=NAX +J+1                                                        UNI00110
  300 A(J)=1.                                                           UNI00120
      GO TO 1000                                                        UNI00130
  999 CALL LNCNT (1)                                                    UNI00140
      WRITE(6, 50)(NA(I),I=1,2)                                         UNI00150
   50 FORMAT  (' DIMENSION ERROR IN UNITY  NA=',2I6)                    UNI00160
 1000 RETURN                                                            UNI00170
      END                                                               UNI00180
```

```
      SUBROUTINE NULL(A,NA)                                             NUL00010
      IMPLICIT REAL*8 (A-H,O-Z)                                         NUL00020
      DIMENSION A(1)                                                    NUL00030
      DIMENSION NA(2)                                                   NUL00040
      N=NA(1)*NA(2)                                                     NUL00050
      IF( NA(1) .LT. 1 .OR.  N .LT. 1 )  GO TO 999                      NUL00060
      DO 10I=1,N                                                        NUL00070
   10 A(I) = 0.0                                                        NUL00080
      RETURN                                                            NUL00090
C                                                                       NUL00100
  999 CONTINUE                                                          NUL00110
      WRITE (6,50) NA                                                   NUL00120
   50 FORMAT(' DIMENSION ERROR IN NULL  NA =',2I6)                      NUL00130
      RETURN                                                            NUL00140
      END                                                               NUL00150
```

```
      SUBROUTINE TRCE  (A,NA,TR)                                        TRC00010
      IMPLICIT REAL*8 (A-H,O-Z)                                         TRC00020
      DIMENSION A(1)                                                    TRC00030
      DIMENSION NA(2)                                                   TRC00040
      IF (NA(1).NE.NA(2)) GO TO 600                                     TRC00050
      N=NA(1)                                                           TRC00060
      TR=0.0                                                            TRC00070
      IF( N .LT. 1 ) GO TO 600                                          TRC00080
      DO 10 I=1,N                                                       TRC00090
      M=I+N*(I-1)                                                       TRC00100
   10 TR=TR+A(M)                                                        TRC00110
      RETURN                                                            TRC00120
  600 CALL LNCNT(1)                                                     TRC00130
      WRITE (6,1600) NA                                                 TRC00140
 1600 FORMAT (' TRACE REQUIRES SQUARE MATRIX     NA=',2I6)             TRC00150
      RETURN                                                            TRC00160
      END                                                               TRC00170
```

```
0       SUBROUTINE ADD (A,NA,B,NB,C,NC)                                  ADD00010
        IMPLICIT REAL*8 (A-H,O-Z)                                        ADD00020
        DIMENSION A(1),B(1),C(1),NA(2),NB(2),NC(2)                       ADD00030
3       IF( (NA(1) .NE. NB(1)) .OR. (NA(2) .NE. NB(2)) )  GO TO 999      ADD00040
4       NC(1)=NA(1)                                                      ADD00050
        NC(2)=NA(2)                                                      ADD00060
6       L=NA(1)*NA(2)                                                    ADD00070
7       IF( NA(1) .LT. 1  .OR.  L .LT. 1 )  GO TO 999                    ADD00080
        DO 300 I=1,L                                                     ADD00090
  300 C(I)=A(I)+B(I)                                                     ADD00100
10      GO TO 1000                                                       ADD00110
  999 CALL LNCNT (1)                                                     ADD00120
        WRITE(6,50) NA,NB                                                ADD00130
13     50 FORMAT  (' DIMENSION ERROR IN ADD     NA=',2I6,5X,'NB=',2I6)  ADD00140
14  1000 RETURN                                                          ADD00150
        END                                                              ADD00160
```

```
      SUBROUTINE SUBT(A,NA,B,NB,C,NC)                                   SUB00010
      IMPLICIT REAL*8 (A-H,O-Z)                                         SUB00020
      DIMENSION A(1),B(1),C(1),NA(2),NB(2),NC(2)                        SUB00030
      IF((NA(1).NE.NB(1)).OR.(NA(2).NE.NB(2))) GO TO 999                SUB00040
      NC(1)=NA(1)                                                       SUB00050
      NC(2)=NA(2)                                                       SUB00060
      L=NA(1)*NA(2)                                                     SUB00070
      IF( NA(1) .LT. 1 .OR. L .LT. 1 ) GO TO 999                        SUB00080
      DO 300 I=1,L                                                      SUB00090
  300 C(I)=A(I)-B(I)                                                    SUB00100
      GO TO 1000                                                        SUB00110
  999 CALL LNCNT (1)                                                    SUB00120
      WRITE(6,50) NA,NB                                                 SUB00130
   50 FORMAT  (' DIMENSION ERROR IN SUBT    NA=',2I6,5X,'NB=',2I6)      SUB00140
 1000 RETURN                                                            SUB00150
      END                                                               SUB00160
```

```
0        SUBROUTINE MULT(A,NA,B,NB,C,NC)                                    MUL0001
1        IMPLICIT REAL*8 (A-H,O-Z)                                          MUL0002
2        DIMENSION A(1),B(1),C(1),NA(2),NB(2),NC(2)                         MUL0003
3        NC(1) = NA(1)                                                      MUL0004
4        NC(2) = NB(2)                                                      MUL0005
5        IF(NA(2).NE.NB(1)) GO TO 999                                       MUL0006
6        NAR = NA(1)                                                        MUL0007
7        NAC = NA(2)                                                        MUL0008
8        NBC = NB(2)                                                        MUL0009
9        NAA=NAR*NAC                                                        MUL0010
0        NBB=NAR*NBC                                                        MUL0011
1        IF ( NAR .LT. 1 .OR. NAA .LT. 1 .OR. NBB .LT. 1 ) GO TO 999        MUL0012
12       IR = 0                                                             MUL0013
13       IK=-NAC                                                            MUL0014
14       DO 350 K=1,NBC                                                     MUL0015
15       IK = IK + NAC                                                      MUL0016
16       DO 350 J=1,NAR                                                     MUL0017
17       IR=IR+1                                                            MUL0018
18       IB=IK                                                              MUL0019
19       JI=J-NAR                                                           MUL0020
20       V1=0.0                                                             MUL0021
21       DO 300 I=1,NAC                                                     MUL0022
22       JI = JI + NAR                                                      MUL0023
23       IB=IB+1                                                            MUL0024
24       V3=A(JI)                                                           MUL0025
25       V4=B(IB)                                                           MUL0026
26       V2=V3*V4                                                           MUL0027
27       V1=V1+V2                                                           MUL0028
28   300 CONTINUE                                                           MUL0029
29       C(IR)=V1                                                           MUL0030
30   350 CONTINUE                                                           MUL0031
31       GO TO 1000                                                         MUL0032
32   999 CALL LNCNT (1)                                                     MUL0033
33       WRITE(6,500) (NA(I),I=1,2),(NB(I),I=1,2)                           MUL0034
34   500 FORMAT  (' DIMENSION ERROR IN MULT    NA=',2I6,5X,'NB=',2I6)       MUL0035
35  1000 RETURN                                                             MUL0036
36       END                                                                MUL0037
```

```
0       SUBROUTINE MAXEL(A,NA,ELMAX)                                    MAX0001
1       IMPLICIT REAL*8 (A-H,O-Z)                                       MAX0002
2       DIMENSION A(1),NA(2)                                            MAX0003
3 C                                                                     MAX0004
4       N = NA(1)*NA(2)                                                 MAX0005
5 C                                                                     MAX0006
6       ELMAX = DABS( A(1))                                             MAX0007
7       DO 100 I = 2,N                                                  MAX0008
8       ELMAXI = DABS( A(I) )                                           MAX0009
9       IF( ELMAXI.GT.ELMAX)  ELMAX=ELMAXI
10    100 CONTINUE                                                      MAX0011
11 C                                                                    MAX0012
12      RETURN                                                          MAX0013
13      END                                                             MAX0014
```

```
      SUBROUTINE NORMS(MAXROW,M,N,A,IOPT,RLNORM)                        NOR0001
      IMPLICIT REAL*8 (A-H,O-Z)                                         NOR0002
      DIMENSION A(1)                                                    NOR0003
C                                                                       NOR0004
C  INITIALIZATION                                                       NOR0005
C                                                                       NOR0006
      RLNORM=0.                                                         NOR0007
      SUM=0.                                                            NOR0008
      I=-MAXROW                                                         NOR0009
C                                                                       NOR0010
C  TRANSFER TO APPROPRIATE LOOP TO COMPUTE THE DESIRED NORM             NOR0011
C                                                                       NOR0012
      IF(IOPT-2)5,20,30                                                 NOR0013
C                                                                       NOR0014
C  THIS LOOP COMPUTES THE ONE-NORM                                      NOR0015
C                                                                       NOR0016
    5 DO 15 K=1,N                                                       NOR0017
      I=I+MAXROW                                                        NOR0018
      DO 10 J=1,M                                                       NOR0019
      L=I+J                                                             NOR0020
   10 SUM=DABS(A(L))+SUM                                                NOR0021
      IF(SUM.GT.RLNORM)RLNORM=SUM                                       NOR0022
   15 SUM=0.                                                            NOR0023
      RETURN                                                            NOR0024
C                                                                       NOR0025
C  THIS LOOP COMPUTES THE EUCLIDEAN NORM                                NOR0026
C                                                                       NOR0027
   20 DO 25 K=1,N                                                       NOR0028
      I=I+MAXROW                                                        NOR0029
      DO 25 J=1,M                                                       NOR0030
      L=I+J                                                             NOR0031
      SUM=A(L)                                                          NOR0032
   25 RLNORM=SUM*SUM+RLNORM                                             NOR0033
      RLNORM=DSQRT(RLNORM)                                              NOR0034
      RETURN                                                            NOR0035
C                                                                       NOR0036
C  THIS LOOP COMPUTES THE INFINITY-NORM                                 NOR0037
C                                                                       NOR0038
   30 DO 40 J=1,M                                                       NOR0039
      L=I+J                                                             NOR0040
      DO 35 K=1,N                                                       NOR0041
      L=L+MAXROW                                                        NOR0042
   35 SUM=DABS(A(L))+SUM                                                NOR0043
      IF(SUM.GT.RLNORM)RLNORM=SUM                                       NOR0044
   40 SUM=0.0                                                           
      RETURN                                                            NOR0045
      END                                                               NOR0046
```

```
      SUBROUTINE JUXTC(A,NA,B,NB,C,NC)                                  JUX0001
      IMPLICIT REAL*8 (A-H,O-Z)                                         JUX0002
      DIMENSION A(1),B(1),C(1),NA(2),NB(2),NC(2)                        JUX0003
      IF (NA(1).NE.NB(1)) GO TO 600                                     JUX0004
      NC(1)=NA(1)                                                       JUX0005
      NC(2)=NA(2)+NB(2)                                                 JUX0006
      L=NA(1)*NA(2)                                                     JUX0007
      NNC=NC(1)*NC(2)                                                   JUX0008
      IF( NA(1) .LT. 1 .OR. L .LT. 1 ) GO TO 600                        JUX0009
      IF( NC(2) .LT. 1 ) GO TO 600                                      JUX0010
      MS=NA(1)*NA(2)                                                    JUX0011
      DO 10 I=1,MS                                                      JUX0012
   10 C(I)=A(I)                                                         JUX0013
      MBS=NA(1)*NB(2)                                                   JUX0014
      DO 20 I=1,MBS                                                     JUX0015
      J=MS+I                                                            JUX0016
   20 C(J)=B(I)                                                         JUX0017
      RETURN                                                            JUX0018
  600 CALL LNCNT(1)                                                     JUX0019
      WRITE (6,1600) NA,NB                                              JUX0020
 1600 FORMAT (' DIMENSION ERROR IN JUXTC,  NA=',2I6,5X,'NB=',2I6)       JUX0021
      RETURN                                                            JUX0022
      END                                                               JUX0023
```

```
0       SUBROUTINE JUXTR(A,NA,B,NB,C,NC)                                JUX0001
1       IMPLICIT REAL*8 (A-H,O-Z)                                       JUX0002
2       DIMENSION A(1),B(1),C(1),NA(2),NB(2),NC(2)                      JUX0003
3       IF(NA(2).NE.NB(2))GO TO 600                                     JUX0004
4       NC(2)=NA(2)                                                     JUX0005
5       NC(1)=NA(1)+NB(1)                                               JUX0006
6       L=NA(1)*NA(2)                                                   JUX0007
7       IF( NA(1) .LT. 1 .OR. L .LT. 1 ) GO TO 600                      JUX0008
8       IF( NC(2) .LT. 1 ) GO TO 600                                    JUX0009
9       MCA=NA(2)                                                       JUX0010
10      MRA=NA(1)                                                       JUX0011
11      MRB=NB(1)                                                       JUX0012
12      MRC=NC(1)                                                       JUX0013
13      DO 10 I=1,MCA                                                   JUX0014
14      DO 10 J=1,MRA                                                   JUX0015
15      K=J+MRA*(I-1)                                                   JUX0016
16      L=J+MRC*(I-1)                                                   JUX0017
17   10 C(L)=A(K)                                                       JUX0018
18      DO 20 I=1,MCA                                                   JUX0019
19      DO 20 J=1,MRB                                                   JUX0020
20      K=J+MRB*(I-1)                                                   JUX0021
21      L=MRA+J+MRC*(I-1)                                               JUX0022
22   20 C(L)=B(K)                                                       JUX0023
23      RETURN                                                          JUX0024
24  600 CALL LNCNT(1)                                                   JUX0025
25      WRITE(6,1600) NA,NB                                             JUX0026
26 1600 FORMAT(' DIMENSION ERROR IN JUXTR,  NA=',2I6,5X,'NB=',2I6)      JUX0027
27      RETURN                                                          JUX0028
28      END                                                             JUX0029
```

```
      SUBROUTINE FACTOR(Q,NQ,D,ND,IOP,IAC,DUMMY)                        FAC0001
      IMPLICIT REAL*8 (A-H,O-Z)                                         FAC0002
      DIMENSION Q(1),D(1),DUMMY(1)                                      FAC0003
      DIMENSION NQ(2),ND(2),NDUM(2)                                     FAC0004
      IOPT = 2                                                          FAC0005
      N = NQ(1)                                                         FAC0006
      M = N**2                                                          FAC0007
      N1 = M + 1                                                        FAC0008
      N2 = N1 + N                                                       FAC0009
C                                                                       FAC0010
      CALL EQUATE(Q,NQ,DUMMY,NQ)                                        FAC0011
      CALL SNVDEC(IOPT,N,N,N,N,DUMMY,NOS,B,IAC,ZTEST,DUMMY(N1),D,IRANK,AFAC0012
     1PLUS,IERR)                                                        FAC0013
      IF( IERR .EQ. 0 ) GO TO 200                                       FAC0014
      CALL LNCNT(5)                                                     FAC0015
      IF( IERR .GT. 0 ) PRINT 100,IERR                                  FAC0016
      IF( IERR .EQ. -1) PRINT 150,ZTEST,IRANK                           FAC0017
  100 FORMAT(//,' IN FACTOR , SNVDEC HAS FAILED TO CONVERGE TO THE ',I4,FAC0018
     1' SINGULAR VALUE AFTER 30 ITERATIONS')                            FAC0019
  150 FORMAT(//,' IN FACTOR, THE MATRIX Q SUBMITTED TO SNVDEC IS CLOSE TFAC0020
     1O A MATRIX OF LOWER RANK USING ZTEST = ',D16.8,/,' IF THE ACCURACYFAC0021
     2 IS REDUCED  THE RANK MAY ALSO BE REDUCED',/,' CURRENT RANK =',I4)FAC0022
      NDUM(1)=N                                                         FAC0023
      NDUM(2)=1                                                         FAC0024
      IF(IERR .EQ. -1)  CALL PRNT(DUMMY(N1),NDUM,4HSNVL,1)              FAC0025
      IF( IERR .GT. 0 ) RETURN                                          FAC0026
C                                                                       FAC0027
  200 CONTINUE                                                          FAC0028
      NDUM(1) = N                                                       FAC0029
C                                                                       FAC0030
      DO 250 J =1,N                                                     FAC0031
      M1 = (J-1)*N + 1                                                  FAC0032
      M2 = J*N                                                          FAC0033
      DO 250 I =M1,M2                                                   FAC0034
      K = N2+I-1                                                        FAC0035
      L = N1+J-1                                                        FAC0036
      IF( DUMMY(L) .EQ. 0.0) GO TO 300                                  FAC0037
      DUMMY(K) = DSQRT(DUMMY(L))*DUMMY(I)                               FAC0038
  250 CONTINUE                                                          FAC0039
      NDUM(2)=N                                                         FAC0040
      GO TO 350                                                         FAC0041
C                                                                       FAC0042
  300 NDUM(2) = J - 1                                                   FAC0043
  350 CONTINUE                                                          FAC0044
      IF( DUMMY(N2) .LT. 0.0 ) CALL SCALE(DUMMY(N2),NDUM,DUMMY(N2),NDUM,FAC0045
     1-1.0)                                                             FAC0046
      CALL TRANP(DUMMY(N2),NDUM,D,ND)                                   FAC0047
C                                                                       FAC0048
      IF( IOP .EQ. 0 ) RETURN                                           FAC0049
      CALL LNCNT(4)                                                     FAC0050
      PRINT 400                                                         FAC0051
  400 FORMAT(//,' FACTOR Q AS (D TRANSPOSE)XD ',/)                      FAC0052
      CALL PRNT(Q,NQ,4H Q  ,1)                                          FAC0053
      CALL PRNT(D,ND,4H D  ,1)                                          FAC0054
      CALL MULT(DUMMY(N2),NDUM,D,ND,DUMMY,NQ)                           FAC0055
      CALL PRNT(DUMMY,NQ,4HDTXD,1)                                      FAC0056
C                                                                       FAC0057
      RETURN                                                            FAC0058
      END                                                               FAC0059
```

```
      SUBROUTINE EIGEN(MAX, N, A, ER, EI, ISV, ILV, V, WK, IERR)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(MAX,N),ER(N),EI(N),V(MAX,1),WK(N,1)
      INTEGER INT(20)
      LOGICAL*1 SELECT(25)
C
C     PRELIMINARY REDUCTION
C
      CALL BALANC (MAX,N,A,LOW,IGH,WK)
      CALL ELMHES (MAX,N,LOW,IGH,A,INT(1))
      IV = ISV + ILV
      IF (IV .NE. 0) GO TO 10
C
C     COMPUTE ALL EIGENVALUES AND NO EIGENVECTORS
C
      CALL HQR (MAX,N,LOW,IGH,A,ER,EI,IERR)
      IF (IERR .NE. 0) GO TO 260
      DO 5 I=1,N
         WK(I,1) = ER(I)
         WK(I,2) = EI(I)
         WK(I,3) = ER(I)**2 + EI(I)**2
    5 CONTINUE
      IC = 0
      GO TO 190
   10 CONTINUE
C
C     SAVE A MATRIX FOR INVERSE ITERATION AND INITIALIZE WK(I,4)
C       ARRAY WHICH WILL BE A LOGICAL ARRAY IN CALLED SUBROUTINES
C
      DO 20 I=1,N
      SELECT(I)=.FALSE.
         JS = 1
         IF (I .GE. 3) JS = I-1
         DO 20 J=JS,N
            WK(I,J+5) = A(I,J)
   20 CONTINUE
C
C     COMPUTE ALL EIGENVALUES (UNORDERED)
C
      CALL HQR (N,N,LOW,IGH,WK(1,6),ER,EI,IERR)
      IF (IERR .NE. 0) GO TO 260
      DO 30 I=1,N
         WK(I,3) = ER(I)**2 + EI(I)**2
   30 CONTINUE
      IF (ILV .EQ. 0) GO TO 60
C
C     FIND LARGEST ILV EIGENVALUES AND FLAG THEM
C
      DO 50 I=1,ILV
         P = -1.D0
         DO 40 J=1,N
            IF (WK(J,3) .LE. P) GO TO 40
            K = J
            P = WK(J,3)
   40    CONTINUE
      SELECT(K)=.TRUE.
         WK(K,3) = -WK(K,3)
   50 CONTINUE
      IF (EI(K) .EQ. 0.) GO TO 60
      IF (EI(K) .GT. 0.) GO TO 55
      IF (SELECT(K-1))GO TO 60
      ILV = ILV+1
      SELECT(K-1)=.TRUE.
```

```
      GO TO 60                                                          EIG006
   55 CONTINUE                                                          EIG006
      IF (.NOT.SELECT(K+1)) ILV = ILV+1                                 EIG006
   60 CONTINUE                                                          EIG006
      IF (ISV .EQ. 0) GO TO 90                                          EIG006
C                                                                       EIG006
C     FIND SMALLEST ISV EIGENVALUES AND FLAG THEM                       EIG007
C                                                                       EIG007
      DO 65 I=1,N                                                       EIG007
         WK(I,3) = DABS(WK(I,3))                                        EIG007
   65 CONTINUE                                                          EIG007
      DO 80 I=1,ISV                                                     EIG007
         P = 1.D74                                                      EIG007
         DO 70 J=1,N                                                    EIG007
            IF (WK(J,3) .GE. P) GO TO 70                                EIG007
            K = J                                                       EIG007
            P = WK(J,3)                                                 EIG007
   70    CONTINUE                                                       EIG007
      SELECT(K)=.TRUE.                                                  EIG007
         WK(K,3) = 1.D74                                                EIG007
   80 CONTINUE                                                          EIG007
      IF (EI(K) .EQ. 0.) GO TO 90                                       EIG007
      IF (EI(K) .GT. 0.) GO TO 85                                       EIG007
      IF (SELECT(K-1)) GO TO 90                                         EIG007
      ISV = ISV+1                                                       EIG007
      SELECT(K-1)=.TRUE.                                                EIG007
      GO TO 90                                                          EIG007
   85 CONTINUE                                                          EIG007
      IF (.NOT.SELECT(K+1)) ISV = ISV+1                                 EIG007
   90 CONTINUE                                                          EIG007
C                                                                       EIG007
C     FIND EIGENVECTORS FOR FLAGGED EIGENVALUES                         EIG007
C                                                                       EIG007
      CALL INVIT (MAX,N,A,ER,EI,SELECT,N,M,V,IERR,WK(1,6),WK(1,3),      EIG007
     1            WK(1,5))                                              EIG007
C                                                                       EIG007
C     BACK TRANSFORM EIGENVECTORS TO ORIGINAL MATRIX                    EIG008
C                                                                       EIG008
      CALL ELMBAK (MAX,LOW,IGH,A,INT(1) ,M,V)                           EIG008
      CALL BALBAK (MAX,N,LOW,IGH,WK,M,V)                                EIG008
C                                                                       EIG008
C     SEPARATE FLAGGED EIGENVALUES FROM UNFLAGGED EIGENVALUES           EIG008
C                                                                       EIG008
      IV = ISV + ILV                                                    EIG008
      IF (IV .LE. N) GO TO 100                                          EIG008
      ILV = N-ISV                                                       EIG008
      IV = N                                                            EIG008
  100 CONTINUE                                                          EIG009
      IC = 0                                                            EIG009
      JC = IV                                                           EIG009
      DO 150 I=1,N                                                      EIG009
         IF (SELECT(I)) GO TO 120                                       EIG009
         IF (EI(I) .GE. 0.) GO TO 110                                   EIG009
         IF (SELECT(I-1)) GO TO 120                                     EIG009
  110    CONTINUE                                                       EIG009
         JC = JC+1                                                      EIG009
         WK(JC,1) = ER(I)                                               EIG010
         WK(JC,2) = EI(I)                                               EIG010
         KC = JC                                                        EIG010
         GO TO 130                                                      EIG010
  120    CONTINUE                                                       EIG010
         IC = IC+1                                                      EIG010
         WK(IC,1) = ER(I)                                               EIG010
```

```
126            WK(IC,2) = EI(I)                                          EIG012
127            KC = IC                                                   EIG012
128     130    CONTINUE                                                  EIG012
129            WK(KC,3) = ER(I)**2 + EI(I)**2                            EIG013
130     150 CONTINUE                                                     EIG013
131 C                                                                    EIG013
132 C       NORMALIZE VECTORS TO UNIT LENGTH AND STORE FOR REORDERING    EIG013
133 C                                                                    EIG013
134         J = 0                                                        EIG013
135     151 CONTINUE                                                     EIG013
136         J = J+1                                                      EIG013
137         IF (WK(J,2) .NE. 0.) GO TO 154                               EIG013
138         SUM = 0.                                                     EIG013
139         DO 152 I=1,N                                                 EIG014
140            SUM = SUM + V(I,J)**2                                     EIG014
141     152 CONTINUE                                                     EIG014
142         IF (SUM .EQ. 0.) GO TO 158                                   EIG014
143         SUM = DSQRT(SUM)                                             EIG014
144         DO 153 I=1,N                                                 EIG014
145            WK(I,J+4) = V(I,J)/SUM                                    EIG014
146     153 CONTINUE                                                     EIG014
147         GO TO 158                                                    EIG014
148     154 CONTINUE                                                     EIG014
149         JP1 = J+1                                                    EIG015
150         SUM = 0.                                                     EIG015
151         DO 155 I=1,N                                                 EIG015
152            SUM = SUM + V(I,J)**2 + V(I,JP1)**2                       EIG015
153     155 CONTINUE                                                     EIG015
154         IF (SUM .EQ. 0.) GO TO 157                                   EIG015
155         SUM = DSQRT(SUM)                                             EIG015
156         DO 156 I=1,N                                                 EIG015
157            WK(I,J+4) = V(I,J)/SUM                                    EIG015
158            WK(I,J+5) = V(I,JP1)/SUM                                  EIG015
159     156 CONTINUE                                                     EIG016
160     157 CONTINUE                                                     EIG016
161         J = JP1                                                      EIG016
162     158 CONTINUE                                                     EIG016
163         IF (J .LT. IV) GO TO 151                                     EIG016
164         IC = 0                                                       EIG016
165         LC = 0                                                       EIG016
166         IF (ISV .EQ. 0) GO TO 190                                    EIG016
167 C                                                                    EIG016
168 C       ORDER SMALLEST ISV EIGENVALUES AND EIGENVECTORS FOR OUTPUT    EIG016
169 C                                                                    EIG017
170         DO 190 I=1,ISV                                               EIG017
171            P = 1.D74                                                 EIG017
172            DO 160 J=1,IV                                             EIG017
173               IF (WK(J,3) .GE. P) GO TO 160                          EIG017
174               K = J                                                  EIG017
175               P = WK(J,3)                                            EIG017
176     160       CONTINUE                                               EIG017
177               IC = IC+1                                              EIG017
178               LC = LC+1                                              EIG017
179               ER(IC) = WK(K,1)                                       EIG018
180               EI(IC) = WK(K,2)                                       EIG018
181               DO 170 J=1,N                                           EIG018
182                  V(J,LC) = WK(J,K+4)                                 EIG018
183     170       CONTINUE                                               EIG018
184               WK(K,3) = 1.D74                                        EIG018
185     180    CONTINUE                                                  EIG018
186     190    CONTINUE                                                  EIG018
187            IF (IV .EQ. N) GO TO 220                                  EIG018
188 C                                                                    EIG018
```

```
C     ORDER UNFLAGGED EIGENVALUES FOR OUTPUT                            EIG0190
C                                                                       EIG0191
      IV1 = IV+1                                                        EIG0192
      IUF = N - IV                                                      EIG0193
      DO 210 I=1,IUF                                                    EIG0194
         P = 1.D74                                                      EIG0195
         DO 200 J=IV1,N                                                 EIG0196
            IF (WK(J,3) .GE. P) GO TO 200                               EIG0197
            K = J                                                       EIG0198
            P = WK(J,3)                                                 EIG0199
  200    CONTINUE                                                       EIG0200
         IC = IC+1                                                      EIG0201
         ER(IC) = WK(K,1)                                               EIG0202
         EI(IC) = WK(K,2)                                               EIG0203
         WK(K,3) = 1.D74                                                EIG0204
  210 CONTINUE                                                          EIG0205
  220 CONTINUE                                                          EIG0206
      IF (ILV .EQ. 0) GO TO 260                                         EIG0207
C                                                                       EIG0208
C     ORDER LARGEST ILV EIGENVALUES AND EIGENVECTORS FOR OUTPUT         EIG0209
C                                                                       EIG0210
      DO 250 I=1,ILV                                                    EIG0211
         P = 1.D74                                                      EIG0212
         DO 230 J=1,IV                                                  EIG0213
            IF (WK(J,3) .GE. P) GO TO 230                               EIG0214
            K = J                                                       EIG0215
            P = WK(J,3)                                                 EIG0216
  230    CONTINUE                                                       EIG0217
         IC = IC+1                                                      EIG0218
         LC = LC+1                                                      EIG0219
         ER(IC) = WK(K,1)                                               EIG0220
         EI(IC) = WK(K,2)                                               EIG0221
         DO 240 J=1,N                                                   EIG0222
            V(J,LC) = WK(J,K+4)                                         EIG0223
  240    CONTINUE                                                       EIG0224
         WK(K,3) = 1.D74                                                EIG0225
  250 CONTINUE                                                          EIG0226
  260 CONTINUE                                                          EIG0227
      RETURN                                                            EIG0228
      END                                                               EIG0229
```

```
      SUBROUTINE SYMPOS (MAXN,N,A,NRHS,B,IOPT,IFAC,DETERM,ISCALE,P,IERR)SYM000
      IMPLICIT REAL*8 (A-H,O-Z)                                       SYM000
      DIMENSION A(MAXN,N),B(MAXN,NRHS),P(N)                           SYM000
C                                                                     SYM000
      DATA R1,R2/1.00+75,1.00-75/                                     SYM000
C                                                                     SYM000
C               TEST FOR A SCALAR MATRIX (IF COEFFICIENT MATRIX IS A  SYM000
C               SCALAR-- SOLVE  AND COMPUTE DETERMINANT IF DESIRED)   SYM000
      IERR = 0                                                        SYM000
      NM1 = N-1                                                       SYM001
      IF (NM1 .GT. 0) GO TO 20                                        SYM001
C                                                                     SYM001
      IF( A(1,1) .LE. 0.0 )  IERR = 1                                 SYM001
      ISCALE = 0                                                      SYM001
      DETERM = A(1,1)                                                 SYM001
      P(1) = 1.0/A(1,1)                                               SYM001
      DO 10 J=1,NRHS                                                  SYM001
      B(1,J) = B(1,J)/DETERM                                          SYM001
   10 CONTINUE                                                        SYM001
      RETURN                                                          SYM002
C                                                                     SYM002
C               TEST TO DETERMINE IF CHOLESKY DECOMPOSITION OF COEFFICIENTSYM002
C               MATRIX IS DESIRED                                     SYM002
   20 IF (IFAC .EQ. 1) GO TO 160                                      SYM002
C                                                                     SYM002
C               INITIALIZE DETERMINANT EVALUATION PARAMETERS          SYM002
      DETERM=1.0                                                      SYM002
      ISCALE=0                                                        SYM002
C                                                                     SYM002
C                'LOOP' TO PERFORM CHOLESKY DECOMPOSTION ON THE COEF- SYM003
C               FICIENT MATRIX A (I.E. MATRIX A WILL BE DECOMPOSED INTO SYM003
C               THE PRODUCT OF A UNIT LOWER TRIANGULAR MATRIX (L), A  SYM003
C               DIAGONAL MATRIX (D), AND THE TRANSPOSE OF L (LTRANSPOSE).)SYM003
C                                                                     SYM003
   30 DO 150 I=1,N                                                    SYM003
      IM1 = I-1                                                       SYM003
C                                                                     SYM003
      DO 150 J=1,I                                                    SYM003
      X = A(J,I)                                                      SYM003
C                                                                     SYM004
C               DETERMINE IF ELEMENTS ARE ABOVE OR BELOW THE DIAGONAL SYM004
      IF (I .GT. J) GO TO 110                                         SYM004
C                                                                     SYM004
C               USING THE DIAGONAL ELEMENTS OF MATRIX A, THIS SECTION SYM004
C               COMPUTES DIAGONAL MATRIX AND DETERMINES IF MATRIX A IS SYM004
C               SYMMETRIC POSITIVE DEFINITE                           SYM004
      IF (IM1 .EQ. 0) GO TO 50                                        SYM004
      DO 40 K=1,IM1                                                   SYM004
      Y = A(I,K)                                                      SYM004
      A(I,K) = Y*P(K)                                                 SYM005
      X = X - Y*A(I,K)                                                SYM005
   40 CONTINUE                                                        SYM005
C                                                                     SYM005
C               TEST IF COEFFICIENT MATRIX IS POSITIVE DEFINITE       SYM005
   50 IF( X .LE. 0.0 )  IERR = 1                                      SYM005
C                                                                     SYM005
C               COMPUTE INVERSE OF DIAGONAL MATRIX D**-1 = 1/P        SYM005
      P(I) = 1.0 / X                                                  SYM005
C                                                                     SYM005
C               TEST TO SEE IF DETERMINANT IS TO BE EVALUATED         SYM006
      IF (IOPT .EQ. 0) GO TO 150                                      SYM006
C                                                                     SYM006
C                                                                     SYM006
```

```
C                SCALE THE DETERMINANT (COMPUTE THE DETERMINANT EVALUATION SYM0064
C                PARAMETERS DETERM AND ISCALE)                             SYM0065
      PIVOTI=X                                                             SYM0066
   60 IF(DABS(DETERM).LT.R1) GO TO 70                                      SYM0067
      DETERM = DETERM*R2                                                   SYM0068
      ISCALE = ISCALE+1                                                    SYM0069
      GO TO 60                                                             SYM0070
   70 IF(DABS(DETERM).GT.R2) GO TO 80                                      SYM0071
      DETERM = DETERM*R1                                                   SYM0072
      ISCALE = ISCALE-1                                                    SYM0073
      GO TO 70                                                             SYM0074
   80 IF(DABS(PIVOTI).LT.R1) GO TO 90                                      SYM0075
      PIVOTI = PIVOTI*R2                                                   SYM0076
      ISCALE = ISCALE+1                                                    SYM0077
      GO TO 80                                                             SYM0078
   90 IF(DABS(PIVOTI).GT.R2) GO TO 100                                     SYM0079
      PIVOTI = PIVOTI*R1                                                   SYM0080
      ISCALE = ISCALE-1                                                    SYM0081
      GO TO 90                                                             SYM0082
  100 DETERM = DETERM*PIVOTI                                               SYM0083
      GO TO 150                                                            SYM0084
C                                                                          SYM0085
C                                                                          SYM0086
C                USING THE LOWER TRIANGULAR ELEMENTS OF MATRIX A, THIS     SYM0087
C                SECTION COMPUTES THE UNIT LOWER TRIANGULAR MATRIX         SYM0088
  110 JM1 = J-1                                                            SYM0089
      IF (JM1 .EQ. 0) GO TO 140                                            SYM0090
      DO 120 K=1,JM1                                                       SYM0091
      X = X - A(I,K)*A(J,K)                                                SYM0092
  120 CONTINUE                                                             SYM0093
C                                                                          SYM0094
  140 A(I,J) = X                                                           SYM0095
C                                                                          SYM0096
  150 CONTINUE                                                             SYM0097
C                                                                          SYM0098
C                SECTION TO APPLY BACK SUBSTITUTION TO SOLVE L*Y = B FOR   SYM0099
C                UNIT LOWER TRIANGULAR MATRIX AND CONSTANT COLUMN VECTOR B SYM0100
C                                                                          SYM0101
  160 IF( IFAC .EQ. 2 )  RETURN                                            SYM0102
      DO 180 I=2,N                                                         SYM0103
      IM1=I-1                                                              SYM0104
C                                                                          SYM0105
      DO 180 J=1,NRHS                                                      SYM0106
      X = B(I,J)                                                           SYM0107
C                                                                          SYM0108
      DO 170 K=1,IM1                                                       SYM0109
      X = X - A(I,K)*B(K,J)                                                SYM0110
  170 CONTINUE                                                             SYM0111
C                                                                          SYM0112
      B(I,J) = X                                                           SYM0113
  180 CONTINUE                                                             SYM0114
C                                                                          SYM0115
C                SECTION TO SOLVE (LTRANSPOSE)*X = (D**-1)*Y FOR TRANSPOSE SYM0116
C                OF UNIT LOWER TRIANGULAR MATRIX AND INVERSE OF DIAGONAL   SYM0117
C                MATRIX                                                    SYM0118
C                                                                          SYM0119
      Y = B(N)                                                             SYM0120
      DO 190 J=1,NRHS                                                      SYM0121
      B(N,J) = B(N,J)*Y                                                    SYM0122
  190 CONTINUE                                                             SYM0123
C                                                                          SYM0124
  200 I = NM1+1                                                            SYM0125
      Y = B(NM1)                                                           SYM0126
```

```
126 C                                                                   SYM012
127       DO 220 J=1,NRHS                                               SYM012
128       X = B(NM1,J)*Y                                                SYM012
129 C                                                                   SYM013
130       DO 210 K=I,N                                                  SYM013
131       X = X - A(K,NM1)*B(K,J)                                       SYM013
132   210 CONTINUE                                                      SYM013
133 C                                                                   SYM013
134       B(NM1,J) = X                                                  SYM013
135   220 CONTINUE                                                      SYM013
136 C                                                                   SYM013
137 C                                                                   SYM013
138 C             TEST TO DETERMINE IF SOLUTIONS HAVE BEEN DETERMINED FOR SYM013
139 C             ALL COLUMN VECTORS                                    SYM014
140       NM1 = NM1-1                                                   SYM014
141       IF (NM1 .GT. 0) GO TO 200                                     SYM014
142 C                                                                   SYM014
143       RETURN                                                        SYM014
144       END                                                           SYM014
```

```
      SUBROUTINE GELIM(NMAX,N,A,NRHS,B,IPIVOT,IFAC,WK,IERR)                GEL000
      IMPLICIT REAL*8 (A-H,O-Z)                                            GEL000
      DIMENSION A(NMAX,1),B(NMAX,1),IPIVOT(1),WK(1)                        GEL000
      IERR=0                                                               GEL000
C                                                                          GEL000
C     TEST FOR L/U FACTORIZATION                                           GEL000
C                                                                          GEL000
      IF(IFAC.EQ.1)GO TO 10                                                GEL000
      CALL DETFAC(NMAX,N,A,IPIVOT,IFAC,DETERM,ISCALE,WK,IERR)              GEL000
      IF(IERR.GT.0)RETURN                                                  GEL001
      IF (IFAC.EQ.2) DETA=DETERM*(10.**(100*ISCALE))
   10 NM1=N-1                                                              GEL001
C                                                                          GEL001
C     TEST FOR SCALAR A MATRIX                                             GEL001
C                                                                          GEL001
      IF(NM1.GT.0)GO TO 40                                                 GEL001
      IF(A(1,1).EQ.0.)GO TO 30                                             GEL001
      DO 20 I=1,NRHS                                                       GEL001
   20 B(1,I)=B(1,I)/A(1,1)                                                 GEL001
      IF (IFAC.EQ.2) WK(1)=DETA
      RETURN                                                               GEL001
   30 IERR=1                                                               GEL002
      RETURN                                                               GEL002
C                                                                          GEL
   40 DO 100 M=1,NRHS                                                      GEL0
C                                                                          GEL00
C     PIVOT THE M-TH COLUMN OF B MATRIX                                    GEL002
C                                                                          GEL002
      DO 50 I=1,NM1                                                        GEL002
      KI=IPIVOT(I)                                                         GEL002
      P=B(KI,M)                                                            GEL002
      B(KI,M)=B(I,M)                                                       GEL003
   50 B(I,M)=P                                                             GEL003
C                                                                          GEL003
C     FORWARD SUBSTITUTION                                                 GEL003
C                                                                          GEL003
      WK(1)=B(1,M)                                                         GEL003
C                                                                          GEL003
      DO 70 I=2,N                                                          GEL003
      IM1=I-1                                                              GEL003
      P=0.0                                                                GEL003
      DO 60 K=1,IM1                                                        GEL004
   60 P=P+A(I,K)*WK(K)                                                     GEL004
   70 WK(I)=B(I,M)-P                                                       GEL004
C                                                                          GEL004
C     BACK SUBSTITUTION                                                    GEL004
C                                                                          GEL004
      B(N,M)=WK(N)/A(N,N)                                                  GEL004
C                                                                          GEL004
      DO 90 J=1,NM1                                                        GEL004
      I=N-J                                                                GEL004
      IP1=I+1                                                              GEL005
      P=WK(I)                                                              GEL005
      DO 80 K=IP1,N                                                        GEL005
   80 P=P-A(I,K)*B(K,M)                                                    GEL005
   90 B(I,M)=P/A(I,I)                                                      GEL005
C                                                                          GEL005
  100 CONTINUE                                                             GEL005
      IF (IFAC.EQ.2) WK(1)=DETA
      RETURN                                                               GEL005
      END                                                                  GEL005
```

```
      SUBROUTINE SNVDEC(IOP,MD,ND,M,N,A,NOS,B,IAC,ZTEST,Q,V,IRANK,APLUS,SNV000
     1IERR)                                                             SNV000
      IMPLICIT REAL*8 (A-H,O-Z)                                         SNV000
      LOGICAL WITHU,WITHV                                               SNV000
      DIMENSION A(MD,N),V(ND,N),Q(N),E(150)                             SNV000
      DIMENSION B(MD,NOS),APLUS(ND,M)                                   SNV000
C                                                                       SNV000
C                                                                       SNV000
C     TEST FOR  SCALAR OR VECTOR A                                      SNV000
C                                                                       SNV001
      IF( N .GE. 2 ) GO TO 3000                                         SNV001
C                                                                       SNV001
      IERR = 0                                                          SNV001
      ZTEST = 10.**(-IAC)                                               SNV001
      SUM = 0.0                                                         SNV001
      DO 1000 I=1,M                                                     SNV001
      SUM = SUM + A(I,1)*A(I,1)                                         SNV001
 1000 CONTINUE                                                          SNV001
      SUM = DSQRT(SUM)                                                  SNV001
      IRANK = 0                                                         SNV002
      IF( SUM .GT. ZTEST ) IRANK = 1                                    SNV002
      Q(1) = SUM                                                        SNV002
C                                                                       SNV002
      IF( IOP .EQ. 1) RETURN                                            SNV002
      V(1,1) = 1.0                                                      SNV002
      IF( IRANK .EQ. 0 ) GO TO 1200                                     SNV002
      DO 1100 I =1,M                                                    SNV002
      A(I,1) = A(I,1)/SUM                                               SNV002
 1100 CONTINUE                                                          SNV002
      GO TO 1300                                                        SNV003
 1200 CONTINUE                                                          SNV003
      A(1,1) = 1.0                                                      SNV003
 1300 CONTINUE                                                          SNV003
C                                                                       SNV003
      IF( IOP .EQ. 2 ) RETURN                                           SNV003
      IF( IOP .EQ. 4 ) GO TO 1850                                       SNV003
      IF( IRANK .EQ. 0 ) GO TO 1600                                     SNV003
      DO 1500 J = 1,NOS                                                 SNV003
      Z = 0                                                             SNV003
      DO 1400 I = 1,M                                                   SNV004
      Z = Z + A(I,1)*B(I,J)/SUM                                         SNV004
 1400 CONTINUE                                                          SNV004
      B(1,J) = Z                                                        SNV004
 1500 CONTINUE                                                          SNV004
      GO TO 1800                                                        SNV004
 1600 CONTINUE                                                          SNV004
      DO 1700 J =1,NOS                                                  SNV004
      B(1,J) = 0.0                                                      SNV004
 1700 CONTINUE                                                          SNV004
 1800 CONTINUE                                                          SNV005
C                                                                       SNV005
      IF( IOP .EQ. 3 ) RETURN                                           SNV005
 1850 CONTINUE                                                          SNV005
      IF( IRANK .EQ. 0 ) GO TO 2000                                     SNV005
      DO 1900 I =1,M                                                    SNV005
      APLUS(1,I) = A(I,1)/SUM                                           SNV005
 1900 CONTINUE                                                          SNV005
      RETURN                                                            SNV005
 2000 CONTINUE                                                          SNV005
      DO 2100 I=1,M                                                     SNV006
      APLUS(1,I) = 0.0                                                  SNV006
 2100 CONTINUE                                                          SNV006
      RETURN                                                            SNV006
```

