NASA CR-137553
AVAILABLE TO THE PUBLIC
ASRL TR 174-2

(NASA-CR-137553) USER'S MANUAL FOR COMPUTER PROGRAM ROTOR (Massachusetts Inst. of Tech.)
235 p HC $7.50 CSCL 01B

N75-14725

Unclas
G3/02 08233

USER'S MANUAL FOR COMPUTER PROGRAM ROTOR

Masahiro Yasue

August 1974

Distribution of this report is provided in the interest of information exchange. Responsibility for the contents resides in the author or organization that prepared it.

Prepared under Contract No. NAS2-7262 by

Aeroelastic and Structures Research Laboratory
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

JAN 1975

for

AMES RESEARCH CENTER
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MOFFETT FIELD, CALIFORNIA 94035

NASA CR-137553
ASRL TR 174-2

USER'S MANUAL FOR COMPUTER PROGRAM ROTOR

Masahiro Yasue

August 1974

Distribution of this report is provided in the interest of information exchange. Responsibility for the contents resides in the author or organization that prepared it.

Prepared under Contract No. NAS2-7262 by

Aeroelastic and Structures Research Laboratory
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

for

AMES RESEARCH CENTER
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MOFFETT FIELD, CALIFORNIA 94035

1. Report No.
NASA CR-137553

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
USER'S MANUAL FOR COMPUTER PROGRAM ROTOR

5. Report Date
August 1974

6. Performing Organization Code

7. Author(s)
Masahiro Yasue

8. Performing Organization Report No.
ASRL TR 174-2

9. Performing Organization Name and Address
Massachusetts Institute of Technology
Aeroelastic and Structures Research Laboratory
Cambridge, Massachusetts 02139

10. Work Unit No.

11. Contract or Grant No.
NAS2-7262

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Moffett Field, California 94035

13. Type of Report and Period Covered
Contractor Report

14. Sponsoring Agency Code

15. Supplementary Notes
NASA Technical Monitors: J.P. Rabbott, Jr., and Wayne R. Johnson

16. Abstract

This report presents a detailed description of a computer program to calculate tilt-rotor aircraft dynamic characteristics. This program (named ROTOR) consists of two separate parts. In the first part, the natural frequencies and corresponding mode shapes of the rotor blade and wing are developed from structural data (mass distribution and stiffness distribution). The second part of the program deals with the frequency response (to gust and blade pitch control inputs) and eigenvalues of the tilt-rotor dynamic system, based on the natural frequencies and mode shapes derived beforehand. Sample problems are included to assist the user.

17. Key Words (Suggested by Author(s))
Rotary Wing
Dynamics
Gust Response
Proprotor
Tilt-Rotor

18. Distribution Statement
Unclassified, Unlimited

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages
234

22. Price*

---

* For sale by the National Technical Information Service, Springfield, Virginia 22151

FOREWORD

This report has been prepared by the Aeroelastic and Structures Research Laboratory (ASRL), Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, under NASA Contract No. NAS2-7262 from the Ames Research Center, National Aeronautics and Space Administration, Moffett Field, California 94035. Mr. John Rabbott and Dr. Wayne Johnson of the Ames Research Center served as technical monitors. The valuable assistance and advice received from these individuals is gratefully acknowledged.

The research described in this report was supervised by Professor Norman D. Ham and Associate Professor Pin Tong. The author would like to express his deep appreciation and gratitude for their invaluable advice and guidance throughout this study. The author is also deeply indebted to Professor John Dugundji and to Dr. Wayne Johnson, who contributed various useful suggestions. Professor E.A. Witmer's advice and assistance in various phases of the work is acknowledged gratefully.

The computations were performed at the Information Processing Center of the Massachusetts Institute of Technology.

ABSTRACT

This report presents a detailed description of a computer program to calculate tilt-rotor aircraft dynamic characteristics. This program (named ROTOR) consists of two separate parts. In the first part, the natural frequencies and corresponding mode shapes of the rotor blade and wing are developed from structural data (mass distribution and stiffness distribution). The second part of the program deals with the frequency response (to gust and blade pitch control inputs) and eigenvalues of the tilt-rotor dynamic system, based on the natural frequencies and mode shapes derived beforehand. Sample problems are included to assist the user.

CONTENTS

Section | Page

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION 1

INTRODUCTION

1.1 Purpose and Scope

Program ROTOR is an in-core program written in FORTRAN IV language for analysis of the dynamic characteristics of the tilt-rotor aircraft.

The analytical model considered here consists of a cantilevered semispan wing with the engine-rotor system at the wing tip (see Fig. 1 of Ref. 1). The dynamic and aeroelastic characteristics of this aircraft are in many ways unique and complicated. The large flexible blades with a large amount of twist have significant coupling between flapping and lagging motion. The engines and gearboxes at the wing tip lead to low wing natural frequencies and possible resonances in the low frequency range.

The purpose of this program is the numerical analysis of the complicated dynamic and aeroelastic behavior of the tilt-rotor aircraft. The first step is the derivation of the equations of motion of the blade and the wing, including inertia forces, blade aerodynamic forces and wing aerodynamic forces. This formulation is described in detail in Ref. 1. Based on the equations of motion, the frequency response of the blade and wing motions to the gust input or blade pitch angle control input are derived. An eigenvalue analysis provides the system stability characteristics.

1.2 Program Outline and Limitations

Program ROTOR consists of two separate parts. One is called FREEVI (free vibration of the tilt-rotor aircraft), which produces the natural frequencies and mode shapes of the free vibration from the blade or from the wing structural characteristic data. The second part is named TILDYN (tilt-rotor dynamics) which calculates the dynamic characteristics of the aircraft, including eigenvalues and frequency response to the gust and to the pitch-control inputs.

The reason for the separation of the program into two parts is that this system gives the user the opportunity to check the results for the natural frequencies and mode shapes without performing the entire calculation. In addition, the dynamic characteristics can be evaluated easily by changing the input natural frequencies or using different assumed modes without changing the structural characteristics, mass distribution, or stiffness distribution. The disadvantage is that the input data for TILDYN are the output data of FREEVI.

The free vibration problem of the wing and blade is solved as an eigenvalue problem in FREEVI by the finite element method. The wing has three degrees of freedom, vertical bending, chordwise bending and torsion. A large wing tip mass represents the rotor, engine, and gearbox.

Only flapping and lagging motions are considered for the blade. Torsion is neglected as a higher-order effect for the blade case. The rotor types treated here are the hingeless rotor and the gimballed rotor. The maximum number of elements is limited to twenty.

With respect to the TILDYN program, it should be mentioned that the flight configuration is restricted to cruising flight only, with the rotor disk plane perpendicular to the free stream. Both powered and autorotation cases can be treated. Total degrees of freedom considered are nine or eighteen for powered flight. The nine degrees of freedom consist of blade flapping and lagging fundamental modes (each has a collective and two cyclic degrees of freedom) and wing vertical bending, chordwise bending, and torsion modes. The eighteen degrees of freedom include two additional blade modes and three additional wing modes. It should be noted that description of the blade motion requires three independent degrees of freedom to reduce one equation with periodic coefficients in the rotating system to three equations with constant coefficients in the non-rotating system.

In the autorotation case, one more degree of freedom is added, the rigid body rotation of the rotor, and the total degrees of freedom become ten or nineteen. After the construction of the equations of motion for each case, the frequency response problem is solved. The excitation inputs consist of the following: vertical gust, lateral gust, longitudinal gust, collective-blade pitch control and two cyclic blade pitch controls. By appropriately specifying the excitation input components, the flight in a cross-wind gust can be considered.

The eigenvalue problem to be solved is the usual eigenvalue problem of a linear system of equations. The EISPACK subroutine developed by the Argonne Code Center is used to treat this problem (Refs. 2 and 3).

SECTION 2

DESCRIPTION OF THE PROGRAMS

2.1 Description of the FREEVI Program

This program consists of one main program and twelve subroutines for the computation of the lowest few eigenvalues and eigenvectors of the proprotor dynamic system modelled by the finite-element method, as described in Ref. 1. The outline flow chart is shown in Fig. 1.

Input data include element-structural characteristics (mass distribution, stiffness distribution and angle of twist), rotational speed, and some instruction data for the computation. The boundary conditions are automatically chosen when the calculation case is selected appropriately. Boundary conditions and degrees of freedom are tabulated in Table 1 for the rotor and wing. Next, the element stiffness and mass matrices are assembled globally. From the input information, the boundary conditions are imposed on the global system. The subspace iteration method is applied to find the eigenvalues and eigenvectors of the system (Refs. 4 and 5). Consider the eigenvalue problem of the n-degree-of-freedom equations:

$$[K][u] = \lambda [M][u] \tag{2.1}$$

where [K] and [M] are square stiffness and mass matrices with order n, [u] is a matrix of the mode shape and λ is an eigenvalue. When m eigenvalues and eigenvectors are required, the main steps of the subspace iteration method are as follows:

(a) Assume mode shape matrix $[u_o]$; nxm matrix containing m vectors

(b) $[M_R] = [u_o]^T[M][u_o]$; $[K_R] = [u_o]^T[K][u_o]$ where $[M_R]$ and $[K_R]$ are reduced square matrices with order m.

(c) Find eigenvectors $[A]$ (mxm matrix) such that $[K_R]\,[A] = [D]\,[M_R]\,[A]$, with $[D]$ denoting a diagonal matrix.

(d) $[\bar{u}_o] = [u_o]\,[A]$

(e) $[u_1] = [K]^{-1}[M]\,[\bar{u}_o]$

(f) $[u_o] = [u_1]$ and go to step (b)

The eigenvalue analysis of the smaller system (order m) in step (c) is achieved by using the Jacobi method. The criterion of terminating the iteration is defined as

$$\left| \frac{\lambda_{i+1} - \lambda_i}{\lambda_i} \right| \leq e \qquad (2.2)$$

Each eigenvalue must satisfy this criterion; the error threshold e can be defined by the user.

Output includes the input data, the eigenvalues, the eigenvectors, and, if required, punched-out cards of the eigenvectors. A built-in message as to whether convergence was achieved is also furnished.

A short description of each of the subroutines is given below:

MAIN	Defines dimensions
TEIGEN	Calculates the normal modes and frequencies
INPUT	Supplies input information
ELEMK	Controls the generation of element stiffness and mass matrices
MESH	Calculates mesh information for the finite element assemblage
ASBV	Applies boundary conditions
FAC	Triple matrix factorization
MTRTR	Matrix multiplication
MULTZ	Matrix multiplication
SOLZ	Forward and backward substitution
DNROOT	Eigen-analysis routine
EIGEN	Eigen-analysis routine needed in DNROOT
OUTPUT	Output routine

The listing is shown in Appendix A.

2.2 Description of the TILDYN Program

This program to solve the equations of motion of the tilt-rotor aircraft derived in Ref. 1, consists of one main program and twenty-four subroutine programs. The outline flow chart is shown in Fig. 2.

Input data are natural frequencies and corresponding mode shapes of the rotor and wing, aerodynamic coefficients, and flight conditions.

Based on such input data, the coefficients of the equations of motion are derived, using the numerical integration method. Finally, the equations of motion are formulated as a matrix:

$$[A]\{\ddot{x}\} + [B]\{\dot{x}\} + [C]\{x\} = [D]\{e\} \qquad (2.3)$$

[A], [B], [C], and [D] are the coefficient matrices, including inertia terms and aerodynamic terms. The matrix {x} is a set of variables and {e} is an exciting force matrix including gust components and blade pitch-control components (see Ref. 1). These equations have nine or eighteen degrees of freedom in the powered flight case. In the autorotation flight case, ten or nineteen degrees of freedom are required, due to the addition of rigid-body rotation (see Table 2).

The dynamic characteristics of these equations are analyzed by two methods. One is the frequency-response analysis and the other is the eigenvalue analysis. In the frequency-response analysis, the accelerations and velocities of the equations are expressed in terms of a given frequency, and the differential equations are transformed into a set of linear algebraic equations. These linear equations are solved by the Gauss-Jordan reduction to obtain the response to the gust or blade pitch-control input.

The eigenvalue problem is formulated in the usual way. Equation 2.3 is rewritten

$$\begin{Bmatrix} \ddot{x} \\ x \end{Bmatrix} = \left[\begin{array}{c|c} -A^{-1}B & -A^{-1}C \\ \hline I & 0 \end{array}\right] \begin{Bmatrix} \dot{x} \\ x \end{Bmatrix} \tag{2.4}$$

to obtain first-order differential equations. The real general matrix eigenvalue problem is solved by the EISPACK package, developed by Argonne National Laboratory to solve a standard matrix eigenvalue-eigenvector problem (Refs. 2 and 3).