```
C                                                                       SNV006
C                                                                       SNV006
 3000 CONTINUE                                                          SNV006
C                                                                       SNV006
C                                                                       SNV006
C                                                                       SNV006
      TOL=1.0D-60                                                       SNV007
      SIZE=0.0                                                          SNV007
      NP1=N+1                                                           SNV007
C                                                                       SNV007
C     COMPUTE THE E-NORM OF MATRIX A AS ZERO TEST FOR SINGULAR VALUES.  SNV007
C                                                                       SNV007
      SUM=0.0                                                           SNV007
      DO 500 I=1,M                                                      SNV007
      DO 500 J=1,N                                                      SNV007
  500 SUM = SUM + A(I,J)**2                                             SNV007
      ZTEST = DSQRT(SUM)                                                SNV008
      ZTEST = ZTEST*10.**(-IAC)                                         SNV008
C                                                                       SNV008
  510 IF (IOP.NE.1 ) GO TO 515                                          SNV008
      WITHU=.FALSE.                                                     SNV008
      WITHV=.FALSE.                                                     SNV008
      GO TO 520                                                         SNV008
  515 WITHU=.TRUE.                                                      SNV008
      WITHV=.TRUE.                                                      SNV008
  520 CONTINUE                                                          SNV008
      G = 0.0                                                           SNV009
      X = 0.0                                                           SNV009
      DO 30 I = 1,N                                                     SNV009
C                                                                       SNV009
C     HOUSEHOLDER REDUCTION TO BIDIAGONAL FORM.                         SNV009
C                                                                       SNV009
      E(I) = G                                                          SNV009
      S = 0.0                                                           SNV009
      L = I+1                                                           SNV009
C                                                                       SNV009
C     ANNIHILATE THE I-TH COLUMN BELOW DIAGONAL.                        SNV010
C                                                                       SNV010
      DO 3 J = I,M                                                      SNV010
    3 S = S + A(J,I)**2                                                 SNV010
      G = 0.0                                                           SNV010
      IF(S .LT. TOL)    GO TO 10                                        SNV010
      G = DSQRT(S)                                                      SNV010
      F = A(I,I)                                                        SNV010
      IF(F .GE. 0.0)   G = -G                                           SNV010
      H = F*G -S                                                        SNV010
      A(I,I) = F-G                                                      SNV011
      IF(I .EQ. N)   GO TO 10                                           SNV011
        DO 9 J = L,N                                                    SNV011
        S = 0.0                                                         SNV011
        DO 7 K = I,M                                                    SNV011
    7   S = S +A(K,I)*A(K,J)                                            SNV011
        F = S/H                                                         SNV011
        DO 8 K = I,M                                                    SNV011
    8   A(K,J) =A(K,J) + F*A(K,I)                                       SNV011
    9   CONTINUE                                                        SNV011
   10 Q(I) = G                                                          SNV012
      IF(I .EQ. N)   GO TO 20                                           SNV012
C                                                                       SNV012
C     ANNIHILATE THE I-TH ROW TO RIGHT OF SUPER-DIAG.                   SNV012
C                                                                       SNV012
      S = 0.0                                                           SNV012
      DO 11 J = L,N                                                     SNV012
```

```
   11 S = S + A(I,J)**2                                                 SNV012
      G = 0.0                                                           SNV012
      IF (S .LT. TOL)     GO TO 20                                      SNV012
        G = DSQRT(S)                                                    SNV013
        F = A(I,I+1)                                                    SNV013
        IF(F .GE. 0.0)    G = -G                                        SNV013
        H = F*G -S                                                      SNV013
        A(I,I+1) = F - G                                                SNV013
        DO 15 J = L,N                                                   SNV013
   15   E(J) = A(I,J)/H                                                 SNV013
        DO 19 J = L,M                                                   SNV013
        S = 0.0                                                         SNV013
        DO 16 K = L,N                                                   SNV013
   16   S = S + A(J,K) * A(I,K)                                         SNV014
        DO 17 K = L,N                                                   SNV014
   17   A(J,K) = A(J,K) + S*E(K)                                        SNV014
   19   CONTINUE                                                        SNV014
   20 Y = DABS(Q(I)) + DABS(E(I))                                       SNV014
      IF(Y .GT. SIZE)     SIZE = Y                                      SNV014
   30 CONTINUE                                                          SNV014
      IF(.NOT. WITHV)    GO TO 41                                      SNV014
C                                                                       SNV014
C     ACCUMULATION OF RIGHT TRANSFORMATIONS.                            SNV014
C                                                                       SNV015
      DO 40 II = 1,N                                                    SNV015
      I = NP1 - II                                                      SNV015
      IF(I .EQ. N)    GO TO 39                                          SNV015
      IF(G .EQ. 0.0)    GO TO 37                                        SNV015
      H = A(I,I+1)*G                                                    SNV015
      DO 32 J = L,N                                                     SNV015
   32 V(J,I) = A(I,J)/H                                                 SNV015
      DO 36 J = L,N                                                     SNV015
      S = 0.0                                                           SNV015
      DO 33 K = L,N                                                     SNV016
   33 S = S + A(I,K)*V(K,J)                                             SNV016
      DO 34 K = L,N                                                     SNV016
   34 V(K,J) = V(K,J) + S*V(K,I)                                        SNV016
   36 CONTINUE                                                          SNV016
   37 DO 38 J = L,N                                                     SNV016
      V(I,J) = 0.0                                                      SNV016
   38 V(J,I) = 0.0                                                      SNV016
   39 V(I,I) = 1.0                                                      SNV016
      G = E(I)                                                          SNV016
   40 L = I                                                             SNV017
   41 CONTINUE                                                          SNV017
      IF(.NOT. WITHU)    GO TO 53                                      SNV017
C                                                                       SNV017
C     ACCUMULATION OF LEFT TRANSFORMATIONS.                             SNV017
C                                                                       SNV017
      DO 52 II = 1,N                                                    SNV017
      I = NP1 -II                                                       SNV017
      L = I + 1                                                         SNV017
      G = Q(I)                                                          SNV017
      IF(I .EQ. N)    GO TO 43                                          SNV018
      DO 42 J = L,N                                                     SNV018
   42 A(I,J) = 0.0                                                      SNV018
   43 CONTINUE                                                          SNV018
      IF(G .EQ. 0.0)    GO TO 49                                        SNV018
      IF(I .EQ. N)    GO TO 47                                          SNV018
        H = A(I,I)*G                                                    SNV018
        DO 46 J = L,N                                                   SNV018
        S = 0.0                                                         SNV018
        DO 44 K = L,M                                                   SNV018
```

```
      44    S = S + A(K,I)*A(K,J)                                          SNV019
            F = S/H                                                        SNV019
            DO 45 K = I,M                                                  SNV019
      45    A(K,J) = A(K,J) + F*A(K,I)                                     SNV019
      46    CONTINUE                                                       SNV019
      47 DO 48 J = I,M                                                     SNV019
      48 A(J,I) = A(J,I)/G                                                 SNV019
         GO TO 51                                                          SNV019
      49 DO 50 J = I,M                                                     SNV019
      50 A(J,I) = 0.0                                                      SNV019
      51 A(I,I) = A(I,I) + 1.0                                             SNV020
      52 CONTINUE                                                          SNV020
      53 CONTINUE                                                          SNV020
C                                                                          SNV020
C     DIAGONALIZATION OF BIDIAGONAL FORM.                                  SNV020
C                                                                          SNV020
         DO 100 KK=1,N                                                     SNV020
           K=NP1-KK                                                        SNV020
           ITCNT=0                                                         SNV020
           KP1=K+1                                                         SNV020
C                                                                          SNV021
C     TEST F SPLITTING.                                                    SNV021
C                                                                          SNV021
      59    CONTINUE                                                       SNV021
            DO 60 LL=1,K                                                   SNV021
              L=KP1-LL                                                     SNV021
              IF((SIZE+DABS(E(L))).EQ.SIZE)    GO TO 64                    SNV021
              LM1=L-1                                                      SNV021
              IF((SIZE+DABS(Q(LM1))).EQ.SIZE)    GO TO 61                  SNV021
      60    CONTINUE                                                       SNV021
C                                                                          SNV022
C     CANCELLATION OF E(L) IF L .GT. 1.                                    SNV022
C                                                                          SNV022
      61    C=0.0                                                          SNV022
            S=1.0                                                          SNV022
            L1=L-1                                                         SNV022
            DO 63 I=L,K                                                    SNV022
              F=S*E(I)                                                     SNV022
              E(I)=C*E(I)                                                  SNV022
              IF((SIZE+DABS(F)).EQ.SIZE)   GO TO 64                        SNV022
              G=Q(I)                                                       SNV023
              Q(I)=DSQRT(F*F+G*G)                                          SNV023
              H=Q(I)                                                       SNV023
              C=G/H                                                        SNV023
              S=-F/H                                                       SNV023
              IF(.NOT.WITHU)   GO TO 63                                    SNV023
                DO 62 J=1,M                                                SNV023
                  Y=A(J,L1)                                                SNV023
                  Z=A(J,I)                                                 SNV023
                  A(J,L1)=Y*C+Z*S                                          SNV023
                  A(J,I)= -Y*S+Z*C                                         SNV024
      62        CONTINUE                                                   SNV024
C                                                                          SNV024
      63    CONTINUE                                                       SNV024
C                                                                          SNV024
C     TEST F CONVERGENCE.                                                  SNV024
C                                                                          SNV024
      64 Z=Q(K)                                                            SNV024
         IF(L.EQ.K)   GO TO 75                                             SNV024
         IF(ITCNT .LE. 30)   GO TO 65                                      SNV024
         IERR = KK                                                         SNV025
         RETURN                                                            SNV025
      65 ITCNT = ITCNT + 1                                                 SNV025
```

```
C                                                                       SNV025
C       SHIFT FROM LOWER 2X2.                                           SNV025
C                                                                       SNV025
        X=Q(L)                                                          SNV025
        Y=Q(K-1)                                                        SNV025
      G=E(K-1)                                                          SNV025
      H=E(K)                                                            SNV025
      F=((Y-Z)*(Y+Z)+(G-H)*(G+H))/(2.0*H*Y)                             SNV026
      G=DSQRT(F*F+1.0)                                                  SNV026
      IF(F.LT.0.0)  G=-G                                                SNV026
      F = ((X-Z)*(X+Z)+H*(Y/(F+G)-H))/X                                 SNV026
C                                                                       SNV026
C                                                                       SNV026
C       NEXT QR TRANSFORMATION.                                         SNV026
C                                                                       SNV026
      C=1.0                                                             SNV026
      S=1.0                                                             SNV027
      LP1=L+1                                                           SNV027
      DO 73 I=LP1,K                                                     SNV027
        G=E(I)                                                          SNV027
        Y=Q(I)                                                          SNV027
        H=S*G                                                           SNV027
        G=C*G                                                           SNV027
        Z=DSQRT(F*F+H*H)                                                SNV027
        E(I-1)=Z                                                        SNV027
        C=F/Z                                                           SNV027
        S=H/Z                                                           SNV028
        F=X*C+G*S                                                       SNV028
        G=-X*S+G*C                                                      SNV028
        H=Y*S                                                           SNV028
        Y=Y*C                                                           SNV028
        IF(.NOT.WITHV)   GO TO 70                                       SNV028
          DO 68 J=1,N                                                   SNV028
            X=V(J,I-1)                                                  SNV028
            Z=V(J,I)                                                    SNV028
            V(J,I-1)=X*C+Z*S                                            SNV028
            V(J,I)=-X*S+Z*C                                             SNV029
   68     CONTINUE                                                      SNV029
C                                                                       SNV029
   70   Z=DSQRT(F*F+H*H)                                                SNV029
        Q(I-1)=Z                                                        SNV029
        C=F/Z                                                           SNV029
        S=H/Z                                                           SNV029
        F=C*G+S*Y                                                       SNV029
        X=-S*G+C*Y                                                      SNV029
        IF(.NOT.WITHU)   GO TO 73                                       SNV029
          DO 72 J=1,M                                                   SNV030
            Y=A(J,I-1)                                                  SNV030
            Z=A(J,I)                                                    SNV030
            A(J,I-1)=Y*C+Z*S                                            SNV030
            A(J,I)=-Y*S+Z*C                                             SNV030
   72     CONTINUE                                                      SNV030
C                                                                       SNV030
C                                                                       SNV030
   73   E(L) = 0.0                                                      SNV030
        E(K)=F                                                          SNV030
        Q(K)=X                                                          SNV030
        GO TO 59                                                        SNV031
C                                                                       SNV031
C       CONVERGENCE.                                                    SNV031
C                                                                       SNV031
   75   CONTINUE                                                        SNV031
        IF(Z.GE.0.0)   GO TO 100                                        SNV031
```

```
            Q(K)=-Z                                                     SNV03160
            IF(.NOT.WITHV) GO TO 100                                    SNV03170
            DO 76 J=1,N                                                 SNV03180
         76 V(J,K)=-V(J,K)                                              SNV03190
        100 CONTINUE                                                    SNV03200
C                                                                       SNV03210
            IERR=0                                                      SNV03220
            DO 280 II=2,N                                               SNV03230
            I=II-1                                                      SNV03240
            K=I                                                         SNV03250
            P=Q(I)                                                      SNV03260
            DO 250  J=II,N                                              SNV03270
            IF (Q(J).LE.P) GO TO 250                                    SNV03280
            K=J                                                         SNV03290
            P=Q(J)                                                      SNV03300
        250 CONTINUE                                                    SNV03310
C                                                                       SNV03320
            IF (K.EQ.I) GO TO 280                                       SNV03330
            Q(K) = Q(I)                                                 SNV03340
            Q(I) = P                                                    SNV03350
C                                                                       SNV03360
            IF(IOP.EQ.1) GO TO 280                                      SNV03370
C                                                                       SNV03380
            DO 260  J=1,N                                               SNV03390
            P= V(J,I)                                                   SNV03400
            V(J,I)= V(J,K)                                              SNV03410
            V(J,K)= P                                                   SNV03420
        260 CONTINUE                                                    SNV03430
C                                                                       SNV03440
            DO 270  J=1,M                                               SNV03450
            P = A(J,I)                                                  SNV03460
            A(J,I)= A(J,K)                                              SNV03470
            A(J,K)= P                                                   SNV03480
        270 CONTINUE                                                    SNV03490
C                                                                       SNV03500
        280 CONTINUE                                                    SNV03510
C                                                                       SNV03520
            J=N                                                         SNV03530
        290 IF (Q(J).GT.ZTEST) GO TO 300                                SNV03540
            Q(J)=0.0                                                    SNV03550
            J=J-1                                                       SNV03560
            GO TO 290                                                   SNV03570
        300 IRANK =J                                                    SNV03580
            TEMP = ZTEST/Q(J)                                           SNV03590
            IF (TEMP.GT..0625)     IERR=-1                              SNV03600
C                                                                       SNV03610
            IF (IOP.LT. 3)  RETURN                                      SNV03620
            IF(IOP.GT.3) GO TO 170                                      SNV03630
            DO 160  L=1,NOS                                             SNV03640
            DO 130  J=1,IRANK                                           SNV03650
            SUM=0.0                                                     SNV03660
            DO 120  I=1,M                                               SNV03670
        120 SUM =SUM + A(I,J)*B(I,L)                                    SNV03680
        130 E(J)= SUM/Q(J)                                              SNV03690
C                                                                       SNV03700
            DO 150  K=1,N                                               SNV03710
            SUM=0.0                                                     SNV03720
            DO 140  I=1,IRANK                                           SNV03730
        140 SUM =SUM  + V(K,I)*E(I)                                     SNV03740
        150 R(K,L)=SUM                                                  SNV03750
        160 CONTINUE                                                    SNV03760
            RETURN                                                      SNV03770
        170 DO 200  J=1,M                                               SNV03780
```

```
      DO 190  I=1,N                                                    SNV03790
      SUM=0.0                                                          SNV03800
      DO 180  K=1,IRANK                                                SNV03810
  180 SUM =SUM + V(I,K)*A(J,K)/Q(K)                                    SNV03820
  190 APLUS(I,J)= SUM                                                  SNV03830
  200 CONTINUE                                                         SNV03840
C                                                                      SNV03850
      IF( IOP .EQ.4) RETURN                                            SNV03860
      DO 230  K=1,NOS                                                  SNV03870
      DO 220  I=1,N                                                    SNV03880
      SUM=0.0                                                          SNV03890
      DO 210  J=1,M                                                    SNV03900
  210 SUM=SUM+ APLUS(I,J)*B(J,K)                                       SNV03910
  220 E(I)=SUM                                                         SNV03920
      DO 225  I=1,N                                                    SNV03930
  225 B(I,K)=E(I)                                                      SNV03940
  230 CONTINUE                                                         SNV03950
      RETURN                                                           SNV03960
      END                                                              SNV03970
```

```
      SUBROUTINE SUM(A,NA,B,NB,C,NC,IOP,SYM,DUMMY)                      SUM00010
      IMPLICIT REAL*8 (A-H,O-Z)                                         SUM00020
      DIMENSION A(1),B(1),C(1),DUMMY(1)                                 SUM00030
      DIMENSION NA(2),NB(2),NC(2)                                       SUM00040
      LOGICAL SYM                                                       SUM00050
      COMMON/CONV/SUMCV,RICTCV,SERCV,MAXSUM                             SUM00060
      IF( IOP  .EQ. 0  ) GO TO 100                                      SUM00070
      PRINT 50                                                          SUM00080
   50 FORMAT(//,' LINEAR EQUATION SOLVER     X = AXC + B ')           SUM00090
      CALL PRNT(A,NA,4H A  ,1)                                          SUM00100
      IF( SYM ) GO TO 75                                                SUM00110
      CALL PRNT(C,NC,4H C  ,1)                                          SUM00120
      GO TO 85                                                          SUM00130
   75 CONTINUE                                                          SUM00140
      PRINT 80                                                          SUM00150
   80 FORMAT(/, ' C = A TRANSPOSE ',/)                                  SUM00160
   85 CONTINUE                                                          SUM00170
      CALL PRNT(B,NB,4H B  ,1)                                          SUM00180
C                                                                       SUM00190
  100 CONTINUE                                                          SUM00200
      N1 = 1 + NA(1)*NC(1)                                              SUM00210
      I=1                                                               SUM00220
  200 CONTINUE                                                          SUM00230
      CALL MULT(A,NA,B,NB,DUMMY,NB)                                     SUM00240
      CALL MULT(DUMMY,NB,C,NC,DUMMY(N1),NB)                             SUM00250
      CALL MAXEL(B,NB,WNS)                                              SUM00260
      CALL MAXEL(DUMMY(N1),NB,WNDX)                                     SUM00270
      IF(WNS .GE. 1.) GO TO 225                                         SUM00280
      IF( WNDX/WNS .LT. SUMCV ) GO TO 300                               SUM00290
      GO TO 235                                                         SUM00300
  225 IF( WNDX .LT. SUMCV ) GO TO 300                                   SUM00310
  235 CONTINUE                                                          SUM00320
      CALL ADD(B,NB,DUMMY(N1),NB,B,NB)                                  SUM00330
      CALL MULT(A,NA,A,NA,DUMMY,NA)                                     SUM00340
      CALL EQUATE(DUMMY,NA,A,NA)                                        SUM00350
      IF( SYM ) GO TO 245                                               SUM00360
      CALL MULT(C,NC,C,NC,DUMMY,NC)                                     SUM00370
      CALL EQUATE(DUMMY,NC,C,NC)                                        SUM00380
      GO TO  250                                                        SUM00390
  245 CONTINUE                                                          SUM00400
      CALL TRANP(A,NA,C,NC)                                             SUM00410
  250 CONTINUE                                                          SUM00420
      I=I+1                                                             SUM00430
      IF( I .LE. MAXSUM ) GO TO 200                                     SUM00440
      CALL LNCNT(3)                                                     SUM00450
      PRINT 275,MAXSUM                                                  SUM00460
  275 FORMAT(//,' IN SUM, THE SEQUENCE OF PARTIAL SUMS HAS EXCEEDED STAGSUM00470
     1 E',I5,' WITHOUT CONVERGENCE')                                    SUM00480
  300 CONTINUE                                                          SUM00490
      IF(IOP .EQ. 0) RETURN                                             SUM00500
      CALL PRNT(B,NB,4H X  ,1)                                          SUM00510
      RETURN                                                            SUM00520
      END                                                               SUM00530
```

```
      SUBROUTINE BILIN(A,NA,B,NB,C,NC,IOP,BETA,SYM,DUMMY)                BIL00010
      IMPLICIT REAL*8 (A-H,O-Z)                                          BIL00020
      DIMENSION A(1),B(1),C(1),DUMMY(1)                                  BIL00030
      DIMENSION NA(2),NB(2),NC(2),NDUM(2)                                BIL00040
      DIMENSION IOP(2)                                                   BIL00050
      LOGICAL SYM                                                        BIL00060
      IF( IOP(1) .EQ. 0 )  GO TO 300                                     BIL00070
      IF(SYM) GO TO 100                                                  BIL00080
      CALL LNCNT(3)                                                      BIL00090
      PRINT 50                                                           BIL00100
   50 FORMAT(//,' LINEAR EQUATION SOLVER  AX + XB = C ')                 BIL00110
      CALL PRNT(A,NA,4H A  ,1)                                           BIL00120
      CALL PRNT(B,NB,4H B  ,1)                                           BIL00130
      GO TO 200                                                          BIL00140
  100 CONTINUE                                                           BIL00150
      CALL LNCNT(3)                                                      BIL00160
      PRINT 150                                                          BIL00170
  150 FORMAT(//,' LINEAR EQUATION SOLVER  ( B TRANSPOSE )X + XB = C ')   BIL00180
      CALL TRANP(A,NA,DUMMY,NDUM)                                        BIL00190
      CALL PRNT(DUMMY,NDUM,4H B  ,1)                                     BIL00200
  200 CONTINUE                                                           BIL00210
      CALL PRNT(C,NC,4H C  ,1)                                           BIL00220
  300 CONTINUE                                                           BIL00230
C                                                                        BIL00240
      IOPTT = 0                                                          BIL00250
      N=NA(1)**2                                                         BIL00260
      M=NB(1)**2                                                         BIL00270
C                                                                        BIL00280
      IF( IOP(2) .EQ. 0 )  GO TO 500                                     BIL00290
C                                                                        BIL00300
      N1 = N + 1                                                         BIL00310
      CALL EQUATE(A,NA,DUMMY,NA)                                         BIL00320
      N2 = N1 + NA(1)                                                    BIL00330
      N3 = N2 + NA(1)                                                    BIL00340
      ISV = 0                                                            BIL00350
      ILV = 0                                                            BIL00360
      NEVL = NA(1)                                                       BIL00370
      CALL EIGEN(NA(1),NA(1),DUMMY,DUMMY(N1),DUMMY(N2),ISV,ILV,V,DUMMY(NBIL00380
     13),IERR)                                                           BIL00390
      IF (IERR .EQ. 0) GO TO 350                                         BIL00400
      CALL LNCNT(3)                                                      BIL00410
      PRINT 325,IERR                                                     BIL00420
  325 FORMAT(//,' IN BILIN, THE ',I4,' EIGENVALUE OF A HAS NOT BEEN  DETBIL00430
     1ERMINED AFTER 30 ITERATIONS')                                      BIL00440
      IERR=1                                                             BIL00450
      CALL NORMS(NEVL,NEVL,NEVL,A,IERR,BETA)                             BIL00460
      BETA=2.*BETA                                                       BIL00470
      GO TO 385                                                          BIL00480
  350 CONTINUE                                                           BIL00490
      J= N1 + NEVL -1                                                    BIL00500
      K = N2 + NEVL -1                                                   BIL00510
      CO = DSQRT(DUMMY(N1)**2 + DUMMY(N2)**2)                            BIL00520
      CN = DSQRT(DUMMY(J)**2 + DUMMY(K)**2)                              BIL00530
      CD = DUMMY(J)-DUMMY(N1)                                            BIL00540
      IF(CD .EQ. 0.0)  GO TO 365                                         BIL00550
      BETA = (DUMMY(N1)*CN-DUMMY(J)*CO)/CD                               BIL00560
      IF(BETA .LE. 0.0)  GO TO 365                                       BIL00570
      BETA = DSQRT(BETA)                                                 BIL00580
      GO TO 385                                                          BIL00590
C                                                                        BIL00600
  365 CONTINUE                                                           BIL00610
C                                                                        BIL00620
      BETA = 0.0                                                         BIL00630
```

```
      DO 375 I = 1,NEVL                                                 BIL00640
      J = N1 + I -1                                                     BIL00650
      K = N2 + I -1                                                     BIL00660
      IF(DUMMY(J) .GE. 0.0)  GO TO 375                                  BIL00670
      BETA = BETA + DSQRT(DUMMY(J)**2 + DUMMY(K)**2)                    BIL00680
  375 CONTINUE                                                          BIL00690
      BETA = BETA/NEVL                                                  BIL00700
C                                                                       BIL00710
  385 CONTINUE                                                          BIL00720
C                                                                       BIL00730
      IF( SYM ) GO TO 500                                               BIL00740
      CALL EQUATE(B,NB,DUMMY,NB)                                        BIL00750
      N1=M+1                                                            BIL00760
      N2 = N1 +NB(1)                                                    BIL00770
      N3 = N2 +NB(1)                                                    BIL00780
      NEVL = NB(1)                                                      BIL00790
      CALL EIGEN(NB(1),NB(1),DUMMY,DUMMY(N1),DUMMY(N2),ISV,ILV,V,DUMMY(NBIL00800
     13),IERR)                                                          BIL00810
      IF(IERR .EQ. 0) GO TO 450                                         BIL00820
      CALL LNCNT(3)                                                     BIL00830
      PRINT 400,IERR                                                    BIL00840
  400 FORMAT(//,' IN BILIN, THE ',I4,' EIGENVALUE OF B HAS NOT BEEN FOUNBIL00850
     1D AFTER 30 ITERATIONS')                                           BIL00860
      IERR=1                                                            BIL00870
      CALL NORMS(NEVL,NEVL,NEVL,B,IERR,BETA1)                           BIL00880
      BETA1=2.*BETA1                                                    BIL00890
      GO TO 485                                                         BIL00900
  450 CONTINUE                                                          BIL00910
      J = N1 + NEVL -1                                                  BIL00920
      K = N2 + NEVL -1                                                  BIL00930
      CO = DSQRT(DUMMY(N1)**2 + DUMMY(N2)**2)                           BIL00940
      CN = DSQRT(DUMMY(J)**2 + DUMMY(K)**2)                             BIL00950
      CD = DUMMY(J)-DUMMY(N1)                                           BIL00960
      IF(CD .EQ. 0.0)  GO TO 465                                        BIL00970
      BETA1 = (DUMMY(N1)*CN - DUMMY(J)*CO)/CD                           BIL00980
      IF(BETA1 .LE. 0.0)  GO TO 465                                     BIL00990
      BETA1 = DSQRT(BETA1)                                              BIL01000
      GO TO 485                                                         BIL01010
C                                                                       BIL01020
  465 CONTINUE                                                          BIL01030
C                                                                       BIL01040
      BETA1 = 0.0                                                       BIL01050
      DO 475 I= 1,NEVL                                                  BIL01060
      J = N1 + I -1                                                     BIL01070
      K = N2 + I -1                                                     BIL01080
      IF(DUMMY(J) .GE. 0.0)  GO TO 475                                  BIL01090
      BETA1 = BETA1 + DSQRT(DUMMY(J)**2 + DUMMY(K)**2)                  BIL01100
  475 CONTINUE                                                          BIL01110
      BETA1 = BETA1/NEVL                                                BIL01120
C                                                                       BIL01130
  485 CONTINUE                                                          BIL01140
      BETA = (BETA + BETA1)/2.                                          BIL01150
C                                                                       BIL01160
  500 CONTINUE                                                          BIL01170
C                                                                       BIL01180
C                                                                       BIL01190
      IF( IOP(1) .EQ. 0 )  GO TO 520                                    BIL01200
      CALL LNCNT(4)                                                     BIL01210
      PRINT 515,BETA                                                    BIL01220
  515 FORMAT(//,' BETA = ',E16.8,/)                                     BIL01230
  520 CONTINUE                                                          BIL01240
C                                                                       BIL01250
      N1 = N+1                                                          BIL01260
```

```
      CALL EQUATE(A,NA,DUMMY,NA)                                        BIL01270
      CALL EQUATE(A,NA,DUMMY(N1),NA)                                    BIL01280
      CALL SCALE(DUMMY,NA,DUMMY,NA,-1.)                                 BIL01290
      L = -NA(1)                                                        BIL01300
      NAX = NA(1)                                                       BIL01310
      DO 525 I=1,NAX                                                    BIL01320
      L = L + NAX +1                                                    BIL01330
      M1 = L + N                                                        BIL01340
      DUMMY(L) = BETA - A(L)                                            BIL01350
      DUMMY(M1)= BETA + A(L)                                            BIL01360
  525 CONTINUE                                                          BIL01370
      N2 = N1 + N                                                       BIL01380
      CALL EQUATE(C,NC,DUMMY(N2),NDUM)                                  BIL01390
      NDUM(2)= NDUM(2) + NA(1)                                          BIL01400
      N3 = N2 + NC(1)*NC(2)                                             BIL01410
      GAM = -2.*BETA                                                    BIL01420
C                                                                       BIL01430
      IF( .NOT. SYM ) GO  TO 600                                        BIL01440
C                                                                       BIL01450
      CALL UNITY(DUMMY(N3),NA)                                          BIL01460
      N4 = N3 + N                                                       BIL01470
      NDUM(2) = NDUM(2) + NA(1)                                         BIL01480
      N5 = N4 + NA(1)                                                   BIL01490
      IFAC = 0                                                          BIL01500
      CALL GELIM(NA(1),NA(1),DUMMY,NDUM(2),DUMMY(N1),DUMMY(N4),IFAC,DUMMBIL01510
     1Y(N5),IERR)                                                       BIL01520
      IF( IERR .EQ. 1 )  PRINT 625                                      BIL01530
      CALL EQUATE(DUMMY(N1),NA,DUMMY,NA)                                BIL01540
      CALL EQUATE(DUMMY(N2),NC,C,NC)                                    BIL01550
      CALL TRANP(DUMMY,NA,DUMMY(N1),NA)                                 BIL01560
      CALL TRANP(DUMMY(N3),NA,DUMMY(N2),NA)                             BIL01570
      CALL MULT(C,NC,DUMMY(N2),NA,DUMMY(N3),NA)                         BIL01580
      CALL SCALE(DUMMY(N3),NC,C,NC,GAM)                                 BIL01590
C                                                                       BIL01600
C                                                                       BIL01610
      CALL SUM(DUMMY,NA,C,NC,DUMMY(N1),NA,IOPTT,SYM,DUMMY(N2))          BIL01620
      GO TO 700                                                         BIL01630
  600 CONTINUE                                                          BIL01640
      N4 = N3 +NA(1)                                                    BIL01650
      IFAC = 0                                                          BIL01660
      CALL GELIM(NA(1),NA(1),DUMMY,NDUM(2),DUMMY(N1),DUMMY(N3),IFAC,DUMMBIL01670
     1Y(N4),IERR)                                                       BIL01680
      IF(IERR .EQ. 1 ) PRINT 625                                        BIL01690
  625 FORMAT(//,' IN BILIN, THE MATRIX  (BETA)I - A IS SINGULAR, INCREASBIL01700
     1E BETA ')                                                         BIL01710
      CALL EQUATE(DUMMY(N1),NA,DUMMY,NA)                                BIL01720
      CALL EQUATE(DUMMY(N2),NC,C,NC)                                    BIL01730
      N2 = M + N                                                        BIL01740
      CALL EQUATE(B,NB,DUMMY(N1),NB)                                    BIL01750
      CALL EQUATE(B,NB,DUMMY(N2),NB)                                    BIL01760
      CALL SCALE(DUMMY(N1),NB,DUMMY(N1),NB,-1.0)                        BIL01770
      L=-NB(1)                                                          BIL01780
      NAX=NB(1)                                                         BIL01790
      DO650I =1,NAX                                                     BIL01800
      L=L + NAX +1                                                      BIL01810
      L1 = L + N                                                        BIL01820
      M1 = L + N2-1                                                     BIL01830
      DUMMY(L1)= BETA- B(L)                                             BIL01840
      DUMMY(M1)= BETA + B(L)                                            BIL01850
  650 CONTINUE                                                          BIL01860
C                                                                       BIL01870
      N3 = N2 + M                                                       BIL01880
      CALL TRANP(DUMMY(N1),NB,DUMMY(N3),NB)                             BIL01890
```

```
      CALL EQUATE(DUMMY(N3),NB,DUMMY(N1),NB)                            BIL01900
      CALL TRANP(DUMMY(N2),NB,DUMMY(N3),NB)                             BIL01910
      CALL EQUATE(DUMMY(N3),NB,DUMMY(N2),NB)                            BIL01920
      CALL TRANP(C,NC,DUMMY(N3),NDUM)                                   BIL01930
      NSDUM = NDUM(2)                                                   BIL01940
      NDUM(2)= NDUM(2) + NB(2)                                          BIL01950
      IFAC = 0                                                          BIL01960
      N4=N3+NC(1)*NC(2)                                                 BIL01970
      N5=N4+NB(1)                                                       BIL01980
      CALL GELIM(NB(1),NB(1),DUMMY(N1),NDUM(2),DUMMY(N2),DUMMY(N4),IFAC,BIL01990
     1DUMMY(N5),IERR)                                                   BIL02000
      IF(IERR .EQ. 1 )  PRINT 675                                       BIL02010
  675 FORMAT(//,'IN BILIN, THE MATRIX (BETA)I - B  IS SINGULAR, INCREASEBIL02020
     1 BETA ')                                                          BIL02030
      CALL TRANP(DUMMY(N2),NB,DUMMY(N1),NB)                             BIL02040
      NDUM(2)= NSDUM                                                    BIL02050
      CALL TRANP(DUMMY(N3),NDUM,C,NC)                                   BIL02060
      CALL SCALE(C,NC,C,NC,GAM)                                         BIL02070
      N2 = N + M + 1                                                    BIL02080
      CALL SUM(DUMMY,NA,C,NC,DUMMY(N1),NB,IOPTT,SYM,DUMMY(N2))          BIL02090
C                                                                       BIL02100
  200 CONTINUE                                                          BIL02110
      IF( IOP(1) .EQ. 0 ) RETURN                                        BIL02120
      CALL PRNT(C,NC,4H X  ,1)                                          BIL02130
      RETURN                                                            BIL02140
      END                                                               BIL02150
```

```
      SUBROUTINE BARSTW(A,NA,B,NB,C,NC,IOP,SYM,EPSA,EPSB,DUMMY)          BAR00010
      IMPLICIT REAL*8 (A-H,O-Z)                                          BAR00020
      DIMENSION A(1),B(1),C(1),DUMMY(1)                                  BAR00030
      DIMENSION NA(2),NB(2),NC(2),NDUM1(2),NDUM2(2),NDUM3(2),NDUM4(2)    BAR00040
      LOGICAL  SYM                                                       BAR00050
      IF ( IOP .EQ. 0 )  GO TO 250                                       BAR00060
      IF(SYM) GO TO 100                                                  BAR00070
      CALL LNCNT(3)                                                      BAR00080
      PRINT 50                                                           BAR00090
   50 FORMAT(//,' LINEAR EQUATION SOLVER     AX + XB = C ')              BAR00100
      CALL PRNT(A,NA,4H A   ,1)                                          BAR00110
      CALL PRNT(B,NB,4H B   ,1)                                          BAR00120
      GO TO 200                                                          BAR00130
  100 CONTINUE                                                           BAR00140
      CALL LNCNT(3)                                                      BAR00150
      PRINT 150                                                          BAR00160
  150 FORMAT(//,' LINEAR EQUATION SOLVER  ( B TRANSPOSE )X + XB = C')    BAR00170
      CALL TRANP(A,NA,DUMMY,NDUM1)                                       BAR00180
      CALL PRNT(DUMMY,NDUM1,4H B   ,1)                                   BAR00190
  200 CONTINUE                                                           BAR00200
      CALL PRNT(C,NC,4H C   ,1)                                          BAR00210
C                                                                        BAR00220
  250 CONTINUE                                                           BAR00230
      CALL EQUATE(A,NA,DUMMY,NDUM1)                                      BAR00240
      N1=(NA(1)**2)+1                                                    BAR00250
      N2=N1+NA(1)-1                                                      BAR00260
      DO 300I=N1,N2                                                      BAR00270
      DUMMY(I)=0.0                                                       BAR00280
  300 CONTINUE                                                           BAR00290
C                                                                        BAR00300
      NDUM1(2)=NDUM1(2)+1                                                BAR00310
      NDUM2(1)=1                                                         BAR00320
      NDUM2(2)=NDUM1(2)                                                  BAR00330
      N1=NDUM1(1)*NDUM1(2)+1                                             BAR00340
      CALL NULL(DUMMY(N1),NDUM2)                                         BAR00350
      LU=(NA(1)+1)**2 + 1                                                BAR00360
      CALL JUXTR(DUMMY,NDUM1,DUMMY(N1),NDUM2,DUMMY(LU),NDUM3)            BAR00370
      CALL EQUATE(DUMMY(LU),NDUM3,DUMMY,NDUM1)                           BAR00380
      N=NA(1)+1                                                          BAR00390
C                                                                        BAR00400
      IF(SYM ) GO TO 500                                                 BAR00410
C                                                                        BAR00420
      CALL EQUATE(B,NB,DUMMY(LU),NDUM2)                                  BAR00430
      M1=LU+NB(1)**2                                                     BAR00440
      M2=M1+NB(1)-1                                                      BAR00450
      DO400I=M1,M2                                                       BAR00460
      DUMMY(I)=0.0                                                       BAR00470
  400 CONTINUE                                                           BAR00480
C                                                                        BAR00490
      NDUM2(2)=NDUM2(2)+1                                                BAR00500
      NDUM3(1)=1                                                         BAR00510
      NDUM3(2)=NDUM2(2)                                                  BAR00520
      M1=NDUM2(1)*NDUM2(2)+LU                                            BAR00530
      CALL NULL(DUMMY(M1),NDUM3)                                         BAR00540
      M2=LU+(NB(1)+1)**2                                                 BAR00550
      CALL JUXTR(DUMMY(LU),NDUM2,DUMMY(M1),NDUM3,DUMMY(M2),NDUM4)        BAR00560
      CALL EQUATE(DUMMY(M2),NDUM4,DUMMY(LU),NDUM2)                       BAR00570
      M=NB(1)+ 1                                                         BAR00580
      LNB = LU                                                           BJR00590
      LU = LU + (NB(1)+1)**2                                             BAR00600
      LV = LU +  NA(1)**2                                                BAR00610
      CALL AXPXB(DUMMY,DUMMY(LU),NA(1),N,NA(1),DUMMY(LNB),DUMMY(LV),NB(1BAR00620
     1),M,NB(1),C,NC(1),EPSA,EPSB,NFAIL)                                 BAR00630
```

```
      GO TO 600                                                          BAR00640
C                                                                        BAR00650
  500 CONTINUE                                                           BAR00660
      CALL TRANP(DUMMY,NDUM1,DUMMY(LU),NDUM2)                            BAR00670
      CALL EQUATE(DUMMY(LU),NDUM2,DUMMY,NDUM1)                           BAR00680
      CALL ATXPXA(DUMMY,DUMMY(LU),C,NA(1),N,NA(1),NC(1),EPSA,NFAIL)      BAR00690
C                                                                        BAR00700
  600 CONTINUE                                                           BAR00710
      IF(NFAIL .EQ. 0 ) GO TO 700                                        BAR00720
      CALL LNCNT(3)                                                      BAR00730
      PRINT 650                                                          BAR00740
  650 FORMAT(//,' IN BARSTW, EITHER THE SUBROUTINE AXPXB  OR  ATXPXA  WABAR00750
     1S UNABLE TO REDUCE A OR B TO SCHUR FORM ')                         BAR00760
      RETURN                                                             BAR00770
C                                                                        BAR00780
  700 CONTINUE                                                           BAR00790
C                                                                        BAR00800
      IF( IOP .NE. 0 )  CALL PRNT(C,NC,4H  X ,1)                         BAR00810
      RETURN                                                             BAR00820
      END                                                                BAR00830
```

```
      SUBROUTINE TESTSA(A,NA,ALPHA,DISC,STABLE,IOP,DUMMY)                TES00010
      IMPLICIT REAL*8 (A-H,O-Z)                                          TES00020
      DIMENSION A(1),DUMMY(1)                                            TES00030
      DIMENSION NA(2),NDUM1(2),NDUM2(2)                                  TES00040
      LOGICAL  DISC,STABLE                                               TES00050
      STABLE = .FALSE.                                                   TES00060
C                                                                        TES00070
      CALL EQUATE(A,NA,DUMMY,NA)                                         TES00080
      N1= NA(1)**2 + 1                                                   TES00090
      N2= N1+NA(1)                                                       TES00100
      N3= N2+NA(1)                                                       TES00110
      ISV = 0                                                            TES00120
      CALL EIGEN(NA(1),NA(1),DUMMY,DUMMY(N1),DUMMY(N2),ISV,ISV,V,DUMMY(NTES00130
     13),IERR)                                                           TES00140
      NEVL = NA(1)                                                       TES00150
      IF( IERR .EQ. 0 ) GO TO 200                                        TES00160
      CALL LNCNT(4)                                                      TES00170
      PRINT 100,IERR                                                     TES00180
  100 FORMAT(//,' IN TESTSA, THE ',I5,' EIGENVALUE OF A HAS NOT BEEN FO  TES00190
     1UND AFTER 30 ITERATIONS',/)                                        TES00200
      RETURN                                                             TES00210
C                                                                        TES00220
  200 CONTINUE                                                           TES00230
      NDUM1(1) = NEVL                                                    TES00240
      NDUM1(2) = 1                                                       TES00250
      CALL JUXTC(DUMMY(N1),NDUM1,DUMMY(N2),NDUM1,DUMMY,NDUM2)            TES00260
C                                                                        TES00270
      IF( DISC ) GO TO 400                                               TES00280
      DO 300 I=1,NEVL                                                    TES00290
      IF( DUMMY(I) .GE. ALPHA ) GO TO 600                                TES00300
  300 CONTINUE                                                           TES00310
      GO TO 550                                                          TES00320
  400 CONTINUE                                                           TES00330
      N = NDUM2(1)*NDUM2(2)+1                                            TES00340
      DO 500 I =1,NEVL                                                   TES00350
      K = I + NEVL                                                       TES00360
      L=N +I -1                                                          TES00370
      DUMMY(L) = DSQRT((DUMMY(I)**2)+(DUMMY(K)**2))                      TES00380
  500 CONTINUE                                                           TES00390
C                                                                        TES00400
      IF( DUMMY(L) .GE. ALPHA ) GO TO 600                                TES00410
C                                                                        TES00420
  550 CONTINUE                                                           TES00430
      STABLE =.TRUE.                                                     TES00440
  600 CONTINUE                                                           TES00450
      IF( IOP .EQ. 0 ) RETURN                                            TES00460
      CALL LNCNT(4)                                                      TES00470
      PRINT 700                                                          TES00480
  700 FORMAT(//,' PROGRAM TO TEST THE RELATIVE STABILITY OF THE MATRIX ATES00490
     1',/)                                                               TES00500
      CALL PRNT(A,NA,4H A  ,1)                                           TES00510
      CALL LNCNT(4)                                                      TES00520
      PRINT 750                                                          TES00530
  750 FORMAT(//,' EIGENVALUES OF A ',/)                                  TES00540
      CALL PRNT(DUMMY,NDUM2,4HEVLA,1)                                    TES00550
      IF(  .NOT. DISC ) GO TO 850                                        TES00560
      CALL LNCNT(4)                                                      TES00570
      PRINT 800                                                          TES00580
  800 FORMAT(//,' MODULI OF EIGENVALUES OF A',/)                         TES00590
      CALL PRNT(DUMMY(N),NDUM1,4HMODA,1)                                 TES00600
C                                                                        TES00610
  850 CONTINUE                                                           TES00620
      CALL LNCNT(4)                                                      TES00630
```

```
      IF(STABLE) PRINT 900,ALPHA                                        TES00640
      IF( .NOT. STABLE) PRINT 950,ALPHA                                 TES00650
  900 FORMAT(//,' MATRIX A  IS STABLE RELATIVE TO ',E16.8,/)             TES00660
  950 FORMAT(//,' MATRIX A  IS UNSTABLE RELATIVE TO ',E16.8,/)           TES00670
C                                                                       TES00680
      RETURN                                                            TES00690
      END                                                               TES00700
```