A short description for each of the subroutines of the TILDYN program is given below:

MAIN	Defines the sequence of the program
BLOCK DATA	Initializes the coefficients of Gaussian quadrature
INITIL	Initialization of the matrices
COEFF	Defines the points and coefficients of the Gaussian quadrature
INPUT	Supplies input information
INTPL	Interpolation for the numerical integration by Gaussian quadrature
AERO	Defines the aerodynamic coefficients at the points of Gaussian quadrature
ORDINT	Defines the order of the numerical integration
INTEG	Numerical integration
F	Defines the integrand function
AINER	Defines the inertia coefficients of the equations in matrix form
AEROMT	Defines the aerodynamic coefficients of the equations in matrix form
EQMTX	Defines the coefficient matrices [A], [B], [C], and [D] in Eq. 2.3.
AUTO	In the autorotation case, another degree of freedom is added

GUSTCO	Defines gust and blade pitch control components
FRQRES	Calculates the frequency response
GAELI	The Gauss-Jordan reduction routine
EIGEN	Routine to form the eigenvalue problem and to call EIPACK subroutine
EIPACK	An eigensystem problem solver for the real general matrix consisting of EISPACK subroutine BALANC, ELMHES, ELTRAN, HQR2 and BALBAK
BALANC	Balances a real general matrix and isolates eigenvalues whenever possible
ELMHES	Obtains an upper Hessenberg matrix from a real general matrix
ELTRAN	Accumulates the elementary similarity transformations for the reduction to upper Hessenberg form
HQR2	Finds the eigenvalues and eigenvectors
BALBAK	Forms the eigenvectors by back-transforming those of the corresponding balanced matrix determined by BALANC
MINV	Inverses a matrix

The listing is shown in Appendix A.

SECTION 3

USER'S GUIDE FOR FREEVI PROGRAM

3.1 Input Data Requirements

The input and output of the computer code is a part of the built-in program with fixed format. This approach requires a minimum knowledge of the programs and programming.

The finite-element model for the input data is shown in Fig. 3. The structure is divided into several elements for application of the finite-element method. The mass distribution, bending and torsional stiffness, and angle of twist are the average values in the element. The unit system used should be consistent throughout the entire program.

This program calculates the natural frequencies and normal modes for five cases, including wing vibration and blade vibration with various boundary conditions (see Table 1). The parameter ICASE specifies the particular case in the program. The parameter IPUNCH specifies whether a punched card deck of the mode shapes is required for input to the program TILDYN.

Uncoupled mode shapes, instead of coupled mode shapes, can be generated, if necessary. The parameter IGUEST controls the initial assumed values for the purpose of generating the uncoupled mode shapes.

The value of M expresses the number of eigenvalues and mode shapes required by the user.

Parameters and variables are described in detail below:

DES	A vector to express the test identifying information. The user can punch the run identification in the first column through the eightieth column of the first card. The format is 20A4.

ICASE	A parameter to specify the calculation case: ICASE=1; wing case with clamped boundary conditions at the root. ICASE=2; blade case with boundary conditions clamped for the flapping motion and clamped for the lagging motion at the root. ICASE=3; blade case, clamped for flapping and hinged for lagging. ICASE=4; blade case, hinged for flapping and clamped for lagging. ICASE=5; blade case, both hinged boundary conditions. It is punched in the integer format as I1 in the first column of the second card.
IPUNCH	A parameter to control whether the mode shapes are punched out in cards for input to the TILDYN program. IPUNCH=0; no punched output IPUNCH=1; punched output The parameter is punched in the integer format as I1 in the first column of the third card.
IGUEST	A parameter to control the mode-shape type coupled or uncoupled, both for the blade and the wing. IGUEST=0; coupled IGUEST=1; uncoupled vertical bending (w) IGUEST=2; uncoupled chordwise bending (v) IGUEST=3; uncoupled torsion (ϕ) It should be noted that the terms PB, RAMDA, COL and THETAE related to coupled motion should be set to zero when uncoupled mode shapes are required. The parameter is punched in the integer format as I1 in the first column of the fourth card.

NET	Total number of elements, maximum number is 20. It is punched out in the integer format as I5 in the first through fifth columns in the fifth card.
NITR	The maximum number of iterations to be performed. If the number of iterations reaches NITR, yet the iteraation is not converged, the program execution is terminated and a built-in message appears. Recommended value for NITR is 20. It is punched in the integer format as I5 in the 6th through the 10th columns of the fifth card.
M	Number of vibration modes required by the user. Recommended value for M is less than 10. If $M = 10$ to 20, NITR is recommended to be 50.
ERR	The error limit used to compare with $\|(\lambda_{i+1} - \lambda_i)/\lambda_i\|$, where λ_i is the eigenvalue calculated at the ith iteration cycle. The iteration terminates if the calculated value is smaller than ERR. Recommended values for ERR are 0.001 to 0.01. It is punched in the real value format (F10.6) at the first through 10th columns of the sixth card.
OMEG	Rotational speed Ω in rad/sec, the direction of rotation is positive for the upward rotational vector when the aircraft configuration is in the helicopter mode. For the wing or non-rotating blade it is set to zero. It is punched in real value format (F10.6) in the first through the 10th columns of the seventh card.
RAMDA	Inflow ratio λ is defined as $(V + v)/\Omega R$. It determines the collective pitch of the blade. For the wing and non-rotating blade, it is set to zero. It is punched in real value format (F10.6) in the first through the 10th columns of the 8th card.

COL	Collective pitch angle in radians (θ_D in Ref. 1). For the wing case it is set to zero. It is punched in real value format (F10.6) in the first through the 10th columns of the 9th card.
SPKB	The flapwise spring constant at the root of the hinged rotor. If there is no spring, it is set to zero. It is punched in real value format (F10.6) in the 1st through the 10th columns of the 10th card.
SPKC	The lagwise spring constant at the root of the hinged rotor. If there is no spring, it is set to zero. It is punched in real value format (F10.6) in the 11th through 20th columns of the 10th card.
ALPHAH	The number is used to avoid a singularity of the stiffness matrix in the case of the hinged blade. The recommend value for ALPHAH is the squared value (λ^2) of the first non-zero eigenvalue. The value 5000.0 is appropriate for the first trial. It is punched in real value format (F10.6) in the first through the 10th columns of the 11th card.
EIBE	A vector to express the vertical bending stiffness $(EI)_B$ of the element. The length of the vector equals NET. It is punched in the exponent format (E15.7) and five data items can be included per card. These data occupy the 12th card through card [10+(NET/5)] if NET is a multiple of 5. Otherwise, up to card [11+(NET/5)] is occupied.
EICE	A vector to express the chordwise bending stiffness $(EI)_C$ of the element. The length of the vector is NET. It is punched in the exponent format (E15.7) and five data items can be included in a card. These data occupy (NET/5) cards if NET is a multiple of 5, otherwise [1+(NET/5)] cards.

THETAE	A vector to express the angle of twist θ_{AT} in radians of the structure element, positive nose up. It should be the average angle of twist over the element. The length of the vector is NET. Eight data items can be punched in real value format (F10.6) in each card. These data occupy (NET/8) cards if NET is a multiple of 8, otherwise [1+(NET/8)] cards.
AMASE	A vector to describe the mass distribution (mass/unit length); its length is NET. It is punched in real value format (F10.6), 8 data items on one card. These data occupy (NET/8) cards if NET is a multiple of 8, otherwise [1+(NET/8)] cards.
ESE	A vector to define the size of the beam element of the blade or wing and its length is NET. It is punched in real value format (F10.6), 8 data items on a card. These data occupy (NET/8) cards if NET is a multiple of 8, otherwise [1+(NET/8)] cards.
AMN	The number to express the tip mass. In the case of the wing, it includes the nacelle and all blade mass as $AMN = M_N + NM_B$ in the symbols of Ref. 1. If a tip mass exists in the blade, it is also appropriate to use AMN. It is punched in real value format (F10.6) in the first through the 10th columns of the card. The next four numbers PIR, PIY, PIP and PBW are punched on the same card with AMN.
PIR	A number to express the rolling moment of inertia of the nacelle and blades at the wing tip: $PIR = I_{P_r} + NI_B$

The format is (F10.6) and it is punched from the 11th to 20th columns.

PIY	A number to express the yawing moment of inertia of the nacelle and blades at the wing tip: $PIY = I_{P_y} + \frac{N}{2} I_B + NM_B h^2$ The format is (F10.6) and it is punched from the 21st to the 30th columns.
PIP	A number to express the pitching moment of inertia of the nacelle and blades at the wing tip: $PIP = I_{P_p} + \frac{N}{2} I_B + NM_B h^2$ The format is (F10.6) and it is punched from the 31st to the 40th columns.
PBW	A number to express the mass coupling effect at the tip between the wing vertical bending and the torsion due to blade mass: $PBW = NM_B h$ The format is (F10.6) and it is punched from the 41st to the 50th columns.

If the calculation case is the wing (ICASE=1), the next three data cards should be added:

GJ	A vector to express the torsional rigidity: its length is NET. The format is (E15.7) and five data items can be included in a card. These data occupy (NET/5) cards if NET is a multiple of 5, otherwise [1+(NET/5)].
PI	A vector to express wing mass moment of inertia about the elastic axis per unit length: its length is NET. The format is (E15.7) and five data items can be included in a card. These data occupy (NET/5) cards if NET is a multiple of 5, otherwise [1+(NET/5)].

PI12	A vector to express wing static mass moment of the segment to define the coupling motion between wing vertical bending and torsion. The vector length is NET and the format is (E15.7). Five data items can be included in a card and data occupy NET/5 cards if NET is a multiple of 5, otherwise [1+(NET/5)].

The data deck setup is shown in Fig. 5 and the example problem data listing is in Appendix B.

A few remarks will now be stated to avoid misuse of the program:

1) A consistent unit system must be adopted.
2) The maximum element number (NET) is 20.
3) Appropriate rotor rotational direction must be chosen. If OMEG is negative, RAMDA (inflow ratio) should be negative. However, COL (the collective pitch) and THETAE (angle of twist) should be positive nose up.
4) If uncoupled mode shapes are required, the coupling terms such as RAMDA, COL, THETA, PBW and PI12 should be set to zero.
5) If there are several cases to be dealt with, the data may consist of several data sets. After the execution of the first case, the computer automatically returns to the beginning of the program and reads the second input data set. Therefore, at the end of the entire calculation, the computer notes the absence of data sets and generates an error message.

3.2 Output Features

All input data are printed out for checking. The built-in messages and outputs in the FREEVI program are as follows:

(a) TENSION DUE TO CENTRIF. FORCE

This prints out the tension force at each nodal

(b) MASS = 0.XXXXXEXXX

This indicates the total mass of the blade or wing.

(c) MOMENT OF INERTIA AT ROOT = 0.XXXXXEXXX

This gives the mass moment of inertia of the blade or wing about the virtual hinge at the root.

(d) MAX. SIZE OF STF IS XXX SPECIFIED SIZE IS XXX

This prints out the specified value of the estimated length of the stiffness matrix and the actual required value for the stiffness matrix. If the specified value is smaller, the program stops. Check the input. If the required value is smaller, no remedy is needed.

(e) THE XXXXTH DIAG. AFTER FACT=0.0, INCOMPLETE FACT

This message appears when the factorization of the mass matrix is not complete. The program also stops. Check the input.

(f) NO. OF NEGATIVE DIAG. = XXXX, FACT COMPLETED

This prints out the number of negative diagonals of the factorized mass matrix. If the printed value is other than zero, the mass matrix is not positive definite. Check the input data.

(g) EIGENVALUES =

At each iteration, calculated eigenvalues are printed out. If eigenvalues satisfy the accuracy requirements, these results are printed out in eigenvalue format (the square value of the natural frequency), radian/second and Hertz.

(h) NO. OF ITERATION = XXXX CONVERGED WITHIN 0.XXXXXEXXX

This indicates that the subspace iteration is completed. The first value printed is the number of the iteration and the second value is the error limit input by the user.

(i) NO. OF ITERATION = XXXX NOT CONVERGED

This appears instead of (h) message if the user specified maximum number of iterations (NITR) has been reached, yet the eigenvalues have not converged to within the error limit set by the user. The user should check the input for possible errors or change the error limit (ERR) because the previous error limit may be too small to be achieved, or increase the maximum number of iteration (NITR).

(j) REDUCED MASS MATRIX

The lower triangular part of the reduced mass matrix is printed out.

(k) REDUCED STIF MATRIX

The lower triangular part of the reduced stiffness matrix is printed out.