```
      SUBROUTINE EXPSER(A,NA,EXPA,NEXPA,T,IOP,DUMMY)                    EXP0001
      IMPLICIT REAL*8 (A-H,O-Z)                                         EXP0002
      DIMENSION A(1),EXPA(1),DUMMY(1)                                   EXP0003
      DIMENSION NA(2),NEXPA(2)                                          EXP0004
      COMMON/CONV/SUMCV,RICTCV,SERCV,MAXSUM                             EXP0005
C                                                                       EXP0006
      N = NA(1)                                                         EXP0007
      L = (N**2) + 1                                                    EXP0008
      TT = T                                                            EXP0009
      NEXPA(1)=NA(1)                                                    EXP0010
      NEXPA(2)=NA(2)                                                    EXP0011
C                                                                       EXP0012
      CALL MAXEL(A,NA,ANAA)                                             EXP0013
      ANAA = ANAA*TT                                                    EXP0014
      ANAA = DABS(ANAA)                                                 EXP0015
      IF( ANAA .GT. 1.E-15 ) GO TO 100                                  EXP0016
      CALL UNITY(EXPA,NEXPA)                                            EXP0017
      GO TO 800                                                         EXP0018
C                                                                       EXP0019
  100 CONTINUE                                                          EXP0020
      IOPT=2                                                            EXP0021
      CALL NORMS(N,N,N,A,IOPT,ZERO)                                     EXP0022
      ZERO=ZERO/(2.**47)                                                EXP0023
      CALL TRCE(A,NA,TR)                                                EXP0024
      TR = TR/N                                                         EXP0025
      DO 200 I =1,N                                                     EXP0026
      M =I+N*(I-1)                                                      EXP0027
      A(M) = A(M) - TR                                                  EXP0028
  200 CONTINUE                                                          EXP0029
C                                                                       EXP0030
      IOPT = 1                                                          EXP0031
      CALL NORMS(N,N,N,A,IOPT,COL)                                      EXP0032
      IOPT = 3                                                          EXP0033
      CALL NORMS(N,N,N,A,IOPT,ROW)                                      EXP0034
      ANORM = ROW                                                       EXP0035
      IF( ANORM .GT. COL )  ANORM = COL                                 EXP0036
      TMAX = 1./ANORM                                                   EXP0037
      K= 0                                                              EXP0038
  300 CONTINUE                                                          EXP0039
      IF( TMAX - TT ) 325,350,350                                       EXP0040
  325 CONTINUE                                                          EXP0041
      K=K+1                                                             EXP0042
      TT = T/(2**K)                                                     EXP0043
      IF( K - 1000 ) 300,700,700                                        EXP0044
  350 CONTINUE                                                          EXP0045
      SC = TT                                                           EXP0046
      CALL SCALE(A,NA,A,NA,TT)                                          EXP0047
      CALL UNITY(EXPA,NEXPA)                                            EXP0048
      II = 2                                                            EXP0049
      CALL ADD(A,NA,EXPA,NEXPA,DUMMY,NA)                                EXP0050
      CALL EQUATE(A,NA,DUMMY(L),NA)                                     EXP0051
  400 CONTINUE                                                          EXP0052
      CALL MULT(A,NA,DUMMY(L),NA,EXPA,NEXPA)                            EXP0053
      S = 1./II                                                         EXP0054
      CALL SCALE(EXPA,NEXPA,DUMMY(L),NA,S)                              EXP0055
      CALL ADD(DUMMY(L),NA,DUMMY,NA,EXPA,NEXPA)                         EXP0056
      CALL MAXEL(DUMMY,NA,TOT)                                          EXP0057
      CALL MAXEL(DUMMY(L),NA,DELT)                                      EXP0058
      IF( TOT .GT. 1.0 ) GO TO 500                                      EXP0059
      IF( DELT/TOT .LT. SERCV )  GO TO 600                              EXP0060
      GO TO 550                                                         EXP0061
  500 CONTINUE                                                          EXP0062
      IF( DELT .LT. SERCV )  GO TO 600                                  EXP0063
```

```
   550 CONTINUE                                                          EXP00640
       CALL EQUATE(EXPA,NEXPA,DUMMY,NA)                                  EXP00650
       II = II + 1                                                       EXP00660
       GO TO 400                                                         EXP00670
C                                                                        EXP00680
   600 CONTINUE                                                          EXP00690
       IF( K ) 625,675,650                                               EXP00700
   625 CONTINUE                                                          EXP00710
       CALL LNCNT(1)                                                     EXP00720
       PRINT 635                                                         EXP00730
   635 FORMAT( '   ERROR IN EXPSER,  K IS NEGATIVE ' )                   EXP00740
       RETURN                                                            EXP00750
C                                                                        EXP00760
   650 CONTINUE                                                          EXP00770
       DO 660 I =1,K                                                     EXP00780
       TT = 2*TT                                                         EXP00790
       CALL EQUATE(EXPA,NEXPA,DUMMY,NA)                                  EXP00800
       CALL EQUATE(DUMMY,NA,DUMMY(L),NA)                                 EXP00810
       CALL MULT(DUMMY(L),NA,DUMMY,NA,EXPA,NEXPA)                        EXP00820
   660 CONTINUE                                                          EXP00830
       T = TT                                                            EXP00840
   675 CONTINUE                                                          EXP00850
       S = 1./SC                                                         EXP00860
       CALL SCALE(A,NA,A,NA,S)                                           EXP00870
       DO 685 I = 1,N                                                    EXP00880
       M = I + N*(I-1)                                                   EXP00890
       A(M) = A(M) + TR                                                  EXP00900
       IF( DABS(A(M)) .LE. ZERO ) A(M) = 0.0                             EXP00910
   685 CONTINUE                                                          EXP00920
C                                                                        EXP00930
       TR=TR*T                                                           EXP00940
       S =  DEXP(TR)                                                     EXP00950
       CALL SCALE(EXPA,NEXPA,EXPA,NEXPA,S)                               EXP00960
       GO TO 800                                                         EXP00970
C                                                                        EXP00980
   700 CONTINUE                                                          EXP00990
       CALL LNCNT(1)                                                     EXP01000
       PRINT 750                                                         EXP01010
   750 FORMAT('  ERROR IN EXPSER,  K = 1000 ')                          EXP01020
       RETURN                                                            EXP01030
C                                                                        EXP01040
   800 CONTINUE                                                          EXP01050
       IF( IOP .EQ. 0 ) RETURN                                           EXP01060
       CALL LNCNT(4)                                                     EXP01070
       PRINT 825                                                         EXP01080
   825 FORMAT(//' COMPUTATION OF THE MATRIX EXPONENTIAL EXP(A T) BY THE SEXP01090
      1ERIES METHOD '/)                                                  EXP01100
       CALL PRNT(A,NA,4H A  ,1)                                          EXP01110
       CALL LNCNT(3)                                                     EXP01120
       PRINT 850,T                                                       EXP01130
   850 FORMAT(/' T = ',D16.8/)                                           EXP01140
       CALL PRNT(EXPA,NEXPA,4HEXPA,1)                                    EXP01150
       RETURN                                                            EXP01160
       END                                                               EXP01170
```

```
      SUBROUTINE EXPADE (MAX, N, A, EA, IDIG, WK, IERR)                 EXP00010
      IMPLICIT REAL*8 (A-H,O-Z)                                         EXP00020
      DIMENSION A(MAX,N),EA(MAX,N),WK(N,1),C(9)                         EXP00030
      REAL*4 SDIGC,ALOG10                                               EXP00040
      IERR = 0                                                          EXP00050
C     CALCULATE NORM OF A                                               EXP00060
      ANORM = 0.                                                        EXP00070
      DO 10 I=1,N                                                       EXP00080
         S = 0.                                                         EXP00090
         DO 5 J=1,N                                                     EXP00100
            S = S + DABS(A(I,J))                                        EXP00110
    5    CONTINUE                                                       EXP00120
         IF (S .GT. ANORM)  ANORM = S                                   EXP00130
   10 CONTINUE                                                          EXP00140
C ****                                                                  EXP00150
C     CALCULATE ACCURACY ESTIMATE                                       EXP00160
C ****                                                                  EXP00170
      DIGC = 24.*DFLOAT(N)                                              EXP00180
      IF (ANORM .GT. 1.) DIGC = DIGC*ANORM                              EXP00190
      SDIGC=DIGC                                                        EXP00200
      IDIG = 15 - IFIX(ALOG10(SDIGC) + .5)                              EXP00210
C     DETERMINE POWER OF TWO AND NORMALIZATION FACTOR                   EXP00220
C ****                                                                  EXP00230
      M = 0                                                             EXP00240
      IF (ANORM .LE. 1.)  GO TO 27                                      EXP00250
      FACTOR =2.                                                        EXP00260
      DO 15 M=1,46                                                      EXP00270
         IF (ANORM .LE. FACTOR) GO TO 20                                EXP00280
         FACTOR = FACTOR*2.                                             EXP00290
   15 CONTINUE                                                          EXP00300
      GO TO 125                                                         EXP00310
   20 CONTINUE                                                          EXP00320
C ****                                                                  EXP00330
C     NORMALIZE MATRIX                                                  EXP00340
C ****                                                                  EXP00350
      DO 25 I=1,N                                                       EXP00360
         DO 25 J=1,N                                                    EXP00370
            A(I,J) = A(I,J)/FACTOR                                      EXP00380
   25 CONTINUE                                                          EXP00390
   27 CONTINUE                                                          EXP00400
C ****                                                                  EXP00410
C     SET COEFFICIENTS FOR (9,9) PADE TABLE ENTRY                       EXP00420
C ****                                                                  EXP00430
      C(1) = .5                                                         EXP00440
      C(2) = 1.17647058823529D-01                                       EXP00450
      C(3) = 1.71568627450980D-02                                       EXP00460
      C(4) = 1.71568627450980D-03                                       EXP00470
      C(5) = 1.22549019607840D-04                                       EXP00480
      C(6) = 6.28456510809450D-06                                       EXP00490
      C(7) = 2.24448753860510D-07                                       EXP00500
      C(8) = 5.10110804228450D-09                                       EXP00510
      C(9) = 5.66789782476050D-11                                       EXP00520
C ****                                                                  EXP00530
C     CALCULATE PADE NUMERATOR AND DENOMINATOR BY COLUMNS               EXP00540
C ****                                                                  EXP00550
      NP1 = N+1                                                         EXP00560
      NP7 = N+7                                                         EXP00570
      DO 95 J=1,N                                                       EXP00580
C ****                                                                  EXP00590
C        COMPUTE JTH COLUMN OF FIRST NINE POWERS OF A                   EXP00600
C ****                                                                  EXP00610
         DO 35 I=1,N                                                    EXP00620
            S = 0.                                                      EXP00630
```

```
      DO 30 L=1,N                                                       EXP00640
         S = S + A(I,L)*A(L,J)                                          EXP00650
30    CONTINUE                                                          EXP00660
      WK(I,NP1) = S                                                     EXP00670
35    CONTINUE                                                          EXP00680
      DO 45 K=NP1,NP7                                                   EXP00690
         KP1 = K+1                                                      EXP00700
         DO 45 I=1,N                                                    EXP00710
            S = 0.                                                      EXP00720
            DO 40 L=1,N                                                 EXP00730
               S = S + A(I,L)*WK(L,K)                                   EXP00740
40          CONTINUE                                                    EXP00750
            WK(I,KP1) = S                                               EXP00760
45    CONTINUE                                                          EXP00770
C ****                                                                  EXP00780
C     COLLECT TERMS FOR JTH COLUMN OF NUMERATOR AND DENOMINATOR         EXP00790
C ****                                                                  EXP00800
      DO 85 I=1,N                                                       EXP00810
         S = 0.                                                         EXP00820
         U = 0.                                                         EXP00830
         DO 65 L=1,8                                                    EXP00840
            K = N+9-L                                                   EXP00850
            KN1 = K-N+1                                                 EXP00860
            P = C(KN1)*WK(I,K)                                          EXP00870
            S = S + P                                                   EXP00880
            IEO = MOD(KN1,2)                                            EXP00890
            IF (IEO.EQ.0) GO TO 55                                      EXP00900
            U = U - P                                                   EXP00910
            GO TO 65                                                    EXP00920
55          CONTINUE                                                    EXP00930
            U = U + P                                                   EXP00940
65       CONTINUE                                                       EXP00950
         P = C(1)*A(I,J)                                                EXP00960
         S = S + P                                                      EXP00970
         U = U - P                                                      EXP00980
         IF (I .NE. J) GO TO 80                                         EXP00990
         S = S + 1.                                                     EXP01000
         U = U + 1.                                                     EXP01010
80       CONTINUE                                                       EXP01020
         EA(I,J) = S                                                    EXP01030
         WK(I,J) = U                                                    EXP01040
85    CONTINUE                                                          EXP01050
95 CONTINUE                                                             EXP01060
C ****                                                                  EXP01070
C     CALCULATE NORMALIZED EXP(A) BY  WK * EXP(A) = EA                  EXP01080
C ****                                                                  EXP01090
      CALL GAUSEL (MAX,N,WK,N,EA,IERR)                                  EXP01100
      IF (IERR .NE. 0) GO TO 130                                        EXP01110
      IF (M .EQ. 0)  GO TO 130                                          EXP01120
C ****                                                                  EXP01130
C     TAKE OUT EFFECT OF NORMALIZATION ON EXP(A)                        EXP01140
C ****                                                                  EXP01150
      DO 120 K=1,M                                                      EXP01160
         DO 110 I=1,N                                                   EXP01170
            DO 110 J=1,N                                                EXP01180
               S = 0.                                                   EXP01190
               DO 105 L=1,N                                             EXP01200
                  S = S + EA(I,L)*EA(L,J)                               EXP01210
105            CONTINUE                                                 EXP01220
               WK(I,J) = S                                              EXP01230
110   CONTINUE                                                          EXP01240
      DO 115 I=1,N                                                      EXP01250
         DO 115 J=1,N                                                   EXP01260
```

```
                EA(I,J) = WK(I,J)                                       EXP01270
  115     CONTINUE                                                      EXP01280
  120   CONTINUE                                                        EXP01290
C ****                                                                  EXP01300
C     UN-NORMALIZE A                                                    EXP01310
C ****                                                                  EXP01320
      DO 122 I=1,N                                                      EXP01330
         DO 122 J=1,N                                                   EXP01340
            A(I,J) = A(I,J)*FACTOR                                      EXP01350
  122 CONTINUE                                                          EXP01360
      GO TO 130                                                         EXP01370
C ****                                                                  EXP01380
C     NORM OF A IS EXCESSIVE                                            EXP01390
C ****                                                                  EXP01400
  125 CONTINUE                                                          EXP01410
      IERR = 1                                                          EXP01420
C ****                                                                  EXP01430
C     EXIT ROUTINE                                                      EXP01440
C ****                                                                  EXP01450
  130 CONTINUE                                                          EXP01460
      RETURN                                                            EXP01470
      END                                                               EXP01480
```

```
      SUBROUTINE EXPINT(A,NA,B,NB,C,NC,T,IOP,DUMMY)                     EXP0C
      IMPLICIT REAL*8 (A-H,O-Z)                                         EXP0C
      DIMENSION A(1),B(1),C(1),DUMMY(1)                                 EXP00
      DIMENSION NA(2),NB(2),NC(2)                                       EXP00
      COMMON/CONV/SUMCV,RICTCV,SERCV,MAXSUM                             EXP00
      N = NA(1)                                                         EXP00
      L = (N**2)+1                                                      EXP00
      NC(1) = NA(1)                                                     EXP00
      NC(2) = NA(2)                                                     EXP00
      NB(1) = NA(1)                                                     EXP00
      NB(2) = NA(2)                                                     EXP00
      TT = T                                                            EXP00
C                                                                       EXP00
      IOPT = 1                                                          EXP00
      CALL NORMS(N,N,N,A,IOPT,COL)                                      EXP00
      IOPT = 3                                                          EXP00
      CALL NORMS(N,N,N,A,IOPT,ROW)                                      EXP00
      ANAA = COL                                                        EXP00
      IF( ANAA .GT. ROW ) ANAA = ROW                                    EXP00
      TMAX = 1./ANAA                                                    EXP00
      K = 0                                                             EXP00
  100 CONTINUE                                                          EXP00
      IF( TMAX - TT ) 125,150,150                                       EXP00
  125 CONTINUE                                                          EXP00
      K = K + 1                                                         EXP00
      TT = T/(2**K)                                                     EXP00
      IF( K - 1000 )100,600,600                                         EXP00
C                                                                       EXP00
  150 CONTINUE                                                          EXP00
      SC = TT                                                           EXP00
      CALL SCALE(A,NA,A,NA,TT)                                          EXP00
      CALL UNITY(B,NB)                                                  EXP00
      CALL SCALE(B,NB,DUMMY,NB,TT)                                      EXP00
      S = TT/2.                                                         EXP00
      CALL SCALE(A,NA,DUMMY(L),NA,S)                                    EXP003
      II = 2                                                            EXP003
      CALL ADD(DUMMY,NA,DUMMY(L),NA,DUMMY(L),NA)                        EXP003
      CALL ADD(A,NA,B,NB,DUMMY,NA)                                      EXP003
      CALL EQUATE(A,NA,C,NC)                                            EXP003
  200 CONTINUE                                                          EXP004
      CALL MULT(A,NA,C,NC,B,NB)                                         EXP004
      S = 1./II                                                         EXP004
      CALL SCALE(B,NB,C,NC,S)                                           EXP004
      CALL MAXEL(DUMMY,NA,TOT)                                          EXP004
      CALL MAXEL(C,NC,DELT)                                             EXP004
      IF( TOT .GT. 1.0 ) GO TO 300                                      EXP004
      IF( DELT/TOT .LT. SERCV )  GO TO 400                              EXP004
      GO TO 350                                                         EXP004
  300 CONTINUE                                                          EXP004
      IF( DELT .LT. SERCV )  GO TO 400                                  EXP005
  350 CONTINUE                                                          EXP005
      S = TT/(II + 1)                                                   EXP005
      CALL SCALE(C,NC,B,NB,S)                                           EXP005
      CALL ADD(B,NB,DUMMY(L),NB,DUMMY(L),NB)                            EXP005
      CALL ADD(C,NC,DUMMY,NC,DUMMY,NC)                                  EXP005
      II = II + 1                                                       EXP005
      GO TO 200                                                         EXP005
C                                                                       EXP005
  400 CONTINUE                                                          EXP005
      CALL EQUATE(DUMMY,NB,B,NB)                                        EXP006
      IF( K ) 425,500,450                                               EXP006
  425 CONTINUE                                                          EXP006
      CALL LNCNT(1)                                                     EXP006
```

```
      PRINT 435                                                          EXP0064(
  435 FORMAT('  ERROR IN EXPINT, K IS NEGATIVE')                         EXP0065(
      RETURN                                                             EXP0066(
C                                                                        EXP0067(
  450 CONTINUE                                                           EXP006A(
      DO 475 J = 1,K                                                     EXP0069(
      TT = 2*TT                                                          EXP0070(
      CALL EQUATE(B,NB,DUMMY,NB)                                         EXP0071(
      CALL MULT(DUMMY,NA,DUMMY(L),NA,C,NC)                               EXP0072(
      CALL ADD(DUMMY(L),NC,C,NC,DUMMY(L),NC)                             EXP0073(
      CALL MULT(DUMMY,NB,DUMMY,NB,B,NB)                                  EXP0074(
  475 CONTINUE                                                           EXP0075(
      T = TT                                                             EXP0076(
C                                                                        EXP0077(
  500 CONTINUE                                                           EXP0078(
      CALL EQUATE(DUMMY(L),NC,C,NC)                                      EXP0079(
      S = 1./SC                                                          EXP0080(
      CALL SCALE(A,NA,A,NA,S)                                            EXP0081(
C                                                                        EXP0082(
      IF( IOP .EQ. 0 ) RETURN                                            EXP0083(
      CALL LNCNT(5)                                                      EXP0084(
      PRINT 550                                                          EXP0085(
  550 FORMAT(//,' COMPUTATION OF THE MATRIX EXPONENTIAL  EXP(A T)',/,' AEXP0086(
     1ND ITS INTEGRAL OVER  (0,T) BY THE SERIES METHOD ',/)              EXP0087(
      CALL PRNT(A,NA,4H A  ,1)                                           EXP0088(
      CALL LNCNT(3)                                                      EXP0089(
      PRINT 575, T                                                       EXP0090(
  575 FORMAT(/,'  T = ',D16.8,/)                                         EXP0091(
      CALL  PRNT(B,NB,4HEXPA,1)                                          EXP0092
      CALL PRNT(C,NC,4HINT ,1)                                           EXP0093
      RETURN                                                             EXP0094
C                                                                        EXP0095
  600 CONTINUE                                                           EXP0096
      CALL LNCNT(1)                                                      EXP0097
      PRINT 650                                                          EXP0098
  650 FORMAT( ' ERROR IN EXPINT, K = 1000 ')                             EXP0099
      RETURN                                                             EXP0100
C                                                                        EXP0101
      END                                                                EXP0102
```

```
      SUBROUTINE VARANC(A,NA,G,NG,Q,NQ,W,NW,IDENT,DISC,IOP,DUMMY)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(1),G(1),Q(1),W(1),DUMMY(1)
      DIMENSION NA(2),NG(2),NQ(2),NW(2),NDUM1(2),IOP(3),IOPT(2)
      LOGICAL IDENT,DISC,SYM
      COMMON/TOL/EPSAM,EPSBM,IACM
      IF( IOP(1) .EQ. 0 ) GO TO 100
      CALL LNCNT(5)
      IF( DISC ) PRINT 25
      IF( .NOT. DISC ) PRINT 35
   25 FORMAT(//,' PROGRAM TO SOLVE FOR THE STEADY-STATE VARIANCE MATRIX'
     1,/,' FOR A LINEAR DISCRETE SYSTEM',/)
   35 FORMAT(//,' PROGRAM TO SOLVE FOR THE STEADY-STATE VARIANCE MATRIX'
     1,/,' FOR A LINEAR CONTINUOUS SYSTEM',/)
      CALL PRNT(A,NA,4H A  ,1)
      IF( .NOT. IDENT ) GO TO 55
      CALL LNCNT(3)
      PRINT 45
   45 FORMAT(/,' G IS AN IDENTITY MATRIX ',/)
      GO TO 65
   55 CONTINUE
      CALL PRNT(G,NG,4H G  ,1)
   65 CONTINUE
      IF ( .NOT. IDENT ) GO TO 85
      CALL LNCNT(3)
      PRINT 75
   75 FORMAT(/,' INTENSITY MATRIX FOR COVARIANCE OF PROCESS NOISE ',/)
C
   85 CONTINUE
      CALL PRNT(Q,NQ,4H Q  ,1)
C
  100 CONTINUE
      IF( IDENT ) GO TO 200
      CALL MULT(G,NG,Q,NQ,DUMMY,NG)
      N1 = NG(1)*NG(2) + 1
      CALL TRANP(G,NG,DUMMY(N1),NDUM1)
      CALL MULT(DUMMY,NG,DUMMY(N1),NDUM1,Q,NQ)
C
      IF( IOP(1) .EQ. 0 ) GO TO 200
      CALL LNCNT(3)
      PRINT 75
      CALL PRNT(Q,NQ,4HGQGT,1)
C
  200 CONTINUE
      IF(.NOT. DISC) CALL SCALE(W,NW,W,NW,-1.0)
      IOPT(1) = IOP(2)
      IOPT(2) = 1
      SYM = .TRUE.
      IF( DISC ) GO TO 300
      IF( IOP(3) .EQ. 0 ) GO TO 250
      CALL BILIN(A,NA,A,NA,W,NW,IOPT,BETA,SYM,DUMMY)
      GO TO 400
C
  250 CONTINUE
      CALL BARSTW(A,NA,A,NA,W,NW,IOPT,SYM,EPSA,EPSA,DUMMY)
      GO TO 400
C
  300 CONTINUE
      CALL EQUATE(A,NA,DUMMY,NA)
      N = NA(1)**2
      N1 = N + 1
      CALL TRANP(A,NA,DUMMY(N1),NA)
      N2 = N1 + N
```

```
      CALL SUM(DUMMY,NA,W,NW,DUMMY(N1),NA,IOPT,SYM,DUMMY(N2))
C
  400 CONTINUE
      IF( IOP(1) .EQ. 0 ) RETURN
      CALL LNCNT(3)
      PRINT 450
  450 FORMAT(/, ' VARIANCE MATRIX ',/)
      CALL PRNT(W,NW,4H W  ,1)
C
      RETURN
      END
```

```
      SUBROUTINE CTROL(A,NA,B,NB,C,NC,IOP,IAC,IRANK,DUMMY)                CTR000
      IMPLICIT REAL*8 (A-H,O-Z)                                           CTR000
      DIMENSION A(1),B(1),C(1),DUMMY(1)                                   CTR000
      DIMENSION NA(2),NB(2),NC(2),NV(2),IOP(5)                            CTR000
      N = NA(1)*NB(2)                                                     CTR000
      N1 = N+1                                                            CTR000
      N2 = N1+N                                                           CTR000
      K = NA(1)-1                                                         CTR000
      J = 1                                                               CTR000
C                                                                         CTR001
      CALL EQUATE(B,NB,DUMMY(N2),NV)                                      CTR001
      CALL EQUATE(B,NB,DUMMY,NB)                                          CTR001
  100 CONTINUE                                                            CTR001
      CALL MULT(A,NA,DUMMY,NB,DUMMY(N1),NB)                               CTR001
      CALL JUXTC(DUMMY(N2),NV,DUMMY(N1),NB,C,NC)                          CTR001
C                                                                         CTR001
      IF( J .EQ. K ) GO TO 200                                            CTR001
C                                                                         CTR001
      CALL EQUATE(DUMMY(N1),NB,DUMMY,NB)                                  CTR001
      CALL EQUATE(C,NC,DUMMY(N2),NV)                                      CTR002
      J = J + 1                                                           CTR002
      GO TO 100                                                           CTR002
C                                                                         CTR002
  200 CONTINUE                                                            CTR002
C                                                                         CTR002
      IF(IOP(1) .EQ. 0 ) GO TO 300                                        CTR002
      CALL PRNT(A,NA,4H A  ,1)                                            CTR002
      CALL PRNT(B,NB,4H B  ,1)                                            CTR002
      CALL LNCNT(4)                                                       CTR002
      PRINT 250                                                           CTR003
  250 FORMAT(//,' THE MATRIX C IS THE CONTROLLABILITY MATRIX FOR THE  A/CTR003
     1B PAIR',/)                                                          CTR003
      CALL PRNT(C,NC,4H C  ,1)                                            CTR003
C                                                                         CTR003
  300  IF( IOP(2) .EQ. 0 ) RETURN                                         CTR003
      NOS = 0                                                             CTR003
      IOPT = 2                                                            CTR003
      K = NC(2)                                                           CTR003
      NC(2) = NB(2)*(NA(2)-NB(2)+1)                                       CTR003
      N = NC(1)*NC(2)                                                     CTR004
      CALL TRANP(C,NC,DUMMY,NV)                                           CTR004
      NC(2) = K                                                           CTR004
      N1 = N + 1                                                          CTR004
      N2 = N1 + NV(2)                                                     CTR004
      CALL SNVDEC(IOPT,NV(1),NV(2),NV(1),NV(2),DUMMY,NOS,B,IAC,ZTEST,DUMCTR004
     1MY(N1),DUMMY(N2),IRANK,A,IERR)                                      CTR004
      IF( IERR .EQ. 0 ) GO TO 340                                         CTR00
      CALL LNCNT(5)                                                       CTR004
      IF( IERR .GT. 0 ) PRINT 310,IERR                                    CTR004
      IF( IERR .EQ. -1 ) PRINT 320, ZTEST,IRANK                           CTR005
  310 FORMAT(//,' IN CTROL, SNVDEC HAS FAILED TO CONVERGE TO THE ',I4, 'CTR005
     1 SINGULAR VALUE AFTER 30 ITERATIONS ')                              CTR005
  320 FORMAT(//,' IN CTROL, THE MATRIX SUBMITTED TO SNVDEC USING ZTEST =CTR005
     1',D16.8,' IS CLOSE TO A MATRIX WHICH IS OF LOWER RANK',/,' IF THE CTR005
     2ACCURACACY IS REDUCED THE RANK MAY ALSO BE REDUCED',/,' CURRENT RACTR005
     3NK =',I4)                                                           CTR005
      IF( IEPR .GT. 0 ) RETURN                                            CTR005
C                                                                         CTR005
  340 CONTINUE                                                            CTR005
      IF( IOP(3) .EQ. 0 ) GO TO 400                                       CTR006
      CALL LNCNT(6)                                                       CTR006
      PRINT 350,ZTEST,IRANK                                               CTR006
  350 FORMAT(//,' BASED ON THE ZERO-TEST ',D16.8,' THE RANK OF THE CONTRCTR006
```

```
      1OLLARILITY MATRIX IS ',I4,/,' THE SINGULAR VALUES ARE ',/)          CTR0
      IOPT = 0                                                             CTR0
      NV(1)= NV(2)                                                         CTR0
      NV(2)= 1                                                             CTR0
      CALL PRNT(DUMMY(N1),NV,IOPT,3)                                       CTR0
C                                                                          CTR0
  400 IF( IOP(4) .EQ. 0 ) RETURN                                           CTR0
      N = NA(1)**2                                                         CTR0
      CALL EQUATE(DUMMY(N2),NA,DUMMY,NA)                                   CTR0
      N1 = N + 1                                                           CTR0
      N2 = N1 + N                                                          CTR0
      CALL MULT(A,NA,DUMMY,NA,DUMMY(N1),NA)                                CTR0
      CALL TRANP(DUMMY,NA,DUMMY(N2),NA)                                    CTR0
      CALL EQUATE(DUMMY(N2),NA,DUMMY,NA)                                   CTR0
      CALL MULT(DUMMY,NA,DUMMY(N1),NA,DUMMY(N2),NA)                        CTR0
      CALL MULT(DUMMY,NA,B,NB,DUMMY(N1),NB)                                CTR0
C                                                                          CTR0
      IF( IOP(5) .EQ. 0 ) RETURN                                           CTR0
      CALL LNCNT(5)                                                        CTR0
      PRINT 500                                                            CTR0
  500 FORMAT(//,' CONTROLLABILITY CANONICAL FORM ',/,' (V TRANSPOSE) A     CTR0
     1 V')                                                                 CTR0
      CALL PRNT(DUMMY(N2),NA,IOPT,3)                                       CTR0
      CALL LNCNT(2)                                                        CTR0
      PRINT 510                                                            CTR0
  510 FORMAT(/,' (V TRANSPOSE ) B ')                                       CTR0
      CALL PRNT(DUMMY(N1),NB,IOPT,3)                                       CTR0
      CALL LNCNT(2)                                                        CTR0
      PRINT 520                                                            CTR0
  520 FORMAT(/,' V TRANSPOSE')                                             CTR0
      CALL PRNT(DUMMY,NA,IOPT,3)                                           CTR0
C                                                                          CTR0
      RETURN                                                               CTR0
      END                                                                  CTR0
```

```
      SUBROUTINE TRNSIT(A,NA,B,NB,H,NH,G,NG,F,NF,V,NV,T,X,NX,DISC,STABL TRN00(
     1E,IOP,DUMMY)                                                      TRN00(
      IMPLICIT REAL*8 (A-H,O-Z)                                         TRN00(
      DIMENSION A(1),B(1),H(1),G(1),F(1),V(1),X(1),DUMMY(1)             TRN00(
      DIMENSION NA(2),NB(2),NH(2),NG(2),NF(2),NV(2),NX(2),T(2),IOP(4)   TRN00(
      DIMENSION NDUM1(2),NDUM2(2)                                       TRN00(
      LOGICAL  DISC,STABLE                                              TRN00(
      N = NA(1)*NA(2)                                                   TRN00(
      N1 = N + 1                                                        TRN00(
      N2 = N + N1                                                       TRN001
      N3 = N + N2                                                       TRN001
      N4 = N + N3                                                       TRN001
      N5 = N + N4                                                       TRN001
      N6 = N + N5                                                       TRN001
C                                                                       TRN001
      CALL LNCNT(4)                                                     TRN001
      IF(DISC) PRINT 100                                                TRN001
      IF( .NOT. DISC ) PRINT 120                                        TRN001
  100 FORMAT(//,' COMPUTATION OF TRANSIENT RESPONSE FOR THE DIGITAL SYSTTRN001
     1EM ',/)                                                           TRN002
  120 FORMAT(//,' COMPUTATION OF TRANSIENT RESPONSE FOR THE CONTINUOUS  TRN002
     1 SYSTEM',/)                                                       TRN002
      CALL PRNT(A,NA,4H A  ,1)                                          TRN002
      CALL PRNT(B,NB,4H B  ,1)                                          TRN002
      IF( (IOP(1) .NE. 1) .AND. (IOP(1) .NE. 0) )  GO TO 180            TRN002
      CALL LNCNT(3)                                                     TRN002
      IF( IOP(1) .EQ. 0 ) PRINT 140                                     TRN002
      IF( IOP(1) .EQ. 1 ) PRINT 160                                     TRN002
  140 FORMAT(//,' H IS A NULL MATRIX ')                                 TRN002
  160 FORMAT(//,' H IS AN IDENTITY MATRIX ')                            TRN003
      GO TO 200                                                         TRN003
  180 CONTINUE                                                          TRN003
      CALL PRNT(H,NH,4H H  ,1)                                          TRN003
  200 CONTINUE                                                          TRN003
      IF( (IOP(2) .NE. 1) .AND. (IOP(2) .NE. 0) )  GO TO 260            TRN003
      CALL LNCNT(3)                                                     TRN003
      IF( IOP(2) .EQ. 0 ) PRINT 220                                     TRN003
      IF( IOP(2) .EQ. 1 ) PRINT 240                                     TRN003
  220 FORMAT(//,' G IS A NULL MATRIX')                                  TRN003
  240 FORMAT(//,' G IS AN IDENTITY MATRIX')                             TRN004
      GO TO 280                                                         TRN004
  260 CONTINUE                                                          TRN004
      CALL PRNT(G,NG,4H G  ,1)                                          TRN004
  280 CONTINUE                                                          TRN004
      CALL PRNT(F,NF,4H F  ,1)                                          TRN004
      IF( (IOP(3) .NE. 0) .AND. (IOP(3) .NE. 1) )  GO TO 295            TRN004
      CALL LNCNT(3)                                                     TRN004
      IF(IOP(3).EQ.0) PRINT 285                                         TRN004
      IF(IOP(3).EQ.1) PRINT 290                                         TRN004
  285 FORMAT(//,' V IS A NULL MATRIX')                                  TRN005
  290 FORMAT(//,' V IS AN IDENTITY MATRIX')                             TRN005
      GO TO 300                                                         TRN005
  295 CONTINUE                                                          TRN005
      CALL PRNT(V,NV,4H V  ,1)                                          TRN005
C                                                                       TRN005
  300 CONTINUE                                                          TRN005
      CALL EQUATE(A,NA,DUMMY(N6),NA)                                    TRN005
      CALL MULT(B,NB,F,NF,DUMMY,NA)                                     TRN005
      CALL SUBT(A,NA,DUMMY,NA,A,NA)                                     TRN005
C                                                                       TRN006
      IF(DISC) GO TO 350                                                TRN006
      NMAX = T(1)/T(2)                                                  TRN006
      IOPT = 1
```

```
      TT = T(2)                                                          TRN00
      IF( IOP(3) .NE. 0 )  GO TO 315                                     TRN00
      CALL EXPSER(A,NA,DUMMY,NA,TT,IOPT,DUMMY(N1))                       TRN00
      GO TO 400                                                          TRN00
  315 CONTINUE                                                           TRN00
      CALL EXPINT(A,NA,DUMMY,NA,DUMMY(N1),NA,TT,IOPT,DUMMY(N2))          TRN00
      CALL MULT(DUMMY(N1),NA,B,NB,DUMMY(N2),NB)                          TRN00
      IF( IOP(3) .NE. 1 ) GO TO 325                                      TRN00
      CALL EQUATE(DUMMY(N2),NB,DUMMY(N1),NX)                             TRN00
      GO TO 400                                                          TRN00
  325 CONTINUE                                                           TRN00
      CALL MULT(DUMMY(N2),NB,V,NV,DUMMY(N1),NX)                          TRN00
      GO TO 400                                                          TRN00
  350 CONTINUE                                                           TRN00
      NMAX = IOP(4)                                                      TRN00
      CALL EQUATE(A,NA,DUMMY,NA)                                         TRN00
      IF( IOP(3) .EQ. 0 ) GO TO 400                                      TRN00
      CALL MULT(B,NB,V,NV,DUMMY(N1),NX)                                  TRN00
C                                                                        TRN00
  400 CONTINUE                                                           TRN00
      CALL LNCNT(4)                                                      TRN00
      PRINT 420                                                          TRN00
  420 FORMAT(//,' STRUCTURE OF PRINTING TO FOLLOW',/)                    TRN00
      CALL LNCNT(6)                                                      TRN00
      PRINT 440                                                          TRN00
  440 FORMAT('   TIME OR STAGE ',//,'  STATE = X TRANSPOSE = FROM DX = AXTRN00
     1 + BU',/,'  OUTPUT = Y TRANSPOSE = FROM Y = HX + GU   IF DIFFERENTTRN00
     2 FROM X',/,'  CONTROL = U TRANSPOSE = FROM U = -FX + V',//)       TRN00
C                                                                        TRN00
      K = 0                                                              TRN00
      L = 0                                                              TRN00
      CALL SCALE(F,NF,F,NF,-1.0)                                         TRN00
C                                                                        TRN00
  450 CONTINUE                                                           TRN00
      IF( K .GT. NMAX ) GO TO 800                                        TRN00
      CALL MULT(F,NF,X,NX,DUMMY(N2),NV)                                  TRN00
      IF( IOP(3) .NE. 0 ) CALL ADD(DUMMY(N2),NV,V,NV,DUMMY(N2),NV)       TRN01
      CALL MULT(DUMMY,NA,X,NX,DUMMY(N3),NX)                              TRN01
      IF( IOP(3) .EQ. 0 ) GO TO 475                                      TRN01
      CALL ADD(DUMMY(N1),NX,DUMMY(N3),NX,DUMMY(N3),NX)                   TRN01
  475 CONTINUE                                                           TRN01
      IF( IOP(2) .EQ. 0 ) GO TO 525                                      TRN01
      IF( IOP(2) .EQ. 1 ) GO TO 500                                      TRN01
      CALL MULT(G,NG,DUMMY(N2),NV,DUMMY(N4),NDUM1)                       TRN01
      GO TO 525                                                          TRN01
  500 CONTINUE                                                           TRN01
      CALL EQUATE(DUMMY(N2),NV,DUMMY(N4),NDUM1)                          TRN01
  525 CONTINUE                                                           TRN01
      IF( IOP(1) .EQ. 0 ) GO TO 575                                      TRN01
      IF( IOP(1) .EQ. 1 ) GO TO 550                                      TRN01
      CALL MULT(H,NH,X,NX,DUMMY(N5),NDUM1)                               TRN01
      GO TO 575                                                          TRN01
  550 CONTINUE                                                           TRN01
      CALL EQUATE(X,NX,DUMMY(N5),NDUM1)                                  TRN01
  575 CONTINUE                                                           TRN01
      IF( IOP(2) .EQ. 0 ) GO TO 600                                      TRN01
      IF( IOP(1) .EQ. 0 ) GO TO 700                                      TRN01
      CALL ADD(DUMMY(N4),NDUM1,DUMMY(N5),NDUM1,DUMMY(N4),NDUM1)          TRN01
      GO TO 700                                                          TRN01
  600 CONTINUE                                                           TRN01
      IF( IOP(1) .NE. 0 ) CALL EQUATE(DUMMY(N5),NDUM1,DUMMY(N4),NDUM1)   TRN01
C                                                                        TRN01
  700 CONTINUE                                                           TRN01
```

```
      CALL LNCNT(5)                                                     TRN0
      IF( .NOT. DISC ) GO TO 720                                        TRN0
      PRINT 710,K                                                       TRN0
  710 FORMAT(////,I5)                                                   TRN0
      GO TO 740                                                         TRN0
  720 CONTINUE                                                          TRN0
      TIME=K*T(2)                                                       TRN0
      PRINT 730,TIME                                                    TRN0
  730 FORMAT(////,D16.7)                                                TRN0
  740 CONTINUE                                                          TRN0
      CALL TRANP(X,NX,DUMMY(N5),NDUM2)                                  TRN0
      CALL PRNT(DUMMY(N5),NDUM2,L,3)                                    TRN0
      IF( (IOP(2) .EQ. 0) .AND. ( (IOP(1) .EQ. 0) .OR. (IOP(1) .EQ. 1) )TRN0
     1) GO TO 750                                                       TRN0
      CALL TRANP(DUMMY(N4),NDUM1,DUMMY(N5),NDUM2)                       TRN0
      CALL PRNT(DUMMY(N5),NDUM2,L,3)                                    TRN0
  750 CONTINUE                                                          TRN0
      CALL TRANP(DUMMY(N2),NV,DUMMY(N5),NDUM2)                          TRN0
      CALL PRNT(DUMMY(N5),NDUM2,L,3)                                    TRN0
C                                                                       TRN0
      CALL EQUATE(DUMMY(N3),NX,X,NX)                                    TRN0
      K = K + 1                                                         TRN0
      GO TO 450                                                         TRN0
C                                                                       TRN0
C                                                                       TRN0
  800 CONTINUE                                                          TRN0
      CALL SCALE(F,NF,F,NF,-1.0)                                        TRN0
      IF( .NOT. STABLE .OR. IOP(3) .EQ. 0 ) GO TO 900                   TRN0
      IF( IOP(3) .EQ. 1 ) GO TO 820                                     TRN0
      CALL MULT(B,NB,V,NV,DUMMY,NX)                                     TRN0
      GO TO 840                                                         TRN0
  820 CONTINUE                                                          TRN0
      CALL EQUATE(B,NB,DUMMY,NX)                                        TRN0
  840 CONTINUE                                                          TRN0
      IF( .NOT. DISC ) GO TO 860                                        TRN0
      CALL UNITY(DUMMY(N1),NA)                                          TRN0
      CALL SUBT(DUMMY(N1),NA,A,NA,A,NA)                                 TRN0
  860 CONTINUE                                                          TRN0
      IFAC = 0                                                          TRN0
      CALL GELIM(NA(1),NA(1),A,NX(2),DUMMY,DUMMY(N1),IFAC,DUMMY(N2),IERRTRN0
     1)                                                                 TRN0
      IF( IERR .EQ. 0 ) GO TO 880                                       TRN0
      CALL LNCNT(3)                                                     TRN0
      IF( .NOT. DISC ) PRINT 865                                        TRN0
      IF( DISC ) PRINT 870                                              TRN0
  865 FORMAT(//,' IN TRNSIT, THE MATRIX A-BF SUBMITTED TO GELIM IS SINGUTRN0
     1LAR')                                                             TRN0
  870 FORMAT(//,' IN TRNSIT, THE MATRIX  I - (A-BF) SUBMITTED TO GELIM ITRN0
     1S SINGULAR')                                                      TRN0
      GO TO 900                                                         TRN0
  880 CONTINUE                                                          TRN0
      IF( .NOT. DISC ) CALL SCALE(DUMMY,NX,DUMMY,NX,-1.0)               TRN0
      CALL LNCNT(5)                                                     TRN0
      PRINT 890                                                         TRN0
  890 FORMAT(////,' STEADY-STATE VALUE OF  X TRANSPOSE')                TRN0
      CALL TRANP(DUMMY,NX,DUMMY(N5),NDUM2)                              TRN0
      CALL PRNT(DUMMY(N5),NDUM2,L,3)                                    TRN0
C                                                                       TRN0
  900 CONTINUE                                                          TRN0
      CALL EQUATE(DUMMY(N6),NA,A,NA)                                    TRN0
C                                                                       TRN0
      RETURN                                                            TRN0
      END                                                               TRN0
```

```
      SUBROUTINE SAMPL(A,NA,B,NB,Q,NQ,R,NR,W,NW,T,IOP,DUMMY)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(1),B(1),Q(1),R(1),W(1),DUMMY(1)
      DIMENSION NA(2),NB(2),NQ(2),NR(2),NW(2),IOP(2),NDUM(2)
      COMMON/CONV/SUMCV,RICTCV,SERCV,MAXSUM
      IF(  IOP(1) .EQ. 0 ) GO TO 100
C
      IF(  IOP(2) .EQ. 0 ) GO TO 50
C
      CALL LNCNT(5)
      PRINT 25
   25 FORMAT(//,' COMPUTATION OF WEIGHTING MATRICES FOR THE OPTIMAL SAMP
     1LED-DATA REGULATOR PROBLEM',//)
      CALL PRNT(A,NA,4H A  ,1)
      CALL PRNT(B,NB,4H B  ,1)
      CALL LNCNT(3)
      PRINT 35
   35 FORMAT(//,' CONTINUOUS PERFORMANCE INDEX WEIGHTING MATRICES',/)
      CALL PRNT(Q,NQ,4H Q  ,1)
      CALL PRNT(R,NR,4H R  ,1)
      CALL LNCNT(3)
      PRINT 45,T
   45 FORMAT(/,' SAMPLE TIME = ',D16.8,/)
C
      GO TO 100
C
   50 CONTINUE
      CALL LNCNT(8)
      PRINT 75
   75 FORMAT(//,' COMPUTATION OF THE RECONSTRUCTIBILITY GRAMIAN',/,' FOR
     1 THE (A,H) SYSTEM OVER THE INTERVAL (0,T) ',/,' THE MATRIX Q IS  (
     2 H TRANSPOSE ) X H',//)
      CALL PRNT(A,NA,4H A  ,1)
      CALL PRNT(Q,NQ,4H Q  ,1)
      CALL LNCNT(3)
      PRINT 85,T
   85 FORMAT(/,' T = ',D16.8,/)
C
  100 CONTINUE
C
      N = NA(1)
      L = ( N**2)
      N1 = L + 1
      N2 = N1 + L
      TT  = T
C
      IOPT = 1
      CALL NORMS(N,N,N,A,IOPT,ANORM)
      IOPT = 3
      CALL NORMS(N,N,N,A,IOPT,ROWA)
      IF( ANORM .GT. ROWA ) ANORM = ROWA
C
      IF( ANORM .LE. 1.E-15 ) GO TO 900
C
      TMAX = 1.0/ANORM
      K = 0
C
  125 CONTINUE
      IF( TMAX - TT ) 150,150,200
  150 CONTINUE
      K = K + 1
      TT = T/( 2**K)
      IF( K - 1000 ) 125,800,800
```

```
      C                                                                 SAM0C
  200 CONTINUE                                                          SAM0C
      C                                                                 SAM0C
          I = 0                                                         SAM0C
          SC = TT                                                       SAM0C
          CALL SCALE(A,NA,A,NA,TT)                                      SAM0C
          CALL SCALE(Q,NQ,Q,NQ,TT)                                      SAM0C
          CALL EQUATE(Q,NQ,DUMMY,NQ)                                    SAM00
      C                                                                 SAM00
          IF( IOP(2) .NE. 0 ) GO TO 500                                 SAM00
      C                                                                 SAM00
  225 CONTINUE                                                          SAM00
          II = I + 2                                                    SAM00
          I = I + 1                                                     SAM00
          F = 1.0/II                                                    SAM00
          CALL SCALE(A,NA,DUMMY(N1),NA,F)                               SAM00
          CALL MULT(DUMMY,NA,DUMMY(N1),NA,DUMMY(N2),NA)                 SAM00
          CALL TRANP(DUMMY(N2),NA,DUMMY(N1),NA)                         SAM00
          CALL ADD(DUMMY(N1),NA,DUMMY(N2),NA,DUMMY,NA)                  SAM00
      C                                                                 SAM00
          CALL MAXEL(Q,NQ,TOT)                                          SAM00
          CALL MAXEL(DUMMY,NA,DELT)                                     SAM00
          IF( TOT .GT. 1.0 ) GO TO 250                                  SAM00
          IF( DELT/TOT .LT. SERCV )  GO TO 300                          SAM00
          GO TO 275                                                     SAM00
  250 CONTINUE                                                          SAM00
          IF( DELT .LT. SERCV )  GO TO 300                              SAM00
  275 CONTINUE                                                          SAM00
          CALL ADD(Q,NQ,DUMMY,NA,Q,NQ)                                  SAM00
          GO TO 225                                                     SAM00
      C                                                                 SAM00
  300 CONTINUE                                                          SAM00
      C                                                                 SAM00
          IF( K .EQ. 0 ) GO TO 400                                      SAM00
          N3 = N2 + L                                                   SAM00
          G = 1.0                                                       SAM00
          IOPT = 0                                                      SAM01
          CALL EXPSER(A,NA,DUMMY,NA,G,IOPT,DUMMY(N1))                   SAM01
      C                                                                 SAM01
  350 CONTINUE                                                          SAM01
          IF( K .EQ. 0 ) GO TO 400                                      SAM01
          K = K-1                                                       SAM01
      C                                                                 SAM01
          CALL TRANP(DUMMY,NA,DUMMY(N1),NA)                             SAM01
          CALL MULT(Q,NQ,DUMMY,NA,DUMMY(N2),NA)                         SAM01
          CALL MULT(DUMMY(N1),NA,DUMMY(N2),NA,DUMMY(N3),NA)             SAM01
          CALL ADD(Q,NQ,DUMMY(N3),NA,Q,NQ)                              SAM011
          CALL MULT(DUMMY,NA,DUMMY,NA,DUMMY(N1),NA)                     SAM011
          CALL EQUATE(DUMMY(N1),NA,DUMMY,NA)                            SAM011
      C                                                                 SAM011
          GO TO 350                                                     SAM011
      C                                                                 SAM011
  400 CONTINUE                                                          SAM011
          S =  1.0/SC                                                   SAM011
          CALL SCALE(A,NA,A,NA,S)                                       SAM011
      C                                                                 SAM011
          IF( IOP(1) .EQ. 0 ) RETURN                                    SAM012
          CALL PRNT(Q,NQ,4HGRAM,1)                                      SAM012
          RETURN                                                        SAM012
      C                                                                 SAM012
  500 CONTINUE                                                          SAM012
          CALL SCALE(B,NB,B,NB,TT)                                      SAM012
          N3 = N2 + L                                                   SAM012
```

```
      N4 = N3 + L                                                       SAM012
      N5 = N4 + L                                                       SAM012
      N6 = N5 + L                                                       SAM012
C                                                                       SAM013
  525 CONTINUE                                                          SAM013
      II = I + 2                                                        SAM013
      I = I + 1                                                         SAM013
      F = 1.0/II                                                        SAM013
      CALL SCALE(A,NA,DUMMY(N1),NA,F)                                   SAM013
      CALL TRANP(DUMMY(N1),NA,DUMMY(N2),NA)                             SAM013
      CALL MULT(DUMMY,NA,DUMMY(N1),NA,DUMMY(N3),NA)                    SJM013
      CALL TRANP(DUMMY(N3),NA,DUMMY(N1),NA)                             SJM013
      CALL MULT(DUMMY,NA,B,NB,DUMMY(N5),NW)                             SAM013
      CALL ADD(DUMMY(N1),NA,DUMMY(N3),NA,DUMMY,NA)                      SAM014
      CALL SCALE(DUMMY(N5),NW,DUMMY(N1),NW,F)                           SAM014
      IF( I .NE. 1 ) GO TO 550                                          SAM014
      CALL EQUATE(DUMMY(N1),NW,W,NW)                                    SJM014
      CALL EQUATE(DUMMY(N1),NW,DUMMY(N6),NW)                            SAM014
      CALL ADD(Q,NQ,DUMMY,NQ,Q,NQ)                                      SAM014
      GO TO 525                                                         SAM014
C                                                                       SAM014
  550 CONTINUE                                                          SAM014
      CALL MULT(DUMMY(N2),NA,DUMMY(N6),NW,DUMMY(N5),NW)                 SAM014
      CALL ADD(DUMMY(N5),NW,DUMMY(N1),NW,DUMMY(N1),NW)                  SAM015
      CALL TRANP(B,NB,DUMMY(N2),NDUM)                                   SAM015
      CALL SCALE(DUMMY(N2),NDUM,DUMMY(N2),NDUM,F)                       SAM015
      CALL MULT(DUMMY(N2),NDUM,DUMMY(N6),NW,DUMMY(N3),NR)               SAM015
      CALL TRANP(DUMMY(N3),NR,DUMMY(N5),NR)                             SAM015
      CALL ADD(DUMMY(N3),NR,DUMMY(N5),NR,DUMMY(N3),NR)                  SAM015
      CALL EQUATE(DUMMY(N1),NW,DUMMY(N6),NW)                            SAM015
      IF( I .NE. 2 ) GO TO 575                                          SAM015
      CALL ADD(Q,NQ,DUMMY,NQ,Q,NQ)                                      SAM015
      CALL ADD(W,NW,DUMMY(N1),NW,W,NW)                                  SAM015
      CALL EQUATE(DUMMY(N3),NR,DUMMY(N4),NR)                            SAM016
      GO TO 525                                                         SJM016
C                                                                       SAM016
  575 CONTINUE                                                          SAM016
      CALL MAXEL(Q,NQ,TOT)                                              SAM016
      CALL MAXEL(DUMMY,NQ,DELT)                                         SAM016
      IF( TOT .GT. 1.0 )  GO TO 580                                     SAM016
      IF( DELT/TOT .LT. SERCV )  GO TO 585                              SAM016
      GO TO 595                                                         SAM016
C                                                                       SAM016
  580 CONTINUE                                                          SAM017
      IF( DELT .LT. SERCV )  GO TO 585                                  SAM017
      GO TO 595                                                         SAM017
C                                                                       SAM017
  585 CONTINUE                                                          SAM017
      CALL MAXEL(DUMMY(N4),NR,TOT)                                      SAM017
      CALL MAXEL(DUMMY(N3),NR,DELT)                                     SAM017
      IF( TOT .GT. 1.0 )  GO TO 590                                     SAM017
      IF( DELT/TOT .LT. SERCV )  GO TO 600                              SAM017
      GO TO 595                                                         SAM017
C                                                                       SAM018
  590 CONTINUE                                                          SAM018
      IF( DELT .LT. SERCV )  GO TO 600                                  SAM018
C                                                                       SAM018
  595 CONTINUE                                                          SAM018
      CALL ADD (Q,NQ,DUMMY,NQ,Q,NQ)                                     SAM018
      CALL ADD(W,NW,DUMMY(N1),NW,W,NW)                                  SAM018
      CALL ADD(DUMMY(N4),NR,DUMMY(N3),NR,DUMMY(N4),NR)                  SAM018
      GO TO 525                                                         SAM018
C                                                                       SAM018
```

```
  600 CONTINUE                                                          SAM01
      IF( K .EQ. 0 ) GO TO 700                                          SAM01
      G = 1.0                                                           SAM01
      IOPT = 0                                                          SAM01
      CALL EXPINT(A,NA,DUMMY,NA,DUMMY(N1),NA,G,IOPT,DUMMY(N2))          SAM01
      CALL MULT(DUMMY(N1),NA,B,NB,DUMMY(N2),NB)                         SAM01
      CALL EQUATE(DUMMY(N2),NB,DUMMY(N1),NB)                            SAM01
C                                                                       SAM01
  650 CONTINUE                                                          SAM01
      IF( K .EQ. 0 ) GO TO 700                                          SAM01
      K = K - 1                                                         SAM02
      CALL MULT(Q,NQ,DUMMY,NA,DUMMY(N2),NA)                             SAM02
      CALL TRANP(DUMMY,NA,DUMMY(N3),NA)                                 SAM02
      CALL MULT(DUMMY(N3),NA,DUMMY(N2),NA,DUMMY(N5),NA)                 SAM02
      CALL MULT(Q,NQ,DUMMY(N1),NB,DUMMY(N2),NB)                         SAM02
      CALL ADD(Q,NQ,DUMMY(N5),NA,Q,NQ)                                  SAM02
      CALL MULT(DUMMY(N3),NA,DUMMY(N2),NB,DUMMY(N5),NB)                 SAM02
      CALL MULT(DUMMY(N3),NA,W,NW,DUMMY(N6),NW)                         SAM02
      CALL ADD(DUMMY(N5),NW,DUMMY(N6),NW,DUMMY(N5),NW)                  SAM02
      CALL TRANP(DUMMY(N1),NB,DUMMY(N6),NDUM)                           SAM02
      CALL MULT(DUMMY(N6),NDUM,W,NW,DUMMY(N3),NR)                       SAM02
      CALL ADD(W,NW,DUMMY(N5),NW,W,NW)                                  SAM02
      CALL MULT(DUMMY(N6),NDUM,DUMMY(N2),NB,DUMMY(N5),NR)               SAM02
      CALL ADD(DUMMY(N5),NR,DUMMY(N3),NR,DUMMY(N5),NR)                  SAM02
      CALL TRANP(DUMMY(N3),NR,DUMMY(N6),NR)                             SAM02
      CALL ADD(DUMMY(N5),NR,DUMMY(N6),NR,DUMMY(N6),NR)                  SAM02
      CALL SCALE(DUMMY(N4),NR,DUMMY(N4),NR,2.0)                         SAM02
      CALL ADD(DUMMY(N6),NR,DUMMY(N4),NR,DUMMY(N4),NR)                  SAM02
      CALL MULT(DUMMY,NA,DUMMY(N1),NB,DUMMY(N3),NB)                     SAM02
      CALL ADD(DUMMY(N3),NB,DUMMY(N1),NB,DUMMY(N1),NB)                  SAM02
      CALL MULT(DUMMY,NA,DUMMY,NA,DUMMY(N3),NA)                         SAM02
      CALL EQUATE(DUMMY(N3),NA,DUMMY,NA)                                SAM02
      GO TO 650                                                         SAM02
C                                                                       SAM02
  700 CONTINUE                                                          SAM02
      CALL SCALE(R,NR,R,NR,T)                                           SAM02
      CALL ADD(R,NR,DUMMY(N4),NR,R,NR)                                  SAM02
      CALL SCALE(W,NW,W,NW,2.0)                                         SAM02
      S = 1.0/SC                                                        SAM02
      CALL SCALE(A,NA,A,NA,S)                                           SAM02
      CALL SCALE(B,NB,B,NB,S)                                           SAM02
      IF( IOP(1) .EQ. 0 ) RETURN                                        SAM02
C                                                                       SAM02
      CALL LNCNT(3)                                                     SAM02
      PRINT 750                                                         SAM02
  750 FORMAT(/,' DISCRETE PERFORMANCE INDEX WEIGHTING MATRICES',/)      SAM02
      CALL PRNT(Q,NQ,4H Q  ,1)                                          SAM02
      CALL PRNT(W,NW,4H W  ,1)                                          SAM02
      CALL PRNT(R,NR,4H R  ,1)                                          SAM02
      RETURN                                                            SAM02
C                                                                       SAM02
  800 CONTINUE                                                          SAM02
      CALL LNCNT(1)                                                     SAM02
      PRINT 850                                                         SAM02
  850 FORMAT(' ERROR IN SAMPL , K = 1000')                              SAM02
      RETURN                                                            SAM02
C                                                                       SAM02
  900 CONTINUE                                                          SAM02
      CALL SCALE(Q,NQ,Q,NQ,T)                                           SAM02
      IF( IOP(2) .NE. 0 ) GO TO 925                                     SAM02
      IF( IOP(1) .NE. 0 ) CALL PRNT(Q,NQ,4HGRAM,1)                      SAM02
      RETURN                                                            SAM02
C                                                                       SAM02
```

```
  925 CONTINUE                                                          SAM02
      CALL MULT(Q,NQ,B,NB,W,NW)                                         SAM02
      CALL SCALE(W,NW,W,NW,T)                                           SAM02
      CALL TRANP(B,NB,DUMMY,NDUM)                                       SAM02
      CALL MULT(DUMMY,NDUM,W,NW,DUMMY(N1),NR)                           SAM02
      TT = T/3.                                                         SJM02
      CALL SCALE(DUMMY(N1),NR,DUMMY,NR,TT)                              SAM02
      CALL SCALE(R,NR,R,NW,T)                                           SAM02
      CALL ADD(R,NR,DUMMY,NR,R,NR)                                      SAM02
      IF( IOP(1) .EQ. 0 ) RETURN                                        SAM02
      CALL LNCNT(3)                                                     SAM02
      PRINT 750                                                         SAM02
      CALL PRNT(Q,NQ,4H Q  ,1)                                          SAM02
      CALL PRNT(W,NW,4H W  ,1)                                          SAM02
      CALL PRNT(R,NR,4H R  ,1)                                          SAM02
      RETURN                                                            SAM02
C                                                                       SAM02
      END                                                               SAM02
```

```
      SUBROUTINE PREFIL(A,NA,B,NB,Q,NQ,W,NW,R,NR,F,NF,IOP,DUMMY)          PREO(
      IMPLICIT REAL*8 (A-H,O-Z)                                           PREO(
      DIMENSION A(1),B(1),Q(1),W(1),R(1),F(1),DUMMY(1)                    PREO(
      DIMENSION NA(2),NB(2),NQ(2),NW(2),NR(2),NF(2),IOP(3)                PREO(
      IF( IOP(1) .EQ. 0 ) GO TO 100                                       PREO(
      CALL LNCNT(5)                                                       PREO(
      PRINT 25                                                            PREO(
   25 FORMAT(//,' PROGRAM TO COMPUTE PREFILTER GAIN F TO ELEMINATE  CROSPREO(
     1S-PRODUCT TERM ',/,' IN QUADRATIC PERFORMANCE INDEX ',/)           PREO(
      IF( IOP(3) .EQ. 0 ) GO TO 50                                        PREO(
      CALL PRNT(A,NA,4H A  ,1)                                            PREO(
      CALL  PRNT(B,NB,4H B  ,1)
   50 CONTINUE                                                            PREO0
      CALL PRNT(Q,NQ,4H Q  ,1)                                            PREO0
      CALL PRNT(W,NW,4H W  ,1)                                            PREO0
      CALL PRNT(R,NR,4H R  ,1)                                            PREO0
C                                                                         PREO0
  100 CONTINUE                                                            PREO0
      CALL TRANP(W,NW,F,NF)                                               PREO0
      CALL SCALE(F,NF,F,NF,0.5)                                           PREO0
      CALL EQUATE(R,NR,DUMMY,NR)                                          PREO0
      IOPT=0                                                              PREO0
      IFAC=0
      N1=NR(1)**2+1
      M = NR(1)                                                           PREO0
      CALL SYMPDS(M,M,DUMMY,NF(2),F,IOPT,IFAC,DETERM,ISCALE,DUMMY(N1),IEPREO0
     1RR)                                                                 PREO0
      IF( IERR .EQ. 0 ) GO TO 200                                         PREO0
      CALL LNCNT(4)                                                       PREO0
      PRINT 150                                                           PREO0
  150 FORMAT(//,' IN PREFIL, THE MATRIX R IS NOT SYMMETRIC POSITIVE DEFIPREO0
     1NITE',/)                                                            PREO0
      RETURN                                                              PREO0
C                                                                         PREO0
  200 CONTINUE                                                            PREO0
      IF( IOP(2) .EQ. 0 ) GO TO 300                                       PREO0
      CALL MULT(W,NW,F,NF,DUMMY,NQ)                                       PREO0
      CALL SCALE(DUMMY,NQ,DUMMY,NQ,0.5)                                   PREO0
      CALL SUBT(Q,NQ,DUMMY,NQ,Q,NQ)                                       PREO0
C                                                                         PREO0
  300 CONTINUE                                                            PREO0
      IF( IOP(3) .EQ. 0 ) GO TO 400                                       PREO0
      CALL MULT(B,NB,F,NF,DUMMY,NA)                                       PREO0
      CALL SUBT(A,NA,DUMMY,NA,A,NA)                                       PREO0
C                                                                         PREO0
  400 CONTINUE                                                            PREO0
      IF( IOP(1) .EQ. 0 ) RETURN                                          PREO0
      CALL PRNT(F,NF,4H F  ,1)                                            PREO0
      IF( IOP(2) .EQ. 0 ) GO TO 500                                       PREO0
      CALL LNCNT(3)                                                       PREO0
      PRINT 450                                                           PREO0
  450 FORMAT(/, ' MATRIX  Q = (W/2)F ',/)                                 PREO0
      CALL PRNT(Q,NQ,4HNEWQ,1)                                            PREO0
C                                                                         PREO0
  500 CONTINUE                                                            PREO0
      IF( IOP(3) .EQ. 0 ) RETURN                                          PREO0
      CALL PRNT(A,NA,4HNEWA,1)                                            PREO0
      RETURN                                                              PREO0
      END
```

```
      SUBROUTINE CSTAB(A,NA,B,NB,F,NF,IOP,SCLE,DUMMY)                    CST0
      IMPLICIT REAL*8 (A-H,O-Z)                                          CST0
      DIMENSION A(1),B(1),F(1),DUMMY(1)                                  CST0
      DIMENSION NA(2),NB(2),NF(2),IOP(3),NDUM(2)                         CST0
      DIMENSION IOPT(2)                                                  CST0
      LOGICAL SYM                                                        CST0
      COMMON/TOL/EPSAM,EPSBM,IACM                                        CST0
      N = NA(1)**2                                                       CST0
      N1=N+1                                                             CST0
C                                                                        CST0
      IF(IOP(2) .EQ. 0 ) GO TO 100                                       CST0
      CALL EQUATE(A,NA,DUMMY,NA)                                         CST0
      N2=N1+NA(1)                                                        CST0
      N3=N2+NA(1)                                                        CST0
      ISV =0                                                             CST 0
      ILV =0                                                             CST00
      CALL EIGEN(NA(1),NA(1),DUMMY,DUMMY(N1),DUMMY(N2),ISV,ILV,V,DUMMY(NCST00
     13),IERR)                                                           CST00
C                                                                        CST00
      M=NA(1)                                                            CST00
      IF(IERR .EQ. 0) GO TO 50                                           CST00
      CALL LNCNT(3)                                                      CST00
      PRINT 25,IERR                                                      CST00
   25 FORMAT(//,' IN CSTAB, THE SUBROUTINE EIGEN FAILED TO DETERMINE THECST00
     1 ',I4,' EIGENVALUE FOR THE MATRIX A AFTER 30 ITERATIONS')          CST00
      IERR=1                                                             CST00
      CALL NORMS(M,M,M,A,IERR,BETA)                                      CST00
      BETA=2.*BETA                                                       CST00
      GO TO 200                                                          CST00
   50 CONTINUE                                                           CST00
C                                                                        CST00
      BETA = 0.0                                                         CST00
      DO 75 I = 1,M                                                      CST00
      J = N1 + I - 1                                                     CST00
      BETA1 = DABS(DUMMY(J))                                             CST00
      IF(BETA1 .GT. BETA)  BETA = BETA1                                  CST00
   75 CONTINUE                                                           CST00
      BETA = SCLE*(BETA + .001)                                          CST00
      GO TO 200                                                          CST00
C                                                                        CST00
  100 CONTINUE                                                           CST00
      BETA = SCLE                                                        CST00
  200 CONTINUE                                                           CST00
C                                                                        CST00
      CALL TRANP(B,NB,DUMMY,NDUM)                                        CST00
      CALL MULT(B,NB,DUMMY,NDUM,DUMMY(N1),NA)                            CST00
      CALL SCALE(DUMMY(N1),NA,DUMMY,NA,-2.0)                             CST00
      CALL SCALE(A,NA,DUMMY(N1),NA,-1.0)                                 CST00
      J = -NA(1)                                                         CST00
      NAX = NA(1)                                                        CST00
      DO 225 I=1,NAX                                                     CST00
      J = J+NAX+1                                                        CST00
      K = N1+J-1                                                         CST00
      DUMMY(K)=DUMMY(K)-BETA                                             CST00
  225 CONTINUE                                                           CST00
      N2 = N1 + N                                                        CST00
      SYM = .TRUE.                                                       CST00
      IOPT(1)=0                                                          CST00
C                                                                        CST00
      IF( IOP(3) .NE. 0 )  GO TO 300                                     CST00
      EPSA=EPSAM                                                         CST00
      CALL BARSTW(DUMMY(N1),NA,A,NA,DUMMY,NA,IOPT,SYM,EPSA,EPSA,DUMMY(N2CST00
     1))                                                                 CST00
```

```
      GO TO 350                                                          CST0(
  300 CONTINUE                                                           CST0(
      IOPT(2) = 1                                                        CST0(
      CALL BILIN(DUMMY(N1),NA,A,NA,DUMMY,NA,IOPT,ASCLE,SYM,DUMMY(N2))    CST0(
  350 CONTINUE                                                           CST0(
C                                                                        CST0(
      CALL EQUATE(B,NB,DUMMY(N1),NB)                                     CST0(
      IOPT(1) = 3                                                        CST0(
      IAC =IACM                                                          CST0(
      N3 = N2 + NA(1)                                                    CST0(
      CALL SNVDEC(IOPT,NA(1),NA(1),NA(1),NA(1),DUMMY,NB(2),DUMMY(N1),IACCST0(
     1,ZTEST,DUMMY(N2),DUMMY(N3),IRANK,APLUS,IERR)                       CST0(
      IF(IERR .EQ. 0 ) GO TO 400                                         CST0(
      CALL LNCNT(5)                                                      CST0(
      IF(IERR .GT. 0 ) PRINT 360,IERR                                    CST00
      IF(IERR .EQ. -1) PRINT 370,ZTEST,IRANK                             CST00
  360 FORMAT(//,' IN CSTAB, SNVDEC HAS FAILED TO CONVERGE TO THE ',I4,' CST00
     1SINGULARVALUE AFTER 30 ITERATIONS',//)                             CST00
  370 FORMAT(//,' IN CSTAB, THE MATRIX SUBMITTED TO SNVDEC USING ZTEST =CST00
     1 ',D16.8,' IS CLOSE TO A  MATRIX OF LOWER  RANK ',/,' IF THE ACCURCST00
     2ACY IAC IS REDUCED THE RANK MAY ALSO BE REDUCED',/,' CURRENT RANK CST00
     3 =',I4)                                                            CST00
      IF( IERR .GT. 0 ) RETURN                                           CST00
      NDUM(1) = NA(1)                                                    CST00
      NDUM(2) =1                                                         CST00
      CALL PRNT(DUMMY(N2),NDUM,4HSGVL,1)                                 CST00
  400 CONTINUE                                                           CST00
C                                                                        CST00
      CALL TRANP(DUMMY(N1),NB,F,NF)                                      CST00
      IF ( IOP(1) .EQ. 0 )   RETURN                                      CST00
      CALL LNCNT(4)                                                      CST00
      PRINT 500                                                          CST00
  500 FORMAT(//,' COMPUTATION OF F MATRIX SUCH THAT A-BF IS ASYMPTOTICALCST00
     1LY STABLE IN THE CONTINUOUS SENSE ',/)                             CST00
      CALL PRNT(A,NA,4H A  ,1)                                           CST00
      CALL LNCNT(4)                                                      CST00
      PRINT 550,BETA                                                     CST01
  550 FORMAT(//,' BETA = ',D16.8,/)                                      CST01
      CALL PRNT(B,NB,4H B  ,1)                                           CST01
      CALL PRNT(F,NF,4H F  ,1)                                           CST01
      CALL MULT(B,NB,F,NF,DUMMY,NA)                                      CST01
      CALL SUBT(A,NA,DUMMY,NA,DUMMY,NA)                                  CST01
      CALL PRNT(DUMMY,NA,4HA-BF,1)                                       CST01
      N2 = N1+NA(1)                                                      CST01
      N3 = N2+NA(1)                                                      CST01
      ISV = 0                                                            CST01
      ILV = 0                                                            CST01
      CALL EIGEN(NA(1),NA(1),DUMMY,DUMMY(N1),DUMMY(N2),ISV,ILV,V,DUMMY(NCST01
     13),IERR)                                                           CST01
      ' = NA(1)                                                          CST01
      IF( IERR .EQ. 0 ) GO TO 600                                        CST01
      M = NA(1)-IERR                                                     CST01
      CALL LNCNT(3)                                                      CST01
      PRINT 25,IERR                                                      CST01
  600 CONTINUE                                                           CST01
      CALL LNCNT(4)                                                      CST01
      PRINT 650                                                          CST01
  650 FORMAT(//,' EIGENVALUES OF A-BF',/)                                CST01
  675 FORMAT(10X,2D16.8)                                                 CST01
      CALL LNCNT(M)                                                      CST01
      DO 700 I=1,M                                                       CST01
      ' = N1+I-1                                                         CST01
      K = N2+I-1                                                         CST01
```

```
126           PRINT 675,DUMMY(J),DUMMY(K)                              CS
127     700   CONTINUE                                                 CS
128   C                                                                CS
129           RETURN                                                   CST
130           END                                                      CST
```

```
      SUBROUTINE DSTAB(A,NA,B,NB,F,NF,SING,IOP,SCLE,DUMMY)                DST
      IMPLICIT REAL*8 (A-H,O-Z)                                           DST
      DIMENSION A(1),B(1),F(1),DUMMY(1)                                   DST
      DIMENSION NA(2),NB(2),NF(2),NDUM(2),IOP(2),IOPT(3),NDUM1(2)         DST
      LOGICAL SING,SYM                                                    DST
      COMMON/TOL/EPSAM,EPSBM,IACM                                         DST
      N = NA(1)**2                                                        DST
      N1 = N + 1                                                          DST
      N2 = N1 + N                                                         DST
      IF( .NOT. SING ) GO TO 100                                          DST
      IOPT(1)=IOP(1)                                                      DST
      IOPT(2) = 1                                                         DST
      IOPT(3) = 0                                                         DST
      CSCLE=1.05                                                          DST
      CALL CSTAB(A,NA,B,NB,F,NF,IOPT,CSCLE,DUMMY)                         DST
      CALL MULT(B,NB,F,NF,DUMMY,NA)                                       DST
      CALL SUBT(A,NA,DUMMY,NA,DUMMY,NA)                                   DST
      CALL EQUATE(DUMMY,NA,DUMMY(N1),NA)                                  DST
      GO TO 200                                                           DST
C                                                                         DST
  100 CONTINUE                                                            DST
      CALL EQUATE(A,NA,DUMMY,NA)                                          DST
      CALL EQUATE(A,NA,DUMMY(N1),NA)                                      DST
C                                                                         DST
  200 CONTINUE                                                            DST
      IF( IOP(2) .EQ. 0 ) GO TO 300                                       DST
      N3 = N2 + NA(1)                                                     DST
      N4 = N3 + NA(1)                                                     DST
      ISV = 0                                                             DST
      CALL EIGEN(NA(1),NA(1),DUMMY(N1),DUMMY(N2),DUMMY(N3),ISV,ISV,V,DUMDST
     1MY(N4),IERR)                                                        DST
      CALL EQUATE(DUMMY,NA,DUMMY(N1),NA)                                  DST
      M = NA(1)                                                           DST
      IF( IERR .EQ. 0 ) GO TO 250                                         DST
      CALL LNCNT(3)                                                       DST
      PRINT  225, IERR
  225 FORMAT(//'IN DSTAB , THE PROGRAM EIGEN FAILED TO DETERMINE',
     1'IS, 'EIGENVALUE FOR THE MATRIX A-BG AFTER 30 ITERATIONS ')
      CALL PRNT(DUMMY,NA,4HA-BG,1)                                        DST
      IF( SING ) CALL PRNT(F,NF,4H G  ,1)                                 DST
      RETURN                                                              DST
C                                                                         DST
  250 CONTINUE                                                            DST
      ALPHA = 1.0                                                         DST
      DO 275 I =1,M                                                       DST
      I1 = N2 + I -1                                                      DST
      I2 = N3 + I -1                                                      DST
      ALPHA1 = DSQRT(DUMMY(I1)**2 + DUMMY(I2)**2)                         DST
      IF( ALPHA1 .LT. ALPHA .AND. ALPHA1 .NE. 0 ) ALPHA = ALPHA1          DST
  275 CONTINUE                                                            DST
      ALPHA = SCLE*ALPHA                                                  DST
      GO TO 400                                                           DST
C                                                                         DST
  300 CONTINUE                                                            DST
      ALPHA = SCLE                                                        DST
C                                                                         DST
  400 CONTINUE                                                            DST
      J = -NA(1)                                                          DST
      NAX = NA(1)                                                         DST
      DO 425 I = 1,NAX                                                    DST
      J = J + NAX + 1                                                     DST
      K = N1 + J -1                                                       DST
      DUMMY(K) = DUMMY(J) - ALPHA                                         DST
```

```
      DUMMY(J) = DUMMY(J) + ALPHA                                       DST
  425 CONTINUE                                                          DST
      CALL EQUATE(B,NB,DUMMY(N2),NB)                                    DST
      N3 = N2 + NA(1)*NB(2)                                             DST
      NRHS = NA(1)+NB(2)                                                DST
      N4 = N3 + NA(1)                                                   DST
      IFAC = 0                                                          DST
      CALL GELIM(NA(1),NA(1),DUMMY,NRHS,DUMMY(N1),DUMMY(N3),IFAC,DUMMY(NDST
     14),IERR)                                                          DST
      IF( IERR .EQ. 0 ) GO TO 500                                       DST
      CALL LNCNT(3)                                                     DST
      IF( .NOT. SING ) GO TO 445                                        DST
      PRINT 435                                                         DST
  435 FORMAT(//,' IN DSTAB, GELIM HAS FOUND THE MATRIX ( A-BG) + (ALPHA)DSTC
     1I SINGULA  ')                                                     DSTC
      CALL PRNT(A,NA,4H A  ,1)                                          DSTC
      CALL PRNT(F,NF,4H G  ,1)                                          DSTC
      GO TO 465                                                         DSTC
  445 CONTINUE                                                          DSTC
      CALL LNCNT(3)                                                     DSTC
      PRINT 455                                                         DSTC
  455 FORMAT(//,' IN DSTAB, GELIM HAS FOUND THE MATRIX A + (ALPHA)I SINGDSTC
     1ULAR ')                                                           DSTC
      CALL PRNT(A,NA,4H A  ,1)                                          DSTC
  465 CONTINUE                                                          DSTC
      CALL LNCNT(3)                                                     DSTC
      PRINT 475,ALPHA                                                   DSTC
  475 FORMAT(//,' ALPHA = ',D16.8)                                      DSTC
      RETURN                                                            DSTC
C                                                                       DSTC
  500 CONTINUE                                                          DSTC
      CALL EQUATE(DUMMY(N1),NA,DUMMY,NA)                                DSTC
      CALL TRANP(DUMMY(N2),NB,DUMMY(N1),NDUM)                           DSTC
      N3 = N2 + N                                                       DSTC
      CALL MULT(DUMMY(N2),NB,DUMMY(N1),NDUM,DUMMY(N3),NA)               DSTC
      CALL SCALE(DUMMY(N3),NA,DUMMY(N1),NA,4.0)                         DSTC
      SYM = .TRUE.                                                      DST1
      IOPT(1) = 0                                                       DST1
      EPSA=EPSAM                                                        DST1
      CALL BARSTW(DUMMY,NA,B,NB,DUMMY(N1),NA,IOPT,SYM,EPSA,EPSA,DUMMY(N2DST1
     1))                                                                DST1
      CALL EQUATE(DUMMY(N1),NA,DUMMY,NA)                                DST1
      CALL TRANP(B,NB,DUMMY(N1),NDUM)                                   DST1
      CALL MULT(B,NB,DUMMY(N1),NDUM,DUMMY(N2),NA)                       DST1
      CALL ADD(DUMMY,NA,DUMMY(N2),NA,DUMMY,NA)                          DST1
      CALL EQUATE(A,NA,DUMMY(N1),NA)                                    DST1
      IF( .NOT. SING ) GO TO 600                                        DST1
      CALL MULT(B,NB,F,NF,DUMMY(N1),NA)                                 DST1
      CALL SUBT(A,NA,DUMMY(N1),NA,DUMMY(N1),NA)                         DST1
C                                                                       DST1
  600 CONTINUE                                                          DST1
      IOPT(1) = 3                                                       DST1
      M = NA(1)                                                         DST1
      IAC=IACM                                                          DST1
      CALL SNVDEC(IOPT,M,M,M,M,DUMMY,M,DUMMY(N1),IAC,ZTEST,DUMMY(N2),DUMDST1
     1MY(N3),IRANK,APLUS,IERR)                                          DST1
      IF( IERR .EQ. 0 ) GO TO 700                                       DST1
      CALL LNCNT(5)                                                     DST1
      IF( IERR .GT. 0 ) PRINT 625,IERR                                  DST1
      IF( IERR .EQ. -1) PRINT 650,ZTEST,IRANK                           DST1
  625 FORMAT(//,' IN DSTAB, SNVDEC HAS FAILED TO CONVERGE TO THE ',I5,' DST1
     1SINGULAR VALUE AFTER 30 ITERATIONS')                              DST1
  650 FORMAT(//,' IN DSTAB, THE MATRIX SUBMITTED TO SNVDEC, USING ZTEST DST1
```

```
     1= ',D16.8,' , IS CLOSE TO A MATRIX OF LOWER RANK',/,' IF THE ACCURDST
    2ACY IAC IS REDUCED THE RANK MAY ALSO BE REDUCED',/,' CURRENT RANK DST
    3 =',I4)                                                            DST
      IF( IERR .GT. 0 ) RETURN                                          DST
      NDUM(1)= NA(1)                                                    DST
      NDUM(2)= 1                                                        DST
      CALL PRNT(DUMMY(N2),NDUM,4HSGVL,1)                                DST
C                                                                       DST
  700 CONTINUE                                                          DST
      CALL TRANP(B,NB,DUMMY(N2),NDUM)                                   DST
      CALL MULT(DUMMY(N2),NDUM,DUMMY(N1),NA,DUMMY,NF)                   DST
      IF( .NOT. SING ) GO TO 800                                        DST
      CALL ADD(F,NF,DUMMY,NF,F,NF)                                      DST
      GO TO 900                                                         DST
C                                                                       DST
  800 CONTINUE                                                          DST
      CALL EQUATE(DUMMY,NF,F,NF)                                        DST
C                                                                       DST
  900 CONTINUE                                                          DST
      IF( IOP(1) .EQ. 0 ) RETURN                                        DST
      CALL LNCNT(4)                                                     DST
      PRINT 1000                                                        DST
 1000 FORMAT(//,' COMPUTATION OF F SUCH THAT A-BF IS ASYMPTOTICALLY STABDST
     1LE IN THE DISCRETE SENSE',/)                                      DST
      CALL PRNT(A,NA,4H A  ,1)                                          DST
      CALL PRNT(B,NB,4H B  ,1)                                          DST
      CALL LNCNT(4)                                                     DST
      PRINT 1100,ALPHA                                                  DST
 1100 FORMAT(//,' ALPHA = ',D16.8,/)                                    DST
      CALL PRNT(F,NF,4H F  ,1)                                          DST
      CALL MULT(B,NB,F,NF,DUMMY,NA)                                     DST
      CALL SUBT(A,NA,DUMMY,NA,DUMMY,NA)                                 DST
      CALL PRNT(DUMMY,NA,4HA-BF,1)                                      DST
      CALL LNCNT(3)                                                     DST
      PRINT 1200                                                        DST
 1200 FORMAT(//,' EIGENVALUES OF A-BF')                                 DST
      NDUM(1) = NA(1)                                                   DST
      NDUM(2) = 1                                                       DST
      N2 = N1 + NA(1)                                                   DST
      N3 = N2 + NA(1)                                                   DST
      ISV = 0                                                           DST
      CALL EIGEN(NA(1),NA(1),DUMMY,DUMMY(N1),DUMMY(N2),ISV,ISV,V,DUMMY(NDST
     13),IERR)                                                          DST
      IF( IERR .EQ. 0 ) GO TO 1300                                      DST
      CALL LNCNT(3)                                                     DST
      PRINT 1250                                                        DST
 1250 FORMAT(//,' IN DSTAB, THE PROGRAM EIGEN FAILED TO DETERMINE THE ',DST
     1I5,' EIGENVALUE FOR THE A-BF MATRIX AFTER 30 ITERATIONS')         DST
      NDUM(1)=NA(1)-IERR                                                DST
C                                                                       DST
 1300 CONTINUE                                                          DST
      CALL JUXTC(DUMMY(N1),NDUM,DUMMY(N2),NDUM,DUMMY,NDUM1)             DST
      CALL PRNT(DUMMY,NDUM1,4HEIGN,1)                                   DST
      CALL LNCNT(4)                                                     DST
      PRINT 1400                                                        DST
 1400 FORMAT(//,' MODULI OF EIGENVALUES OF A-BF',/)                     DST
      M =NDUM(1)                                                        DST
      DO 1500 I = 1,M                                                   DST
      J = N1 + I - 1                                                    DST
      K = N2 + I - 1                                                    DST
      DUMMY(I)=DSQRT(DUMMY(J)**2 + DUMMY(K)**2)                         DST
 1500 CONTINUE                                                          DST
      CALL PRNT(DUMMY,NDUM,4HMOD ,1)                                    DST
```

```
18  C                                                          DST
19                RETURN                                       DST
191               END                                          DST
```

```
      SUBROUTINE DISREG(A,NA,B,NB,H,NH,Q,NQ,R,NR,F,NF,P,NP,IOP,IDENT,DU DIS
     1MMY)                                                              DIS
      IMPLICIT REAL*8 (A-H,O-Z)                                         DIS
      DIMENSION A(1),B(1),Q(1),R(1),F(1),P(1),DUMMY(1)                  DIS
      DIMENSION NA(2),NB(2),NQ(2),NR(2),NF(2),NP(2)                     DIS
      DIMENSION IOP(3)                                                  DIS
      DIMENSION H(1),NH(2),NDUM(2)                                      DIS
      LOGICAL  IDENT                                                    DIS
      COMMON/TOL/EPSAM,EPSBM,IACM                                       DIS
      COMMON/CONV/SUMCV,RICTCV,SERCV,MAXSUM                             DIS
      N = NA(1)**2                                                      DIS
      N1= N +1                                                          DIS
      N2= N1+N                                                          DIS
      N3= N2+N                                                          DIS
C                                                                       DIS
      KSS = 0                                                           DIS0
      I=IOP(3)                                                          DIS0
C                                                                       DIS0
      IF(IOP(1) .EQ. 0)  GO TO 85                                       DIS0
      CALL LNCNT(5)                                                     DIS0
      PRINT 25                                                          DIS0
   25 FORMAT(//,' PROGRAM TO SOLVE THE TIME-INVARIANT FINITE-DURATION OPDIS0
     1TIMAL',/,' DIGITAL REGULATOR PROBLEM WITH NOISE-FREE MEASUREMENTS'DIS0
     2,/)                                                               DIS0
      CALL PRNT(A,NA,4H A  ,1)                                          DIS0
      CALL PRNT(B,NB,4H B  ,1)                                          DIS0
      CALL PRNT(Q,NQ,4H Q  ,1)                                          DIS0
      IF( .NOT. IDENT )  GO TO 45                                       DIS0
      CALL LNCNT(3)                                                     DIS0
      PRINT 35                                                          DIS0
   35 FORMAT(/,' H IS AN IDENTITY MATRIX',/)                            DIS0
      GO TO 65                                                          DIS0
   45 CONTINUE                                                          DIS0
      CALL PRNT(H,NH,4H H  ,1)                                          DIS0
      CALL MULT(Q,NQ,H,NH,DUMMY,NH)                                     DIS0
      CALL TRANP(H,NH,DUMMY(N1),NF)                                     DIS0
      CALL MULT(DUMMY(N1),NF,DUMMY,NH,Q,NQ)                             DIS0
      CALL LNCNT(3)                                                     DIS0
      PRINT 55                                                          DIS0
   55 FORMAT(/,' MATRIX ( H TRANSPOSE )QH',/)                           DIS0
      CALL PRNT(Q,NQ,4HHTQH,1)                                          DIS0
   65 CONTINUE                                                          DIS0
      CALL PRNT(R,NR,4H R  ,1)                                          DIS0
      CALL LNCNT(4)                                                     DIS0
      PRINT 75                                                          DIS0
   75 FORMAT(//,' WEIGHTING ON TERMINAL VALUE OF STATE VECTOR',/)       DIS0
      CALL PRNT(P,NP,4H P  ,1)                                          DIS0
C                                                                       DIS0
   85 CONTINUE                                                          DIS0
      IF((IOP(1) .NE. 0)  .OR. IDENT)  GO TO 100                        DIS0
      CALL MULT(Q,NQ,H,NH,DUMMY,NH)                                     DIS0
      CALL TRANP(H,NH,DUMMY(N1),NF)                                     DIS0
      CALL MULT(DUMMY(N1),NF,DUMMY,NH,Q,NQ)                             DIS0
C                                                                       DIS0
  100 CONTINUE                                                          DIS0
      I=I-1                                                             DIS0
      CALL EQUATE(P,NP,DUMMY,NP)                                        DIS0
      CALL MULT(P,NP,A,NA,DUMMY(N1),NA)                                 DIS00
      CALL TRANP(B,NB,DUMMY(N2),NF)                                     DIS00
      CALL MULT(DUMMY(N2),NF,DUMMY(N1),NA,F,NF)                         DIS00
      CALL MULT(P,NP,B,NB,DUMMY(N1),NB)                                 DIS00
      CALL MULT(DUMMY(N2),NF,DUMMY(N1),NB,DUMMY(N3),NR)                 DIS00
      CALL ADD(R,NR,DUMMY(N3),NR,DUMMY(N1),NR)                          DIS00
```