(ℓ) ****BLADE MODE SHAPES**** or ****WING MODE SHAPES****
Mode shapes are printed out. Index I indicates the eigenvalue to which the mode shape corresponds and index J indicates the station number of the node. Symbols W(I,J), V(I,J), PW(I,J), and DV(I,J) are vertical bending (flapping motion), chordwise bending (lagging motion), slope of the vertical bending, and slope of the chordwise bending, for the case of blade vibration. In addition to those symbols, PHI(I,J) and DPHI(I,J) are used for the torsion and slope of the torsion for the wing vibration. The coordinate system is shown in Fig. 4.

(m) Punched Card Output

If IPUNCH is set equal to one, the punched card output is also performed. The format is E13.5 with six data items on a card. The order is W(1,1) of the first mode to last W(1,NET+1), punched on [(NET+1)/6] cards if (NET+1) is a multiple of 6, otherwise [(NET+1)/6+1] cards. Next V(1,J), DW(1,J) and DV(1,J) groups are punched.After the first mode shape the group of the second mode shape is punched and it continues to the Mth mode shape. In the case of the wing, the output data of PHI(I,J), DPHI(I,J) are added in the same fashion after the set of DV(I,J).

3.3 Example Problems

Example problems cited here for the FREEVI program are taken from Ref. 1.

3.3.1 Application to the Wing

The free vibration of the Bell wing is considered in this report. Structural characteristics (mass distribution, bending stiffness and so on) are shown in Fig. 6. Mass moments of inertia of the nacelle and blades are tabulated in the "Bell" column of Table 3. The listing of both input and output are shown in Appendix B.

3.3.2 Application to the Blade

The hingeless rotor of the Boeing Vertol is studied here. It should be noted that the rotation direction is negative in this case. Structural characteristics are shown in Fig. 7. The listing of both input and output are shown in Appendix B.

SECTION 4

USER'S GUIDE FOR THE TILDYN PROGRAM

4.1 Input Data Requirements

This program has several parameters to designate the case being considered by the user. The first one is ITYPE, which specifies whether collective mode shapes of the blade different from the cyclic mode shapes are needed. If it is the gimballed rotor, the parameter instructs the computer to read more data for the collective mode shapes of the gimballed rotor. The parameter IFLT defines whether the case considered is powered flight or autorotation flight. The next parameter IDOF specifies the number of degrees of freedom. If it is nine, two blade modes (giving six degrees of freedom in the non-rotating system) and three wing modes are necessary. If IDOF is eighteen, two more modes for the blade and three more modes for the wing should be added to the nine degrees-of-freedom system. The parameter IRES specifies execution of the frequency response analysis. The user should decide whether the response based in terms of normal mode shapes or in terms of mode shapes normalized to unity at the blade tip. This is determined by the parameter IFRMAG. The last parameter IEIGEN specifies execution of the eigenvalue analysis.

Parameters and variables are described in input order below:

DES	A vector to express the test identifying information. The user can punch the run identification in the first column through the eightieth column of the first card. The format is 80A1.
ITYPE	A parameter to control the reading of input data depending upon the type of rotor.

ITYPE=0; the hingeless rotor in powered flight
ITYPE=1; the hingeless rotor in autorotational flight.

the gimballed rotor both in powered and autorotational flight.

The format is (I1) and it is punched in the first column of the second card.

IFLT A parameter to determine the flight condition.

IFLT=0; powered flight
IFLT=1; autorotation flight

The format is (I1) and it is punched in the first column of the third card.

IDOF A parameter to define the number of basic elastic deformation degrees of freedom and how many mode shapes are needed. It should be noted that the same number IDOF is used for both powered flight and autorotation flight.

IDOF=9; In the powered flight case, nine equations are constructed and two mode shapes for the blade and three mode shapes for the wing are necessary. In the autorotation flight (IFLT=1), ten equations are constructed, due to the addition of the rigid-body rotation of the blades.
The same number of mode shapes as for the powered flight is necessary.

IDOF=18; In powered flight, eighteen equations are formulated. In autorotation flight they become nineteen. In total, four mode shapes are necessary for the blade and six for the wing.

The format is (I1) and it is punched in the first two columns of the fourth card.

IRES A parameter to control whether the frequency response analysis is carried out.

IRES=0; it is not carried out.
IRES=1; it is carried out.

The format is (I1) and it is punched in the first column of the fifth card.

IFRMAG	A parameter to control the type of mode shapes used for the output results. IFRMAG=0; the frequency response and eigenvector results are based on mode shapes as follows: the predominant components of the blade-coupled-mode shape are normalized to R(rotor radius) at the maximum deflection point. The wing-bending-mode shapes are normalized to L (wing semispan)at the the maximum deflection point, and the wing-torsion-mode shape is normalized to unity at the maximum deflection point. This type of normalization is for the purpose of obtaining results comparable with those described in Ref. 6. IFRMAG=1; the frequency response and eigenvector results are based on the normal modes used as input data. The format is (I1), and it is punched out in the first column of the sixth card.
IEIGEN	A parameter to control whether the eigenvalue analysis is executed. IEIGEN=0; it is not executed. IEIGEN=1; it is executed. The format is (I1) and it is punched in the first column of the seventh card.
NOBLD	The blade number. The format is (I1) and it is punched in the first column of the eighth card.
ROH	The air density. The format is (E10.0); the user can put a datum in either F format or E format in the first ten columns of the ninth card.

OMEGA	The rotor rotational speed (radian/sec). The format is E10.0; the user can choose either E or F type. The datum is put in the eleventh column through the twentieth column of the ninth card. OMEGA can take positive or negative values corresponding to the rotational direction. The sign definition is the same as that of the FREEVI program.
RAMDA	The inflow ratio. The sign should be consistent with the rotational direction of the rotor. The format is either in E or F type. The datum is put in the 21st through 30th column of the ninth card.
VEL	The cruising speed of the aircraft. The format is either E or F type. The datum is put in the 31st through the 40th column of the ninth card.
R	The blade radius. The format is either E or F type. It is punched in the first through the tenth column of the tenth card.
AIB	The blade flapping moment of inertia. The format is either E or F type. It is punched in the eleventh through the 20th column of the tenth card.
CHOD	The mean chord length of the blade. The format is either E or F type. It is punched in the 21st through the 30th column of the tenth card.
CL	The lift-curve slope of the blade. The format is either E or F type. It is punched in the 31st through the 40th column of the tenth card.
CD	The drag coefficient of the blade (C_{D_o}). The format is either E or F type. It is punched in the 41st through the 50th column of the tenth card.

HMAST	The mast height. The format is either E or F type. It is punched in the 51st through the 60th column of the tenth card.
DEL3	The rotor blade pitch-flap coupling (δ_3). The unit is radians. The format is either E or F type. It is punched in the 61st through the 70th column of the tenth card.
WL	The wing semispan length. The format is either E or F type. It is punched in the first through the tenth column of the eleventh card.
WCOD	The mean wing chord length. The format is either E or F type. It is punched in the eleventh through the 20th column of the eleventh card.
WCL	The wing lift curve slope. The format is either E or F type. It is punched in the 21st through the 30th column of the eleventh card.
WCD	The wing drag coefficient (C_{D_0} of the wing). The format is either E or F type. It is punched in the 31st through the 40th column of the eleventh card.
WCMO	The wing pitching moment coefficient (C_{m_0}). The format is either E or F type. It is punched out in the 41st through the 50th column of the eleventh card.
WCMA	The wing pitching moment curve slope (C_{m_α}). The format is either E or F type. It is punched in the 51st through the 60th column of the eleventh card.
EDIS	The distance (nondimensionalized by the wing chord) between the elastic axis and the aerodynamic center of the wing (positive if the aerodynamic center is ahead of the elastic axis). The format is either E or F type. It is punched in the 61st through the 70th column of the 11th card.

WTHET	The wing trim angle of attack in radians. The format is either E or F type. It is punched in the 71st through the 80th column of the eleventh card.
CGUST	A vector to express the magnitudes of the exciting force components shown in Eq. 2.3 as {e}. CGUST(1); vertical gust u_G/V CGUST(2): lateral gust v_G/V CGUST(3); longitudinal gust w_G/V CGUST(4); collective pitch control θ_o CGUST(5); cyclic cosine pitch control θ_{1c} CGUST(6); cyclic sine pitch control θ_{1s} If the user specifies 1.0 for one of these quantities, that gust or pitch control quantity becomes the exciting force. Each vector component has either E or F format and occupies ten columns each of the twelfth card.
BRAM	A vector to express the blade eigenvalues and its length is 4. The values are the squared values of the natural frequencies (rad^2/sec^2). If IDOF is set equal to 9, the latter two eigenvalue columns may have blanks. If the calculation case is the gimballed rotor (ITYPE=1), BRAM should include the cyclic mode eigenvalues of the blade. The format is either E or F type and each component occupies ten columns in order of the 13th card.
WRAM	A vector to express the wing eigenvalues and its length is 6. The values are the squared values of the natural frequencies (rad^2/sec^2). If IDOF is set equal to 9, the latter three eigenvalue columns may have blanks. The format is either E or F type and each component occupies ten columns in order of the 14th card.
NW	A number to specify the wing element quantity. The format is (I2) and it is punched in the first two columns of the 15th card.

EMSW	A vector to describe the wing element size. The vector length is NW and the element size, nondimensionalized by the wing semispan,should be input. The format is either E or F type (E10.0). Each vector component occupies ten columns from the 16th card. The number of the card for EMSW is NW/8 if NW is a multiple of 8. Otherwise [(NW/8) + 1] cards.
G	A matrix to describe the wing vertical bending mode shape (γ in Ref. 1) at the nodes. The size of the matrix is MWx(NW+1), where MW is 3 if IDOF=9 and 6 if IDOF=18. The format is E or F type (E13.5) and G(1,1) expresses the vertical deflection at the root node of the first mode. G(1,NW+1) is the one at the tip node of the first mode (Fig. 4). The data should be punched in order from the root node to the tip node value. One card can include 6 data.
Z	A matrix to describe the wing chordwise bending mode shape (ζ in Ref. 1) at the node. Other comments are the same as for G.
DG	A matrix to describe the wing vertical bending slope ($d\gamma/dy$ in Ref. 1). Other comments are the same as for G.
DZ	A matrix to describe the wing chordwise bending slope ($d\zeta/dy$ in Ref. 1). Other comments are the same as for G.
WPHI	A matrix to describe the wing torsion deflection (ϕ in Ref. 1). Other comments are the same as for G.
DWPHI	A matrix to describe the wing torsion slope ($d\phi/dy$ in Ref. 1). Other comments are the same as for G.

The output of the FREEVI program automatically satisfies the deck setup for the TILDYN program, but for convenience the card setup for the wing mode shapes are repeated as follows:

1st card contains G(1,1), G(1,2)......... G(1,6)

Next card contains G(1,7) ....G(1,NW+1)

New card contains Z(1,1)................ Z(1,6)

Z(1,7)............. Z(1,NW+1)

New card contains DG(1,1)............

DG(1,7)............

New card contains DZ(1,1)............

DZ(1,7).............

New card contains WPHI(1,1)..........

WPHI(1,7)..........

New card contains DWPHI(1,1).........

DWPHI(1,7).........

New card contains G(2,1)............. G(2,6)

G(2,7)

Z(2,1)

Z(2,7)

.

.

New card contains DWPHI(MW,1)........

Last card for the wing mode shape contains DWPHI(MW,7)...

DWPHI(MW,NW+1)

The wing mode shapes occupy N_W cards where

$$N_W = \begin{cases} 6MW[(NW+1)/6] & \text{if } (NW+1) \text{ is a multiple of } 6 \\ 6MW[(NW+1)/6+1] & \text{if } (NW+1) \text{ is not a multiple of } 6 \end{cases}$$

N A number to specify the blade element quantity. The format is (I2) and it is punched in the first two columns of the next card to the wing mode shapes.

EMS	A vector to describe the blade element size. The vector length is N and the element size, nondimensionalized by the rotor radius, should be input. The format is either E or F type (E10.0). Each vector component occupies ten columns, and the card number is N/8 if the N is a multiple of 8. Otherwise [N/8+1] cards.
AMASS	A vector to describe the mass distribution of the blade. The vector length is (N+1). The value should be the mass distribution (mass per unit length) expressed at the node. The format is either E or F type (E10.0). Each vector component occupies ten columns, and the card number is (N+1)/8 if (N+L) is a multiple of 8. Otherwise [(N+1)/8+1] cards.
THETN	A vector to describe the angle of twist of the blade. The vector length is (N+1). The value should be the angle of twist at the node and positive nose up. The format is either E or F type (E10.0). Each vector component occupies ten columns, and the card number is (N+1)/8 if (N+1) is a multiple of 8. Otherwise, [(N+1)/8+1] cards.
COL	A number to express the collective pitch angle (θ_D in Ref. 1) defined from the performance (trim) calculation. The format is either E or F type (E10.0), and it occupies the first ten columns of the next card to THETN.
W	A matrix to describe the blade out-of-plane bending mode shape (W_j in Ref. 1) at the node. The size of the matrix is MBx(N+1) where MB is 2 if IDOF=9 and 4 if IDOF=18. The format is E or F type (E13.5) and W(1,1) expresses the out-of-plane deflection at the root node of the first mode. W(1,N+1) is the one at the tip node of the first mode. The data should be punched in order from the root node to the tip node value. One card can include 6 data.