```
      IOPT = 3                                                          DIS
      IAC=IACM                                                          DIS
      MF = NR(1)                                                        DIS
      CALL SNVDEC(IOPT,MF,MF,MF,MF,DUMMY(N1),NF(2),F,IAC,ZTEST,DUMMY(N2)DIS
     1,DUMMY(N3),IRANK,APLUS,IERR)                                      DIS
      IF( IERR .EQ.0) GO TO 300                                         DIS
      CALL LNCNT(5)                                                     DIS
      IF(IERR .GT. 0) PRINT 200,IERR                                    DIS
      IF(IERR  .EQ. -1) PRINT 250,ZTEST,IRANK                           DIS
  200 FORMAT(//,' IN DISREG, SNVDEC HAS FAILED TO CONVERGE TO THE ',I4, DIS
     1'SINGULARVALUE AFTER 30 ITERATIONS',//)                           DIS
  250 FORMAT(//,' IN DISREG, THE MATRIX SUBMITTED TO SNVDEC USING ZTEST DIS
     1=',D16.8,' IS CLOSE TO A MATRIX OF LOWER RANK',/,'IF  THE ACCURACYDIS
     2 IAC IS REDUCED THE RANK MAY ALSO BE REDUCED',/,' CURRENT RANK = 'DIS
     3 ,I4)                                                             DIS
      IF( IERR .GT. 0 ) RETURN                                          DIS
      NDUM(1) = NA(1)                                                   DIS
      NDUM(2) =  1                                                      DIS
      CALL PRNT(DUMMY(N2),NDUM,4HSGVL,1)                                DIS
C                                                                       DIS
  300 CONTINUE                                                          DIS
      CALL MULT(R,NR,F,NF,DUMMY(N1),NF)                                 DIS
      CALL TRANP(F,NF,DUMMY(N2),NB)                                     DIS
      CALL MULT(DUMMY(N2),NB,DUMMY(N1),NF,P,NP)                         DIS
      CALL ADD(Q,NQ,P,NP,P,NP)                                          DIS
      CALL MULT(B,NB,F,NF,DUMMY(N1),NA)                                 DIS
      CALL SUBT(A,NA,DUMMY(N1),NA,DUMMY(N1),NA)                         DIS
      CALL MULT(DUMMY,NA,DUMMY(N1),NA,DUMMY(N2),NA)                     DIS
      CALL TRANP(DUMMY(N1),NA,DUMMY(N3),NA)                             DIS
      CALL MULT(DUMMY(N3),NA,DUMMY(N2),NA,DUMMY(N1),NA)                 DIS
      CALL ADD(P,NP,DUMMY(N1),NA,P,NP)                                  DIS
C                                                                       DIS
      IF( IOP(2) .EQ. 0 )  GO TO 400                                    DIS
      CALL LNCNT(5)                                                     DIS
      PRINT 350,I                                                       DIS
  350 FORMAT(///,' STAGE ',I5,/)                                        DIS
      CALL PRNT(F,NF,4H F  ,1)                                          DIS
      CALL PRNT(P,NP,4H P  ,1)                                          DIS
C                                                                       DIS
  400 CONTINUE                                                          DIS
      IF( I .EQ. 0 )  GO TO 600                                         DIS
      CALL MAXEL(DUMMY,NP,ANORM1)                                       DIS
      CALL SUBT(DUMMY,NP,P,NP,DUMMY(N2),NP)                             DIS
      CALL MAXEL(DUMMY(N2),NP,ANORM2)                                   DIS
      IF( ANORM1 .NE. 0.0 )  GO TO 500                                  DIS
      GO TO 100                                                         DIS
C                                                                       DIS
  500 CONTINUE                                                          DIS
      IF(ANORM1 .GT. 1.0 ) GO TO 550                                    DIS
      IF( ANORM2/ANORM1 .LT. RICTCV ) KSS = 1                           DIS
      GO TO 575                                                         DIS
  550 CONTINUE                                                          DIS
      IF( ANORM2 .LT. RICTCV ) KSS=1                                    DIS
  575 CONTINUE                                                          DIS
      IF( KSS .EQ. 1) GO TO 600                                         DIS
      GO TO 100                                                         DIS
C                                                                       DIS
  600 CONTINUE                                                          DIS
      K = IOP(1) + IOP(2)                                               DIS
      IF( K .EQ. 0 ) RETURN                                             DIS
      IF( KSS .EQ. 0) GO TO 700                                         DIS
      CALL LNCNT(4)                                                     DIS
      PRINT 650                                                         DIS
```

```
  650 FORMAT(//,' STEADY-STATE SOLUTION HAS BEEN REACHED IN  DISREG',/) DIS01
C                                                                       DIS01
  700 CONTINUE                                                          DIS01
      IF( IOP(2) .NE. 0 )  RETURN                                       DIS01
      IF( IOP(1) .EQ. 0 )  RETURN                                       DIS01
      CALL LNCNT(3)                                                     DIS01
      I = IOP(3)-I                                                      DIS01
      PRINT 800, I                                                      DIS01
  800 FORMAT(/,' F AND P AFTER ',I5, ' STEPS',/)                        DIS01
      CALL PRNT(F,NF,4H F  ,1)                                          DIS01
      CALL PRNT(P,NP,4H P  ,1)                                          DIS01
      RETURN                                                            DIS01
      END                                                               DIS01
```

```
      SUBROUTINE CNTREG(A,NA,B,NB,H,NH,Q,NQ,R,NR,Z,W,LAMBDA,S,F,NF,P,NP CNT
     1,T,IOP,IDENT,DUMMY)                                               CNT
      IMPLICIT REAL*8 (A-H,O-Z)                                         CNT
C                                                                       CNT
      DIMENSION A(1),B(1),H(1),Q(1),R(1),Z(1),W(1),LAMBDA(1),S(1),F(1),PCNT
     1(1),T(1),DUMMY(1)                                                 CNT
      DIMENSION NA(2),NB(2),NH(2),NQ(2),NR(2),NF(2),NP(2),IOP(3),NDUM1(2CNT
     1),NDUM2(2)                                                        CNT
      LOGICAL IDENT                                                     CNT
      REAL*8 LAMBDA                                                     CNT
      COMMON/CONV/SUMCV,RICTCV,SERCV,MAXSUM                             CNT
C                                                                       CNT
      IF( IOP(1) .EQ. 0 ) GO TO 65                                      CNT
      CALL LNCNT(5)                                                     CNT
      IF( IOP(3) .EQ. 0 ) PRINT 25                                      CNT
   25 FORMAT(//' PROGRAM TO SOLVE THE TIME-INVARIANT FINITE-DURATION CONCNT
     1TINUOUS OPTIMAL'/' REGULATOR PROBLEM WITH NOISE-FREE MEASUREMENTS'CNT
     2)                                                                 CNT
      IF( IOP(3) .NE. 0 ) PRINT 30                                      CNT
   30 FORMAT(//' PROGRAM TO SOLVE THE TIME-INVARIANT INFINITE-DURATION CCNT
     1ONTINUOUS OPTIMAL'/' REGULATOR PROBLEM  WITH NOISE-FREE MEASUREMENCNT
     2TS')                                                              CNT
      CALL PRNT(A,NA,4H A  ,1)                                          CNT
      CALL PRNT(B,NB,4H B  ,1)                                          CNT
      CALL PRNT(Q,NQ,4H Q  ,1)                                          CNT
      IF( .NOT. IDENT ) GO TO 45                                        CNT
      CALL LNCNT(3)                                                     CNT
      PRINT 35                                                          CNT
   35 FORMAT(/' H IS AN IDENTITY MATRIX'/)                              CNT
      GO TO 55                                                          CNT
C                                                                       CNT
   45 CONTINUE                                                          CNT
      CALL PRNT(H,NH,4H H  ,1)                                          LNT
      CALL MULT(Q,NQ,H,NH,DUMMY,NH)                                     CNT
      N1= NH(1)*NH(2)+1                                                 CNT
      CALL TRANP(H,NH,DUMMY(N1),NDUM1)                                  CNT
      CALL MULT(DUMMY(N1),NDUM1,DUMMY,NH,Q,NQ)                          CNT
      CALL LNCNT(3)                                                     CNT
      PRINT 50                                                          CNT
   50 FORMAT(//' MATRIX (H TRANSPOSE)QH')                               CNT
      CALL PRNT(Q,NQ,0,3)                                               CNT
   55 CONTINUE                                                          CNT
      CALL PRNT(R,NR,4H R  ,1)                                          CNT
C                                                                       CNT
      IF( IOP(3) .NE. 0 ) GO TO 65                                      CNT
      CALL LNCNT(4)                                                     CNT
      PRINT 60                                                          CNT
   60 FORMAT(//' WEIGHTING ON TERMINAL VALUE OF STATE VECTOR'/)         CNT
      CALL PRNT(P,NP,4H P  ,1)                                          CNT
C                                                                       CNT
   65 CONTINUE                                                          CNT
      CALL EQUATE(R,NR,DUMMY,NR)                                        CNT
      N = NA(1)**2                                                      CNT
      N1 = NB(1)*NB(2)+1                                                CNT
      CALL TRANP(B,NB,DUMMY(N1),NDUM1)                                  CNT
      N2 = N1 + N                                                       CNT
      L = NR(1)                                                         CNT
      IOPT = 0                                                          CNT
      IFAC = 0                                                          CNT
      CALL SYMPDS(L,L,DUMMY,NB(1),DUMMY(N1),IOPT,IFAC,DET,ISCALE,DUMMY(NCNT
     12),IERR)                                                          CNT
C                                                                       CNT
      IF( IERR .EQ. 0 ) GO TO 100                                       CNT
```

```
      CALL LNCNT(4)                                                     CNT0
      PRINT 75                                                          CNT0
   75 FORMAT(//' IN CNTREG, THE SUBROUTINE  SYMPOS HAS FOUND THE MATRIX CNT0
     1 R NOT SYMMETRIC POSITIVE DEFINITE'/)                             CNT0
      RETURN                                                            CNT0
C                                                                       CNT0
  100 CONTINUE                                                          CNT0
      CALL EQUATE(DUMMY(N1),NDUM1,DUMMY,NDUM1)                          CNT0
      CALL MULT(B,NB,DUMMY(N1),NDUM1,DUMMY(N2),NA)                      CNT0
      CALL SCALE(DUMMY(N2),NA,DUMMY(N1),NA,-1.0)                        CNT0
      N3 = N2 + N                                                       CNT0
      IF( IDENT .OR. (IOP(1) .NE. 0) ) GO TO 200                        CNT0
      CALL MULT(Q,NQ,H,NH,DUMMY(N2),NH)                                 CNT0
      CALL TRANP(H,NH,DUMMY(N3),NDUM1)                                  CNT0
      CALL MULT(DUMMY(N3),NDUM1,DUMMY(N2),NH,Q,NQ)                      CNT0
C                                                                       CNT0
  200 CONTINUE                                                          CNT0
      CALL SCALE(Q,NQ,Q,NQ,-1.0)                                        CNT0
      CALL JUXTR(A,NA,Q,NQ,Z,NDUM1)                                     CNT0
      CALL TRANP(A,NA,DUMMY(N2),NA)                                     CNT0
      CALL SCALE( DUMMY(N2),NA,DUMMY(N2),NA,-1.0)                       CNT0
      L = 2*N + 1                                                       CNT0
      CALL JUXTR(DUMMY(N1),NA,DUMMY(N2),NA,Z(L),NDUM1)                  CNT0
      CALL SCALE(Q,NQ,Q,NQ,-1.0)                                        CNT0
      NDUM2(1) = 2*NA(1)                                                CNT0
      NDUM2(2) = NDUM2(1)                                               CNT0
      IF( IOP(1) .NE. 0 )  CALL PRNT(Z,NDUM2,4H Z  ,1)                  CNT0
      CALL EQUATE(Z,NDUM2,DUMMY(N1),NDUM2)                              CNT0
      M = 4*N                                                           CNT0
      N2 = M + N1                                                       CNT0
      L = 2*NA(1)                                                       CNT0
      N3 = N2 + L                                                       CNT0
      N4 = N3 + L                                                       CNT0
      ISV = L                                                           CNT0
      ILV = 0                                                           CNT0
      CALL EIGEN(L,L,DUMMY(N1),DUMMY(N2),DUMMY(N3),ISV,ILV,W,DUMMY(N4),ICNT0
     1IERR)                                                             CNT0
      IF( IERR .EQ. 0 ) GO TO 300                                       CNT0
      CALL LNCNT(4)                                                     CNT0
      IF( IERR .GT. 0 ) GO TO 250                                       CNT0
      PRINT 225,IERR                                                    CNT0
  225 FORMAT(//' IN CNTREG, EIGEN FAILED TO COMPUTE THE ',I6,' EIGENVEC CNT0
     1TOR OF Z '/)                                                      CNT0
      RETURN                                                            CNT0
  250 CONTINUE                                                          CNT0
      PRINT 275,IERR                                                    CNT0
  275 FORMAT(//' IN CNTREG, THE ',I6,' EIGENVALUE OF Z HAS NOT BEEN FO  CNT0
     1UND AFTER 30 ITERATIONS IN EIGEN'/)                               CNT0
      RETURN                                                            CNT0
C                                                                       CNT0
  300 CONTINUE                                                          CNT0
      IF( IOP(1) .EQ. 0 ) GO TO 400                                     CNT0
      CALL LNCNT(3)                                                     CNT0
      PRINT 325                                                         CNT0
  325 FORMAT(//' EIGENVALUES OF Z')                                     CNT0
      NDUM1(1) = L                                                      CNT0
      NDUM1(2) = 2                                                      CNT0
      CALL PRNT(DUMMY(N2),NDUM1,0,3)                                    CNT0
      CALL LNCNT(3)                                                     CNT0
      PRINT 350                                                         CNT0
  350 FORMAT(//' CORRESPONDING EIGENVECTORS')                           CNT0
      CALL PRNT(W,NDUM2,0,3)                                            CNT0
C                                                                       CNT0
```

```
  400 CONTINUE                                                          CNT
      CALL EQUATE(W,NDUM2,DUMMY(N1),NDUM2)                              CNT
      J1 = 1                                                            CNT
      J2 = 1                                                            CNT
      M = 2*N                                                           CNT
      NDUM1(1) = L                                                      CNT
      NDUM1(2) = 1                                                      CNT
      K4 = N4                                                           CNT
C                                                                       CNT
      I=1                                                               CNT
  415 CONTINUE                                                          CNT
      IF( I .GT. L )  GO TO 515                                         CNT
      K1 = N2+I-1                                                       CNT
      K2 = N1+(I-1)*L                                                   CNT
      K3 = N3+I-1                                                       CNT
      IF(DUMMY(K1) .GT. 0.0 ) GO TO 425                                 CNT
      J = (J1-1)*L+M+1                                                  CNT
      J1 = J1+1                                                         CNT
      IF(DUMMY(K3).NE. 0.0) J1=J1+1                                     CNT
      GO TO 450                                                         CNT
  425 CONTINUE                                                          CNT
      DUMMY(K4)=I                                                       CNT
      K4 = K4+1                                                         CNT
      J = (J2-1)*L+1                                                    CNT
      J2 = J2+1                                                         CNT
      IF( DUMMY(K3) .NE. 0.0 )  J2 = J2 + 1                             CNT
  450 CONTINUE                                                          CNT
      CALL EQUATE(DUMMY(K2),NDUM1,W(J),NDUM1)                           CNT
      IF(DUMMY(K3) .EQ. 0.0) GO TO 500                                  CNT
      I = I+1                                                           CNT
      K2 = K2+L                                                         CNT
      J = J+L                                                           CNT
      CALL EQUATE(DUMMY(K2),NDUM1,W(J),NDUM1)                           CNT
  500 CONTINUE                                                          CNT
      I=I+1                                                             CNT
      GO TO 415                                                         CNT
  515 CONTINUE                                                          CNT
C                                                                       CNT
      CALL NULL(LAMBDA,NA)                                              CNT
      K0 = -1                                                           CNT
      J = -NA(1)                                                        CNT
      NAX = NA(1)                                                       CNT
      I=1                                                               CNT
  520 CONTINUE                                                          CNT
      IF( I .GT. NAX )  GO TO 530                                       CNT
      J = NAX + J + 1                                                   CNT
      K0 = K0 + 1                                                       CNT
      K1 = N4 + K0                                                      CNT
      K2 = DUMMY(K1)                                                    CNT
      K = N2+K2-1                                                       CNT
      LAMBDA(J) = DUMMY(K)                                              CNT
      K3 = N3+K2-1                                                      CNT
      IF( DUMMY(K3) .EQ. 0.0 ) GO TO 525                                CNT
      K4 = J+1                                                          CNT
      LAMBDA(K4) = -DUMMY(K3)                                           CNT
      K4 = K4+NAX                                                       CNT
      LAMBDA(K4) = DUMMY(K)                                             CNT
      K4 = K4-1                                                         CNT
      LAMBDA(K4) = DUMMY(K3)                                            CNT
      K5 = M + (I-1)*L + 1                                              CNT
      K6 = K5 + L                                                       CNT
      CALL EQUATE(W(K5),NDUM1,DUMMY(N1),NDUM1)                          CNT
      CALL EQUATE(W(K6),NDUM1,W(K5),NDUM1)                              CNT
```

```
      CALL EQUATE(DUMMY(N1),NDUM1,W(K6),NDUM1)                          CNTO
      I = I+1                                                           CNTO
      J = NAX + J +1                                                    CNTO
  525 CONTINUE                                                          CNTO
      I=I+1                                                             CNTO
      GO TO 520                                                         CNTO
  530 CONTINUE                                                          CNTO
C                                                                       CNTO
      IF( IOP(1) .EQ. 0 ) GO TO 700                                     CNTO
      CALL LNCNT(3)                                                     CNTO
      PRINT 535                                                         CNTO
  535 FORMAT(//' REORDERED EIGENVECTORS')                               CNTO
      CALL PRNT(W,NDUM2,0,3)                                            CNTO
      CALL LNCNT(4)                                                     CNTO
      PRINT 545                                                         CNTO
  545 FORMAT(//' LAMBDA MATRIX OF EIGENVALUES OF Z WITH POSITIVE REAL PACNTO
     1RTS')                                                             CNTO
      CALL PRNT(LAMBDA,NA,0,3)                                          CNTO
C                                                                       CNTO
      CALL MULT(Z,NDUM2,W,NDUM2,DUMMY(N1),NDUM2)                        CNTO
      L = NDUM2(1)                                                      CNTO
      M = L**2                                                          CNTO
      N2 = N1+M                                                         CNTO
      CALL EQUATE(W,NDUM2,DUMMY(N2),NDUM2)                              CNTO
      N3 = N2+M                                                         CNTO
      N4 = N3+L                                                         CNTO
      IFAC = 0                                                          CNTO
      CALL GELIM(L,L,DUMMY(N2),L,DUMMY(N1),DUMMY(N3),IFAC,DUMMY(N4),IERRCNTO
     1)                                                                 CNTO
      IF( IERR .EQ. 0 ) GO TO 600                                       CNTO
      CALL LNCNT(4)                                                     CNTO
      PRINT 550                                                         CNTO
  550 FORMAT(//' IN CNTREG, GELIM HAS FOUND THE REORDERED MATRIX W TO B CNTO
     1E SINGULAR ')                                                     CNTO
  600 CONTINUE                                                          CNTO
      CALL PRNT(DUMMY(N1),NDUM2,4HWIZW,1)                               CNTO
C                                                                       CNTO
  700 CONTINUE                                                          CNTO
      NDUM1(1) = 2*NA(1)                                                CNTO
      NDUM1(2) = NA(1)                                                  CNTO
      N2 = 2*N + N1                                                     CNTO
      CALL TRANP(W,NDUM1,DUMMY(N2),NDUM2)                               CNTO
      NW11 = N1                                                         CNTO
      NDUM1(1) = NA(1)                                                  CNTO
      CALL TRANP(DUMMY(N2),NDUM1,DUMMY(NW11),NDUM1)                     CNTO
      L = N2+N                                                          CNTO
      NW21 = NW11+N                                                     CNTO
      CALL TRANP(DUMMY(L),NDUM1,DUMMY(NW21),NDUM1)                      CNTO
      L = 2*N+1                                                         CNTO
      NDUM1(1)=2*NA(1)                                                  CNTO
      N3 = N2 + 2*N                                                     CNTO
      CALL TRANP(W(L),NDUM1,DUMMY(N3),NDUM2)                            CNTO
      NDUM1(1) = NA(1)                                                  CNTO
      NW12 = NW21+N                                                     CNTO
      CALL TRANP(DUMMY(N3),NDUM1,DUMMY(NW12),NDUM1)                     CNTO
      L = N3 + N                                                        CNTO
      NW22 = NW12 + N                                                   CNTO
      CALL TRANP(DUMMY(L),NDUM1,DUMMY(NW22),NDUM1)                      CNTO
C                                                                       CNTO
      IF( IOP(1) .EQ. 0 ) GO TO 800                                     CNTO
      CALL PRNT(DUMMY(NW11),NA,4HW11 ,1)                                CNTO
      CALL PRNT(DUMMY(NW21),NA,4HW21 ,1)                                CNTO
      CALL PRNT(DUMMY(NW12),NA,4HW12 ,1)                                CNTO
```

```
      CALL PRNT(DUMMY(NW22),NA,4HW22 ,1)                                CNTC
C                                                                       CNTC
  900 CONTINUE                                                          CNTC
      IF( IOP(3) .NE. 0 ) GO TO 900                                     CNTC
      N2 = N1+4*N                                                       CNTC
      CALL MULT(P,NP,DUMMY(NW12),NA,S,NA)                               CNTC
      CALL MULT(P,NP,DUMMY(NW11),NA,DUMMY(N2),NA)                       CNTC
      CALL SUBT(S,NA,DUMMY(NW22),NA,S,NA)                               CNTC
      CALL SUBT(DUMMY(NW21),NA,DUMMY(N2),NA,DUMMY(N2),NA)               CNTC
      N3 = N2+N                                                         CNTC
      L = NA(1)                                                         CNTC
      IFAC = 0                                                          CNTC
      N4 = N3+NA(1)                                                     CNTC
      CALL GELIM(L,L,DUMMY(N2),L,S,DUMMY(N3),IFAC,DUMMY(N4),IERR)       CNTC
      IF( IERR .EQ. 0 ) GO TO 850                                       CNTC
      CALL LNCNT(4)                                                     CNTC
      PRINT 825                                                         CNTC
  825 FORMAT(//' IN CNTREG, GELIM HAS FOUND THE MATRIX  W21 - P1XW11 TO CNTC
     1 BE SINGULAR'/)                                                   CNTC
      RETURN                                                            CNTC
C                                                                       CNTC
  850 CONTINUE                                                          CNTC
      IF( IOP(1) .EQ. 0 ) GO TO 1000                                    CNTC
      CALL PRNT(S,NA,4H S  ,1)                                          CNTC
      NDUM1(1) = NR(1)                                                  CNTC
      NDUM1(2) = NA(1)                                                  CNTC
      CALL LNCNT(3)                                                     CNTC
      PRINT 875                                                         CNTC
  875 FORMAT(//' MATRIX (R INVERSE)X(B TRANSPOSE)')                     CNTC
      CALL PRNT(DUMMY,NDUM1,0,3)                                        CNTC
      GO TO 1000                                                        CNTC
C                                                                       CNTC
  900 CONTINUE                                                          CNTC
      N2 = N1+4*N                                                       CNTC
      CALL TRANP(DUMMY(NW12),NA,DUMMY(N2),NA)                           CNTC
      CALL TRANP(DUMMY(NW22),NA,P,NP)                                   CNTC
      N3 = N2+N                                                         CNTC
      IFAC = 0                                                          CNTC
      L = NA(1)                                                         CNTC
      N4 = N3 + NA(1)                                                   CNTC
      CALL GELIM(L,L,DUMMY(N2),L,P,DUMMY(N3),IFAC,DUMMY(N4),IERR)       CNTC
      IF( IERR .EQ. 0 ) GO TO 950                                       CNTC
      CALL LNCNT(4)                                                     CNTC
      PRINT 925                                                         CNTC
  925 FORMAT(//' IN CNTREG, GELIM HAS FOUND THE MATRIX W12 TO BE SINGUL CNTC
     1AR'/)                                                             CNTC
      RETURN                                                            CNTC
  950 CONTINUE                                                          CNTC
      NDUM1(1) = NR(1)                                                  CNTC
      NDUM1(2) = NA(1)                                                  CNTC
      CALL MULT(DUMMY,NDUM1,P,NP,F,NF)                                  CNTC
      IF( IOP(1) .EQ. 0 ) RETURN                                        CNTC
      CALL PRNT(P,NP,4H P  ,1)                                          CNTC
      CALL PRNT(F,NF,4H F  ,1)                                          CNTC
      RETURN                                                            CNTC
C                                                                       CNTC
 1000 CONTINUE                                                          CNTC
      NMAX = T(1)/T(2)                                                  CNTC
      I = NMAX                                                          CNTC
      CALL EQUATE(LAMBDA,NA,DUMMY(N2),NA)                               CNTC
      TT = -T(2)                                                        LNT0
      N4 = N3+N                                                         CNTC
      N5 = N4+N                                                         CNTC
```

```
      N6 = N5+N                                                         CNT
      N7 = N6+NA(1)                                                     CNT
      KSS = 0                                                           CNT
      NDUM1(1) = NR(1)                                                  CNT
      NDUM1(2) = NA(1)                                                  CNT
      CALL EXPSER(DUMMY(N2),NA,DUMMY(N3),NA,TT,KSS,DUMMY(N4))           CNT
      CALL EQUATE(DUMMY(N3),NA,DUMMY(N2),NA)                            CNT
      IF( IOP(1) .EQ. 0 )  GO TO 1075                                   CNT
      CALL LNCNT(3)                                                     CNT
      PRINT 1050,T(2)                                                   CNT
 1050 FORMAT(//' EXP(-LAMBDA X ',D16.8,')')                             CNT
      CALL PRNT(DUMMY(N2),NA,0,3)                                       CNT
 1075 CONTINUE                                                          CNT
      IF( NMAX .LE. 0 )  RETURN                                         CNT
      CALL EQUATE(S,NA,DUMMY(N3),NA)                                    CNT
 1100 CONTINUE                                                          CNT
      TIME = I*T(2)                                                     CNT
      IF( I .NE. NMAX )  CALL EQUATE(DUMMY(N5),NA,P,NP)                 CNT
      CALL MULT(DUMMY(N3),NA,DUMMY(N2),NA,DUMMY(N4),NA)                 CNT
      CALL MULT(DUMMY(N2),NA,DUMMY(N4),NA,DUMMY(N3),NA)                 CNT
      CALL MULT(DUMMY(NW11),NA,DUMMY(N3),NA,DUMMY(N4),NA)               CNT
      CALL ADD(DUMMY(NW12),NA,DUMMY(N4),NA,DUMMY(N4),NA)                CNT
      CALL TRANP(DUMMY(N4),NA,DUMMY(N5),NA)                             CNT
      CALL EQUATE(DUMMY(N5),NA,DUMMY(N4),NA)                            CNT
      CALL MULT(DUMMY(NW21),NA,DUMMY(N3),NA,DUMMY(N5),NA)               CNT
      CALL ADD(DUMMY(NW22),NA,DUMMY(N5),NA,DUMMY(N5),NA)                CNT
      CALL TRANP(DUMMY(N5),NA,DUMMY(N6),NA)                             CNT
      CALL EQUATE(DUMMY(N6),NA,DUMMY(N5),NA)                            CNT
      L = NA(1)                                                         CNT
      IFAC = 0                                                          CNT
      CALL GELIM(L,L,DUMMY(N4),L,DUMMY(N5),DUMMY(N6),IFAC,DUMMY(N7),IERRCNT
     1)                                                                 CNT
      IF( IERR .EQ. 0 ) GO TO 1200                                      CNT
      CALL LNCNT(3)                                                     CNT
      PRINT 1150,TIME                                                   CNT
 1150 FORMAT(//' IN CNTREG AT TIME ',D16.8,' P CANNOT BE COMPUTED DUE T CNT
     1O MATRIX SINGULARITY IN GELIM')                                   CNT
      RETURN                                                            CNT
C                                                                       CNT
 1200 CONTINUE                                                          CNT
      CALL MAXEL(P,NP,ANORM1)                                           CNT
      CALL SUBT(DUMMY(N5),NA,P,NP,DUMMY(N4),NA)                         CNT
      CALL MAXEL(DUMMY(N4),NA,ANORM2)                                   CNT
      IF( ANORM1 .NE. 0.0 ) GO TO 1225                                  CNT
      GO TO 1300                                                        CNT
C                                                                       CNT
 1225 CONTINUE                                                          CNT
      IF(ANORM1 .GT. 1.0 ) GO TO 1250                                   CNT
      IF( ANORM2/ANORM1 .LT. RICTCV ) KSS=1                             CNT
      GO TO 1300                                                        CNT
 1250 CONTINUE                                                          CNT
      IF( ANORM2 .LT. RICTCV ) KSS=1                                    CNT
C                                                                       CNT
 1300 CONTINUE                                                          CNT
      CALL MULT(DUMMY,NDUM1,P,NP,F,NF)                                  CNT
      IF( IOP(2) .EQ. 0 ) GO TO 1400                                    CNT
      CALL LNCNT(5)                                                     CNT
      PRINT 1350,TIME                                                   CNT
 1350 FORMAT(///' TIME = ',D16.8/)                                      CNT
      CALL PRNT(P,NP,4H P  ,1)                                          CNT
      IF( I .NE. NMAX )  CALL PRNT(F,NF,4H F  ,1)                       CNT
C                                                                       CNT
 1400 CONTINUE                                                          CNT
```

```
      IF( KSS .EQ. 1 ) GO TO 1500                                       CNT
      I = I-1                                                           CNT
      IF( I .GE. 0 ) GO TO 1100                                         CNT
      GO TO 1600                                                        CNT
 1500 CONTINUE                                                          CNT
      CALL LNCNT(4)                                                     CNT
      PRINT 1550                                                        CNT
 1550 FORMAT(//' STEADY-STATE SOLUTION HAS BEEN REACHED IN CNTREG'/)    CNT
C                                                                       CNT
 1600 CONTINUE                                                          CNT
      IF( IOP(2) .NE. 0 ) RETURN                                        CNT
      IF( IOP(1) .EQ. 0 ) RETURN                                        CNT
      CALL LNCNT(5)                                                     CNT
      PRINT 1350,TIME                                                   CNT
      CALL PRNT(P,NP,4H P  ,1)                                          CNT
      CALL PRNT(F,NF,4H F  ,1)                                          CNT
C                                                                       CNT
      RETURN                                                            LNT
      END                                                               CNT
```

```
      SUBROUTINE RICNWT(A,NA,B,NB,H,NH,Q,NQ,R,NR,F,NF,P,NP,IOP,IDENT,DI RIC00
     1SC,FNULL,DUMMY)                                                   RIC00
      IMPLICIT REAL*8 (A-H,O-Z)                                         RIC00
      DIMENSION A(1),B(1),Q(1),R(1),F(1),P(1),DUMMY(1)                  RIC00
      DIMENSION NA(2),NB(2),NQ(2),NR(2),NF(2),NP(2),IOP(3)              RIC00
      DIMENSION H(1),NH(2),IOPT(2)                                      RIC00
      LOGICAL  IDENT,DISC,FNULL,SYM                                     RIC00
      COMMON/TOL/EPSAM,EPSBM,IACM                                       RIC00
      COMMON/CONV/SUMCV,RICTCV,SERCV,MAXSUM                             RIC00
      I=1                                                               RIC00
      IOPT(1)=0                                                         RIC00
      SYM = .TRUE.                                                      RIC00
C                                                                       RIC00
      N = NA(1)**2                                                      RIC00
      N1 = N +1                                                         RIC00
      IF( .NOT. DISC) N1 = NA(1)*NR(1) + 1                              RIC00
      N2= N1+N                                                          RIC00
      N3= N2+N                                                          RIC00
      N4 = N3+N                                                         RIC00
C                                                                       RIC00
      IF( IOP(1) .EQ. 0 )  GO TO 210                                    RIC00
      CALL LNCNT(4)                                                     RIC00
      IF(.NOT. DISC)PRINT 100                                           RIC00
      IF( DISC )PRINT 150                                               RIC00
  100 FORMAT(//,' PROGRAM TO SOLVE CONTINUOUS STEADY-STATE RICCATI EQUATRIC00
     1ION BY THE NEWTON ALGORITHM',/)                                   RIC00
  150 FORMAT(//,' PROGRAM TO SOLVE DISCRETE STEADY-STATE RICCATI EQUATIORIC00
     1N BY THE NEWTON ALGORITHM',/)                                     RIC00
      CALL PRNT(A,NA,4H A  ,1)                                          RIC00
      CALL PRNT(B,NB,4H B  ,1)                                          RIC00
      CALL PRNT(Q,NQ,4H Q  ,1)                                          RIC00
      IF( .NOT. IDENT )GO TO 185                                        RIC00
      CALL LNCNT(3)                                                     RIC00
      PRINT 180                                                         RIC00
  180 FORMAT(/,' H IS AN IDENTITY MATRIX',/)                            RIC00
      GO TO 200                                                         RIC00
  185 CONTINUE                                                          RIC00
      CALL PRNT(H,NH,4H H  ,1)                                          RIC00
      CALL MULT(Q,NQ,H,NH,DUMMY,NH)                                     RIC0
      CALL TRANP(H,NH,DUMMY(N2),NP)                                     RIC0
      CALL MULT(DUMMY(N2),NP,DUMMY,NH,Q,NQ)                             RIC0
      CALL LNCNT(3)                                                     RIC0
      PRINT 195                                                         RIC0
  195 FORMAT(/,' MATRIX (H TRANSPOSE)QH ',/)                            RIC0
      CALL PRNT(Q,NQ,4HHTQH,1)                                          RIC0
  200 CONTINUE                                                          RIC0
      CALL PRNT(R,NR,4H R  ,1)                                          RIC0
      IF( FNULL )  GO TO 210                                            RIC0
      CALL LNCNT(3)                                                     RIC0
      PRINT 205                                                         RIC0
  205 FORMAT(/,' INITIAL F MATRIX',/)                                   RIC0
      CALL PRNT(F,NF,4H F  ,1)                                          RIC0
C                                                                       RIC0
  210 CONTINUE                                                          RIC0
      IF((IOP(1) .NE. 0)  .OR. IDENT)  GO TO 220                        RIC0
      CALL MULT(Q,NQ,H,NH,DUMMY,NH)                                     RIC0
      CALL TRANP(H,NH,DUMMY(N2),NP)                                     RIC0
      CALL MULT(DUMMY(N2),NP,DUMMY,NH,Q,NQ)                             RIC0
  220 CONTINUE                                                          RIC0
C                                                                       RIC0
      IF (DISC) GO TO 900                                               RIC0
C                                                                       RIC0
      CALL TRANP(B,NB,P,NP)                                             RIC0
```

```
      CALL EQUATE(R,NR,DUMMY,NR)                                        RIC
      CALL SYMPOS(NR(1),NR(1),DUMMY,NP(2),P,IOPT,IOPT,DET,ISCALE,DUMMY(NRIC
     11),IERR)                                                          RIC
      IF(IERR .EQ. 0) GO TO 250                                         RIC
      CALL LNCNT(3)                                                     RIC
      PRINT 225                                                         RIC
  225 FORMAT(/,' IN RICNWT, A MATRIX WHICH IS  NOT SYMMETRIC POSITIVE DERIC
     1FINITE HAS BEEN SUBMITTED TO  SYMPOS',/)                          RIC
      RETURN                                                            RIC
C                                                                       RIC
  250 CONTINUE                                                          RIC
      CALL EQUATE(P,NP,DUMMY,NF)                                        RIC
      CALL MULT(B,NB,DUMMY,NF,DUMMY(N1),NA)                             RIC
      CALL TRANP(DUMMY(N1),NA,DUMMY(N2),NA)                             RIC
      CALL ADD(DUMMY(N1),NA,DUMMY(N2),NA,DUMMY(N1),NA)                  RIC
      CALL SCALE(DUMMY(N1),NA,DUMMY(N1),NA,0.5)                         RIC
C                                                                       RIC
      IF(FNULL) GO TO 300                                               RIC
C                                                                       RIC
      CALL MULT(B,NB,F,NF,DUMMY(N2),NA)                                 RIC
      CALL SUBT(A,NA,DUMMY(N2),NA,DUMMY(N2),NA)                         RIC
      CALL TRANP(DUMMY(N2),NA,DUMMY(N3),NA)                             RIC
      CALL EQUATE(DUMMY(N3),NA,DUMMY(N2),NA)                            RIC
      CALL MULT(R,NR,F,NF,DUMMY(N3),NF)                                 RIC
      CALL TRANP(F,NF,P,NP)                                             RIC
      CALL MULT(P,NP,DUMMY(N3),NF,DUMMY(N4),NA)                         RIC
      CALL TRANP(DUMMY(N4),NA,DUMMY(N3),NA)                             RIC
      CALL ADD(DUMMY(N4),NA,DUMMY(N3),NA,DUMMY(N3),NA)                  RIC
      CALL SCALE(DUMMY(N3),NA,DUMMY(N3),NA,0.5)                         RIC
      CALL ADD(DUMMY(N3),NA,Q,NQ,P,NP)                                  RIC
      CALL SCALE(P,NP,P,NP,-1.0)                                        RIC
      GO TO 350                                                         RIC
C                                                                       RIC
  300 CONTINUE                                                          RIC
      CALL TRANP(A,NA,DUMMY(N2),NA)                                     RIC
      CALL SCALE(Q,NQ,P,NP,-1.0)                                        RIC
C                                                                       RIC
  350 CONTINUE                                                          RIC
      IF(IOP(3) .NE. 0) GO TO 400                                       RIC
      EPSA= EPSAM                                                       RIC
      CALL BARSTW(DUMMY(N2),NA,B,NB,P,NP,IOPT,SYM,EPSA,EPSA,DUMMY(N3))  RIC
      GO TO 450                                                         RIC
C                                                                       RIC
  400 CONTINUE                                                          RIC
      IOPT(2)=1                                                         RIC
      CALL BILIN(DUMMY(N2),NA,B,NB,P,NP,IOPT,SCLE,SYM,DUMMY(N3))        RIC
C                                                                       RIC
  450 CONTINUE                                                          RIC
      CALL EQUATE(P,NP,DUMMY(N2),NP)                                    RIC
      IF(IOP(2).EQ. 0) GO TO 550                                        RIC
      CALL LNCNT(3)                                                     RIC
      PRINT 500,I                                                       RIC
  500 FORMAT(/,' ITERATION  ',I5,/)                                     RIC
      CALL PRNT(P,NP,4H P  ,1)                                          RIC
C                                                                       RIC
  550 CONTINUE                                                          RIC
      CALL MULT(DUMMY(N1),NA,P,NP,DUMMY(N3),NA)                         RIC
      CALL MULT(P,NP,DUMMY(N3),NA,DUMMY(N4),NA)                         RIC
      CALL TPANP(DUMMY(N4),NA,P,NA)                                     RIC
      CALL ADD(P,NP,DUMMY(N4),NA,P,NP)                                  RIC
      CALL SCALE(P,NP,P,NP,0.5)                                         RIC
      CALL ADD(Q,NQ,P,NP,P,NP)                                          RIC
      CALL SCALE(P,NP,P,NP,-1.0)                                        RIC
```

```
      CALL SUBT(A,NA,DUMMY(N3),NA,DUMMY(N4),NA)                         RIC01
      CALL TRANP(DUMMY(N4),NA,DUMMY(N3),NA)                             RIC01
C                                                                       RIC01
      IF(IOP(3) .NE. 0 ) GO TO 650                                      RIC01
      CALL BARSTW(DUMMY(N3),NA,B,NB,P,NP,IOPT,SYM,EPSA,EPSA,DUMMY(N4))  RIC01
      GO TO 675                                                         RIC01
C                                                                       RIC01
  650 CONTINUE                                                          RIC01
      CALL BILIN(DUMMY(N3),NA,B,NB,P,NP,IOPT,SCLE,SYM,DUMMY(N4))        RIC01
C                                                                       RIC01
  675 CONTINUE                                                          RIC01
      I=I+1                                                             RIC01
      CALL MAXEL(DUMMY(N2),NA,ANORM1)                                   RIC01
      CALL SUBT(P,NP,DUMMY(N2),NA,DUMMY(N3),NA)                         RIC01
      CALL MAXEL(DUMMY(N3),NA,ANORM2)                                   RIC01
      IF(ANORM1 .GT. 1.0) GO TO 700                                     RIC01
      IF( ANORM2/ANORM1 .LT. RICTCV ) GO TO 800                         RIC01
      GO TO 750                                                         RIC01
C                                                                       RIC01
  700 CONTINUE                                                          RIC01
      IF( ANORM2 .LT. RICTCV ) GO TO 800                                RIC01
C                                                                       RIC01
  750 CONTINUE                                                          RIC01
      IF( I .LE. 101) GO TO 450                                         RIC01
      CALL LNCNT(3)                                                     RIC01
      PRINT 775                                                         RIC01
  775 FORMAT(/,' THE SUBROUTINE RICNWT HAS EXCEEDED 100 ITERATIONS WITHORIC01
     1UT CONVERGENCE',/)                                                RIC01
      IOP(1) = 1                                                        RIC01
C                                                                       RIC01
  800 CONTINUE                                                          RIC01
      CALL MULT(DUMMY,NF,P,NP,F,NF)                                     RIC01
      GO TO 1300                                                        RIC01
C                                                                       RIC01
  900 CONTINUE                                                          RIC01
      IF( .NOT. FNULL ) GO TO 950                                       RIC01
C                                                                       RIC01
      CALL EQUATE(Q,NQ,P,NP)                                            RIC01
      CALL EQUATE(A,NA,DUMMY(N1),NA)                                    RIC01
      CALL TRANP(A,NA,DUMMY(N2),NA)                                     RIC01
      GO TO 1000                                                        RIC01
  925 CONTINUE                                                          RIC01
C                                                                       RIC01
      I=I+1                                                             RIC01
      CALL EQUATE(P,NP,DUMMY,NP)                                        RIC01
  950 CONTINUE                                                          RIC01
C                                                                       RIC01
      CALL MULT(R,NR,F,NF,DUMMY(N1),NF)                                 RIC01
      CALL TRANP(F,NF,P,NP)                                             RIC01
      CALL MULT(P,NP,DUMMY(N1),NF,DUMMY(N2),NA)                         RIC01
      CALL TRANP(DUMMY(N2),NA,DUMMY(N1),NA)                             RIC01
      CALL ADD(DUMMY(N1),NA,DUMMY(N2),NA,DUMMY(N1),NA)                  RIC01
      CALL SCALE(DUMMY(N1),NA,DUMMY(N  ,NA,0.5)                         RIC01
      CALL ADD(Q,NQ,DUMMY(N1),NA,P,NP)                                  RIC01
      CALL MULT(B,NB,F,NF,DUMMY(N1),NA)                                 RIC01
      CALL SUBT(A,NA,DUMMY(N1),NA,DUMMY(N1),NA)                         RIC01
      CALL TRANP(DUMMY(N1),NA,DUMMY(N2),NA)                             RIC01
C                                                                       RIC01
 1000 CONTINUE                                                          RIC01
      CALL SUM(DUMMY(N2),NA,P,NP,DUMMY(N1),NA,IOPT,SYM,DUMMY(N3))       RIC01
      IF(IOP(2) .EQ. 0) GO TO 1100                                      RIC01
      CALL LNCNT(3)                                                     RIC01
      PRINT 500,I                                                       RIC01
```

```
      CALL PRNT(P,NP,4H P  ,1)                                          RIC0
C                                                                       RIC0
 1100 CONTINUE                                                          RIC0
      CALL MULT(P,NP,A,NA,DUMMY(N1),NA)                                 RIC0
      CALL MULT(P,NP,B,NB,DUMMY(N2),NB)                                 RIC0
      CALL TRANP(B,NB,DUMMY(N3),NF)                                     RIC0
      CALL MULT(DUMMY(N3),NF,DUMMY(N1),NA,F,NF)                         RIC0
      CALL MULT(DUMMY(N3),NF,DUMMY(N2),NB,DUMMY(N1),NR)                 RIC0
      CALL TRANP(DUMMY(N1),NR,DUMMY(N2),NR)                             RIC0
      CALL ADD(DUMMY(N1),NR,DUMMY(N2),NR,DUMMY(N1),NR)                  RIC0
      CALL SCALE(DUMMY(N1),NR,DUMMY(N1),NR,0.5)                         RIC0
      CALL ADD(R,NR,DUMMY(N1),NR,DUMMY(N1),NR)                          RIC0
      CALL SYMPDS(NR(1),NR(1),DUMMY(N1),NA(1),F,IOPT,IOPT,DET,ISCALE,DUMRIC0
     1MY(N2),IERR)                                                      RIC0
      IF(IERR .EQ. 0) GO TO 1150                                        RIC0
      CALL LNCNT(3)                                                     RIC0
      PRINT 225                                                         RIC0
      RETURN                                                            RIC0
C                                                                       RIC0
 1150 CONTINUE                                                          RIC0
      IF( I .EQ. 1) GO TO 925                                           RIC0
      CALL MAXEL(DUMMY,NA,ANORM1)                                       RIC0
      CALL SUBT(P,NP,DUMMY,NA,DUMMY(N1),NA)                             RIC0
      CALL MAXEL(DUMMY(N1),NA,ANORM2)                                   RIC0
      IF( ANORM1 .GT. 1.) GO TO 1200                                    RIC0
      IF( ANORM2/ANORM1 .LT. RICTCV ) GO TO 1300                        RIC0
      GO TO 1250                                                        RIC0
 1200 CONTINUE                                                          RIC0
      IF( ANORM2 .LT. RICTCV ) GO TO 1300                               RIC0
C                                                                       RIC0
 1250 CONTINUE                                                          RIC0
      IF( I .LE. 101) GO TO  925                                        RIC0
      CALL LNCNT(3)                                                     RIC0
      PRINT 775                                                         RIC0
      IOP(1) = 1                                                        RIC0
C                                                                       RIC0
 1300 CONTINUE                                                          RIC0
      IF(IOP(1) .EQ. 0 ) RETURN                                         RIC0
      CALL LNCNT(4)                                                     RIC0
      PRINT 1350,I                                                      RIC0
 1350 FORMAT(//,' FINAL VALUES OF P AND F AFTER',I5,' ITERATIONS TO CONVRIC0
     1ERGE',/)                                                          RIC0
      CALL PRNT(P,NP,4H P  ,1)                                          RIC0
      CALL PRNT(F,NF,4H F  ,1)                                          RIC0
C                                                                       RIC0
      RETURN                                                            RIC0
      END                                                               RIC0
```

```
      SUBROUTINE ASMREG(A,NA,B,NB,H,NH,Q,NQ,R,NR,F,NF,P,NP,IDENT,DISC,N ASM0
     1EWT,STABLE,FNULL,ALPHA,IOP,DUMMY)                                 ASM0
      IMPLICIT REAL*8 (A-H,O-Z)                                         ASM0
      DIMENSION A(1),B(1),H(1),Q(1),R(1),F(1),P(1),DUMMY(1)             ASM0
      DIMENSION NA(2),NB(2),NH(2),NQ(2),NR(2),NF(2),NP(2),IOP(5),IOPT(3)ASM0
     1,NDUM1(2),NDUM2(2),NDUM3(2)                                       ASM0
      LOGICAL IDENT,DISC,NEWT,STABLE,FNULL,SING                         ASM0
      N = NA(1)**2                                                      ASM0
      N1= N+1                                                           ASM0
      IOPTT=0                                                           ASM0
      IF ( .NOT. NEWT ) GO TO 600                                       ASM0
      IF( STABLE )  GO TO 500                                           ASM0
      IF ( FNULL ) GO TO 100                                            ASM0
      CALL MULT(B,NB,F,NF,DUMMY,NA)                                     ASM0
      CALL SUBT(A,NA,DUMMY,NA,DUMMY,NA)                                 ASM0
      CALL TESTSA(DUMMY,NA,ALPHA,DISC,STABLE,IOPTT,DUMMY(N1))           ASM0
      GO TO 200                                                         ASM0
  100 CONTINUE                                                          ASM0
      CALL TESTSA(A,NA,ALPHA,DISC,STABLE,IOPTT,DUMMY)                   ASM0
C                                                                       ASM0
  200 CONTINUE                                                          ASM0
      IF( STABLE ) GO TO 500                                            ASM0
      IF( DISC ) GO TO 230                                              ASM0
      J = -NA(1)                                                        ASM0
      NAX = NA(1)                                                       ASM0
      DO 210 I =1,NAX                                                   ASM0
      J = J + NAX +1                                                    ASM0
      A(J) = A(J)-ALPHA                                                 ASM0
  210 CONTINUE                                                          ASM0
      SCLE = 3.                                                         ASM0
      IOPT(1)=IOP(1)                                                    ASM0
      IOPT(2) = 1                                                       ASM0
      IOPT(3)=1                                                         ASM0
      CALL CSTAB(A,NA,B,NB,F,NF,IOPT,SCLE,DUMMY)                        ASM0
      J = -NA(1)                                                        ASM0
      DO 220 I=1,NAX                                                    ASM0
      J = J + NAX + 1                                                   ASM0
      A(J) = A(J) + ALPHA                                               ASM0
  220 CONTINUE                                                          ASM0
  225 CONTINUE                                                          ASM0
      CALL MULT(B,NB,F,NF,DUMMY,NA)                                     ASM0
      CALL SUBT(A,NA,DUMMY,NA,DUMMY,NA)                                 ASM0
      CALL TESTSA(DUMMY,NA,ALPHA,DISC,STABLE,IOPTT,DUMMY(N1))           ASM0
      GO TO 300                                                         ASM0
C                                                                       ASM0
  230 CONTINUE                                                          ASM0
      J = 2*NA(1) + 1                                                   ASM0
      IF( .NOT. FNULL )  J = J + N                                      ASM0
      SING = .FALSE.                                                    ASM0
      IF( DUMMY(J) .EQ. 0.0 )  SING = .TRUE.                            ASM0
      IOPT(1) = IOP(1)                                                  ASM0
      IOPT(2) = 1                                                       ASM0
      DSCLE = 0.5                                                       ASM0
      ALPHAT = 1./ALPHA                                                 ASM0
      CALL SCALE(A,NA,A,NA,ALPHAT)                                      ASM0
      CALL SCALE(B,NB,B,NB,ALPHAT)                                      ASM0
      CALL DSTAB(A,NA,B,NB,F,NF,SING,IOPT,DSCLE,DUMMY)                  ASM0
      CALL SCALE(A,NA,A,NA,ALPHA)                                       ASM0
      CALL SCALE(B,NB,B,NB,ALPHA)                                       ASM0
      GO TO 225                                                         ASM0
C                                                                       ASM0
  300 CONTINUE                                                          ASM0
      IF( STABLE) GO TO 400                                             ASM0
```

```
      CALL LNCNT(5)                                                     ASM0(
      IF( DISC ) GO TO 330                                              ASM0(
      PRINT 310,ALPHA                                                   ASM0(
  310 FORMAT(//' IN ASMREG, CSTAB HAS FAILED TO FIND A STABILIZING GAIN ASM0(
     1 MATRIX (F) RELATIVE TO ',/,' ALPHA = ',D16.8/)                   ASM0
      RETURN                                                            ASM0(
  330 CONTINUE                                                          ASM0(
      PRINT 340,ALPHA                                                   ASM0(
  340 FORMAT(//' IN ASMREG, DSTAB HAS FAILED TO FIND A STABILIZING GAIN ASM0(
     1 MATRIX (F) RELATIVE TO ',/,' ALPHA = ',D16.8/)                   ASM0(
      RETURN                                                            ASM0(
C                                                                       ASM0(
  400 CONTINUE                                                          ASM0(
      FNULL = .FALSE.                                                   ASM0(
C                                                                       ASM0(
  500 CONTINUE                                                          ASM0(
      CALL RICNWT(A,NA,B,NB,H,NH,Q,NQ,R,NR,F,NF,P,NP,IOP,IDENT,DISC,FNU ASM0(
     1LL,DUMMY)                                                         ASM0(
      GO TO 750                                                         ASM0(
C                                                                       ASM0(
  600 CONTINUE                                                          ASM0(
      IF( DISC ) GO TO 700                                              ASM0(
      NW = 4*N + 1                                                      ASM0(
      NLAM = NW + 4*N                                                   ASM0(
      NDUM = NLAM + N                                                   ASM0(
      IOP(3) = 1                                                        ASM0(
      CALL CNTREG(A,NA,B,NB,H,NH,Q,NQ,R,NR,DUMMY,DUMMY(NW),DUMMY(NLAM), ASM00
     1S,F,NF,P,NP,T,IOP,IDENT,DUMMY(NDUM))                              ASM00
      GO TO 750                                                         ASM00
  700 CONTINUE                                                          ASM00
      CALL DISREG(A,NA,B,NB,H,NH,Q,NQ,R,NR,F,NF,P,NP,IOP,IDENT,DUMMY)   ASM00
C                                                                       ASM00
  750 CONTINUE                                                          ASM00
C                                                                       ASM00
      IF( IOP(4) .EQ. 0 ) GO TO 1100                                    ASM00
C                                                                       ASM00
      N2= N1 + N                                                        ASM01
      N3= N2 + N                                                        ASM01
C                                                                       ASM01
      IF( DISC ) GO TO 800                                              ASM01
      CALL MULT(P,NP,B,NB,DUMMY,NB)                                     ASM01
      CALL MULT(DUMMY,NB,F,NF,DUMMY(N1),NP)                             ASM01
      CALL TRANP(DUMMY(N1),NP,DUMMY,NP)                                 ASM01
      CALL ADD(DUMMY,NP,DUMMY(N1),NP,DUMMY,NP)                          ASM01
      CALL SCALE(DUMMY,NP,DUMMY,NP,0.5)                                 ASM01
      CALL SUBT(Q,NQ,DUMMY,NP,DUMMY,NP)                                 ASM01
      CALL MULT(P,NP,A,NA,DUMMY(N1),NP)                                 ASM01
      CALL ADD(DUMMY,NP,DUMMY(N1),NP,DUMMY,NP)                          ASM01
      CALL TRANP(DUMMY(N1),NP,DUMMY(N2),NP)                             ASM01
      CALL ADD(DUMMY,NP,DUMMY(N2),NP,DUMMY,NP)                          ASM01
      GO TO 900                                                         ASM01
C                                                                       ASM01
  800 CONTINUE                                                          ASM01
      CALL MULT(R,NR,F,NF,DUMMY,NF)                                     ASM01
      CALL TRANP(F,NF,DUMMY(N1),NB)                                     ASM01
      CALL MULT(DUMMY(N1),NB,DUMMY,NF,DUMMY(N2),NA)                     ASM01
      CALL ADD(DUMMY(N2),NA,Q,NQ,DUMMY,NA)                              ASM01
      CALL MULT(B,NB,F,NF,DUMMY(N1),NA)                                 ASM01
      CALL SUBT(A,NA,DUMMY(N1),NA,DUMMY(N1),NA)                         ASM01
      CALL MULT(P,NP,DUMMY(N1),NA,DUMMY(N2),NA)                         ASM01
      CALL TRANP(DUMMY(N1),NA,DUMMY(N3),NA)                             ASM01
      CALL MULT(DUMMY(N3),NA,DUMMY(N2),NA,DUMMY(N1),NA)                 ASM01
      CALL ADD(DUMMY,NA,DUMMY(N1),NA,DUMMY,NA)                          ASM01
```

```
      CALL SUBT(P,NP,DUMMY,NA,DUMMY,NA)                                 ASM01
C                                                                       ASM01
  900 CONTINUE                                                          ASM01
      CALL LNCNT(4)                                                     ASM01
      PRINT 1000                                                        ASM01
 1000 FORMAT(//' RESIDUAL ERROR IN RICCATI EQUATION '/)                 ASM01
      CALL PRNT(DUMMY,NP,4HERROR,1)                                     ASM01
C                                                                       ASM01
 1100 CONTINUE                                                          ASM01
      N2= N1+NA(1)                                                      ASM01
      N3= N2+NA(1)                                                      ASM01
      ISV = 0                                                           ASM01
      CALL EQUATE(P,NP,DUMMY,NP)                                        ASM01
      CALL EIGEN(NA(1),NA(1),DUMMY,DUMMY(N1),DUMMY(N2),ISV,ISV,V,DUMMY(NASM01
     13),IERR)                                                          ASM01
      NEVL = NA(1)                                                      ASM01
      IF( IERR .EQ. 0) GO TO 1300                                       ASM01
      NEVL=NA(1)-IERR                                                   ASM01
      CALL LNCNT(4)                                                     ASM01
      PRINT 1200,IERR                                                   ASM01
 1200 FORMAT(//' IN ASMREG, THE ',I5,' EIGENVALUE OF P  HAS NOT BEEN COASM01
     1PUTED AFTER 30 ITERATIONS ')                                      ASM01
C                                                                       ASM01
 1300 CONTINUE                                                          ASM01
      NDUM1(1) = NEVL                                                   ASM01
      NDUM1(2) = 1                                                      ASM01
      CALL EQUATE(DUMMY(N1),NDUM1,DUMMY,NDUM1)                          ASM01
      N1 = NDUM1(1) +1                                                  ASM01
      CALL MULT(B,NB,F,NF,DUMMY(N1),NA)                                 ASM01
      CALL SUBT(A,NA,DUMMY(N1),NA,DUMMY(N1),NA)                         ASM01
      N2 = N1+N                                                         ASM01
      CALL EQUATE(DUMMY(N1),NA,DUMMY(N2),NA)                            ASM01
      N3=N2+N                                                           ASM01
      N4=N3+NA(1)                                                       ASM01
      N5=N4+NA(1)                                                       ASM01
      CALL EIGEN(NA(1),NA(1),DUMMY(N2),DUMMY(N3),DUMMY(N4),ISV,ISV,V,DUMASM01
     1MY(N5),IERR)                                                      ASM01
      NEVL = NA(1)                                                      ASM01
      IF( IERR .EQ. 0 ) GO TO 1500                                      ASM01
      NEVL=NA(1)-IERR                                                   ASM01
      CALL LNCNT(4)                                                     ASM01
      PRINT 1400,IERR                                                   ASM01
 1400 FORMAT(//' IN ASMREG, THE ',I5,' EIGENVALUE OF A-BF HAS NOT BEEN CASM01
     1OMPUTED AFTER 30 ITERATIONS'/)                                    ASM01
C                                                                       ASM01
 1500 CONTINUE                                                          ASM01
      NDUM2(1) = NEVL                                                   ASM01
      NDUM2(2) = 1                                                      ASM01
      CALL JUXTC(DUMMY(N3),NDUM2,DUMMY(N4),NDUM2,DUMMY(N2),NDUM3)       ASM01
C                                                                       ASM01
      IF ( IOP(5) .EQ. 0 ) RETURN                                       ASM01
C                                                                       ASM01
      CALL LNCNT(4)                                                     ASM01
      PRINT 1600                                                        ASM01
 1600 FORMAT(//' EIGENVALUES OF P '/)                                   ASM01
      CALL PRNT(DUMMY,NDUM1,4HEVLP,1)                                   ASM01
      CALL LNCNT(4)                                                     ASM01
      PRINT 1700                                                        ASM01
 1700 FORMAT(//' CLOSED-LOOP RESPONSE MATRIX A-BF '/)                   ASM01
      CALL PRNT(DUMMY(N1),NA,4HA-BF,1)                                  ASM01
      CALL LNCNT(3)                                                     ASM01
      PRINT 1800                                                        ASM01
 1800 FORMAT(//' EIGENVALUES OF A-BF')                                  ASM01
```

```
      CALL PRNT('UMMY(N2)',NDUM3,0,3)                                   ASM0
C                                                                       ASM0
      RETURN                                                            ASM0
      END                                                               ASM0
```

```
      SUBROUTINE ASMFIL(A,NA,G,NG,H,NH,Q,NQ,R,NR,F,NF,P,NP,IDENT,DISC,N
     1EWT,STABLE,FNULL,ALPHA,IOP,DUMMY)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(1),G(1),H(1),Q(1),R(1),F(1),P(1),DUMMY(1)
      DIMENSION NA(2),NG(2),NH(2),NQ(2),NR(2),NF(2),NP(2),IOPT(5),NDUM1(
     12),IOP(1)
      LOGICAL  IDENT,DISC,NEWT,STABLE,FNULL
      IF( IOP(1) .EQ. 0 ) GO TO 100
      CALL LNCNT(4)
      IF(DISC)  PRINT 15
      IF( .NOT. DISC )  PRINT 25
   15 FORMAT(//,' PROGRAM TO SOLVE THE DISCRETE INFINITE-DURATION OPTIMA
     1L FILTER PROBLEM',/)
   25 FORMAT(//,' PROGRAM TO SOLVE THE CONTINUOUS INFINITE-DURATION OPTI
     1MAL FILTER PROBLEM',/)
      CALL PRNT(A,NA,4H A  ,1)
      IF( .NOT.  IDENT )  GO TO 35
      CALL LNCNT(3)
      PRINT 30
   30 FORMAT(/,' G IS AN IDENTITY MATRIX',/)
      GO TO 40
   35 CONTINUE
      CALL PRNT(G,NG,4H G  ,1)
   40 CONTINUE
      CALL PRNT(H,NH,4H H  ,1)
      CALL LNCNT(3)
      PRINT 45
   45 FORMAT(/,'INTENSITY MATRIX FOR COVARIANCE OF MEASUREMENT NOISE',/)
      CALL PRNT(R,NR,4H R  ,1)
C
      IF( .NOT. IDENT ) GO TO 65
      CALL LNCNT(3)
      PRINT 55
   55 FORMAT(/,' INTENSITY MATRIX FOR COVARIANCE OF PROCESS NOISE',/)
C
   65 CONTINUE
      CALL PRNT(Q,NQ,4H Q  ,1)
C
  100 CONTINUE
      IOPT(1)=IOP(2)
      IOPT(2)=IOP(3)
      IOPT(3)=IOP(4)
      IOPT(4)=IOP(5)
      IOPT(5)=0
      K = 0
C
  200 CONTINUE
      CALL TRANP(A,NA,DUMMY,NA)
      CALL EQUATE(DUMMY,NA,A,NA)
      CALL TRANP(H,NH,DUMMY,NDUM1)
      CALL EQUATE(DUMMY,NDUM1,H,NH)
      IF( IDENT )  GO TO 250
      CALL TRANP(G,NG,DUMMY,NDUM1)
      CALL EQUATE(DUMMY,NDUM1,G,NG)
  250 CONTINUE
      IF ( K .EQ. 1 ) RETURN
C
      K = K+1
      CALL ASMREG(A,NA,H,NH,G,NG,Q,NQ,R,NR,F,NF,P,NP,IDENT,DISC,NEWT,ST
     1ABLE,FNULL,ALPHA,IOPT,DUMMY)
C
      N1=(NA(1)**2)+3*NA(1)+1
      CALL TRANP(F,NF,DUMMY(N1),NDUM1)
```

```
      CALL EQUATE(DUMMY(N1),NDUM1,F,NF)                                 ASM0
C                                                                       ASM0
      IF( IOP(1) .EQ. 0 ) GO TO 200                                     ASM0
C                                                                       ASM0
      IF(IDENT) GO TO 300                                               ASM0
      CALL LNCNT(3)                                                     ASM0
      PRINT 55                                                          ASM0
      CALL PRNT(Q,NQ,4HGOGT,1)                                          ASM0
C                                                                       ASM0
  300 CONTINUE                                                          ASM0
      CALL LNCNT(3)                                                     ASM0
      PRINT 325                                                         ASM0
  325 FORMAT(/,' FILTER GAIN',/)                                        ASM0
      CALL PRNT(F,NF,4H F  ,1)                                          ASM0
      CALL LNCNT(3)                                                     ASM0
      PRINT 350                                                         ASM0
  350 FORMAT(/,'STEADY-STATE VARIANCE MATRIX OF RECONSTRUCTION ERROR',/)ASM0
      CALL PRNT(P,NP,4H P  ,1)                                          ASM0
      NDUM1(1)=NP(1)                                                    ASM0
      NDUM1(2)=1                                                        ASM0
      CALL LNCNT(3)                                                     ASM0
      PRINT 375                                                         ASM0
  375 FORMAT(/,' EIGENVALUES OF P ',/)                                  ASM0
      CALL PRNT(DUMMY,NDUM1,4HEVLP,1)                                   ASM0
      N1 = NP(1) + 1                                                    ASM0
      N = NA(1)**2                                                      ASM0
      N2 = N1 + N + 2*NA(1)                                             ASM0
      CALL TRANP(DUMMY(N1),NA,DUMMY(N2),NA)                             ASM0
      CALL PRNT(DUMMY(N2),NA,4HA-FH,1)                                  ASM0
      N2 = N1 + N                                                       ASM0
      CALL LNCNT(3)                                                     ASM0
      PRINT 385                                                         ASM0
  385 FORMAT(/,' EIGENVALUES OF A-FH MATRIX',/)                         ASM0
      NDUM1(1) = NA(1)                                                  ASM0
      NDUM1(2) =  2                                                     ASM0
      CALL PRNT(DUMMY(N2),NDUM1,0,3)                                    ASM0
C                                                                       ASM0
      GO TO 200                                                         ASM0
C                                                                       ASM0
      END                                                               ASM0
```

```
      SUBROUTINE EXPMDF (A,NA,B,NB,H,NH,AM,NAM,HM,NHM,Q,NQ,R,NR,F,NF,P, EXP0
     1NP,HIDENT,HMDENT,DISC,NEWT,STABLE,FNULL,ALPHA,IOP,DUMMY)          EXP0
      IMPLICIT REAL*8 (A-H,O-Z)                                         EXP0
      DIMENSION A(1),B(1),H(1),AM(1),HM(1),Q(1),R(1),F(1),P(1),DUMMY(1) EXP0
      DIMENSION NA(2),NB(2),NH(2),NAM(2),NHM(2),NQ(2),NR(2),NF(2),NP(2),EXP0
     1IOP(1),IOPT(5),NDUM1(2),NDUM2(2),NDUM3(2)                         EXP0
      LOGICAL  HIDENT,HMDENT,DISC,NEWT,STABLE,FNULL,SYM                 EXP0
      COMMON/TOL/EPSAM,EPSBM,IACM                                       EXP0
      IF( IOP(1) .EQ. 0 ) GO TO 300                                     EXP0
      CALL LNCNT(6)                                                     EXP0
      IF( DISC ) PRINT 25                                               EXP0
      IF( .NOT. DISC ) PRINT 50                                         EXP0
   25 FORMAT(/,' PROGRAM TO SOLVE ASYMPTOTIC DISCRETE EXPLICIT MODEL-FOLEXP0
     1LOWING PROBLEM',//,' PLANT DYNAMICS',/)                           EXP0
   50 FORMAT(/,' PROGRAM TO SOLVE ASYMPTOTIC CONTINUOUS EXPLICIT MODEL-FEXP0
     1OLLOWING PROBLEM',//,' PLANT DYNAMICS',/)                         EXP0
      CALL PRNT(A,NA,4H A  ,1)                                          EXP0
      CALL PRNT(B,NB,4H B  ,1)                                          EXP0
      IF( HIDENT ) GO TO 75                                             EXP0
      CALL PRNT(H,NH,4H H  ,1)                                          EXP0
      GO TO 100                                                         EXP0
   75 CONTINUE                                                          EXP0
      CALL LNCNT(3)                                                     EXP0
      PRINT 85                                                          EXP0
   85 FORMAT(/,' H IS AN IDENTITY MATRIX',/)                            EXP0
C                                                                       EXP0
  100 CONTINUE                                                          EXP0
      CALL LNCNT(4)                                                     EXP0
      PRINT 125                                                         EXP0
  125 FORMAT(//,' MODEL DYNAMICS',/)                                    EXP0
      CALL PRNT(AM,NAM,4H AM ,1)                                        EXP0
      IF( HMDENT ) GO TO 175                                            EXP0
      CALL PRNT(HM,NHM,4H HM ,1)                                        EXP0
      GO TO 200                                                         EXP0
  175 CONTINUE                                                          EXP0
      CALL LNCNT(3)                                                     EXP0
      PRINT 185                                                         EXP0
  185 FORMAT(/,' HM IS AN IDENTITY MATRIX ',/)                          EXP0
C                                                                       EXP0
  200 CONTINUE                                                          EXP0
      CALL LNCNT(4)                                                     EXP0
      PRINT 225                                                         EXP0
  225 FORMAT(//,' WEIGHTING MATRICES ',/)                               EXP0
      CALL PRNT(Q,NQ,4H Q  ,1)                                          EXP0
      CALL PRNT(R,NR,4H R  ,1)                                          EXP0
C                                                                       EXP0
  300 CONTINUE                                                          EXP0
      IF( IOP(2) .EQ. 0 ) GO TO 400                                     EXP0
      NF(1) = NB(2)                                                     EXP0
      NF(2) = NA(1)                                                     EXP0
      NP(1) = NA(1)                                                     EXP0
      NP(2) = NA(1)                                                     EXP0
      IOPT(1) = IOP(3)                                                  EXP0
      IOPT(2) = IOP(4)                                                  EXP0
      IOPT(3) = IOP(5)                                                  EXP0
      IOPT(4) =  0                                                      EXP0
      IOPT(5) =  0                                                      EXP0
      N1 = NA(1)*NA(2) + 1                                              EXP0
      CALL EQUATE(Q,NQ,DUMMY,NQ)                                        EXP0
      CALL ASMREG(A,NA,B,NB,H,NH,DUMMY,NQ,R,NR,F,NF,P,NP,HIDENT,DISC,NE EXP0
     1WT,STABLE,FNULL,ALPHA,IOPT,DUMMY(N1))                             EXP0
C                                                                       EXP0
  400 CONTINUE                                                          EXP0
```

```
      IF( IOP(1) .EQ. 0 ) GO TO 600                                     EXPOC
      CALL LNCNT(4)                                                     EXPOC
      PRINT 425                                                         EXPOC
  425 FORMAT(//,' CONTROL LAW U = -F( COL.(X,XM) ),  F = (F11,F12)',/)  EXPOC
      CALL LNCNT(3)                                                     EXPOC
      PRINT 450                                                         EXPOC
  450 FORMAT(/,' PART OF F MULTIPLYING  X ',/)                          EXPOC
      CALL PRNT(F,NF,4H F11,1)                                          EXPOC
      IF( .NOT. DISC  .AND. IOP(2) .EQ. 0 ) GO  TO 600                  EXPOC
      CALL PRNT(P,NP,4H P11,1)                                          EXPOC
      IF(  IOP(2) .EQ. 0 ) GO TO 600                                    EXPOC
      CALL LNCNT(2)                                                     EXPOC
      PRINT 475                                                         EXPOC
  475 FORMAT(/,' EIGENVALUES OF P11')                                   EXPOC
      NDUM1(1) = NA(1)                                                  EXPOC
      NDUM1(2) = 1                                                      EXPOC
      CALL PRNT(DUMMY(N1),NDUM1,0,3)                                    EXPOC
      N1 = N1 + NDUM1(1)                                                EXPOC
      NDUM1(2) = NA(1)                                                  EXPOC
      CALL LNCNT(2)                                                     EXPOC
      PRINT 500                                                         EXPOC
  500 FORMAT(/,' PLANT CLOSED-LOOP RESPONSE MATRIX A - BF11')           EXPOC
      CALL PRNT(DUMMY(N1),NDUM1,0,3)                                    EXPOC
      CALL LNCNT(2)                                                     EXPOC
      PRINT 525                                                         EXPOC
  525 FORMAT(/,' EIGENVALUES OF CLOSED-LOOP RESPONSE MATRIX')           EXPOC
      N1 = N1 + NDUM1(1)*NDUM1(2)                                       EXPOC
      NDUM1(2) = 2                                                      EXPOC
      CALL PRNT(DUMMY(N1),NDUM1,0,3)                                    EXPOC
C                                                                       EXPOC
  600 CONTINUE                                                          EXPOC
      NF(1)= NB(2)                                                      EXPOC
      NF(2)= NA(1)                                                      EXPOC
      CALL MULT(B,NB,F,NF,DUMMY,NA)                                     EXPOC
      CALL SUBT(A,NA,DUMMY,NA,DUMMY,NA)                                 EXPOC
      IF(  IOP(1).EQ. 0  .OR.  IOP(2) .NE. 0 ) GO TO 700                EXPOC
      CALL LNCNT(2)                                                     EXPO
      PRINT 500                                                         EXPO
      CALL PRNT(DUMMY,NA,0,3)                                           EXPO
C                                                                       EXPO
  700 CONTINUE                                                          EXPO
      N1 =  NA(1)**2 +1                                                 EXPO
      CALL TRANP(DUMMY,NA,DUMMY(N1),NA)                                 EXPO
      CALL EQUATE(DUMMY(N1),NA,DUMMY,NA)                                EXPO
      NF(2) = NA(1) + NAM(1)                                            EXPO
      NP(2) = NF(2)                                                     EXPO
      IF( .NOT. DISC .AND. IOP(2).EQ. 0 ) NP(2) = NAM(2)                EXPO
      IOPTT=0                                                           EXPO
      SYM = .FALSE.                                                     EXPO
      CALL EQUATE( Q,NQ,DUMMY(N1),NDUM2)                                EXPO
      IF( HMDENT ) GO TO 725                                            EXPO
      CALL MULT(Q,NQ,HM,NHM,DUMMY(N1),NDUM2)                            EXPO
  725 CONTINUE                                                          EXPO
      IF( HIDENT ) GO TO 750                                            EXPO
      N2 = N1 + NQ(1)*NHM(2)                                            EXPO
      CALL TRANP(H,NH,DUMMY(N2),NDUM1)                                  EXPO
      N3 = N2 + NH(1)*NH(2)                                             EXPO
      CALL MULT(DUMMY(N2),NDUM1,DUMMY(N1),NHM,DUMMY(N3),NDUM2)          EXPO
      CALL EQUATE(DUMMY(N3),NDUM2,DUMMY(N1),NDUM2)                      EXPO
  750 CONTINUE                                                          EXPO
      N2 = NA(1)**2 + NA(1)*NHM(2) + 1                                  EXPO
      N3 = NA(1)**2 + 1                                                 EXPO
      IF( .NOT. DISC .AND. IOP(2) .EQ. 0 ) N3 = 1                       EXPO
```

```
      CALL EQUATE(DUMMY(N1),NDUM2,P(N3),NDUM2)                          EXP0
      IF( DISC ) GO TO 800                                              EXP0
      EPSA = EPSAM                                                      EXP0
      CALL BARSTW(DUMMY,NA,AM,NAM,P(N3),NDUM2,IOPTT,SYM ,EPSA,EPSA,DUMMYEXP0
     1(N2))                                                             EXP0
      GO TO 900                                                         EXP0
C                                                                       EXP0
  800 CONTINUE                                                          EXP0
      CALL SCALE(P(N3),NDUM2,P(N3),NDUM2,-1.0)                          EXP0
      N4 = N2 +NAM(1)**2                                                EXP0
      CALL EQUATE(AM,NAM,DUMMY(N2),NAM)                                 EXP0
      CALL SUM(DUMMY,NA,P(N3),NDUM2,DUMMY(N2),NAM,IOPTT,SYM,DUMMY(N4))  EXP0
C                                                                       EXP0
  900 CONTINUE                                                          EXP0
      N2 = NB(2)*NA(1) + 1                                              EXP0
      CALL TRANP(B,NB,DUMMY,NDUM1)                                      EXP0
      CALL MULT(DUMMY,NDUM1,P(N3),NDUM2,F(N2),NDUM3)                    EXP0
      IF( .NOT. DISC ) GO TO 1000                                       EXP0
      N1 = NB(1)*NB(2) + 1                                              EXP0
      CALL MULT(DUMMY,NDUM1,P,NA,DUMMY(N1),NDUM2)                       EXP0
      CALL MULT(DUMMY(N1),NDUM2,B,NB,DUMMY,NR)                          EXP0
      CALL ADD(R,NR,DUMMY,NR,DUMMY,NR)                                  EXP0
      GO TO 1100                                                        EXP0
C                                                                       EXP0
 1000 CONTINUE                                                          EXP0
      CALL EQUATE(R,NR,DUMMY,NR)                                        EXP0
C                                                                       EXP0
 1100 CONTINUE                                                          EXP0
      N1 = NR(1)**2 + 1                                                 EXP0
      CALL SYMPDS(NR(1),NR(1),DUMMY,NHM(2),F(N2),IOPTT,IOPTT,DETERM,ISCAEXP0
     1LE,DUMMY(N1),IERR)                                                EXP0
      IF( IERR .EQ. 0 ) GO TO 1200                                      EXP0
      CALL LNCNT(3)                                                     EXP0
      PRINT 1150                                                        EXP0
 1150 FORMAT(/,' IN EXPMDF, THE COEFFICIENT MATRIX FOR SYMPDS IS NOT SY EXP0
     1MMETRIC POSITIVE DEFINITE ',/)                                    EXP0
      RETURN                                                            EXP0
C                                                                       EXP0
 1200 CONTINUE                                                          EXP0
      IF( .NOT. DISC ) GO TO 1300                                       EXP0
      CALL MULT(F(N2),NDUM3,AM,NAM,DUMMY,NDUM1)                         EXP0
      CALL EQUATE(DUMMY,NDUM1,F(N2),NDUM1)                              EXP0
 1300 CONTINUE                                                          EXP0
      IF( IOP(1) .EQ. 0 ) RETURN                                        EXP0
      CALL LNCNT(3)                                                     EXP0
      PRINT 1325                                                        EXP0
 1325 FORMAT(/,' PART OF F MULTIPLYING XM ',/)                          EXP0
      CALL PRNT(F(N2),NDUM3,4H F12,1)                                   EXP0
      NDUM1(1) = NA(1)                                                  EXP0
      NDUM1(2) = NAM(1)                                                 EXP0
      CALL PRNT(P(N3),NDUM1,4H P12,1)                                   EXP0
      RETURN                                                            EXP0
      END                                                               EXP0
```

```
      SUBROUTINE IMPMDF(A,NA,B,NB,H,NH,AM,NAM,BM,NBM,Q,NQ,R,NR,F,NF,P,N IMP0
     1P,IDENT,DISC,NEWT,STABLE,FNULL,ALPHA,IOP,DUMMY)                   IMP0
      IMPLICIT REAL*8 (A-H,O-Z)                                         IMP0
      DIMENSION A(1),B(1),H(1),AM(1),BM(1),Q(1),R(1),F(1),P(1),DUMMY(1) IMP0
      DIMENSION NA(2),NB(2),NH(2),NAM(2),NBM(2),NQ(2),NR(2),NF(2),NP(2),IMP0
     1IOP(1),IOPT(5),NDUM1(2)                                           IMP0
      LOGICAL IDENT,DISC,NEWT,STABLE,FNULL,HIDENT                       IMP0
      IF( IOP(1) .EQ. 0 ) GO TO 200                                     IMP0
      CALL LNCNT(6)                                                     IMP0
      IF( DISC ) PRINT 25                                               IMP0
      IF( .NOT. DISC ) PRINT 50                                         IMP0
   25 FORMAT(/,'PROGRAM TO SOLVE ASYMPTOTIC DISCRETE IMPLICIT MODEL-FOLLIMP0
     1OWING PROBLEM',//,' PLANT DYNAMICS ',/)                           IMP0
   50 FORMAT(/,' PROGRAM TO SOLVE ASYMPTOTIC CONTINUOUS IMPLICIT MODEL-FIMP0
     1OLLOWING PROBLEM',//,' PLANT DYNAMICS',/)                         IMP0
      CALL PRNT(A,NA,4H A  ,1)                                          IMP0
      CALL PRNT(B,NB,4H B  ,1)                                          IMP0
      IF( IDENT ) GO TO 75                                              IMP0
      CALL PRNT(H,NH,4H H  ,1)                                          IMP0
      GO TO 100                                                         IMP0
   75 CONTINUE                                                          IMP0
      CALL LNCNT(3)                                                     IMP0
      PRINT 85                                                          IMP0
   85 FORMAT(/,' H IS AN IDENTITY MATRIX',/)                            IMP0
C                                                                       IMP0
  100 CONTINUE                                                          IMP0
      CALL LNCNT(4)                                                     IMP0
      PRINT 125                                                         IMP0
  125 FORMAT(//,' MODEL DYNAMICS',/)                                    IMP0
      CALL PRNT(AM,NAM,4H AM ,1)                                        RMP0
      CALL PRNT(BM,NBM,4H BM ,1)                                        IMP0
      CALL LNCNT(4)                                                     IMP0
      PRINT 150                                                         IMP0
  150 FORMAT(//,' WEIGHTING MATRICES',/)                                IMP0
      CALL PRNT(Q,NQ,4H Q  ,1)                                          IMP0
      CALL PRNT(R,NR,4H R  ,1)                                          IMP0
C                                                                       IMP0
  200 CONTINUE                                                          IMP0
      N = NA(1)**2                                                      IMP0
      N1 = N + 1                                                        IMP0
      IF( .NOT. IDENT ) GO TO 300                                       IMP0
      CALL SUBT(A,NA,AM,NAM,DUMMY,NA)                                   IMP0
      CALL SUBT(B,NB,BM,NBM,DUMMY(N1),NB)                               IMP0
      GO TO 400                                                         IMP0
C                                                                       IMP0
  300 CONTINUE                                                          IMP0
      CALL MULT(H,NH,A,NA,DUMMY,NH)                                     IMP0
      CALL MULT(AM,NAM,H,NH,DUMMY(N1),NH)                               IMP0
      CALL SUBT(DUMMY,NH,DUMMY(N1),NH,DUMMY,NH)                         IMP0
      CALL MULT(H,NH,B,NB,DUMMY(N1),NBM)                                IMP0
      CALL SUBT(DUMMY(N1),NBM,BM,NBM,DUMMY(N1),NBM)                     IMP0
C                                                                       IMP0
  400 CONTINUE                                                          IMP0
      IF( IOP(1) .EQ. 0 ) GO TO 500                                     IMP0
      CALL LNCNT(3)                                                     IMP0
      PRINT 450                                                         IMP0
  450 FORMAT(//,' MATRIX HA - AMH')                                     IMP0
      CALL PRNT(DUMMY,NH,0,3)                                           IMP0
      CALL LNCNT(3)                                                     IMP0
      PRINT 475                                                         IMP0
  475 FORMAT(//,' MATRIX HB - BM')                                      IMP0
      CALL PRNT(DUMMY(N1),NBM,0,3)                                      IMP0
C                                                                       IMP0
```

```
  500 CONTINUE                                                          IMP
      N2 = N1 + N                                                       IMP
      N3 = N2 + N                                                       IMP
      N4 = N3 + N                                                       IMP
      CALL MULT(Q,NQ,DUMMY,NH,DUMMY(N2),NH)                             IMP
      CALL MULT(Q,NQ,DUMMY(N1),NBM,DUMMY(N3),NBM)                       IMP
      CALL TRANP(DUMMY,NH,DUMMY(N4),NDUM1)                              IMP
      CALL MULT(DUMMY(N4),NDUM1,DUMMY(N2),NH,DUMMY,NA)                  IMP
      CALL MULT(DUMMY(N4),NDUM1,DUMMY(N3),NBM,DUMMY(N2),NB)             IMP
      CALL TRANP(DUMMY(N1),NBM,DUMMY(N4),NDUM1)                         IMP
      CALL SCALE(DUMMY(N2),NB,DUMMY(N1),NB,2.0)                         IMP
      CALL MULT(DUMMY(N4),NDUM1,DUMMY(N3),NBM,DUMMY(N2),NR)             IMP
      CALL ADD(DUMMY(N2),NR,R,NR,DUMMY(N2),NR)                          IMP
      IF( IOP(1) .EQ. 0 ) GO TO 600                                     IMP
      CALL LNCNT(3)                                                     IMP
      PRINT 525                                                         IMP
  525 FORMAT(//,' MATRIX (HA - AMH TRANSPOSE)Q( HA - AMH)')             IMP
      CALL PRNT(DUMMY,NA,0,3)                                           IMP
      CALL LNCNT(3)                                                     IMP
      PRINT 550                                                         IMP
  550 FORMAT(//,' MATRIX 2( HA - AMH  TRANSPOSE)Q( HB - BM)')           IMP
      CALL PRNT(DUMMY(N1),NB,0,3)                                       IMP
      CALL LNCNT(3)                                                     IMP
      PRINT 575                                                         IMP
  575 FORMAT(//,' MATRIX ( HB - BM TRANSPOSE)Q( HB - BM ) + R')         IMP
      CALL PRNT(DUMMY(N2),NR,0,3)                                       IMP
C                                                                       IMP
  600 CONTINUE                                                          IMP
      IOPT(1)= 0                                                        IMP
      IOPT(2)= 1                                                        IMP
      IOPT(3)= 1                                                        IMP
      N5 = N4 + N                                                       IMP
      CALL EQUATE(A,NA,DUMMY(N3),NA)                                    IMP
      CALL PREFIL(DUMMY(N3),NA,B,NB,DUMMY,NA,DUMMY(N1),NB,DUMMY(N2),NR,DI IMP
     1UMMY(N4),NF,IOPT,DUMMY(N5))                                       IMP
      IF(IOP(1) .EQ. 0 ) GO TO 700                                      IMP
      CALL LNCNT(3)                                                     IMP
      PRINT 625                                                         IMP
  625 FORMAT(//,' PREFILTER GAIN')                                      IMP
      CALL PRNT(DUMMY(N4),NF,0,3)                                       IMP
      CALL LNCNT(3)                                                     IMP
      PRINT 650                                                         IMP
  650 FORMAT(//,' MATRIX A - B(PREFILTER)')                             IMP
      CALL PRNT(DUMMY(N3),NA,0,3)                                       IMP
      CALL LNCNT(3)                                                     IMP
      PRINT 675                                                         IMP
  675 FORMAT(//,' MODIFIED STATE VECTOR WEIGHTING MATRIX')              IMP
      CALL PRNT(DUMMY,NA,0,3)                                           IMP
C                                                                       IMP
  700 CONTINUE                                                          IMP
      CALL EQUATE(DUMMY(N4),NF,DUMMY(N1),NF)                            IMP
C                                                                       IMP
      IF( IOP(2) .EQ. -1000 ) RETURN                                    IMP
C                                                                       IMP
      IOPT(1) = IOP(2)                                                  IMP
      IOPT(2) = IOP(3)                                                  IMP
      IOPT(3) = IOP(4)                                                  IMP
      IOPT(4) = 0                                                       IMP
      IOPT(5) = 0                                                       IMP
      HIDENT = .TRUE.                                                   IMP
      CALL ASMREG(DUMMY(N3),NA,B,NB,H,NH,DUMMY,NA,DUMMY(N2),NR,F,NF,P,N IMP
     1P,HIDENT,DISC,NEXT,STABLE,FNULL,ALPHA,IOPT,DUMMY(N4))             IMP
      IF( IOP(1) .EQ. 0 ) GO TO 800                                     IMP
```

```
      CALL LNCNT(3)                                                     IMPO
      PRINT 725                                                         IMPO
  725 FORMAT(//,' GAIN FROM ASMREG')                                    IMPO
      CALL PRNT(F,NF,0,3)                                               IMPO
      CALL LNCNT(3)                                                     IMPO
      PRINT 750                                                         IMPO
  750 FORMAT(//,' SOLUTION OF ASSOCIATED STEADY-STATE RICCATI EQUATION')IMPO
      CALL PRNT(P,NP,0,3)                                               IMPO
      CALL LNCNT(3)                                                     IMPO
      PRINT 775                                                         IMPO
  775 FORMAT(//,' EIGENVALUES OF P')                                    IMPO
      NDUM1(1)= NA(1)                                                   IMPO
      NDUM1(2)= 1                                                       IMPO
      CALL PRNT(DUMMY(N4),NDUM1,0,3)                                    IMPO
C                                                                       IMPO
  800 CONTINUE                                                          IMPO
      CALL ADD(F,NF,DUMMY(N1),NF,F,NF)                                  IMPO
      IF( IOP(1) .EQ. 0 ) RETURN                                        IMPO
      CALL LNCNT(4)                                                     IMPO
      PRINT 825                                                         IMPO
  825 FORMAT(//,' GAIN FOR MODEL-FOLLOWING CONTROL LAW, U = - F X   , F =IMPO
     1(PREFILTER) + (ASMREG)',/)                                        IMPO
      CALL PRNT(F,NF,4H F   ,1)                                         IMPO
      N6 = N4 + NA(1)                                                   IMPO
      CALL PRNT(DUMMY(N6),NA,4HA-BF,1)                                  IMPO
      NDUM1(2) = 2                                                      IMPO
      N6 = N6 + N                                                       IMPO
      CALL LNCNT(3)                                                     IMPO
      PRINT 850                                                         IMPO
  850 FORMAT(//,' EIGENVALUES OF A-BF')                                 IMPO
      CALL PRNT(DUMMY(N6),NDUM1,0,3)                                    IMPO
C                                                                       IMPO
      RETURN                                                            IMPO
      END                                                               IMPO
```

```
  0        SUBROUTINE READ1 (A,NA,NZ,NAM)                                  REA
  1        IMPLICIT REAL*8 (A-H,O-Z)                                       REA
  2        DIMENSION A(1),NA(2),NZ(2)                                      REA
  3        IF  (NZ(1).EQ.0)  GO TO 410                                     REA
  4        NR=NZ(1)                                                        REA
  5        NC=NZ(2)                                                        REA
  6        NLST=NR*NC                                                      REA
  7        IF( NLST .LT. 1 .OR. NR .LT. 1 ) GO TO 16                       REA
  8        DO 400 I = 1, NR                                                REA
  9    400 READ (5,101) (A(J), J = I,NLST,NR)                              REA
 10        NA(1)=NR                                                        REA
 11        NA(2)=NC                                                        REA
 12    410 CALL  PRNT (A,NA,NAM,1)                                         REA
 13    101 FORMAT(8D10.2)                                                  REA
 14        RETURN                                                          REA
 15     16 CALL LNCNT(1)                                                   REA
 16        WRITE  (6,916)  NAM,NR,NC                                       REA
 17    916 FORMAT  (' ERROR IN READ1   MATRIX ',A4,' HAS NA=',2I6)         REA
 18        RETURN                                                          REA
 19        END                                                             REA
```