V	A matrix to describe the blade inplane bending mode shape (V_j in Ref. 1) at the node. Other comments are the same as those for W.
DW	A matrix to describe the blade out-of-plane bending mode shape slope (dW_j/dr in Ref. 1) at the node. Other comments are the same as those for W.
DV	A matrix to describe the blade inplane bending mode shape slope (dV_j/dr in Ref. 1) at the node. Other comments are the same as those for W.

If the calculation case is the gimballed rotor (ITYPE=1), or autorotational flight (ITYPE=1 and IFLT=1), the above blade-mode shapes correspond to the cyclic mode shapes.

The output of the FREEVI program automatically satisfies the deck setup for the TILDYN program; however, for convenience, the card setup for blade mode shapes is repeated below:

```
1st card contains W(1,1), W(1,2), ..........W(1,6)
Next card contains W(1,7)................W(1,N+1)
New card contains V(1,1).................
                  V(1,7)................
                  DW(1,1) ...............
                  DW(1,7) ...............
                  DV(1,1)                  DV(1,6)
                  DV(1,7)............... DV(1,N+1)
                  W(2,1) ...............
                  W(2,7) ...............
                  V(2,1)................
                  V(2,7)................
                      .
                      .
                      .
                      .
                      .
                      .
                 DV(MB,1) ..............
```

Last card contains DV(MB,7) .... DV(MB,N+1)

The blade mode shapes occupy n_B cards where

$$n_B = \begin{cases} 4MB[(N+1)/6] & \text{if } (N+1) \text{ is a multiple of } 6 \\ 4MB[(N+1)/6+1] & \text{if } (N+1) \text{ is not a multiple of } 6 \end{cases}$$

If the computer calculation case is the gimballed rotor (ITYPE=1) or autorotational flight (ITYPE=1 and IFLT=1), the next five data cards as for the collective mode shapes of the blade should be added.

BRAMO	A vector to express the blade collective mode eigenvalues and its length is 4. The values are the squared values of the natural frequencies (rad^2/sec^2). If IDOF is set equal to 9, the latter two eigenvalue columns may have blanks. The format is either E or F type and each vector component occupies ten columns in order of the next card to the blade mode shapes.
WCOL	A matrix to describe the blade collective out-of-plane bending mode shape(W_j^o in Ref. 1) at the node. The size and other comments are the same as those for W.
VCOL	A matrix to describe the blade collective inplane bending mode shape (V_j^o in Ref. 1) at the node. Other comments are the same as those for W.
DWCOL	A matrix to describe the blade collective out-of-plane mode shape slope (dW_j^o/dr in Ref. 1). Other comments are the same as those for W.
DVCOL	A matrix to describe the blade collective inplane mode shape slope (dV_j^o/dr in Ref. 1). Other comments are the same as those for W.

The data deck setup is shown in Fig. 8, and an example problem data listing is given in Appendix B.

A few notes to supplement the input data definitions:

a) The maximum element number (N or NW) is 20

b) If rotor rotational direction is negative, RAMDA should be negative. However, THETN and COL are positive nose up as in the FREEVI program.

c) The mode shapes used in the TILDYN program should be defined as normal modes. Those definitions appear in Eq. 4.7 for the blade and in Eq. 4.12 for the wing in Ref. 1. If modes are not normalized in this way, the calculation will give wrong answers.

d) Output mode shapes from the FREEVI program sometimes include unnecessary mode shapes; for example, the rigid body mode for the collective mode shapes of the gimballed rotor if the user uses the clamped boundary condition for the flapping motion and the hinged boundary condition for the lagging motion to derive the collective mode for autorotational flight.

e) If there are several cases, the data may consist of several data sets. The computer execution continues until it finds the absence of data.

4.2 Output Features

In the output, the identifying title is printed first, as punched in by the user. All input data are printed out below.

After the mode shape listing, the matrices A, B, C, and D of the equations of motion in Eq. 2.3 are listed. When the degrees of freedom of the equations are 18 or 19, the first 9 columns of the coefficient matrix are printed out, followed by the latter 9 or 10 columns of the matrix.

If the user has chosen the frequency response analysis (IRES=1), the results of that calculation appear next. Each response magnitude is showed corresponding to a nondimensional frequency. If IFRMAG is set equal to zero, the response magnitudes are based on the mode shapes normalized to rotor radius and wing semispan (refer to the explanation of IFRMAG in Section 4.1). In autorotation flight, the rigid-body rotation response is added to the basic form. It should be noted that the rigid-body rotation response is the rotational speed perturbation response, not the deflection response. Therefore, it is termed D(NUR)/DT (to express $\dot{\nu}_R$).

The eigenvalue analysis consists of the eigenvalue and eigenvector listing. All eigenvalues of the system are printed in the form of complex values with damping ratios, including pairs of complex conjugate values. The eigenvectors corresponding to the eigenvalue are printed. The maximum absolute values of the eigenvector components are normalized to unity. Real parts and imaginary parts express the phase angle between each eigenvector component. If IFRMAG is set to zero, the eigenvectors are expressed based on the mode shapes normalized by rotor radius and wing semispan (refer to the explanation of IFRMAG in Subsection 4.1) as in the frequency response. On the other hand, if IFRMAG is set to unity, all are in length, except ν_R (rigid-body rotation in autorotation flight). ν_R is an angle, in radians. Therefore, some attention should be paid to comparing the role of each eigenvector.

Only one built-in message is furnished for this program. If an error occurs in the eigenvalue analysis, the message below is automatically printed out after the title "EIGENVALUES":

IERROR=XXXXX

The error code is shown as follows:

Value of IERROR	Error Significance
I	The calculation of the Ith eigenvalue failed to converge. Eigenvalues I+1 ... N should be correct.
-I	The calculation of one or more eigenvectors, including the Ith, failed to converge. All eigenvalues and non-zero eigenvectors are correct.

4.3 Example Problems

Sample problems are carried out here for the Bell and Boeing tilt rotor wings. The flight condition is normal level flight cruising (around 200 kt) at sea level. The detail data is shown in Table 3. The input data listing in Appendix B includes the autorotation flight case for the Bell model and the powered flight case for the Boeing model. However, the output listing of only the Bell autorotation flight case is shown as an example in Appendix B.

REFERENCES

1. Yasue, M., "A Study of Gust Response for a Rotor-Propeller in Cruising Flight", ASRL TR 174-1, Massachusetts Institute of Technology, August 1974. Also NASA CR-137537.

2. Klema, V.K., Garbow, B.S., and Moler, C.B., "EISPACK, User's Information", Argonne National Laboratory, Applied Mathematics Division, June 1973.

3. "EISPACK", Applications Program Series, AP-42, Information Processing Center, Massachusetts Institute of Technology, September 14, 1973.

4. Mau, S., Pian, T.H.H., "Linear Dynamic Analyses of Laminated Plates and Shells by the Hybrid-Stress Finite-Element Method", ASRL TR 172-2, Massachusetts Institute of Technology or AMMRC CTR 73-40, Army Materials and Mechanics Research Center, October 1973.

5. Rutishauser, H., "Computational Aspects of F.L. Bauer's Simultaneous Iteration Method", Numerical Method, Vol. 13, 1969, pp 4-13.

6. Johnson, W., "Dynamics of Tilting Proprotor Aircraft in Cruising Flight", NASA TN-D 3677, May 1974.

TABLE 1 BOUNDARY CONDITIONS AND OUTPUT DEFLECTIONS OF THE ROTOR AND WING IN THE FREEVI PROGRAM

	BOUNDARY CONDITIONS		
	ROOT	TIP	OUTPUT DEFLECTIONS
Wing	Clamped for all Deflections	Lumped Mass with Mass and Mass Moment of Inertia	Vertical Bending Chordwise Bending Torsion
Blade			
Powered			
Hingeless Rotor	Clamped for all Deflections	Free or Tip Mass if Necessary	Out-of-Plane Bending Inplane Bending
Gimballed Rotor			
Collective Mode	Clamped for all Deflections	"	"
Cyclic Mode	Hinged for Flapping, Clamped for Lagging	"	"
Autorotation			
Hingeless Rotor			
Collective Mode	Clamped for Flapping, Hinged for Lagging	"	"
Cyclic Mode	Clamped for all Deflections	"	"
Gimballed Rotor			
Collective Mode	Clamped for Flapping, Hinged for Lagging	"	"
Cyclic Mode	Hinged for Flapping, Clamped for Lagging	"	"

TABLE 2 DESCRIPTION OF VARIABLES

(a) Description of {x} in Eq. 2.3

Total Degrees of Freedom				Description
Powered Flight		Autorotation Flight		
9 DOF	18DOF	10DOF	19DOF	
Q_{10}	Q_{10}	Q_{10}	Q_{10}	Blade Collective Motion of 1st Natural Frequency
Q_{1c}	Q_{1c}	Q_{1c}	Q_{1c}	Blade Cyclic Cosine Motion of 1st Natural Frequency
Q_{1s}	Q_{1s}	Q_{1s}	Q_{1s}	Blade Cyclic Sine Motion of 1st Natural Frequency
Q_{20}	Q_{20}	Q_{20}	Q_{20}	Blade Collective Motion of 2nd Natural Frequency
Q_{2c}	Q_{2c}	Q_{2c}	Q_{2c}	Blade Cyclic Cosine Motion of 2nd Natural Frequency
Q_{2s}	Q_{2s}	Q_{2s}	Q_{2s}	Blade Cyclic Sine Motion of 2nd Natural Frequency
	Q_{30}		Q_{30}	Blade Collective Motion of 3rd Natural Frequency
	Q_{3c}		Q_{3c}	Blade Cyclic Cosine Motion of 3rd Natural Frequency
	Q_{3s}		Q_{3s}	Blade Cyclic Sine Motion of 3rd Natural Frequency
	Q_{40}		Q_{40}	Blade Collective Motion of 4th Natural Frequency
	Q_{4c}		Q_{4c}	Blade Cyclic Cosine Motion of 4th Natural Frequency
	Q_{4s}		Q_{4s}	Blade Cyclic Sine Motion of 4th Natural Frequency
a_1	a_1	a_1	a_1	Wing Motion of 1st Natural Frequency
a_2	a_2	a_2	a_2	Wing Motion of 2nd Natural Frequency
a_3	a_3	a_3	a_3	Wing Motion of 3rd Natural Frequency
	a_4		a_4	Wing Motion of 4th Natural Frequency
	a_5		a_5	Wing Motion of 5th Natural Frequency
	a_6		a_6	Wing Motion of 6th Natural Frequency
		ν_R	ν_R	Rotor Rigid-Body Rotation

TABLE 2 CONCLUDED

(b) Description of {e} in Eq. 2.3

Symbol	Description
u_G/V	Nondimensional Vertical Gust
v_G/V	Nondimensional Lateral Gust
w_G/V	Nondimensional Longitudinal Gust
θ_o	Collective Pitch Control
θ_{1c}	Lateral Cyclic Pitch Control
θ_{1s}	Longitudinal Cyclic Pitch Control

TABLE 3

DESCRIPTION OF THE BELL AND THE BOEING PROPROTOR DESIGNS IN POWERED FLIGHT CONSIDERED IN THIS REPORT

	BELL	BOEING
ROTOR		
Type	gimballed, stiff inplane	cantilever, soft inplane
Number of blades, N	3	3
Radius, R	156 in.	150 in.
Chord, C_B	18.9 in.	14 in.
Lock number, γ	3.83	4.04
Solidity, σ	0.089	0.115
Pitch/flap coupling, δ_3	-15 deg.	0
Collective pitch, θ_D	1.25 deg.	1.0 deg.
Lift-curve slope, a	5.7	5.7
Drag Coefficient, C_{Do}	0.0065	0.0065
Rotor rotation direction, $\overline{\Omega}$	+1	-1
Inflow ratio,	0.7	-0.7
Rotational speed, $\lvert\Omega\rvert$	458 RPM 48.9 rad/sec	386 RPM 40.4 rad/sec
Blade Natural Frequencies		
first, $\lambda_1/\lvert\Omega\rvert$	1.02/rev (7.78Hz)	0.827/rev. (5.32Hz)
second, $\lambda_2/\lvert\Omega\rvert$	1.34/rev (10.2Hz)	1.32/rev (8.49Hz)
third, $\lambda_3/\lvert\Omega\rvert$	4.35/rev (33.2Hz)	3.40/rev (21.9Hz)