```
      SUBROUTINE BALANC(NM,N,A,LOW,IGH,SCALE)                           BAL00(
      IMPLICIT REAL*8 (A-H,O-Z)                                         BAL00(
      INTEGER I,J,K,L,M,N,JJ,NM,IGH,LOW,IEXC                            BAL00(
      DIMENSION A(NM,N),SCALE(N)                                        BAL00(
C     REAL C,F,G,R,S,B2,RADIX                                           BAL00(
C     REAL DABS                                                         BAL00(
      LOGICAL NOCONV                                                    BAL00(
C                                                                       BAL00(
C                                                                       BAL00(
C     ********** RADIX IS A MACHINE DEPENDENT PARAMETER SPECIFYING      BAL00
C                THE BASE OF THE MACHINE FLOATING POINT REPRESENTATION. BAL00
C                                                                       BAL00
C                                                                       BAL00
      RADIX = 16.                                                       BAL00
C                                                                       BAL00
      B2 = RADIX * RADIX                                                BAL00
      K = 1                                                             BAL00
      L = N                                                             BAL00
      GO TO 100                                                         BAL00
C     ********** IN-LINE PROCEDURE FOR ROW AND                          BAL00
C                COLUMN EXCHANGE **********                            BAL00
   20 SCALE(M) = J                                                      BJL00
      IF (J .EQ. M) GO TO 50                                            BAL00
C                                                                       BAL00
      DO 30 I = 1, L                                                    BAL00
         F = A(I,J)                                                     BAL00
         A(I,J) = A(I,M)                                                BAL00
         A(I,M) = F                                                     BAL00
   30 CONTINUE                                                          BAL00
C                                                                       BAL00
      DO 40 I = K, N                                                    BAL00
         F = A(J,I)                                                     BAL00
         A(J,I) = A(M,I)                                                BAL00
         A(M,I) = F                                                     BAL00
   40 CONTINUE                                                          BAL00
C                                                                       BAL00
   50 GO TO (80,130), IEXC                                              BAL00
C     ********** SEARCH FOR ROWS ISOLATING AN EIGENVALUE                BAL00
C                AND PUSH THEM DOWN **********                         BAL00
   80 IF (L .EQ. 1) GO TO 280                                           BAL00
      L = L - 1                                                         BAL00
C     ********** FOR J=L STEP -1 UNTIL 1 DO -- **********              BAL00
  100 DO 120 JJ = 1, L                                                  BAL00
         J = L + 1 - JJ                                                 BAL00
C                                                                       BAL00
         DO 110 I = 1, L                                                BAL00
            IF (I .EQ. J) GO TO 110                                     BAL00
            IF (A(J,I) .NE. 0.0D0) GO TO 120                            BAL00
  110    CONTINUE                                                       BAL00
C                                                                       BAL00
         M = L                                                          BAL00
         IEXC = 1                                                       BAL00
         GO TO 20                                                       BAL00
  120 CONTINUE                                                          BAL00
C                                                                       BAL00
      GO TO 140                                                         BAL00
C     ********** SEARCH FOR COLUMNS ISOLATING AN EIGENVALUE             BAL00
C                AND PUSH THEM LEFT **********                         BAL00
  130 K = K + 1                                                         BAL00
C                                                                       BJL00
  140 DO 170 J = K, L                                                   BAL00
C                                                                       BAL00
         DO 150 I = K, L                                                BAL00
```

```
                  IF (I .EQ. J) GO TO 150                                BAL00
                  IF (A(I,J) .NE. 0.0D0) GO TO 170                       BJL00
  150     CONTINUE                                                       BAL00
C                                                                        BAL00
                  M = K                                                  BAL00
                  IEXC = 2                                               BAL00
                  GO TO 20                                               BAL00
  170 CONTINUE                                                           BAL00
C     ********** NOW BALANCE THE SUBMATRIX IN ROWS K TO L **********     BAL00
            DO 180 I = K, L                                              BAL00
  180 SCALE(I) = 1.0D0                                                   BAL00
C     ********** ITERATIVE LOOP FOR NORM REDUCTION **********           BJL00
  190 NOCONV = .FALSE.                                                   BAL00
C                                                                        BAL00
            DO  270  I=K, L
               C = 0.0D0                                                 BAL00
               R = 0.0D0                                                 BAL00
C                                                                        BAL00
               DO 200 J = K, L                                           BAL00
                  IF (J .EQ. I) GO TO 200                                BAL00
                  C = C + DABS(A(J,I))                                   BAL00
                  R = R + DABS(A(I,J))                                   BAL00
  200          CONTINUE                                                  BAL00
C     ********** GUARD AGAINST ZERO C OR R DUE TO UNDERFLOW **********   BAL00
               IF (C .EQ. 0.0D0 .OR. R .EQ. 0.0D0) GO TO 270             BAL00
               G = R / RADIX                                             BAL00
               F = 1.0D0                                                 BAL00
               S = C + R                                                 BAL00
  210          IF (C .GE. G) GO TO 220                                   BAL00
               F = F * RADIX                                             BAL00
               C = C * B2                                                BAL00
               GO TO 210                                                 BAL00
  220          G = R * RADIX                                             BAL00
  230          IF (C .LT. G) GO TO 240                                   BAL00
               F = F / RADIX                                             BAL00
               C = C / B2                                                BAL00
               GO TO 230                                                 BAL01
C     ********** NOW BALANCE **********                                  BAL01
  240          IF ((C + R) / F .GE. 0.95 * S) GO TO 270                  BAL01
               G = 1.0D0 / F                                             BAL01
               SCALE(I) = SCALE(I) * F                                   BAL01
               NOCONV = .TRUE.                                           BAL01
C                                                                        BAL01
               DO 250 J = K, N                                           BAL01
  250          A(I,J) = A(I,J) * G                                       BAL01
C                                                                        BAL01
               DO 260 J = 1, L                                           BAL01
  260          A(J,I) = A(J,I) * F                                       BAL01
C                                                                        BAL01
  270 CONTINUE                                                           BAL01
C                                                                        BAL01
            IF (NOCONV) GO TO 190                                        BAL01
C                                                                        BAL01
  280 LOW = K                                                            BAL01
            IGH = L                                                      BAL01
            RETURN                                                       BAL01
C     ********** LAST CARD OF BALANC **********                         BAL01
            END                                                          BAL01
```

```
      SUBROUTINE ELMHES(NM,N,LOW,IGH,A,INT)                             ELM000
      IMPLICIT REAL*8 (A-H,O-Z)                                         ELM000
      INTEGER I,J,M,N,LA,NM,IGH,KP1,LOW,MM1,MP1                         ELM000
      DIMENSION A(NM,N)                                                 ELM000
C     REAL X,Y                                                          ELM000
C     REAL DABS                                                         ELM000
      INTEGER INT(IGH)                                                  ELM000
C                                                                       ELM000
      LA = IGH - 1                                                      ELM000
      KP1 = LOW + 1                                                     ELM001
      IF (LA .LT. KP1) GO TO 200                                        ELM001
C                                                                       ELM001
      DO 180 M = KP1, LA                                                ELM001
         MM1 = M - 1                                                    ELM001
         X = 0.0D0                                                      ELM001
         I = M                                                          ELM001
C                                                                       ELM001
         DO 100 J = M, IGH                                              ELM001
            IF (DABS(A(J,MM1)) .LE. DABS(X)) GO TO 100                  ELM001
            X = A(J,MM1)                                                ELM002
            I = J                                                       ELM002
  100    CONTINUE                                                       ELM002
C                                                                       ELM002
         INT(M) = I                                                     ELM002
         IF (I .EQ. M) GO TO 130                                        ELM002
C     ********** INTERCHANGE ROWS AND COLUMNS OF A **********          ELM002
         DO 110 J = MM1, N                                              ELM002
            Y = A(I,J)                                                  ELM002
            A(I,J) = A(M,J)                                             ELM002
            A(M,J) = Y                                                  ELM003
  110    CONTINUE                                                       ELM003
C                                                                       ELM003
         DO 120 J = 1, IGH                                              ELM003
            Y = A(J,I)                                                  ELM003
            A(J,I) = A(J,M)                                             ELM003
            A(J,M) = Y                                                  ELM003
  120    CONTINUE                                                       ELM003
C     ********** END INTERCHANGE **********                            ELM003
  130    IF (X .EQ. 0.0D0) GO TO 180                                    ELM003
         MP1 = M + 1                                                    ELM004
C                                                                       ELM004
         DO 160 I = MP1, IGH                                            ELM004
            Y = A(I,MM1)                                                ELM004
            IF (Y .EQ. 0.0D0) GO TO 160                                 ELM004
            Y = Y / X                                                   ELM004
            A(I,MM1) = Y                                                ELM004
C                                                                       ELM004
            DO 140 J = M, N                                             ELM004
  140       A(I,J) = A(I,J) - Y * A(M,J)                                ELM00.
C                                                                       ELM00
            DO 150 J = 1, IGH                                           ELM00
  150       A(J,M) = A(J,M) + Y * A(J,I)                                ELM00
C                                                                       ELM00
  160    CONTINUE                                                       ELM00
C                                                                       ELM00
  180 CONTINUE                                                          ELM00
C                                                                       ELM00
  200 RETURN                                                            ELM00
C     ********** LAST CARD OF ELMHES **********                        ELM00
      END                                                               ELM00
```

```
      SUBROUTINE HQR(NM,N,LOW,IGH,H,WR,WI,IERR)                         HQR0
      IMPLICIT REAL*8 (A-H,O-Z)                                         HQR00
       REAL*8 NORM,MACHEP                                               HQR0
      INTEGER I,J,K,L,M,N,EN,LL,MM,NA,NM,IGH,ITS,LOW,MP2,ENM2,IERR      HQR0
      DIMENSION H(NM,N),WR(N),WI(N)                                     HQR00
C     REAL P,Q,R,S,T,W,X,Y,ZZ,NORM,MACHEP                               HQR00
      REAL*8 DSQRT,DABS,DSIGN                                           HQR0
C     INTEGER MIN0                                                      HQR0
      LOGICAL NOTLAS                                                    HQR00
C                                                                       HQR0
C                                                                       HQR0
C     ********** MACHEP IS A MACHINE DEPENDENT PARAMETER SPECIFYING     HQR00
C                THE RELATIVE PRECISION OF FLOATING POINT ARITHMETIC.   HQR0
C                                                                       HQR0
C                                                                       HQR00
      MACHEP = 16.**(-13)                                               HQR00
C                                                                       HQR0
      IERR = 0                                                          HQR0
      NORM = 0.0D0                                                      HQR00
      K = 1                                                             HQR0
C     ********** STORE ROOTS ISOLATED BY BALANC                         HQR0
C                AND COMPUTE MATRIX NORM **********                     HQR00
      DO 50 I = 1, N                                                    HQR00
C                                                                       HQR0
         DO 40 J = K, N                                                 HQR0
   40    NORM = NORM + DABS(H(I,J))                                     HQR00
C                                                                       HQR0
         K = I                                                          HQR0
         IF (I .GE. LOW .AND. I .LE. IGH) GO TO 50                      HQR0
         WR(I) = H(I,I)                                                 HQR0
         WI(I) = 0.0D0                                                  HQR0
   50 CONTINUE                                                          HQR0
C                                                                       HQR00
      EN = IGH                                                          HQR0
      T = 0.0D0                                                         HQR0
C     ********** SEARCH FOR NEXT EIGENVALUES **********                 HQR00
   60 IF (EN .LT. LOW) GO TO 1001                                       HQR0
      ITS = 0                                                           HQR0
      NA = EN - 1                                                       HQR00
      ENM2 = NA - 1                                                     HQR0
C     ********** LOOK FOR SINGLE SMALL SUB-DIAGONAL ELEMENT             HQR0
C                FOR L=EN STEP -1 UNTIL LOW DO -- **********            HQR0
   70 DO 80 LL = LOW, EN                                                HQR0
         L = EN + LOW - LL                                              HQR0
         IF (L .EQ. LOW) GO TO 100                                      HQR0
         S = DABS(H(L-1,L-1)) + DABS(H(L,L))                            HQR0
         IF (S .EQ. 0.0D0) S = NORM                                     HQR0
         IF (DABS(H(L,L-1)) .LE. MACHEP * S) GO TO 100                  HQR0
   80 CONTINUE                                                          HQR0
C     ********** FORM SHIFT **********                                  HQR0
  100 X = H(EN,EN)                                                      HQR0
      IF (L .EQ. EN) GO TO 270                                          HQR0
      Y = H(NA,NA)                                                      HQR0
      W = H(EN,NA) * H(NA,EN)                                           HQR0
      IF (L .EQ. NA) GO TO 280                                          HQR0
      IF (ITS .EQ. 30) GO TO 1000                                       HQR0
      IF (ITS .NE. 10 .AND. ITS .NE. 20) GO TO 130                      HQR0
C     ********** FORM EXCEPTIONAL SHIFT **********                      HQR0
      T = T + X                                                         HQR0
C                                                                       HQR0
      DO 120 I = LOW, EN                                                HQR0
  120 H(I,I) = H(I,I) - X                                               HQR0
C                                                                       HQR00
```

```
      S = DABS(H(EN,NA)) + DABS(H(NA,ENM2))                              HQR0
      X = 0.75 * S                                                       HQR0
      Y = X                                                              HQR0
      W = -0.4375 * S * S                                                HQR0
  130 ITS = ITS + 1                                                      HQR0
C     ********** LOOK FOR TWO CONSECUTIVE SMALL                          HQR0
C                SUB-DIAGONAL ELEMENTS.                                 HQR0
C                FOR M=EN-2 STEP -1 UNTIL L DO -- **********            HQR0
      DO 140 MM = L, ENM2                                                HQR0
         M = ENM2 + L - MM                                               HQR0
         ZZ = H(M,M)                                                     HQR0
         R = X - ZZ                                                      HQR0
         S = Y - ZZ                                                      HQR0
         P = (R * S - W) / H(M+1,M) + H(M,M+1)                           HQR0
         Q = H(M+1,M+1) - ZZ - R - S                                     HQR0
         R = H(M+2,M+1)                                                  HQR0
         S = DABS(P) + DABS(Q) + DABS(R)                                 HQR0
         P = P / S                                                       HQR0
         Q = Q / S                                                       HQR0
         R = R / S                                                       HQR0
         IF (M .EQ. L) GO TO 150                                         HQR0
         IF (DABS(H(M,M-1)) * (DABS(Q) + DABS(R)) .LE. MACHEP * DABS(P) HQR0
     X      * (DABS(H(M-1,M-1)) + DABS(ZZ) + DABS(H(M+1,M+1)))) GO TO 150 HQR0
  140 CONTINUE                                                           HQR0
C                                                                        HQR0
  150 MP2 = M + 2                                                        HQR0
C                                                                        HQR0
      DO 160 I = MP2, EN                                                 HQR0
         H(I,I-2) = 0.0D0                                                HQR0
         IF (I .EQ. MP2) GO TO 160                                       HQR0
         H(I,I-3) = 0.0D0                                                HQR0
  160 CONTINUE                                                           HQR0
C     ********** DOUBLE QR STEP INVOLVING ROWS L TO EN AND               HQR0
C                COLUMNS M TO EN **********                             HQR0
      DO 260 K = M, NA                                                   HQR0
         NOTLAS = K .NE. NA                                              HQR0
         IF (K .EQ. M) GO TO 170                                         HQR0
         P = H(K,K-1)                                                    HQR0
         Q = H(K+1,K-1)                                                  HQR0
         R = 0.0D0                                                       HQR0
         IF (NOTLAS) R = H(K+2,K-1)                                      HQR0
         X = DABS(P) + DABS(Q) + DABS(R)                                 HQR0
         IF (X .EQ. 0.0D0) GO TO 260                                     HQR0
         P = P / X                                                       HQR0
         Q = Q / X                                                       HQR0
         R = R / X                                                       HQR0
  170    S = DSIGN(DSQRT(P*P+Q*Q+R*R),P)                                 QQR0
         IF (K .EQ. M) GO TO 180                                         HQR0
         H(K,K-1) = -S * X                                               HQR0
         GO TO 190
  180    IF (L .NE. M) H(K,K-1) = -H(K,K-1)                              QQR0
  190    P = P + S                                                       HQR0
         X = P / S                                                       HQR0
         Y = Q / S                                                       HQR0
         ZZ = R / S                                                      HQR0
         Q = Q / P                                                       HQR0
         R = R / P                                                       HQR0
C        ********** ROW MODIFICATION **********                         HQR0
         DO 210 J = K, EN                                                HQR0
            P = H(K,J) + Q * H(K+1,J)                                    HQR0
            IF (.NOT. NOTLAS) GO TO 200                                  HQR0
            P = P + R * H(K+2,J)                                         HQR0
            H(K+2,J) = H(K+2,J) - P * ZZ                                 HQR0
```

```
      200        H(K+1,J) = H(K+1,J) - P * Y                                    HQR
                 H(K,J) = H(K,J) - P * X                                        HQR
      210     CONTINUE                                                          HQR0
C     ------------------------------------------------------------------------  HQR
            J = MINO(EN,K+3)                                                    HQR
C     ********** COLUMN MODIFICATION **********                                 HQR0
            DO 230 I = L, J                                                     HQR
               P = X * H(I,K) + Y * H(I,K+1)                                    HQR
               IF (.NOT. NOTLAS) GO TO 220                                      HQR0
               P = P + ZZ * H(I,K+2)                                            HQR0
               H(I,K+2) = H(I,K+2) - P * R                                      HQR
      220      H(I,K+1) = H(I,K+1) - P * Q                                      HQR
               H(I,K) = H(I,K) - P                                              HQR0
      230   CONTINUE                                                            HQR
C     ------------------------------------------------------------------------  HQR
      260 CONTINUE                                                              HQR0
C     ------------------------------------------------------------------------  HQR0
            GO TO 70                                                            HQR
C     ********** ONE ROOT FOUND **********                                     HQR0
      270   WR(EN) = X + T                                                      HQR0
            WI(EN) = 0.0D0                                                      HQR
            EN = NA                                                             HQR
            GO TO 60                                                            HQR0
C     ********** TWO ROOTS FOUND **********                                    HQR
      280   P = (Y - X) / 2.0D0                                                 HQR
            Q = P * P + W                                                       HQR0
            ZZ = DSQRT(DABS(Q))                                                 HQR0
            X = X + T                                                           HQR
            IF (Q .LT. 0.0D0) GO TO 320                                         HQR
C     ********** REAL PAIR **********                                          QQR0
            ZZ = P + DSIGN(ZZ,P)                                                HQR
            WR(NA) = X + ZZ                                                     HQR
            WR(EN) = WR(NA)                                                     HQR0
            IF (ZZ .NE. 0.0D0) WR(EN) = X - W / ZZ                              HQR
            WI(NA) = 0.0D0                                                      HQR
            WI(EN) = 0.0D0                                                      QQR0
            GO TO 330                                                           HQR0
C     ********** COMPLEX PAIR **********                                       HQR
      320   WR(NA) = X + P                                                      HQR
            WR(EN) = X + P                                                      HQR0
            WI(NA) = ZZ                                                         QQR0
            WI(EN) = -ZZ                                                        HQR
      330   EN = ENM2                                                           HQR0
            GO TO 60                                                            HQR0
C     ********** SET ERROR -- NO CONVERGENCE TO AN                             HQR
C                EIGENVALUE AFTER 30 ITERATIONS **********                    HQR0
 1000 IERR = EN                                                                 HQR0
 1001 RETURN                                                                    HQR
C     ********** LAST CARD OF HQR **********                                   HQR
      END                                                                       HQR0
```

```
      SUBROUTINE INVIT(NM,N,A,WR,WI,SELECT,MM,M,Z,IERR,RM1,RV1,RV2)      INVO
      IMPLICIT REAL*8 (A-H,O-Z)                                         INVO
      REAL*8 NORM,NORMV,ILAMBD,MACHEP                                   INVO
      INTEGER I,J,K,L,M,N,S,II,IP,MM,MP,NM,NS,N1,UK,IP1,ITS,KM1,IERR    RNVO
      DIMENSION A(NM,N),WR(N),WI(N),Z(NM,MM),RM1(N,N),RV1(N),RV2(N)     INVO
C     REAL T,W,X,Y,EPS3,NORM,NORMV,GROWTO,ILAMBD,MACHEP,RLAMBD,UKROOT   INVO
      REAL*8 DSQRT,CDABS,DABS,DFLOAT                                    INVO
      INTEGER IABS                                                      RNVO
      LOGICAL*1 SELECT(N)                                               INVO
      COMPLEX*16 Z3,DCMPLX                                              INVO
      REAL*8 DREAL,DIMAG                                                INVO
C                                                                       INVO
      MACHEP = 16.**(-13)                                               INVO
C                                                                       INVO
      IERR = 0                                                          INVO
      UK = 0                                                            RNVO
      S = 1                                                             INVO
C     ********** IP = 0, REAL EIGENVALUE                                INVO
C                     1, FIRST OF CONJUGATE COMPLEX PAIR                INVO
C                    -1, SECOND OF CONJUGATE COMPLEX PAIR **********    INVO
      IP = 0                                                            INVO
      N1 = N - 1                                                        RNVO
C                                                                       INVO
      DO 980 K = 1, N                                                   INVO
         IF (WI(K) .EQ. 0.0D0 .OR. IP .LT. 0) GO TO 100                 INVO
         IP = 1                                                         INVO
         IF (SELECT(K) .AND. SELECT(K+1)) SELECT(K+1) = .FALSE.         INVO
  100    IF (.NOT. SELECT(K)) GO TO 960                                 INVO
         IF (WI(K) .NE. 0.0D0) S = S + 1                                INVO
         IF (S .GT. MM) GO TO 1000                                      INVO
         IF (UK .GE. K) GO TO 200                                       INVO
C        ********** CHECK FOR POSSIBLE SPLITTING **********           INVO
         DO 120 UK = K, N                                               INVO
            IF (UK .EQ. N) GO TO 140                                    INVO
            IF (A(UK+1,UK) .EQ. 0.0D0) GO TO 140                        INVO
  120    CONTINUE                                                       INVO
C        ********** COMPUTE INFINITY NORM OF LEADING UK BY UK          INVO
C                   (HESSENBERG) MATRIX **********                     INVO
  140    NORM = 0.0D0                                                   INVO
         MP = 1                                                         INVO
C                                                                       INVO
         DO 180 I = 1, UK                                               INVO
            X = 0.0D0                                                   INVO
C                                                                       INVO
            DO 160 J = MP, UK                                           INVO
  160       X = X + DABS(A(I,J))                                        INVO
C                                                                       INVO
            IF (X .GT. NORM) NORM = X                                   INVO
            MP = I                                                      INVO
  180    CONTINUE                                                       INVO
C        ********** EPS3 REPLACES ZERO PIVOT IN DECOMPOSITION          INVO
C                   AND CLOSE ROOTS ARE MODIFIED BY EPS3-**********   &RNVO
         IF (NORM .EQ. 0.0D0) NORM = 1.0D0                              INVO
         EPS3 = MACHEP * NORM                                           INVO
C        ********** GROWTO IS THE CRITERION FOR THE GROWTH **********  INVO
         UKROOT = DSQRT(DFLOAT(UK))                                     INVO
         GROWTO = 1.0D-1 / UKROOT                                       INVO
  200    RLAMBD = WR(K)                                                 INVO
         ILAMBD = WI(K)                                                 INVO
         IF (K .EQ. 1) GO TO 280                                        INVO
         KM1 = K - 1                                                    INVO
         GO TO 240                                                      INVO
C        ********** PERTURB EIGENVALUE IF IT IS CLOSE                  INVO
```

```
C                  TO ANY PREVIOUS EIGENVALUE **********                RNVC
  220    RLAMBD = RLAMBD + EPS3                                           INVC
C     ********** FOR I=K-1 STEP -1 UNTIL 1 DO -- **********              INVC
  240    DO 260 II = 1, KM1                                               INVC
            I = K - II                                                    INVC
            IF (SELECT(I) .AND. DABS(WR(I)-RLAMBD) .LT. EPS3 .AND.       INVC
     X         DABS(WI(I)-ILAMBD) .LT. EPS3) GO TO 220                   INVC
  260    CONTINUE                                                         INVC
C                                                                         INVC
         WR(K) = RLAMBD                                                   INVC
C     ********** PERTURB CONJUGATE EIGENVALUE TO MATCH **********        INVC
         IP1 = K + IP                                                     INVC
         WR(IP1) = RLAMBD                                                 INVC
C     ********** FORM UPPER HESSENBERG A-RLAMBD*I (TRANSPOSED)            RNVC
C                  AND INITIAL REAL VECTOR **********                    INVC
  280    MP = 1                                                           INVC
C                                                                         INVC
         DO 320 I = 1, UK                                                 INVC
C                                                                         RNVC
            DO 300 J = MP, UK                                             INVC
  300       RM1(J,I) = A(I,J)                                             INVC
C                                                                         INVC
            RM1(I,I) = RM1(I,I) - RLAMBD                                  INVC
            MP = I                                                        INVC
            RV1(I) = EPS3                                                 RNVC
  320    CONTINUE                                                         INVC
C                                                                         INVC
         ITS = 0                                                          INVC
         IF (ILAMBD .NE. 0.0D0) GO TO 520                                 INVC
C     ********** REAL EIGENVALUE,                                        INVC
C                  TRIANGULAR DECOMPOSITION WITH INTERCHANGES,           INVC
C                  REPLACING ZERO PIVOTS BY EPS3 **********              INVC
         IF (UK .EQ. 1) GO TO 420                                         INVC
C                                                                         INVC
         DO 400 I = 2, UK                                                 INVC
            MP = I - 1                                                    INVC
            IF (DABS(RM1(MP,I)) .LE. DABS(RM1(MP,MP))) GO TO 360         INVC
C                                                                         INVC
            DO 340 J = MP, UK                                             INVC
               Y = RM1(J,I)                                               INVC
               RM1(J,I) = RM1(J,MP)                                       INVC
               RM1(J,MP) = Y                                              INVC
  340       CONTINUE                                                      INVC
C                                                                         INVC
  360       IF (RM1(MP,MP) .EQ. 0.0D0) RM1(MP,MP) = EPS3                 INVC
            X = RM1(MP,I) / RM1(MP,MP)                                    INVC
            IF (X .EQ. 0.0D0) GO TO 400                                   INVC
C                                                                         INVC
            DO 380 J = I, UK                                              INVC
  380       RM1(J,I) = RM1(J,I) - X * RM1(J,MP)                           INVC
C                                                                         INVC
  400    CONTINUE                                                         INVC
C                                                                         INVC
  420    IF (RM1(UK,UK) .EQ. 0.0D0) RM1(UK,UK) = EPS3                     INVC
C     ********** BACK SUBSTITUTION FOR REAL VECTOR                       INVC
C                  FOR I=UK STEP -1 UNTIL 1 DO -- **********             RNVC
  440    DO 500 II = 1, UK                                                INVC
            I = UK + 1 - II                                               INVC
            Y = RV1(I)                                                    INVC
            IF (I .EQ. UK) GO TO 480                                      INVC
            IP1 = I + 1                                                   INVC
C                                                                         INVC
            DO 460 J = IP1, UK                                            INVC
```

```
126    460        Y = Y - RM1(J,I) * RV1(J)                                    INVO
127 C                                                                          INV01
128    480        RV1(I) = Y / RM1(I,I)                                        INVO
129    500     CONTINUE                                                        INVO
130 C                                                                          INV01
131            GO TO 740                                                       INV01
132 C     ********** COMPLEX EIGENVALUE,                                       INVO
133 C                 TRIANGULAR DECOMPOSITION WITH INTERCHANGES,             RNVO
134 C                 REPLACING ZERO PIVOTS BY EPS3.  STORE IMAGINARY         INV01
135 C                 PARTS IN UPPER TRIANGLE STARTING AT (1,3) **********    INVO
136    520     NS = N - S                                                      RNVO
137            Z(1,S-1) = -ILAMBD                                              INV01
138            Z(1,S) = 0.0D0                                                  INV01
139            IF (N .EQ. 2) GO TO 550                                         INVO
140            RM1(1,3) = -ILAMBD                                              INV01
141            Z(1,S-1) = 0.0D0                                                INV01
142            IF (N .EQ. 3) GO TO 550                                         INVO
143 C                                                                          INVO
144            DO 540 I = 4, N                                                 INV01
145    540     RM1(1,I) = 0.0D0                                                INVO
146 C                                                                          INVO
147    550     DO 640 I = 2, UK                                                INV01
148               MP = I - 1                                                   INV01
149               W = RM1(MP,I)                                                INVO
150               IF (I .LT. N) T = RM1(MP,I+1)                                INVO
151               IF (I .EQ. N) T = Z(MP,S-1)                                  INV01
152               X = RM1(MP,MP) * RM1(MP,MP) + T * T                          INVO
153               IF (W * W .LE. X) GO TO 580                                  INVO
154               X = RM1(MP,MP) / W                                           INV01
155               Y = T / W                                                    INV01
156               RM1(MP,MP) = W                                               INVO
157               IF (I .LT. N) RM1(MP,I+1) = 0.0D0                            INV01
158               IF (I .EQ. N) Z(MP,S-1) = 0.0D0                              INV01
159 C                                                                          INVO
160               DO 560 J = I, UK                                             INVO
161                  W = RM1(J,I)                                              INV01
162                  RM1(J,I) = RM1(J,MP) - X * W                              INV01
163                  RM1(J,MP) = W                                             INVO
164                  IF (J .LT. N1) GO TO 555                                  INV01
165                  L = J - NS                                                INV01
166                  Z(I,L) = Z(MP,L) - Y * W                                  INVO
167                  Z(MP,L) = 0.0D0                                           INVO
168                  GO TO 560                                                 INV01
169    555           RM1(I,J+2) = RM1(MP,J+2) - Y * W                          INVO
170                  RM1(MP,J+2) = 0.0D0                                       INVO
171    560        CONTINUE                                                     INV01
172 C                                                                          INV01
173               RM1(I,I) = RM1(I,I) - Y * ILAMBD                             INVO
174               IF (I .LT. N1) GO TO 570                                     INV01
175               L = I - NS                                                   INV01
176               Z(MP,L) = -ILAMBD                                            INVO
177               Z(I,L) = Z(I,L) + X * ILAMBD                                 INVO
178               GO TO 640                                                    INV01
179    570        RM1(MP,I+2) = -ILAMBD                                        RNVO
180               RM1(I,I+2) = RM1(I,I+2) + X * ILAMBD                         INVO
181               GO TO 640                                                    INV01
182    580        IF (X .NE. 0.0D0) GO TO 600                                  INV01
183               RM1(MP,MP) = EPS3                                            INVO
184               IF (I .LT. N) RM1(MP,I+1) = 0.0D0                            INVO
185               IF (I .EQ. N) Z(MP,S-1) = 0.0D0                              INV01
186               T = 0.0D0                                                    INVO
187               X = EPS3 * EPS3                                              INVO
188    600        W = W / X                                                    INV01
```

```
                X = RM1(MP,MP) * W                                              INV01
                Y = -T * W                                                      INV01
C                                                                               INV01
                DO 620 J = I, UK                                                INV01
                   IF (J .LT. N1) GO TO 610                                     INV01
                   L = J - NS                                                   INV01
                   T = Z(MP,L)                                                  INV01
                   Z(I,L) = -X * T - Y * RM1(J,MP)                              INV01
                   GO TO 615                                                    INV01
  610           T = RM1(MP,J+2)                                                 INV01
                RM1(I,J+2) = -X * T - Y * RM1(J,MP)                             INV02
  615           RM1(J,I) = RM1(J,I) - X * RM1(J,MP) + Y * T                     INV02
  620       CONTINUE                                                            INV02
C                                                                               INV02
                IF (I .LT. N1) GO TO 630                                        INV02
                L = I - NS                                                      INV02
                Z(I,L) = Z(I,L) - ILAMBD                                        INV02
                GO TO 640                                                       INV02
  630           RM1(I,I+2) = RM1(I,I+2) - ILAMBD                                INV02
  640       CONTINUE                                                            INV02
C                                                                               INV02
            IF (UK .LT. N1) GO TO 650                                           INV02
            L = UK - NS                                                         INV02
            T = Z(UK,L)                                                         INV02
            GO TO 655                                                           INV02
  650       T = RM1(UK,UK+2)                                                    INV02
  655       IF (RM1(UK,UK) .EQ. 0.0D0 .AND. T .EQ. 0.0D0) RM1(UK,UK) = EPS3   INV02
C     ********** BACK SUBSTITUTION FOR COMPLEX VECTOR                           INV02
C                    FOR I=UK STEP -1 UNTIL 1 DO -- **********                  INV02
  660       DO 720 II = 1, UK                                                   INV02
                I = UK + 1 - II                                                 INV02
                X = RV1(I)                                                      INV02
                Y = 0.0D0                                                       RNV02
                IF (I .EQ. UK) GO TO 700                                        INV02
                IP1 = I + 1                                                     INV02
C                                                                               INV02
                DO 680 J = IP1, UK                                              INV02
                   IF (J .LT. N1) GO TO 670                                     INV02
                   L = J - NS                                                   INV02
                   T = Z(I,L)                                                   INV02
                   GO TO 675                                                    INV02
  670              T = RM1(I,J+2)                                              INV02
  675              X = X - RM1(J,I) * RV1(J) + T * RV2(J)                      INV02
                   Y = Y - RM1(J,I) * RV2(J) - T * RV1(J)                      RNV02
  680           CONTINUE                                                        INV02
C                                                                               INV02
  700           IF (I .LT. N1) GO TO 710                                      INV02
                L = I - NS                                                      INV02
                T = Z(I,L)                                                      INV02
                GO TO 715                                                       INV02
  710           T = RM1(I,I+2)                                                INV02
  715           Z3 = DCMPLX(X,Y) / DCMPLX(RM1(I,I),T)                         INV02
                RV1(I) = DREAL(Z3)                                              INV02
                RV2(I) = DIMAG(Z3)                                              INV02
  720       CONTINUE                                                            INV02
C     ********** ACCEPTANCE TEST FOR REAL OR COMPLEX                            INV02
C                    EIGENVECTOR AND NORMALIZATION **********                   INV02
  740       ITS = ITS + 1                                                       INV02
            NORM = 0.0D0                                                        INV02
            NORMV = 0.0D0                                                       INV02
C                                                                               INV02
            DO 780 I = 1, UK                                                    INV02
               IF (ILAMBD .EQ. 0.0D0) X = DABS(RV1(I))                          INV02
```

```
252              IF (ILAMBD .NE. 0.0D0) X = CDABS(DCMPLX(RV1(I),RV2(I)))      INV0
253              IF (NORMV .GE. X) GO TO 760                                  INV02
254              NORMV = X                                                    INV02
255              J = I                                                        INV0
256     760      NORM = NORM + X                                              INV02
257     780   CONTINUE                                                        INV02
258 C                                                                         INV0
259           IF (NORM .LT. GROWTO) GO TO 840                                INV0
260 C     ********** ACCEPT VECTOR **********                                INV02
261           X = RV1(J)                                                     INV02
262           IF (ILAMBD .EQ. 0.0D0) X = 1.0D0 / X                           INV0
263           IF (ILAMBD .NE. 0.0D0) Y = RV2(J)                              RNV02
264 C                                                                        INV02
265           DO 820 I = 1, UK                                               I0
266              IF (ILAMBD .NE. 0.0D0) GO TO 800                            INV0
267              Z(I,S) = RV1(I) * X                                         INV02
268              GO TO 820                                                   INV0
269     800      Z3 = DCMPLX(RV1(I),RV2(I)) / DCMPLX(X,Y)                    INV0
270              Z(I,S-1) = DREAL(Z3)                                        RNV02
271              Z(I,S) = DIMAG(Z3)                                          INV02
272     820   CONTINUE                                                       INV0
273 C                                                                        INV02
274           IF (UK .EQ. N) GO TO 940                                       INV02
275           J = UK + 1                                                     INV0
276           GO TO 900                                                      INV0
277 C     ********** IN-LINE PROCEDURE FOR CHOOSING                          INV02
278 C                A NEW STARTING VECTOR **********                      INV0
279     840   IF (ITS .GE. UK) GO TO 880                                     INV0
280           X = UKROOT                                                     INV02
281           Y = EPS3 / (X + 1.0D0)                                         RNV02
282           RV1(1) = EPS3                                                  INV0
283 C                                                                        INV0
284           DO 860 I = 2, UK                                               INV02
285     860   RV1(I) = Y                                                     INV0
286 C                                                                        INV0
287           J = UK - ITS + 1                                               INV02
288           RV1 J) = RV1(J) - EPS3 * X                                     INV02
289           IF (ILAMBD .EQ. 0.0D0) GO TO 440                               INV0
290           GO TO 660                                                      INV02
291 C     ********** SET ERROR -- UNACCEPTED EIGENVECTOR **********          INV02
292     880   J = 1                                                          INV0
293           IERR = -K                                                      INV0
294 C     ********** SET REMAINING VECTOR COMPONENTS TO ZERO **********      INV02
295     900   DO 920 I = J, N                                                INV0
296              Z(I,S) = 0.0D0                                              INV0
297              IF (ILAMBD .NE. 0.0D0) Z(I,S-1) = 0.0D0                     INV02
298     920   CONTINUE                                                       INV02
299 C                                                                        INV0
300     940   S = S + 1                                                      INV0
301     960   IF (IP .EQ. (-1)) IP = 0                                       INV03
302           IF (IP .EQ. 1) IP = -1                                         INV0
303     980 CONTINUE                                                         INV0
304 C                                                                        INV03
305         GO TO 1001                                                       INV0
306 C     ********** SET ERROR -- UNDERESTIMATE OF EIGENVECTOR               INV0
307 C                SPACE REQUIRED **********                             INV05
308    1000 IF (IERR .NE. 0) IERR = IERR - N                                 INV03
309         IF (IERR .EQ. 0) IERR = -(2 * N + 1)                             INV0
310    1001 M = S - 1 - IABS(IP)                                             INV0
311         RETURN                                                           INV03
312 C     ********** LAST CARD OF INVIT **********                           INV0
313         END                                                              INV0
```

```
      SUBROUTINE ELMBAK(NM,LOW,IGH,A,INT,M,Z)                          ELM00
      IMPLICIT REAL*8 (A-H,O-Z)                                        ELM000
      INTEGER I,J,M,LA,MM,MP,NM,IGH,KP1,LOW,MP1                        ELM000
      DIMENSION A(NM,IGH),Z(NM,M)                                      ELM00
C     REAL X                                                           ELM000
      INTEGER INT(IGH)                                                 ELM000
C                                                                      ELM00
C                                                                      ELM00
      IF (M .EQ. 0) GO TO 200                                          ELM000
      LA = IGH - 1                                                     ELM00
      KP1 = LOW + 1                                                    ELM00
      IF (LA .LT. KP1) GO TO 200                                       ELM001
C     ********** FOR MP=IGH-1 STEP -1 UNTIL LOW+1 DO -- **********     ELM001
      DO 140 MM = KP1, LA                                              ELM00
         MP = LOW + IGH - MM                                           ELM001
         MP1 = MP + 1                                                  ELM001
C                                                                      ELM00
         DO 110 I = MP1, IGH                                           ELM00
            X = A(I,MP-1)                                              ELM001
            IF (X .EQ. 0.0) GO TO 110                                  ELM00
C                                                                      ELM00
            DO 100 J = 1, M                                            ELM002
  100       Z(I,J) = Z(I,J) + X * Z(MP,J)                              ELM002
C                                                                      ELM00
  110    CONTINUE                                                      ELM00
C                                                                      ELM002
         I = INT(MP)                                                   ELM00
         IF (I .EQ. MP) GO TO 140                                      ELM00
C                                                                      ELM002
         DO 130 J = 1, M                                               ELM003
            X = Z(I,J)                                                 ELM00
            Z(I,J) = Z(MP,J)                                           ELM00
            Z(MP,J) = X                                                ELM003
  130    CONTINUE                                                      ELM00
C                                                                      ELM00
  140 CONTINUE                                                         ELM003
C                                                                      ELM00
  200 RETURN                                                           ELM00
C     ********** LAST CARD OF ELMBAK **********                       ELM00
      END                                                              ELM004
```

```
  0      SUBROUTINE BALBAK(NM,N,LOW,IGH,SCALE,M,Z)                        BAL0
  1      IMPLICIT REAL*8 (A-H,O-Z)                                        BAL00
  2      INTEGER I,J,K,M,N,II,NM,IGH,LOW                                  BAL0
  3      DIMENSION SCALE(N),Z(NM,M)                                       BAL0
  4 C    REAL S                                                           BAL00
  5      IF (M .EQ. 0) GO TO 200                                          BAL0
  6      IF (IGH .EQ. LOW) GO TO 120                                      BAL0
  7 C                                                                     BAL00
  8      DO 110 I = LOW, IGH                                              BAL00
  9         S = SCALE(I)                                                  BAL0
 10 C    ********** LEFT HAND EIGENVECTORS ARE BACK TRANSFORMED           BAL0
 11 C                IF THE FOREGOING STATEMENT IS REPLACED BY           BAL00
 12 C                S=1.0/SCALE(I). **********                          BAL0
 13         DO 100 J = 1, M                                              BAL0
 14   100   Z(I,J) = Z(I,J) * S                                          BAL00
 15 C                                                                    BAL00
 16   110 CONTINUE                                                       BAL0
 17 C    ********** FOR I=LOW-1 STEP -1 UNTIL 1,                         BAL0
 18 C                IGH+1 STEP 1 UNTIL N DO -- **********              BAL00
 19   120 DO 140 II = 1, N                                               BJL0
 20         I = II                                                       BAL0
 21         IF (I .GE. LOW .AND. I .LE. IGH) GO TO 140                   BAL00
 22         IF (I .LT. LOW) I = LOW - II                                 BAL0
 23         K = SCALE(I)                                                 BAL0
 24         IF (K .EQ. I) GO TO 140                                      BAL00
 25 C                                                                    BAL00
 26         DO 130 J = 1, M                                              BAL0
 27            S = Z(I,J)                                                BAL0
 28            Z(I,J) = Z(K,J)                                           BAL00
 29         .  Z(K,J) = S                                                BAL0
 30   130   CONTINUE                                                     BAL0
 31 C                                                                    BAL00
 32   140 CONTINUE                                                       BAL0
 33 C                                                                    BAL0
 34   200 RETURN                                                         BAL0
 35 C    ********** LAST CARD OF BALBAK **********                       BAL00
 36      END                                                            BAL0
```

```
      SUBROUTINE DETFAC(NMAX,N,A,IPIVOT,IDET,DETERM,ISCALE,WK,IERR)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(NMAX,1),IPIVOT(1),WK(1)
C
C
      ISCALE=0
      NM1=N-1
      IERR=0
C
C     DETERMINANT CALCULATION TEST
C
      IF(IDET.EQ.1)GO TO 230
C
C     TEST FOR A SCALAR MATRIX
C
      IF(NM1.GT.0)GO TO 20
      DETERM=A(1,1)
      RETURN
C
C     COMPUTE SCALING FACTORS
C
   20 CONTINUE
      DO 60 I=1,N
      P=0.0
      DO 30 J=1,N
      Q=DMAX1(P,DABS(A(I,J)))
      IF(Q.GT.P)P=Q
   30 CONTINUE
      IF(P)60,40,60
   40 DETERM=0.0
      IERR=1
      RETURN
   60 WK(I)=P
C
      DO 210 M=1,NM1
C
C     PIVOTAL LOGIC SETUP
C
      P=0.0
      DO 110 I=M,N
      Q=DABS(A(I,M)/WK(I))
      IF(Q-P)110,110,100
  100 P=Q
      IP=I
  110 CONTINUE
C
      IPIVOT(M)=IP
C
      IF(P.EQ.0.)GO TO 40
      IF(M.EQ.IP)GO TO 155
C
C     PIVOT THE M-TH ROW OF THE A MATRIX
C
C
      DO 150 I=1,N
      P=A(IP,I)
      A(IP,I)=A(M,I)
  150 A(M,I)=P
C
      P=WK(IP)
      WK(IP)=WK(M)
      WK(M)=P
C
  155 MP1=M+1
```

```
C                                                                       DET0
C      L/U FACTORIZATION LOGIC                                          DET0
C                                                                       DET00
      P=A(M,M)                                                          DET0
      DO 180 I=MP1,N                                                    DET0
      A(I,M)=A(I,M)/P                                                   DET00
      Q=A(I,M)                                                          DET00
      DO 180 K=MP1,N                                                    DET0
  180 A(I,K)=A(I,K)-Q*A(M,K)                                            DET0
C                                                                       DET00
  210 CONTINUE                                                          DET0
C                                                                       DET0
      IPIVOT(N)=N                                                       DET00
      IF (A(N,N) .EQ. 0.0) GO TO 40                                     DET0
C                                                                       DET0
C     CALCULATION OF THE DETERMINANT OF A                               DET00
C                                                                       DET00
      IF(IDET.EQ.0)RETURN                                               DET0
C                                                                       DET0
  230 SIGN=1.0                                                          DET00
      DETERM=1.0                                                        DET0
C                                                                       DET0
C     ADJUST SIGN OF DETERMINANT DUE TO PIVOTAL STRATEGY                DET00
C                                                                       DET00
      DO 250 I=1,NM1                                                    DET0
      IF(I-IPIVOT(I))240,250,240                                        DET0
  240 SIGN=-SIGN                                                        DET00
  250 CONTINUE                                                          DET0
C                                                                       DET0
      DO 340 I=1,N                                                      DET00
      P=A(I,I)                                                          DET0
C                                                                       DET0
  260 CONTINUE                                                          DET00
      IF(R1.GT.DABS(P))GO TO 280                                        DET00
      P=P*R2                                                            DET0
      ISCALE=ISCALE+1                                                   DET0
      GO TO 260                                                         DET0
C                                                                       DET0
  280 CONTINUE                                                          DET0
      IF(R2.LT.DABS(P))GO TO 290                                        DET0
      P=P*R1                                                            DET0
      ISCALE=ISCALE-1                                                   DET0
      GO TO 280                                                         DET0
C                                                                       DET0
  290 DETERM=DETERM*P                                                   DET0
C                                                                       DET0
  300 CONTINUE                                                          DET0
      IF(R1.GT.DABS(DETERM))GO TO 320                                   DET0
      DETERM=DETERM*R2                                                  DET0
      ISCALE=ISCALE+1                                                   DET0
      GO TO 300                                                         DET0
C                                                                       DET0
  320 CONTINUE                                                          DET0
      IF(R2.LT.DABS(DETERM))GO TO 340                                   DET0
      DETERM=DETERM*R1                                                  DET0
      ISCALE=ISCALE-1                                                   DET0
      GO TO 320                                                         DET0
C                                                                       DET0
  340 CONTINUE                                                          DET0
C                                                                       DET0
      DETERM=DETERM*SIGN                                                DET0
C                                                                       DET0
      RETURN                                                            DET0
```

```
      SUBROUTINE AXPXB(A,U,M,NA,NU,B,V,N,NB,NV,C,NC,EPSA,
     1EPSB,FAIL)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION
     1A(NA,1),U(NU,1),B(NB,1),V(NV,1),C(NC,1)
      INTEGER
C
     1 FAIL
      M1 = M+1
      MM1 = M-1
      N1 = N+1
      NM1 = N-1
C IF REQUIRED, REDUCE A TO UPPER REAL SCHUR FORM.
C
      IF(EPSA .LT. 0.) GO TO 35
      DO 10 I=1,M
        DO 10 J=I,M
          TEMP = A(I,J)
          A(I,J) = A(J,I)
          A(J,I) = TEMP
   10 CONTINUE
      CALL HSHLDR(A,M,NA)
      CALL BCKMLT(A,U,M,NA,NU)
      IF(MM1 .EQ. 0) GO TO 25
      DO 20 I=1,MM1
        A(I+1,I) = A(I,M1)
   20 CONTINUE
      CALL SCHUR(A,U,M,NA,NU,EPSA,FAIL)
      IF(FAIL .NE. 0) RETURN
   25 DO 30 I=1,M
        DO 30 J=I,M
          TEMP = A(I,J)
          A(I,J) = A(J,I)
          A(J,I) = TEMP
   30 CONTINUE
C
C IF REQUIRED, REDUCE B TO UPPER REAL SCHUR FORM.
C
   35 IF(EPSB .LT. 0.) GO TO 45
      CALL HSHLDR(B,N,NB)
      CALL BCKMLT(B,V,N,NB,NV)
      IF(NM1 .EQ. 0) GO TO 45
      DO 40 I=1,NM1
        B(I+1,I) = B(I,N1)
   40 CONTINUE
      CALL SCHUR(B,V,N,NB,NV,EPSB,FAIL)
      FAIL = -FAIL
      IF(FAIL .NE. 0) RETURN
C
C TRANSFORM C.
C
   45 DO 60 J=1,N
        DO 50 I=1,M
          A(I,M1) = 0.
          DO 50 K=1,M
            A(I,M1) = A(I,M1) + U(K,I)*C(K,J)
   50 CONTINUE
      DO 60 I=1,M
        C(I,J) = A(I,M1)
   60 CONTINUE
      DO 80 I=1,M
        DO 70 J=1,N
          B(N1,J) = 0.
```

```
      DO 70 K=1,N                                                       AXPC
        B(N1,J) = B(N1,J) + C(I,K)*V(K,J)                               AXPC
   70 CONTINUE                                                          AXPC
      DO 80 J=1,N                                                       AXPC
        C(I,J) = B(N1,J)                                                AXPC
   80 CONTINUE                                                          AXPC
C                                                                       AXPC
C SOLVE THE TRANSFORMED SYSTEM.                                         AXPC
C                                                                       AXPC
      CALL SHRSLV(A,B,C,M,N,NA,NB,NC)                                   AXPC
C                                                                       AXPC
C TRANSFORM C BACK TO THE SOLUTION.                                     AXPC
C                                                                       AXPC
      DO 100 J=1,N                                                      AXPC
        DO 90 I=1,M                                                     AXPC
          A(I,M1) = 0.                                                  AXPC
          DO 90 K=1,M                                                   AXPC
            A(I,M1) = A(I,M1) + U(I,K)*C(K,J)                           AXPC
   90 CONTINUE                                                          AXPC
      DO 100 I=1,M                                                      AXPC
        C(I,J) = A(I,M1)                                                AXPC
  100 CONTINUE                                                          AXPC
      DO 120 I=1,M                                                      AXPC
        DO 110 J=1,N                                                    AXPC
          B(N1,J) = 0.                                                  AXPC
          DO 110 K=1,N                                                  AXPC
            B(N1,J) = B(N1,J) + C(I,K)*V(J,K)                           AXPC
  110   CONTINUE                                                        AXPC
        DO 120 J=1,N                                                    AXPC
          C(I,J) = B(N1,J)                                              AXPC
  120 CONTINUE                                                          AXPC
      RETURN                                                            AXPC
      END                                                               AXPC
```

```
      SUBROUTINE SHRSLV(A,B,C,M,N,NA,NB,NC)                          SHR0
      IMPLICIT REAL*8 (A-H,O-Z)                                      SHR0
      DIMENSION                                                      SHR0
     1A(NA,1),B(NB,1),C(NC,1)                                        SHR0
      INTEGER                                                        SHR0
     1 DK,DL                                                         SHR0
      COMMON/SLVBLK/T(5,5),P(5),NSYS                                 SHR0
      L = 1                                                          SHR0
   10   LM1 = L-1                                                   SHR0
        DL = 1                                                       SHR0
        IF(L .EQ. N) GO TO 15                                        SHR0
        IF(B(L+1,L) .NE. 0.) DL = 2                                  SHR0
   15   LL = L+DL-1                                                 SHR0
        IF(L .EQ. 1) GO TO 30                                        SHR0
        DO 20 J=L,LL                                                 SHR0
          DO 20 I=1,M                                                SHR0
            DO 20 IB=1,LM1                                           SHR0
              C(I,J) = C(I,J) - C(I,IB)*B(IB,J)                      SHR0
   20   CONTINUE                                                    SHR0
   30   K = 1                                                       SHR0
   40     KM1 = K-1                                                 SHR0
          DK = 1                                                    SHR0
          IF(K .EQ. M) GO TO 45                                     SHR0
          IF(A(K,K+1) .NE. 0.) DK = 2                               SHR0
   45     KK = K+DK-1                                               SHR0
          IF(K .EQ. 1) GO TO 60                                     SHR0
          DO 50 I=K,KK                                              SHR0
            DO 50 J=L,LL                                            SHR0
              DO 50 JA=1,KM1                                        SHR0
                C(I,J) = C(I,J) - A(I,JA)*C(JA,J)                   SHR0
   50     CONTINUE                                                  SHR0
   60     IF(DL .EQ. 2) GO TO 80                                    SHR0
          IF(DK .EQ. 2) GO TO 70                                    SHR0
          T(1,1) = A(K,K) + B(L,L)                                  SHR0
          IF(T(1,1) .EQ. 0.) STOP                                   SHR0
          C(K,L) = C(K,L)/T(1,1)                                    SHR0
          GO TO 100                                                 SHR0
   70     T(1,1) = A(K,K) + B(L,L)                                  SHR0
          T(1,2) = A(K,KK)                                          SHR0
          T(2,1) = A(KK,K)                                          SHR0
          T(2,2) = A(KK,KK) + B(L,L)                                SHR0
          P(1) = C(K,L)                                             SHR0
          P(2) = C(KK,L)                                            SHR0
          NSYS = 2                                                  SHR0
          CALL SYSSLV                                               SHR0
          C(K,L) = P(1)                                             SHR0
          C(KK,L) = P(2)                                            SHR0
          GO TO 100                                                 SHR0
   80     IF(DK .EQ. 2) GO TO 90                                    SHR0
          T(1,1) = A(K,K) + B(L,L)                                  SHR0
          T(1,2) = B(LL,L)                                          SHR0
          T(2,1) = B(L,LL)                                          SHR0
          T(2,2) = A(K,K) + B(LL,LL)                                SHR0
          P(1) = C(K,L)                                             SHR0
          P(2) = C(K,LL)                                            SHR0
          NSYS = 2                                                  SHR0
          CALL SYSSLV                                               SHR0
          C(K,L) = P(1)                                             SHR0
          C(K,LL) = P(2)                                            SHR0
          GO TO 100                                                 SHR0
   90     T(1,1) = A(K,K) + B(L,L)                                  SHR0
          T(1,2) = A(K,KK)                                          SHR0
          T(1,3) = B(LL,L)                                          SHR0
```

```
      T(1,4) = 0.                                                       SHR0(
      T(2,1) = A(KK,K)                                                  SHR0(
      T(2,2) = A(KK,KK) + B(L,L)                                        SHR0(
      T(2,3) = 0.                                                       SHR0(
      T(2,4) = T(1,3)                                                   SHR0(
      T(3,1) = B(L,LL)                                                  SHR0(
      T(3,2) = 0.                                                       SHR0(
      T(3,3) = A(K,K) + B(LL,LL)                                        SHR0(
      T(3,4) = T(1,2)                                                   SHR0(
      T(4,1) = 0.                                                       SHR0(
      T(4,2) = T(3,1)                                                   SHR0(
      T(4,3) = T(2,1)                                                   SHR0(
      T(4,4) = A(KK,KK) + B(LL,LL)                                      SHR0(
      P(1) = C(K,L)                                                     SHR0(
      P(2) = C(KK,L)                                                    SHR0(
      P(3) = C(K,LL)                                                    SHR0(
      P(4) = C(KK,LL)                                                   SHR0(
      NSYS = 4                                                          SHR0(
      CALL SYSSLV                                                       SHR0(
      C(K,L) = P(1)                                                     SHR0(
      C(KK,L) = P(2)                                                    SHR0(
      C(K,LL) = P(3)                                                    SHR0(
      C(KK,LL) = P(4)                                                   SHR0(
  100 K = K + DK                                                        SHR0(
      IF(K .LE. M) GO TO 40                                             SHR0(
      L = L + DL                                                        SHR0(
      IF(L .LE. N) GO TO 10                                             SHR0(
      RETURN                                                            SHR0(
      END                                                               SHR0(
```

```
      SUBROUTINE ATXPXA(A,U,C,N,NA,NU,NC,EPS,FAIL)                      ATX0
      IMPLICIT REAL*8 (A-H,O-Z)                                         ATX0
      DIMENSION                                                         ATX0
     1A(NA,1),U(NU,1),C(NC,1)                                           ATX0
      INTEGER                                                           ATX0
     1 FAIL                                                             ATX0
      N1 = N+1                                                          ATX0
      NM1 = N-1                                                         ATX0
C                                                                       ATX0
C IF REQUIRED, REDUCE A TO LOWER REAL SCHUR FORM.                       ATX0
C                                                                       ATX0
      IF(EPS .LT. 0.) GO TO 15                                          ATX0
      CALL HSHLDR(A,N,NA)                                               ATX0
      CALL BCKMLT(A,U,N,NA,NU)                                          ATX0
      DO 10 I=1,NM1                                                     ATX0
        A(I+1,I) = A(I,N1)                                              ATX0
   10 CONTINUE                                                          ATX0
      CALL SCHUR(A,U,N,NA,NU,EPS,FAIL)                                  ATX0
      IF(FAIL .NE. 0) RETURN                                            ATX0
C                                                                       ATX0
C TRANSFORM C.                                                          ATX0
C                                                                       ATX0
   15 DO 20 I=1,N                                                       ATX0
        C(I,I)=C(I,I)/2.                                                ATX0
   20 CONTINUE                                                          ATX0
      DO 40 I=1,N                                                       ATX0
        DO 30 J=1,N                                                     ATX0
          A(N1,J) = 0.                                                  ATX0
          DO 30 K=I,N                                                   ATX0
            A(N1,J) = A(N1,J) + C(I,K)*U(K,J)                           ATX0
   30   CONTINUE                                                        ATX0
          DO 40 J=1,N                                                   ATX0
          C(I,J) = A(N1,J)                                              ATX0
   40 CONTINUE                                                          ATX0
      DO 60 J=1,N                                                       ATX0
        DO 50 I=1,N                                                     ATX0
          A(I,N1) = 0.                                                  ATX0
          DO 50 K=1,N                                                   ATX0
            A(I,N1) = A(I,N1) + U(K,I)*C(K,J)                           ATX0
   50   CONTINUE                                                        ATX0
        DO 60 I=1,N                                                     ATX0
          C(I,J) = A(I,N1)                                              ATX0
   60 CONTINUE                                                          ATX0
      DO 70 I=1,N                                                       ATX0
        DO 70 J=I,N                                                     ATX0
          C(I,J) = C(I,J) + C(J,I)                                      ATX0
          C(J,I) = C(I,J)                                               ATX0
   70 CONTINUE                                                          ATX0
C                                                                       ATX0
C SOLVE THE TRANSFORMED SYSTEM.                                         ATX0
C                                                                       ATX0
      CALL SYMSLV(A,C,N,NA,NC)                                          ATX0
C                                                                       ATX0
C TRANSFORM C BACK TO THE SOLUTION.                                     ATX0
C                                                                       ATX0
      DO 80 I=1,N                                                       ATX0
        C(I,I) = C(I,I)/2.                                              ATX0
   80 CONTINUE                                                          ATX0
      DO 100 I=1,N                                                      ATX0
        DO 90 J=1,N                                                     ATX0
          A(N1,J) = 0.                                                  ATX0
          DO 90 K=I,N                                                   ATX0
            A(N1,J) = A(N1,J) + C(I,K)*U(J,K)                           ATX0
```

```
 90   CONTINUE                                                          ATX0C
      DO 100 J=1,N                                                      ATX0C
        C(I,J) = A(N1,J)                                                ATX0C
100 CONTINUE                                                            ATX0C
      DO 120 J=1,N                                                      ATX0C
        DO 110 I=1,N                                                    ATX0C
          A(I,N1) = 0.                                                  ATX0C
          DO 110 K=1,N                                                  ATX0C
            A(I,N1) = A(I,N1) + U(I,K)*C(K,J)                           ATX0C
110   CONTINUE                                                          ATX0C
        DO 120 I=1,N                                                    ATX0C
          C(I,J) = A(I,N1)                                              ATX0C
120 CONTINUE                                                            ATX0C
      DO 130 I=1,N                                                      ATX0C
        DO 130 J=I,N                                                    ATX0C
          C(I,J) = C(I,J) + C(J,I)                                      ATX0C
          C(J,I) = C(I,J)                                               ATX0C
130 CONTINUE                                                            ATX0C
      RETURN                                                            ATX0C
      END                                                               ATX0C
```

```
      SUBROUTINE SYMSLV(A,C,N,NA,NC)                                    SYMC
      IMPLICIT REAL*8 (A-H,O-Z)                                         SYMC
      DIMENSION                                                         SYMC
     1A(NA,1),C(NC,1)                                                   SYMC
      INTEGER                                                           SYMC
     1 DK,DL                                                            SYMC
      COMMON/SLVBLK/T(5,5),P(5),NSYS                                    SYMC
      L = 1                                                             SYMC
   10   DL = 1                                                          SYMC
        IF(L .EQ. N) GO TO 20                                           SYMC
        IF(A(L+1,L) .NE. 0.) DL = 2                                     SYMC
   20   LL = L+DL-1                                                     SYMC
        K = L                                                           SYMC
   30     KM1 = K-1                                                     SYMC
          DK = 1                                                        SYMC
          IF(K .EQ. N) GO TO 35                                         SYMC
          IF(A(K+1,K) .NE. 0.) DK = 2                                   SYMC
   35     KK = K+DK-1                                                   SYMC
          IF(K .EQ. L) GO TO 45                                         SYMC
          DO 40 I=K,KK                                                  SYMC
            DO 40 J=L,LL                                                SYMC
              DO 40 IA=L,KM1                                            SYMC
                C(I,J) = C(I,J) - A(IA,I)*C(IA,J)                       SYMC
   40     CONTINUE                                                      SYMC
   45   IF(DL .EQ. 2) GO TO 60                                          SYMC
        IF(DK .EQ. 2 ) GO TO 50                                         SYMC
        T(1,1) = A(K,K) + A(L,L)                                        SYMC
        IF(T(1,1) .EQ. 0.) STOP                                         SYMC
        C(K,L) = C(K,L)/T(1,1)                                          SYMC
        GO TO 90                                                        SYMC
   50   T(1,1) = A(K,K) + A(L,L)                                        SYMC
        T(1,2) = A(KK,K)                                                SYMC
        T(2,1) = A(K,KK)                                                SYMC
        T(2,2) = A(KK,KK) + A(L,L)                                      SYMC
        P(1) = C(K,L)                                                   SYMC
        P(2) = C(KK,L)                                                  SYMC
        NSYS = 2                                                        SYMC
        CALL SYSSLV                                                     SYMC
        C(K,L) = P(1)                                                   SYMC
        C(KK,L) = P(2)                                                  SYMC
        GO TO 90                                                        SYMC
   60   IF(DK .EQ. 2) GO TO 70                                          SYMC
        T(1,1) = A(K,K) + A(L,L)                                        SYMC
        T(1,2) = A(LL,L)                                                SYMC
        T(2,1) = A(L,LL)                                                SYMC
        T(2,2) = A(K,K) + A(LL,LL)                                      SYMC
        P(1) = C(K,L)                                                   SYMC
        P(2) = C(K,LL)                                                  SYMC
        NSYS = 2                                                        SYMC
        CALL SYSSLV                                                     SYMC
        C(K,L) = P(1)                                                   SYMC
        C(K,LL) = P(2)                                                  SYMC
        GO TO 90                                                        SYMC
   70   IF(K .NE. L) GO TO 80                                           SYMC
        T(1,1) = A(L,L)                                                 SYMC
        T(1,2) = A(LL,L)                                                SYMC
        T(1,3) = 0.                                                     SYMC
        T(2,1) = A(L,LL)                                                SYMC
        T(2,2) = A(L,L) + A(LL,LL)                                      SYMC
        T(2,3) = T(1,2)                                                 SYMC
        T(3,1) = 0.                                                     SYMC
        T(3,2) = T(2,1)                                                 SYMC
        T(3,3) = A(LL,LL)                                               SYMC
```

```
      P(1) = C(L,L)/2.                                                  SYM0
      P(2) = C(LL,L)                                                    SYM0
      P(3) = C(LL,LL)/2.                                                SYM0
      NSYS = 3                                                          SYM0
      CALL SYSSLV                                                       SYM0
      C(L,L) = P(1)                                                     SYM0
      C(LL,L) = P(2)                                                    SYM0
      C(L,LL) = P(2)                                                    SYM0
      C(LL,LL) = P(3)                                                   SYM0
      GO TO 90                                                          SYM0
 80   T(1,1) = A(K,K) + A(L,L)                                          SYM0
      T(1,2) = A(KK,K)                                                  SYM0
      T(1,3) = A(LL,L)                                                  SYM0
      T(1,4) = 0.                                                       SYM0
      T(2,1) = A(K,KK)                                                  SYM0
      T(2,2) = A(KK,KK) + A(L,L)                                        SYM0
      T(2,3) = 0.                                                       SYM0
      T(2,4) = T(1,3)                                                   SYM0
      T(3,1) = A(L,LL)                                                  SYM0
      T(3,2) = 0.                                                       SYM0
      T(3,3) = A(K,K) + A(LL,LL)                                        SYM0
      T(3,4) = T(1,2)                                                   SYM0
      T(4,1) = 0.                                                       SYM0
      T(4,2) = T(3,1)                                                   SYM0
      T(4,3) = T(2,1)                                                   SYM0
      T(4,4) = A(KK,KK) + A(LL,LL)                                      SYM0
      P(1) = C(K,L)                                                     SYM0
      P(2) = C(KK,L)                                                    SYM0
      P(3) = C(K,LL)                                                    SYM0
      P(4) = C(KK,LL)                                                   SYM0
      NSYS = 4                                                          SYM0
      CALL SYSSLV                                                       SYM0
      C(K,L) = P(1)                                                     SYM0
      C(KK,L) = P(2)                                                    SYM0
      C(K,LL) = P(3)                                                    SYM0
      C(KK,LL) = P(4)                                                   SYM0
 90   K = K + DK                                                        SYM1
      IF(K .LE. N) GO TO 30                                             SYM1
      LDL = L + DL                                                      SYM1
      IF(LDL .GT. N) RETURN                                             SYM1
      DO 120 J=LDL,N                                                    SYM1
        DO 100 I=L,LL                                                   SYM1
          C(I,J) = C(J,I)                                               SYM1
100     CONTINUE                                                        SYM1
        DO 120 I=J,N                                                    SYM1
          DO 110 K=L,LL                                                 SYM1
            C(I,J) = C(I,J) - C(I,K)*A(K,J) - A(K,I)*C(K,J)             SYM1
110     CONTINUE                                                        SYM1
        C(J,I) = C(I,J)                                                 SYM1
120   CONTINUE                                                          SYM1
      L = LDL                                                           SYM1
      GO TO 10                                                          SYM1
      END                                                               SY'101
```

```
      SUBROUTINE HSHLDR(A,N,NA)                                         HSH0
      IMPLICIT REAL*8 (A-H,O-Z)                                         HSH0
      DIMENSION A(NA,1)                                                 HSH0
      REAL*8 MAX                                                        HSH0
C                                                                       HSH0
      NM2 = N-2                                                         HSH0
      N1 = N+1                                                          HSH0
      IF(N .EQ. 1) RETURN                                               HSH0
      IF(N .GT. 2) GO TO 5                                              HSH0
      A(1,N1) = A(2,1)                                                  HSH0
      RETURN                                                            HSH0
    5 DO 80 L=1,NM2                                                     HSH0
        L1 = L+1                                                        HSH0
        MAX = 0.                                                        HSH0
        DO 10 I=L1,N                                                    HSH0
          MAX = DMAX1(MAX,DABS(A(I,L)))                                 HSH0
   10   CONTINUE                                                        HSH0
        IF(MAX .NE. 0.) GO TO 20                                        HSH0
        A(L,N1) = 0.                                                    HSH0
        A(N1,L) = 0.                                                    HSH0
        GO TO 80                                                        HSH0
   20   SUM = 0.                                                        HSH0
        DO 30 I=L1,N                                                    HSH0
          A(I,L) = A(I,L)/MAX                                           HSH0
          SUM = SUM + A(I,L)**2                                         HSH0
   30   CONTINUE                                                        HSH0
        S = DSIGN(DSQRT(SUM),A(L1,L))                                   HSH0
        A(L,N1) = -MAX*S                                                HSH0
        A(L1,L) = S + A(L1,L)                                           HSH0
        A(N1,L) = S*A(L1,L)                                             HSH0
        DO 50 J=L1,N                                                    HSH0
          SUM = 0.                                                      HSH0
          DO 40 I=L1,N                                                  HSH0
            SUM = SUM + A(I,L)*A(I,J)                                   HSH0
   40     CONTINUE                                                      HSH0
          P = SUM/A(N1,L)                                               HSH0
          DO 50 I=L1,N                                                  HSH0
            A(I,J) = A(I,J) - A(I,L)*P                                  HSH0
   50   CONTINUE                                                        HSH0
        DO 70 I=1,N                                                     HSH0
          SUM = 0.                                                      HSH0
          DO 60 J=L1,N                                                  HSH0
            SUM = SUM + A(I,J)*A(J,L)                                   HSH0
   60     CONTINUE                                                      HSH0
          P = SUM/A(N1,L)                                               HSH0
          DO 70 J=L1,N                                                  HSH0
            A(I,J) = A(I,J) - P*A(J,L)                                  HSH0
   70   CONTINUE                                                        HSH0
   80 CONTINUE                                                          HSH0
      A(N-1,N1) = A(N,N-1)                                              HSH0
      RETURN                                                            HSH0
      END                                                               HSH0
```

```
      SUBROUTINE BCKMLT(A,U,N,NA,NU)                                    BCK0
      IMPLICIT REAL*8 (A-H,O-Z)                                         BCK0
      DIMENSION                                                         BCK0
     1A(NA,1),U(NU,1)                                                   BCK0
C                                                                       BCK0
      N1 = N+1                                                          BCK0
      NM1 = N-1                                                         BCK0
      NM2 = N-2                                                         BCK0
      U(N,N) = 1.                                                       BCK0
      IF(NM1 .EQ. 0) RETURN                                             BCK0
      U(NM1,N) = 0.                                                     BCK0
      U(N,NM1) = 0.                                                     BCK0
      U(NM1,NM1) = 1.                                                   BCK0
      IF(NM2 .EQ. 0) RETURN                                             BCK0
      DO 40 LL=1,NM2                                                    BCK0
        L = NM2-LL+1                                                    BCK0
        L1 = L+1                                                        BCK0
        IF(A(N1,L) .EQ. 0.) GO TO 25                                    BCK0
        DO 20 J=L1,N                                                    BCK0
          SUM = 0.                                                      BCK0
          DO 10 I=L1,N                                                  BCK0
            SUM = SUM + A(I,L)*U(I,J)                                   BCK0
   10     CONTINUE                                                      BCK0
          P = SUM/A(N1,L)                                               BCK0
          DO 20 I=L1,N                                                  BCK0
            U(I,J) = U(I,J) - A(I,L)*P                                  BCK0
   20   CONTINUE                                                       BCK0
   25   DO 30 I=L1,N                                                   BCK0
          U(I,L) = 0.                                                   BCK0
          U(L,I) = 0.                                                   BCK0
   30   CONTINUE                                                       BCK0
        U(L,L) = 1.                                                     BCK0
   40 CONTINUE                                                         BCK0
      RETURN                                                            BCK0
      END                                                               BCK0
```

```
      SUBROUTINE SCHUR(H,U,NN,NH,NU,EPS,FAIL)                          SC
      IMPLICIT REAL*8 (A-H,O-Z)                                        SC
       DIMENSION                                                       SC
     1H(NH,1),U(NU,1)                                                  SC
      INTEGER                                                          SC
     1 FAIL                                                            SC
      LOGICAL                                                          SC
     1LAST                                                             SC
      N = NN                                                           SC
      HN = 0.                                                          SC
      DO 20 I=1,N                                                      SC
        JL = MAX0(1,I-1)                                               SC
        RSUM = 0.                                                      SC
      DO 10 J=JL,N                                                     SC
          RSUM = RSUM + DABS(H(I,J))                                   SC
   10 CONTINUE                                                         SC
        HN = DMAX1(HN,RSUM)                                            SC
   20 CONTINUE                                                         SC
      TEST = EPS*HN                                                    SCH
      IF(HN .EQ. 0.) GO TO 230                                         SCH
   30 IF(N .LE. 1) GO TO 230                                           SCH
      ITS = 0                                                          SCH
      NA = N-1                                                         SCH
      NM2 = N-2                                                        SCH
   40 DO 50 LL=2,N                                                     SCH
      L = N-LL+2                                                       SCH
        IF(DABS(H(L,L-1)) .LE. TEST) GO TO 60                          SCH
   50 CONTINUE                                                         SCH
      L = 1                                                            SCH
      GO TO 70                                                         SCH
   60 H(L,L-1) = 0.                                                    SCH
   70 IF(L .LT. NA) GO TO 72                                           SCH
      N = L-1                                                          SCH
      GO TO 30                                                         SCH
   72 X = H(N,N)/HN                                                    SCH
      Y = H(NA,NA)/HN                                                  SCH
      R = (H(N,NA)/HN)*(H(NA,N)/HN)                                    SCH
      IF(ITS .LT. 30) GO TO 75                                         SCH
      FAIL = N                                                         SCH
      RETURN                                                           SCH
   75 IF(ITS.EQ.10 .OR. ITS.EQ.20) GO TO 80                            SCH
      S = X + Y                                                        SCH
      Y = X*Y - R                                                      SCH
      GO TO 90                                                         SCH
   80 Y = (DABS(H(N,NA)) + DABS(H(NA,NM2)))/HN                         SCH
      S = 1.5*Y                                                        SCH
      Y = Y**2                                                         SCH
   90 ITS = ITS + 1                                                    SCH
      DO 100 MM=L,NM2                                                  SCH
        M = NM2-MM+L                                                   SCH
        X = H(M,M)/HN                                                  SCH
        R = H(M+1,M)/HN                                                SCH
        Z = H(M+1,M+1)/HN                                              SCH
        P = X*(X-S) + Y + R*(H(M,M+1)/HN)                              SCH
        Q = R*(X+Z-S)                                                  SCH
        R = R*(H(M+2,M+1)/HN)                                          SCH
        W = DABS(P) + DABS(Q) + DABS(R)                                SCH
        P = P/W                                                        SCH
        Q = Q/W                                                        SCH
        R = R/W                                                        SCH
        IF(M .EQ. L) GO TO 110                                         SCH
      IF(DABS(H(M,M-1))*(DABS(Q)+DABS(R)) .LE. DABS(P)*TEST)           SCH
     1GO TO 110
```

```
  63   100 CONTINUE                                                          SCH0C
  64   110 M2 = M+2                                                          SCH0C
  65       M3 = M+3                                                          SCH0C
  66       DO 120 I=M2,N                                                     SCH0C
  67         H(I,I-2) = 0.                                                   SCH0C
  68   120 CONTINUE                                                          SCH0C
  69       IF(M3 .GT. N) GO TO 140                                           SCH0C
  70       DO 130 I=M3,N                                                     SCH0C
  71         H(I,I-3) = 0.                                                   SCH0C
  72   130 CONTINUE                                                          SCH0C
  73   140 DO 220 K=M,NA                                                     SCH0C
  74         LAST = K.EQ.NA                                                  SCH0C
  75         IF(K .EQ. M) GO TO 150                                          SCH0C
  76         P = H(K,K-1)                                                    SCH0C
  77         Q = H(K+1,K-1)                                                  SCH0C
  78         R = 0.                                                          SCH0C
  79         IF(.NOT.LAST) R = H(K+2,K-1)                                    SCH0C
  80         X = DABS(P) + DABS(Q) + DABS(R)                                 SCH0C
  81         IF(X .EQ. 0.) GO TO 220                                         SCH0C
  82         P = P/X                                                         SCH0C
  83         Q = Q/X                                                         SCH0C
  84         R = R/X                                                         SCH0C
  85   150   S = DSQRT(P**2 + Q**2 + R**2)                                   SCH0C
  86         IF(P .LT. 0.) S = -S                                            SCH0C
  87         IF(K .NE. M) H(K,K-1) = -S*X                                    SCH0C
  88         IF(K.EQ.M .AND. L.NE.M) H(K,K-1) = -H(K,K-1)                    SCH0C
  89         P = P + S                                                       SCH0C
  90         X = P/S                                                         SCH0C
  91         Y = Q/S                                                         SCH0C
  92         Z = R/S                                                         SCH0C
  93         Q = Q/P                                                         SCH0C
  94         R = R/P                                                         SCH0C
  95         DO 170 J=K,NN                                                   SCH0C
  96           P = H(K,J) + Q*H(K+1,J)                                       SCH0C
  97           IF(LAST) GO TO 160                                            SCH0C
  98           P = P + R*H(K+2,J)                                            SCH0C
  99           H(K+2,J) = H(K+2,J) - P*Z                                     SCH01
 100   160     H(K+1,J) = H(K+1,J) - P*Y                                     SCH01
 101           H(K,J) = H(K,J) - P*X                                         SCH01
 102   170   CONTINUE                                                        SCH01
 103         J = MIN0(K+3,N)                                                 SCH01
 104         DO 190 I=1,J                                                    SCH01
 105           P = X*H(I,K) + Y*H(I,K+1)                                     SCH01
 106           IF(LAST) GO TO 180                                            SCH01
 107           P = P + Z*H(I,K+2)                                            SCH01
 108           H(I,K+2) = H(I,K+2) - P*R                                     SCH01
 109   180     H(I,K+1) = H(I,K+1) - P*Q                                     SCH01
 110           H(I,K) = H(I,K) - P                                           SCH01
 111   190   CONTINUE                                                        SCH01
 112         DO 210 I=1,NN                                                   SCH01
 113           P = X*U(I,K) + Y*U(I,K+1)                                     SCH01
 114           IF(LAST) GO TO 200                                            SCH01
 115           P = P + Z*U(I,K+2)                                            SCH01
 116           U(I,K+2) = U(I,K+2) - P*R                                     SCH01
 117   200     U(I,K+1) = U(I,K+1) - P*Q                                     SCH01
 118           U(I,K) = U(I,K) - P                                           SCH01
 119   210   CONTINUE                                                        SCH01
 120   220 CONTINUE                                                          SCH01
 121       GO TO 40                                                          SCH01
 122   230 FAIL = 0                                                          SCH01
 123       RETURN                                                            SCH01
 124       END                                                               SCH01
```

```
      SUBROUTINE SYSSLV                                                 SYS0
      IMPLICIT REAL*8 (A-H,O-Z)                                         SYS0
C                                                                       SYS0
      COMMON/SLVBLK/A(5,5),B(5),N                                       SYS0
      REAL*8 MAX                                                        SYS0
    1 NM1 = N - 1                                                       SYS0
      N1 = N+1                                                          SYS0
C                                                                       SYS0
C COMPUTE THE LU FACTORIZATION OF A.                                    SYS0
      DO 80 K=1,N                                                       SYS0
        KM1 = K-1                                                       SYS0
        IF(K.EQ.1) GO TO 20                                             SYS0
        DO 10 I=K,N                                                     SYS0
          DO 10 J=1,KM1                                                 SYS0
            A(I,K) = A(I,K) - A(I,J)*A(J,K)                             SYS0
   10   CONTINUE                                                        SYS0
   20   IF(K.EQ.N) GO TO 100                                            SYS0
        KP1 = K+1                                                       SYS0
        MAX = DABS(A(K,K))                                              SYS0
        INTR = K                                                        SYS0
        DO 30 I=KP1,N                                                   SYS0
          AA = DABS(A(I,K))                                             SYS0
          IF(AA .LE. MAX) GO TO 30                                      SYS0
          MAX = AA                                                      SYS0
          INTR = I                                                      SYS0
   30   CONTINUE                                                        SYS0
        IF(MAX .EQ. 0.) STOP                                            SYS0
        A(N1,K) = INTR                                                  SYS0
        IF(INTR .EQ. K) GO TO 50                                        SYS0
        DO 40 J=1,N                                                     SYS0
          TEMP = A(K,J)                                                 SYS0
          A(K,J) = A(INTR,J)                                            SYS0
          A(INTR,J) = TEMP                                              SYS0
   40   CONTINUE                                                        SYS0
   50   DO 80 J=KP1,N                                                   SYS0
          IF(K.EQ.1) GO TO 70                                           SYS0
          DO 60 I=1,KM1                                                 SYS0
            A(K,J) = A(K,J) - A(K,I)*A(I,J)                             SYS0
   60     CONTINUE                                                      SYS0
   70     A(K,J) = A(K,J)/A(K,K)                                        SYS0
   80 CONTINUE                                                          SYS0
C                                                                       SYS0
C INTERCHANGE THE COMPONENTS OF B.                                      SYS0
C                                                                       SYS0
  100 DO 110 J=1,NM1                                                    SYS0
        INTR = A(N1,J)                                                  SYS0
        IF(INTR .EQ. J) GO TO 110                                       SYS0
        TEMP = B(J)                                                     SYS0
        B(J) = B(INTR)                                                  SYS0
        B(INTR) = TEMP                                                  SYS0
  110 CONTINUE                                                          SYS0
C                                                                       SYS0
C SOLVE LX = B.                                                         SYS0
C                                                                       SYS0
  200 B(1) = B(1)/A(1,1)                                                SYS0
      DO 220 I=2,N                                                      SYS0
        IM1 = I-1                                                       SYS0
        DO 210 J=1,IM1                                                  SYS0
          B(I) = B(I) - A(I,J)*B(J)                                     SYS0
  210   CONTINUE                                                        SYS0
        B(I) = B(I)/A(I,I)                                              SYS0
  220 CONTINUE                                                          SYS0
C                                                                       SYS0
```

```
C SOLVE UX = B.                                                    SYS0
C                                                                  SYS0
  300 DO 310 II=1,NM1                                              SYS0
        I = NM1-II+1                                               SYS0
        I1 = I+1                                                   SYS0
        DO 310 J=I1,N                                              SYS0
          B(I) = B(I) - A(I,J)*B(J)                                SYS0
  310 CONTINUE                                                     SYS0
      RETURN                                                       SYS0
      END                                                          SYS0
```

```
      SUBROUTINE GAUSEL (MAX, N, A, NR, B, IERR)                        GAU
      IMPLICIT REAL*8 (A-H,O-Z)                                         GAU
C     FUNCTION              - COMPUTES SOLUTION TO A SET OF SIMULTANEOUS GAU
C                             LINEAR EQUATIONS (DOES NOT GIVE PIVOT OR  GAU
C                             DETERMINANT DATA)                         GAU
C     USAGE                 - CALL GAUSEL (MAX,N,A,NR,B,IERR)         GAU
C     PARAMETERS  MAX       - MAXIMUM ROW DIMENSION OF B              GAU
C                 N         - ORDER OF A                              GAU
C                 A(N,N)    - INPUT MATRIX OF COEFFICIENTS (DESTROYED) GAU
C                 NR        - NUMBER OF COLUMNS IN B                  GAU
C                 B(MAX,NR) - MATRIX OF CONSTANTS (REPLACED BY SOLUTIONS) GAU
C                 IERR      - INTEGER ERROR CODE                      GAU
C                             = 0   NORMAL RETURN                     GAU
C                             = 2   INPUT MATRIX IS SINGULAR          GAU
C     REQUIRED ROUTINES     - NONE                                    GAU
C                                                                     GAU
C     SOURCE                                                          GAU
C                 NASA, LRC, ANALYSIS AND COMPUTATION DIVISION SUBPROGRAM GAU
C                 LIBRARY                                             GAU
C ****                                                                GAU
      DIMENSION A(N,N),B(MAX,NR)                                      GAU
      NM1 = N-1                                                       GAU
      IF (NM1 .EQ. 0) GO TO 140                                       GAU
C ****                                                                GAU
C     FIND LARGEST REMAINING ELEMENT IN I-TH COLUMN FOR PIVOT         GAU
C ****                                                                GAU
      DO 100 I=1,NM1                                                  GAU
         BIG = 0.                                                     GAU
         DO 20 K=I,N                                                  GAU
            TERM = DABS(A(K,I))                                       GAU
            IF (TERM - BIG) 20,20,10                                  GAU
   10       BIG = TERM                                                GAU
            L = K                                                     GAU
   20    CONTINUE                                                     GAU
         IF (BIG) 40,30,40                                            GAU
   30    IERR = 2                                                     GAU
         RETURN                                                       GAU
   40    IF (I-L) 50,80,50                                            GAU
C ****                                                                GAU
C     PIVOT ROWS OF A AND B                                           GAU
C ****                                                                GAU
   50    CONTINUE                                                     GAU
         DO 60 J=1,N                                                  GAU
            TEMP = A(I,J)                                             GAU
            A(I,J) = A(L,J)                                           GAU
            A(L,J) = TEMP                                             GAU
   60    CONTINUE                                                     GAU
         DO 70 J=1,NR                                                 GAU
            TEMP = B(I,J)                                             GAU
            B(I,J) = B(L,J)                                           GAU
            B(L,J) = TEMP                                             GAU
   70    CONTINUE                                                     GAU
   80    CONTINUE                                                     GAU
C ****                                                                GAU
C     STORE PIVOT AND PERFORM COLUMN OPERATIONS ON A AND B            GAU
C ****                                                                GAU
         IP1 = I+1                                                    GAU
         DO 100 II=IP1,N                                              GAU
            A(II,I) = A(II,I)/A(I,I)                                  GAU
            X3 = A(II,I)                                              GAU
            DO 90 K=IP1,N                                             GAU
               A(II,K) = A(II,K) - X3*A(I,K)                          GAU
   90       CONTINUE                                                  GAU
```

```
63                DO 100 K=1,NR                                          GAUO
64                   B(II,K) = B(II,K) - X3*B(I,K)                       GAUO
65   100  CONTINUE                                                       GAUO
66   C ****                                                             GAUO
67   C     PERFORM BACK SUBSTITUTION                                    GAUO
68   C ****                                                             GAUO
69         DO 110 IC=1,NR                                               GAUO
70            B(N,IC) = B(N,IC)/A(N,N)                                  GAUO
71   110  CONTINUE                                                       GAUO
72         DO 130 KK=1,NM1                                              GAUO
73            I = N-KK                                                  GAUO
74            IP1 = I+1                                                 GAUO
75            DO 130 J=1,NR                                             GAUO
76               SUM = B(I,J)                                           GAUO
77               DO 120 K=IP1,N                                         GAUO
78                  SUM = SUM - A(I,K)*B(K,J)                           GAUO
79   120          CONTINUE                                               GAUO
80               B(I,J) = SUM/A(I,I)                                    GAUO
81   130  CONTINUE                                                       GAUO
82         RETURN                                                       GAUO
83   140  CONTINUE                                                       GAUO
84         IF (A(1,1) .EQ. 0.) GO TO 300                                GAUO
85         DO 150 J=1,NR                                                GAUO
86            B(1,J) = B(1,J)/A(1,1)                                    GAUO
87   150  CONTINUE                                                       GAUO
88         RETURN                                                       GAUO
89    300  IERR = 2                                                     GAUO
90         RETURN                                                       GAUO
91         END                                                          GAUO
```

```
  0         SUBROUTINE PNCH (A,NA,NAM,IOP)
  1 C IOP(1)=0, SKIP TITLE; IOP(2)=N, SKIP LINES; IOP(3)=1, TAB 25 SPACES.
  2         IMPLICIT REAL*8 (A-H,O-Z)
  3         DIMENSION A(1),IOP(4),NA(2)
  4         NR=NA(1)
  5         NC=NA(2)
  6         NMAX=NR*NC
  7         NSKIP=IOP(2)
  8         IF (IOP(2).EQ.0) GO TO 205
  9         DO 200 I=1,NSKIP
 10     200 WRITE(7,150)
 11     150 FORMAT(2X)
 12     205 CONTINUE
 13         IF (IOP(1).EQ.0) GO TO 210
 14         WRITE(7,151) NAM,NR,NC
 15     151 FORMAT(A4,/,2I5)
 16     210 CONTINUE
 17         DO 250 I=1,NR
 18         IF (IOP(3).EQ.0) WRITE(7,152) (A(J),J=I,NMAX,NR)
 19         IF (IOP(3).NE.0) WRITE(7,153) (A(J),J=I,NMAX,NR)
 20     250 CONTINUE
 21     152 FORMAT(6(1PD13.5))
 22     153 FORMAT(25X,6(1PD13.5))
 23         RETURN
 24         END
```

```
0       FUNCTION DIMAG(Z)                                       DIMC
1       REAL*8 A(2),DIMAG                                       DIMC
2       COMPLEX*16 Z,B                                          DIMC
3       EQUIVALENCE (A,B)                                       DIMC
4       B=Z                                                     DIMC
5       DIMAG=A(2)                                              DIMC
6       RETURN                                                  DIMC
7       END                                                     DIMC
```

```
0       FUNCTION DREAL(Z)                                       DRE00
1       REAL*8 A(2),DREAL                                       DRE00
2       COMPLEX*16 Z,B                                          DRE00
3       EQUIVALENCE (A,B)                                       DRE00
4       B=Z                                                     DRE00
5       DREAL=A(1)                                              DRE00
6       RETURN                                                  DRE00
7       END                                                     DRE00
```

```
      BLOCK DATA                                                        MODD0
      IMPLICIT REAL*8 (A-H,O-Z)                                         MODD0
      COMMON/LINES/TITLE(10),TIL(3),NLP,LIN                             MODD0
      COMMON/FORM/FMT1(2),FMT2(2),NEPR                                  MODD0
      COMMON/TOL/EPSAM,EPS8M,IACM                                       MODD0
      COMMON/CONV/SUMCV,RICTCV,SERCV,MAXSUM                             MODD0
      DATA LIN,NLP/1,58/                                                MODD0
      DATA NEPR,FMT1/7,8H(1P7D16.,8H7)      /                           MODD0
      DATA TIL/8H      ORA,8HCLS  PRO,8HGRAM    /                       MODD0
      DATA FMT2/8H(3X,1P7D,8H16.7)  /                                   MODD0
      DATA EPSAM/1.E-10/                                                MODD0
      DATA EPS8M/1.E-10/                                                MODD0
      DATA IACM/12/
      DATA SUMCV/1.E-8/                                                 MODD0
      DATA RICTCV/1.E-8/                                                MODD0
      DATA SERCV/1.E-8/                                                 MODD0
      DATA MAXSUM/50/                                                   MODD0
      END                                                               MODD0
```