TABLE 3 CONTINUED

ROTOR(cont'd)	BELL	BOEING
fourth, $\lambda_4/\lvert\Omega\rvert$	10.1/rev(77.1Hz)	6.77/rev(43.5Hz)
Collective Natural Frequency		
first, $\lambda_1^{(o)}/\lvert\Omega\rvert$	1.31/rev (10.0Hz)	
second, $\lambda_2^{(o)}/\lvert\Omega\rvert$	2.12/rev (16.2Hz)	
third, $\lambda_3^{(o)}/\lvert\Omega\rvert$	4.93/rev (37.7Hz)	
fourth $\lambda_4^{(o)}/\lvert\Omega\rvert$	10.6/rev (80.9Hz)	
Blade flapping inertia, I_B	105 slug-ft^2	150 slug-ft^2
One blade weight, M_B	133 lb	124 lb
WING		
Semispan, L	200 in.	200 in.
Chord, c_W	62.2 in.	62.2 in.
Mast height, h	51.3 in.	51.3 in.
Sweep	0	0
Dihedral	0	0
Lift-curve slope, a_W	5.7	5.7
Drag coefficient, C_{Dow}	0.004	0.004
Moment coefficient C_{mo}	-0.005	-0.005
Aerodynamic center, $\bar{e} = x_{A_W}/C_W$	0.01	0.01
Angle of attack, α_{WO}	2.0 deg	2.0 deg

TABLE 3. CONCLUDED

	BELL	BOEING
WING (cont'd)		
Natural Frequencies		
first, $\Lambda_1/\lvert\Omega\rvert$	0.347/rev(2.65Hz)	0.365/rev(2.35Hz)
second, $\Lambda_2/\lvert\Omega\rvert$	0.622/rev(4.75Hz)	0.653/rev(4.20Hz)
third, $\Lambda_3/\lvert\Omega\rvert$	1.09/rev(8.32Hz)	1.11/rev(7.14Hz)
fourth, $\Lambda_4/\lvert\Omega\rvert$	2.37/rev(18.1Hz)	2.47/rev(15.9Hz)
fifth, $\Lambda_5/\lvert\Omega\rvert$	3.76/rev(28.7Hz)	3.95/rev(25.4Hz)
sixth, $\Lambda_6/\lvert\Omega\rvert$	10.6/rev(80.9Hz)	12.5/rev(80.4Hz)
PYLON		
Weight, M_p	1420 lb	2000 lb
Yaw inertia, I_{py}	164.8 slug-ft^2	250.0 slug-ft^2
Pitch inertia, I_{pp}	190.0 slug-ft^2	250.0 slug-ft^2
Roll inertia, I_{pr}	42.4 slug-ft^2	30.0 slug-ft^2
FLIGHT CONDITION FOR CALCULATIONS, $\lambda = 0.7$		
Cruising speed, V	250 kt	218 kt
Cruising altitude	sea level	sea level

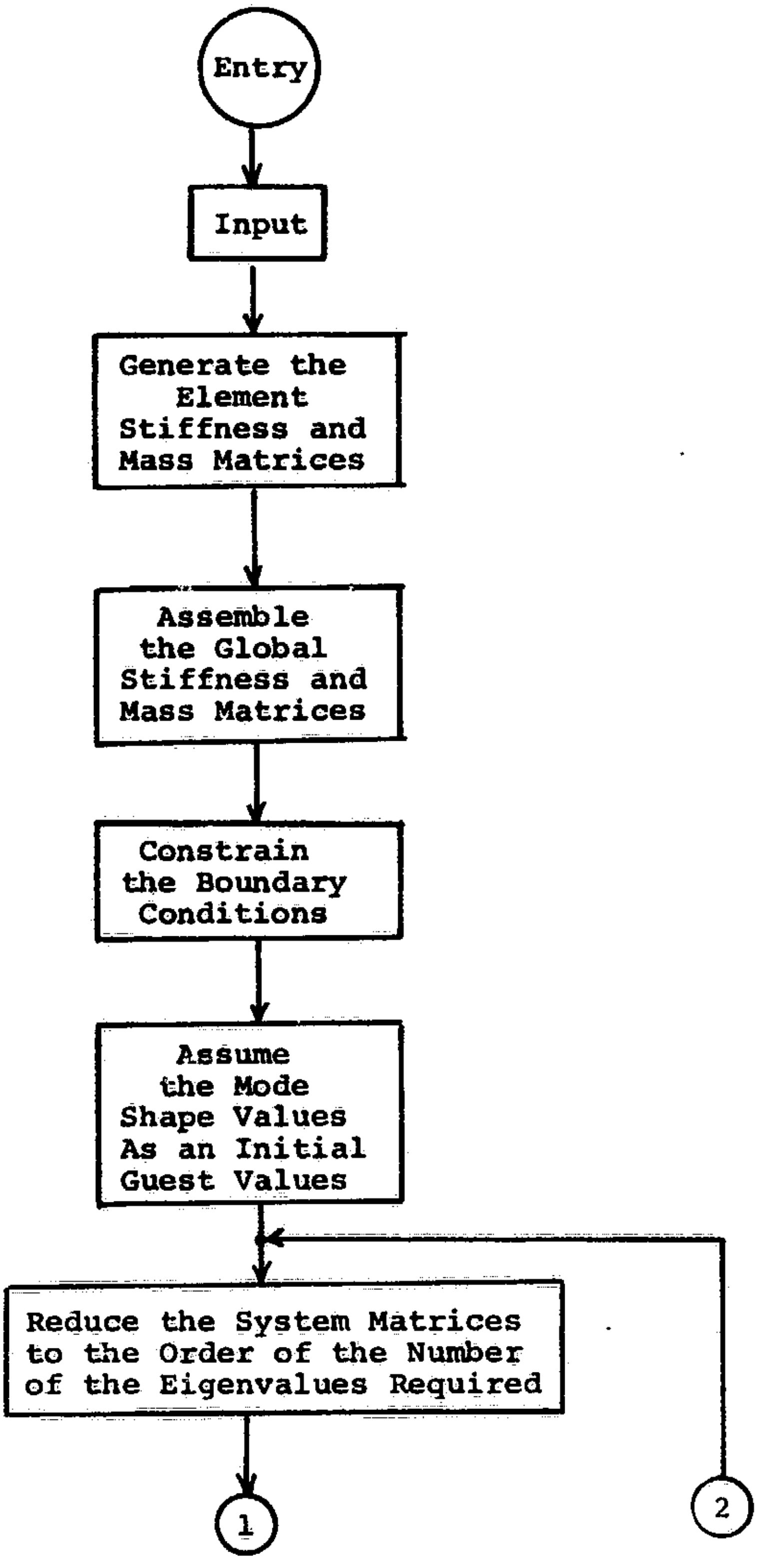


FIG. 1 FLOW CHART OF FREEVI PROGRAM FOR BLADE AND WING OF NATURAL FREQUENCIES AND MODE SHAPES

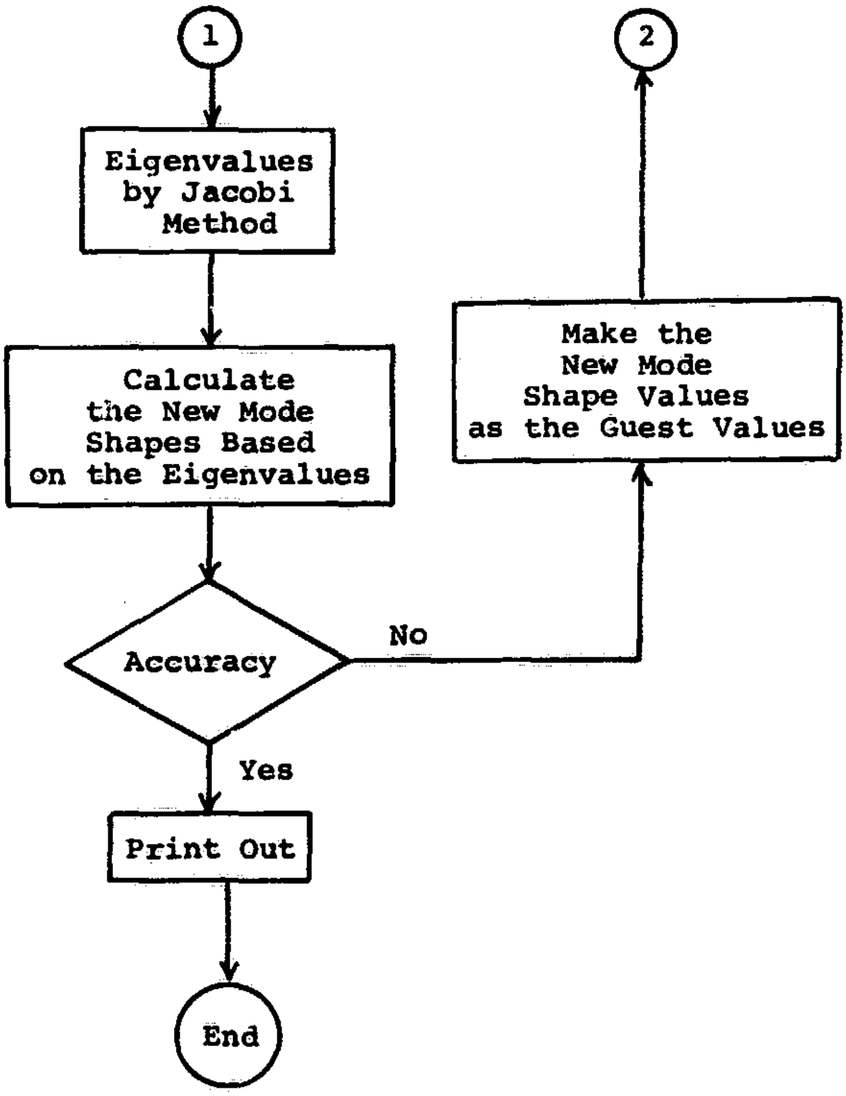


FIG. 1 CONCLUDED

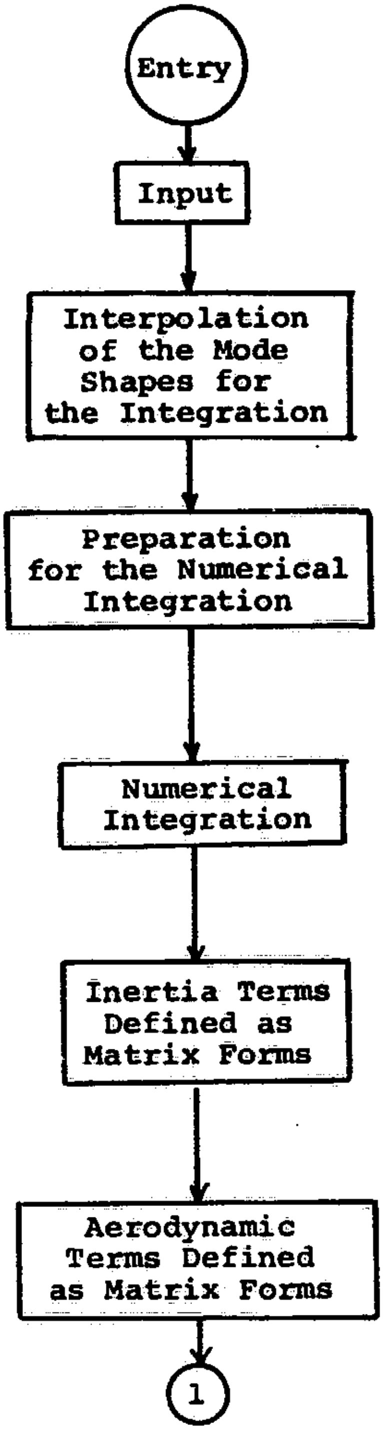


FIG. 2 FLOW CHART OF TILDYN PROGRAM FOR ANALYSIS OF TILT ROTOR AIRCRAFT DYNAMICS

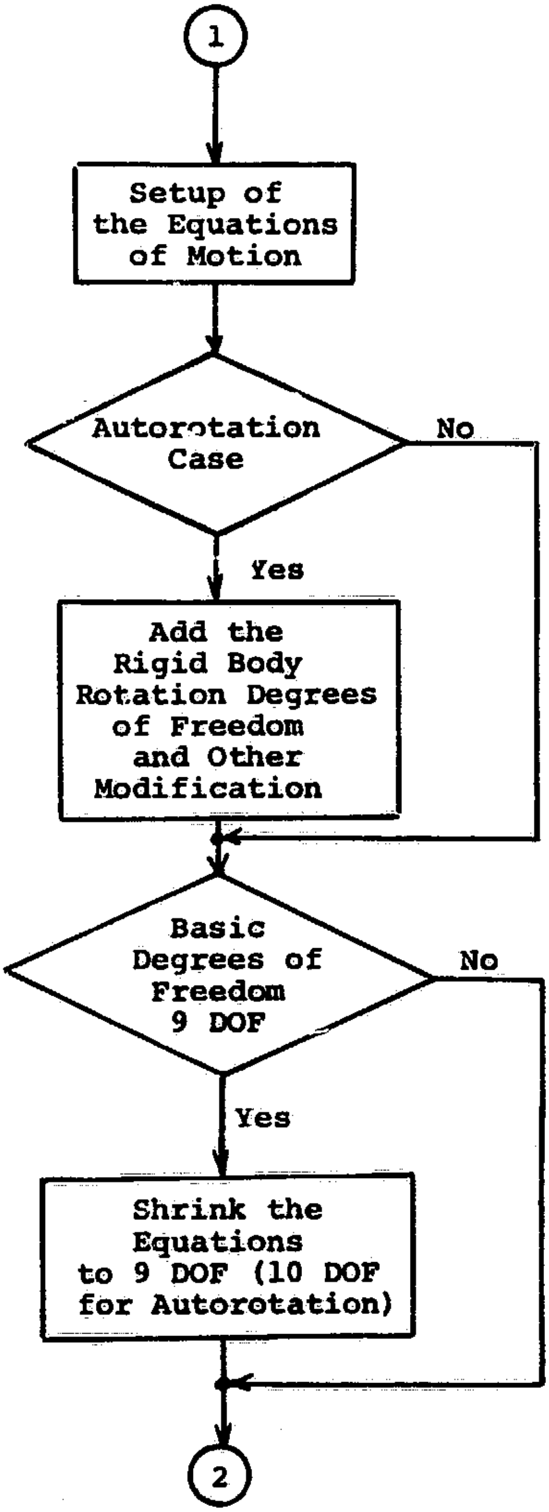


FIG. 2 CONTINUED

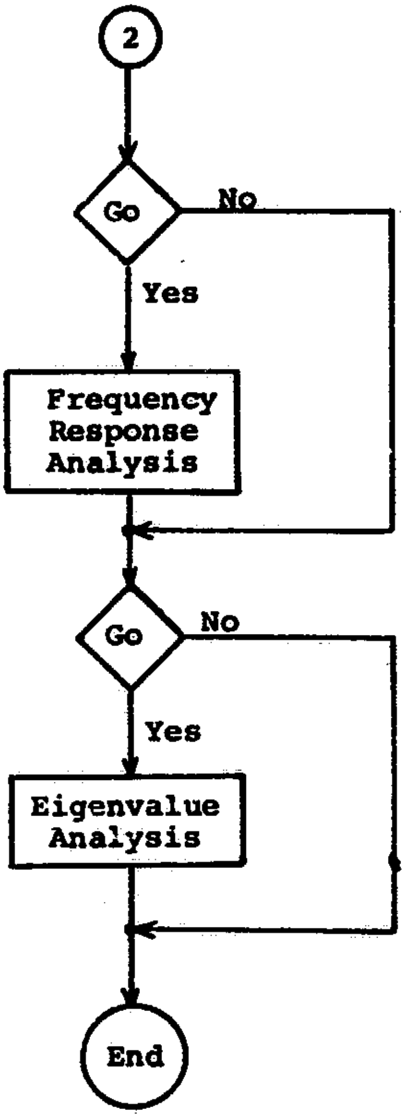


FIG. 2 CONCLUDED

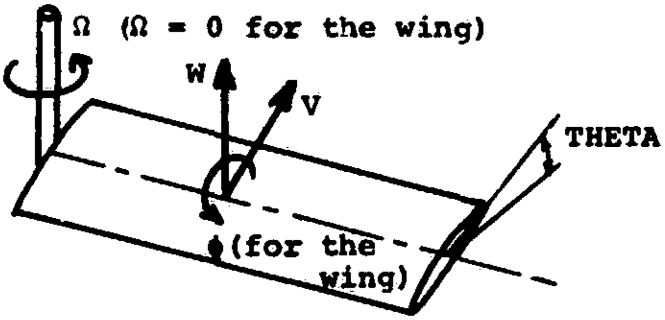


FIG. 3 COORDINATE SYSTEM FOR FREEVI PROGRAM

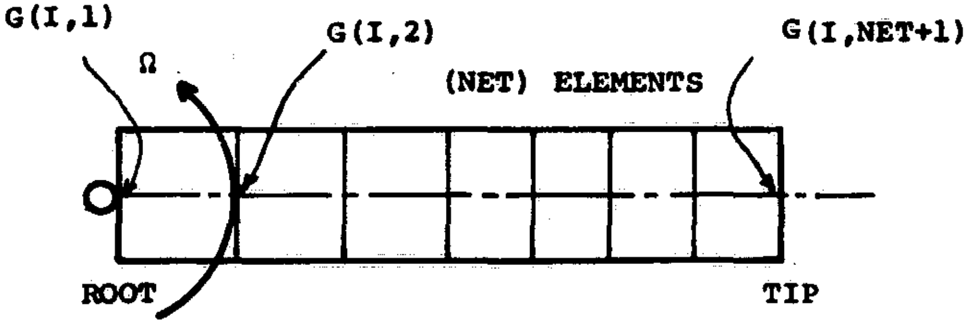


FIG. 4 FINITE ELEMENT REPRESENTATION WITH BEAM ELEMENTS

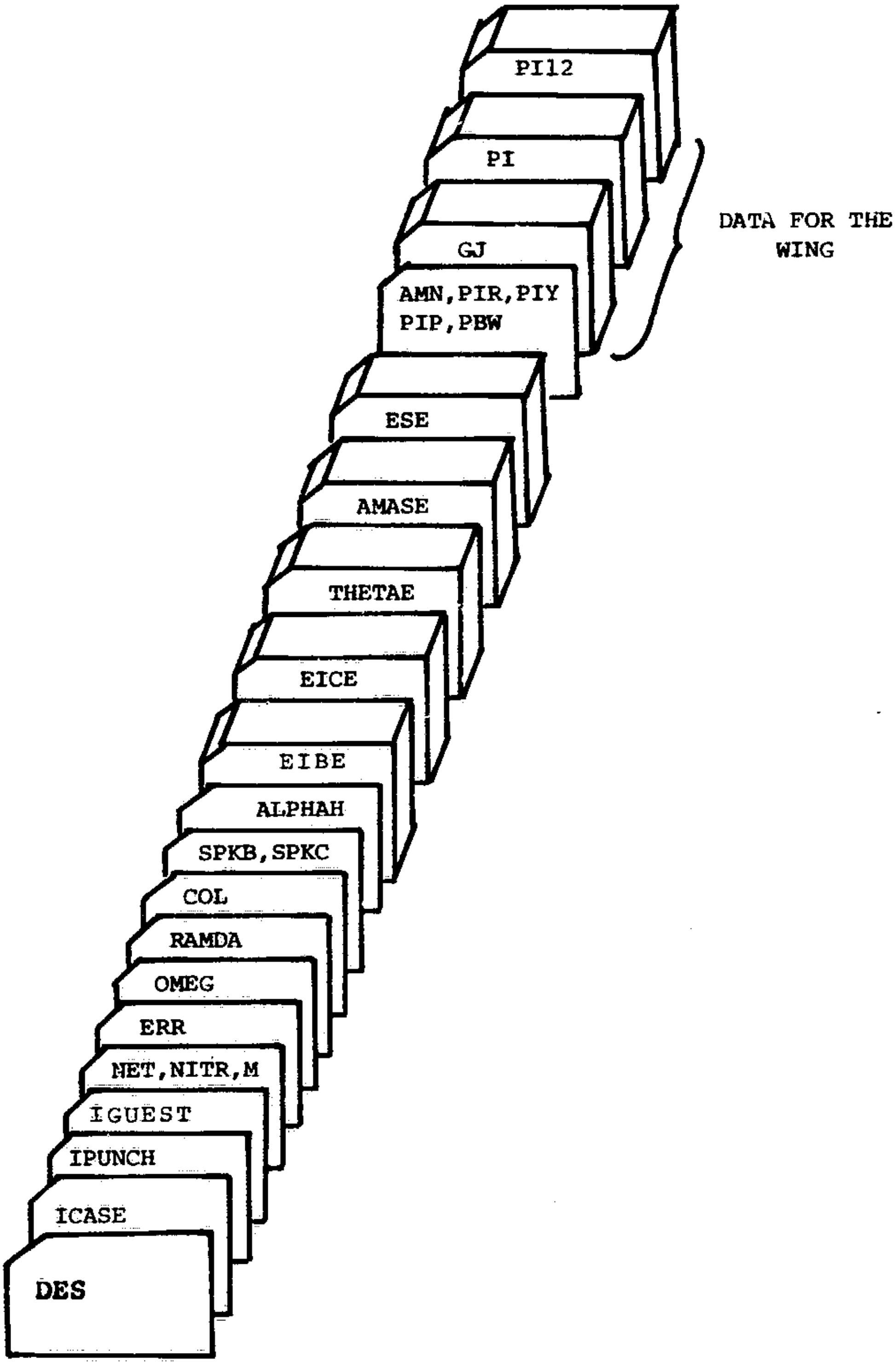
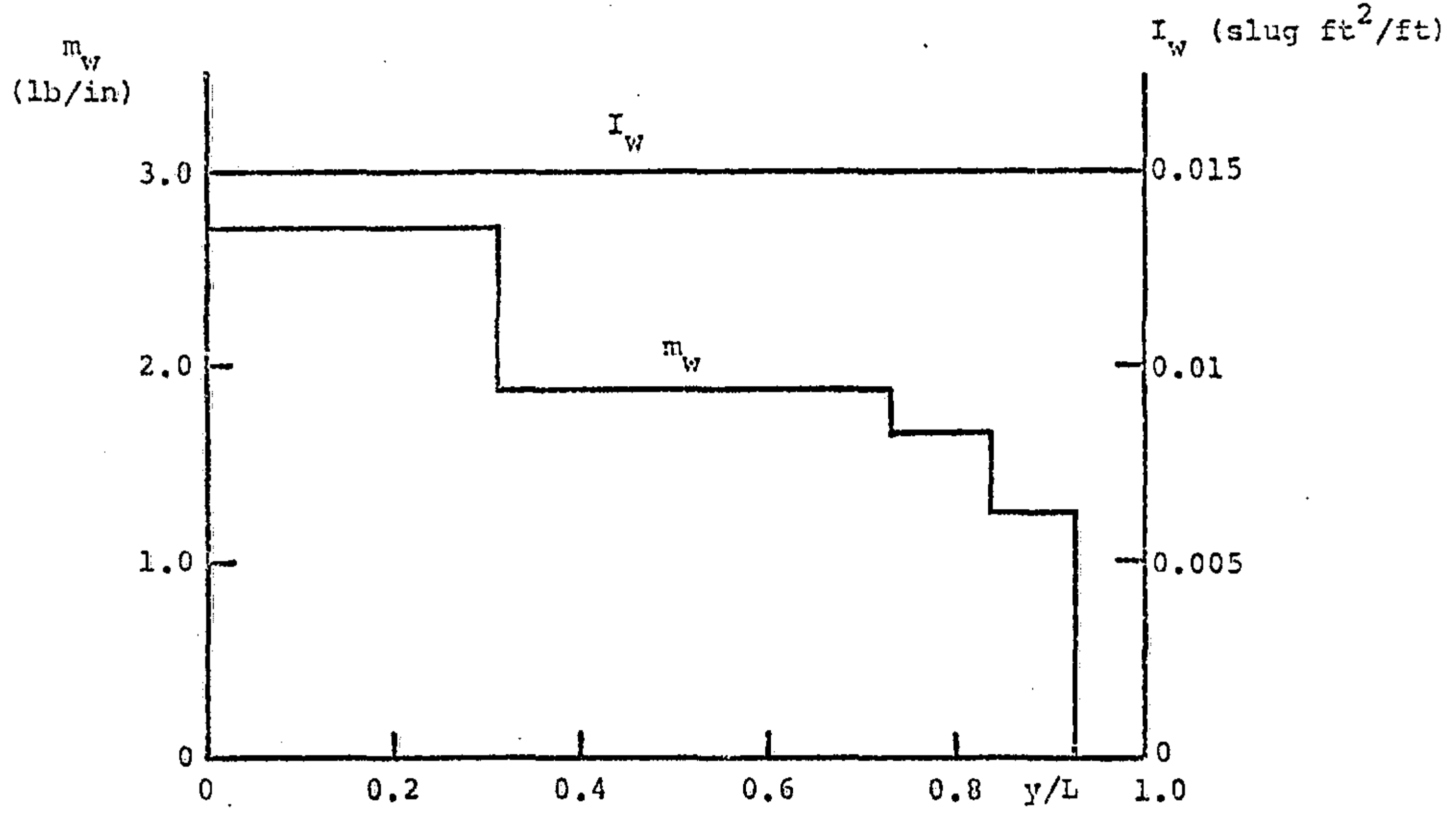
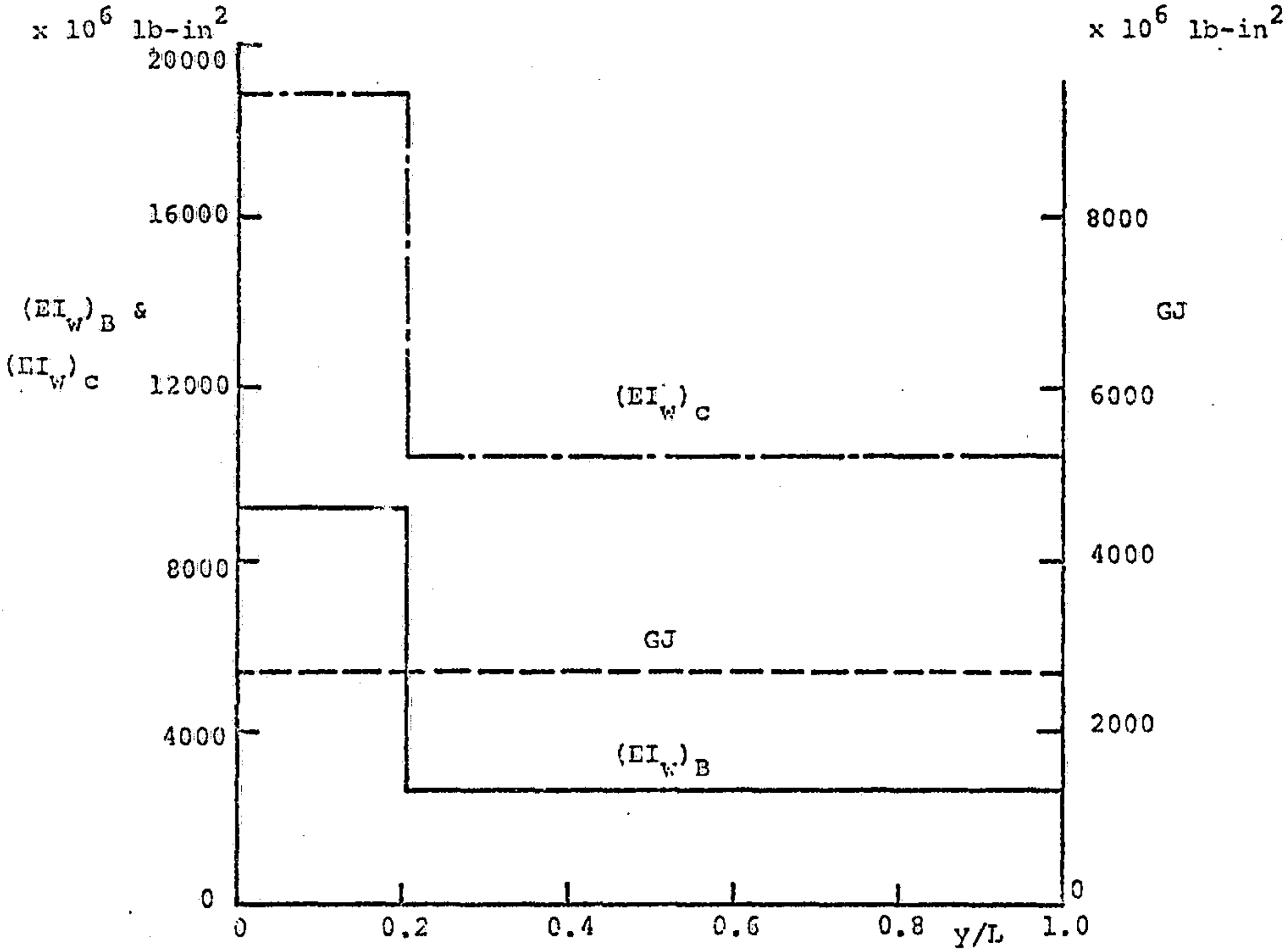


FIG. 5 DATA DECK SETUP FOR THE FREEVI PROGRAM



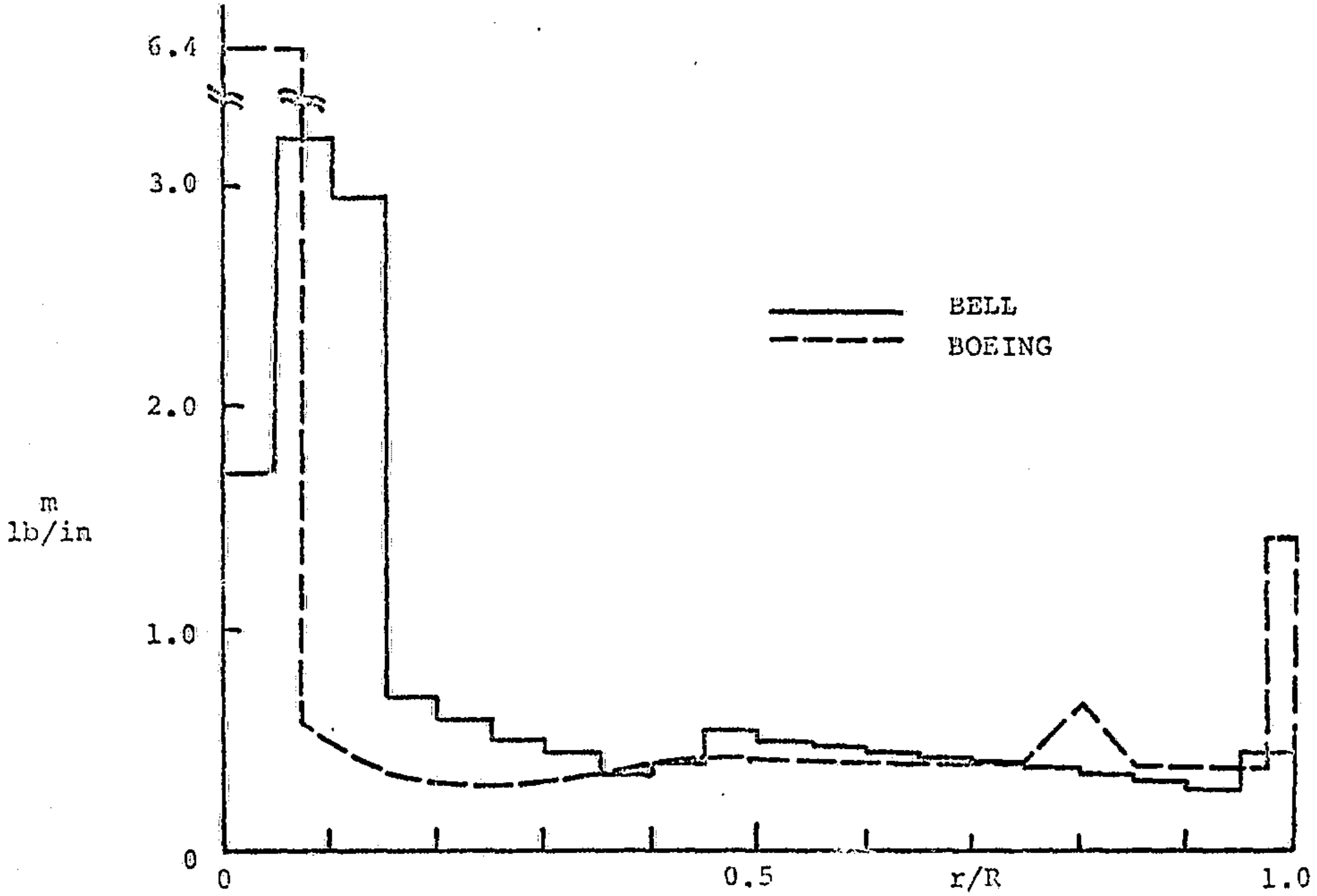
(a) Mass and Cross-Sectional Moment of Inertia Distribution

FIG. 6 STRUCTURAL CHARACTERISTICS OF THE WING



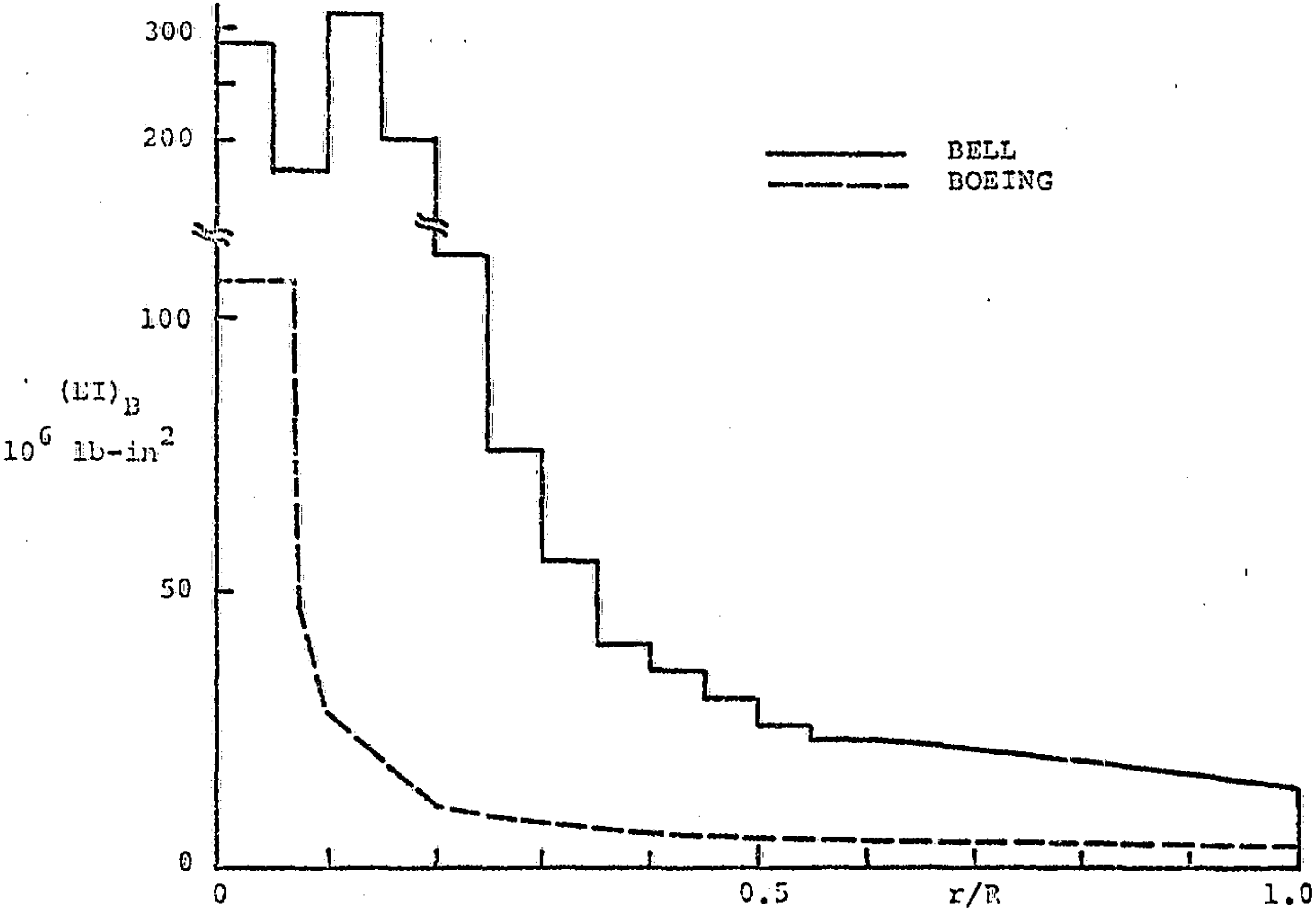
(b) Stiffness Distribution: Vertical Bending Stiffness $(EI_W)_B$, Chordwise Bending Stiffness $(EI_W)_C$, and Torsional Rigidity GJ

FIG. 6 CONCLUDED



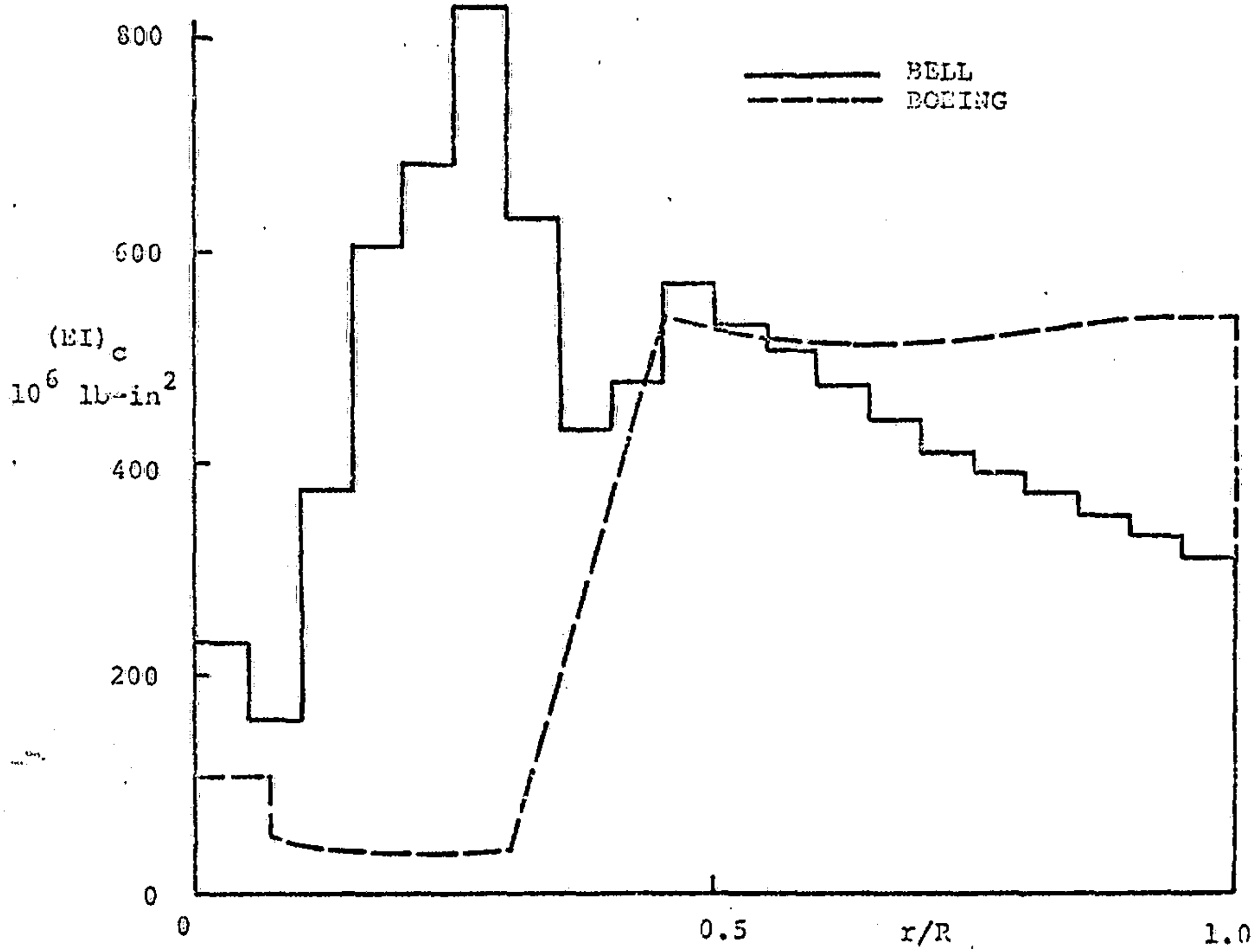
(a) Section Mass Distribution

FIG. 7 STRUCTURAL CHARACTERISTICS OF TWO PROPROTOR BLADES



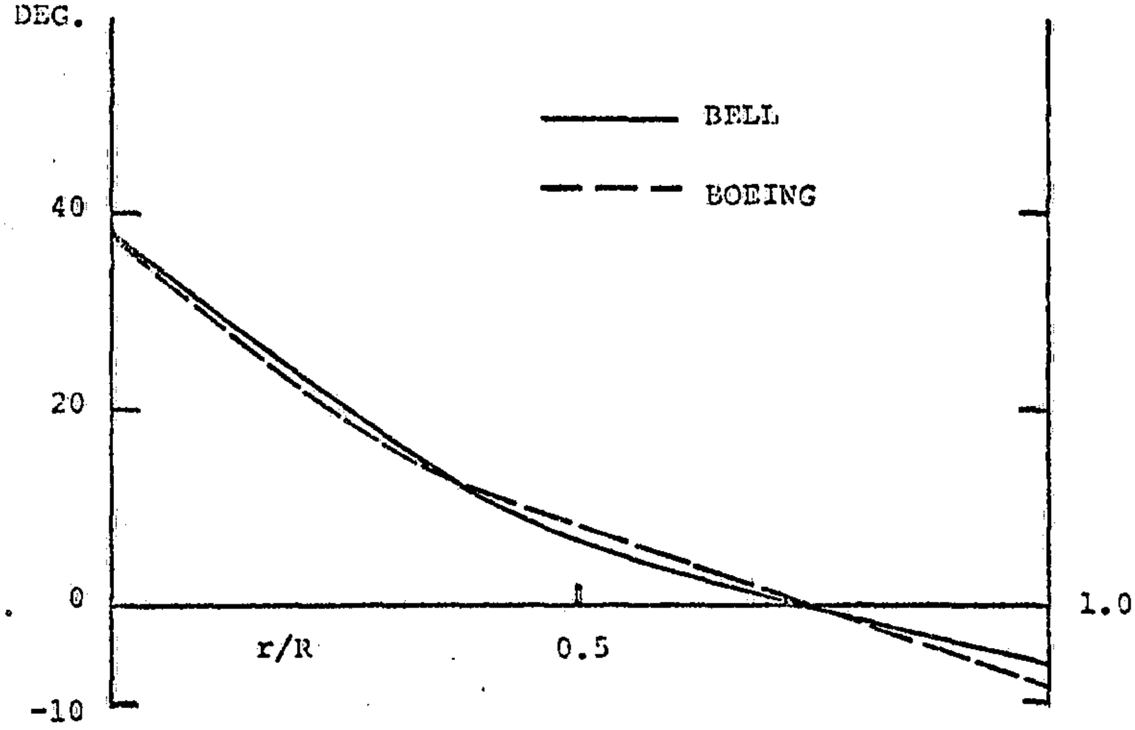
(b) Section Flapwise Bending Stiffness Distribution

FIG. 7 CONTINUED



(c) Section Chordwise Bending Stiffness Distribution

FIG. 7 CONTINUED



(d) Angle of Twist

FIG. 7 CONCLUDED

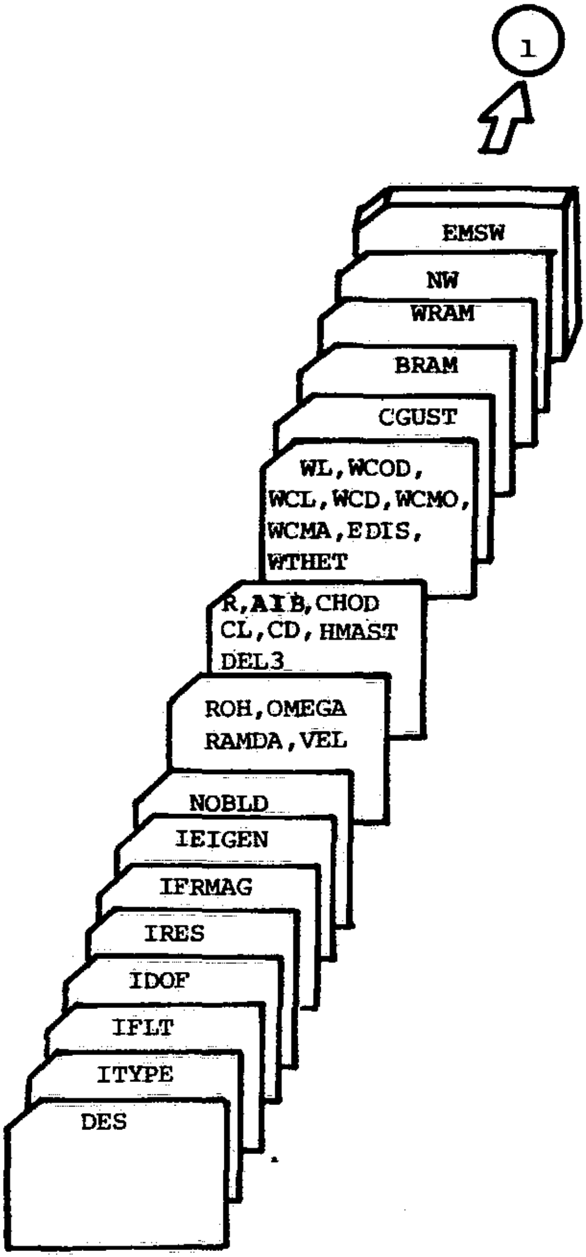


FIG. 8 DATA DECK SETUP FOR THE TILDYN PROGRAM

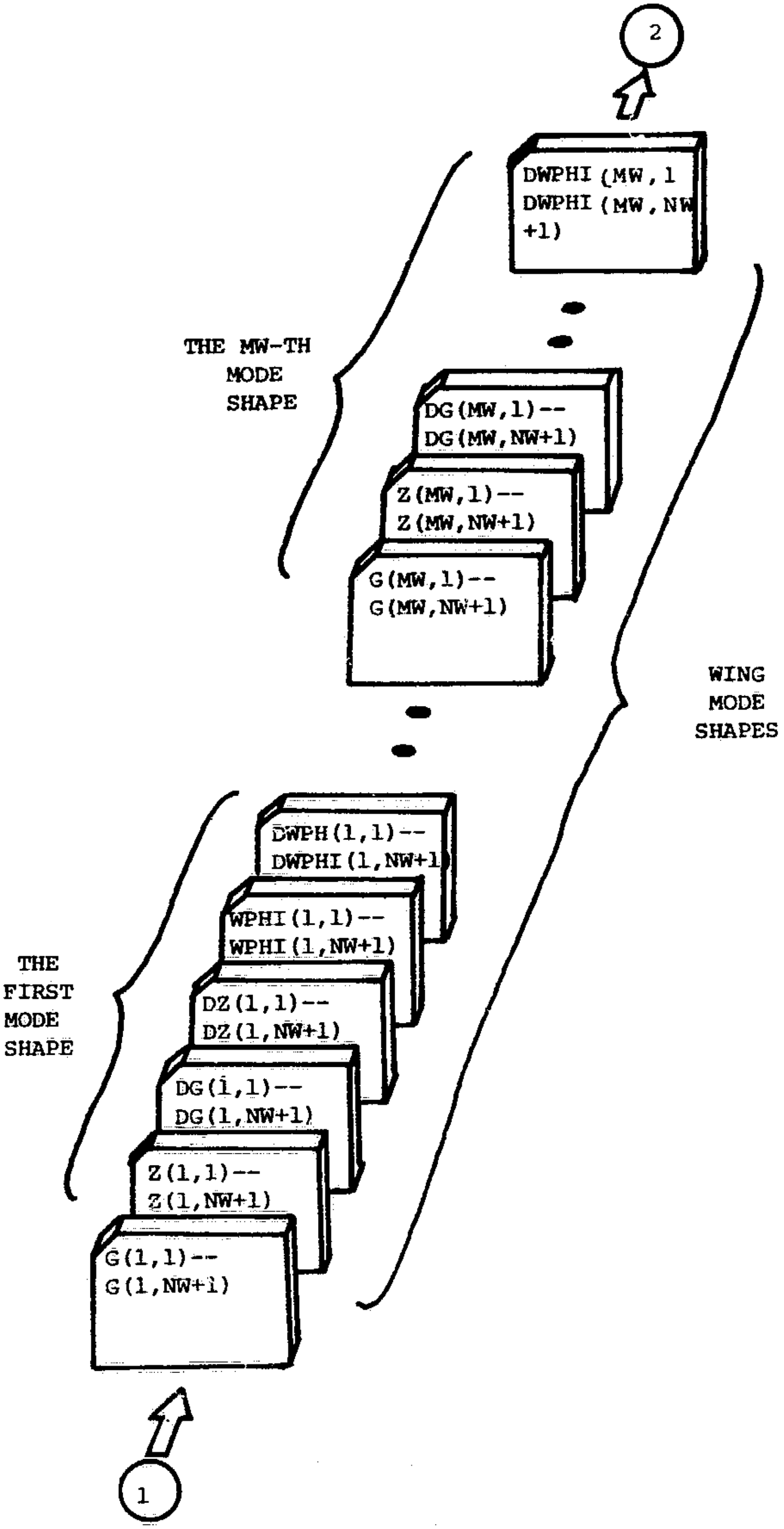


FIG. 8 CONTINUED

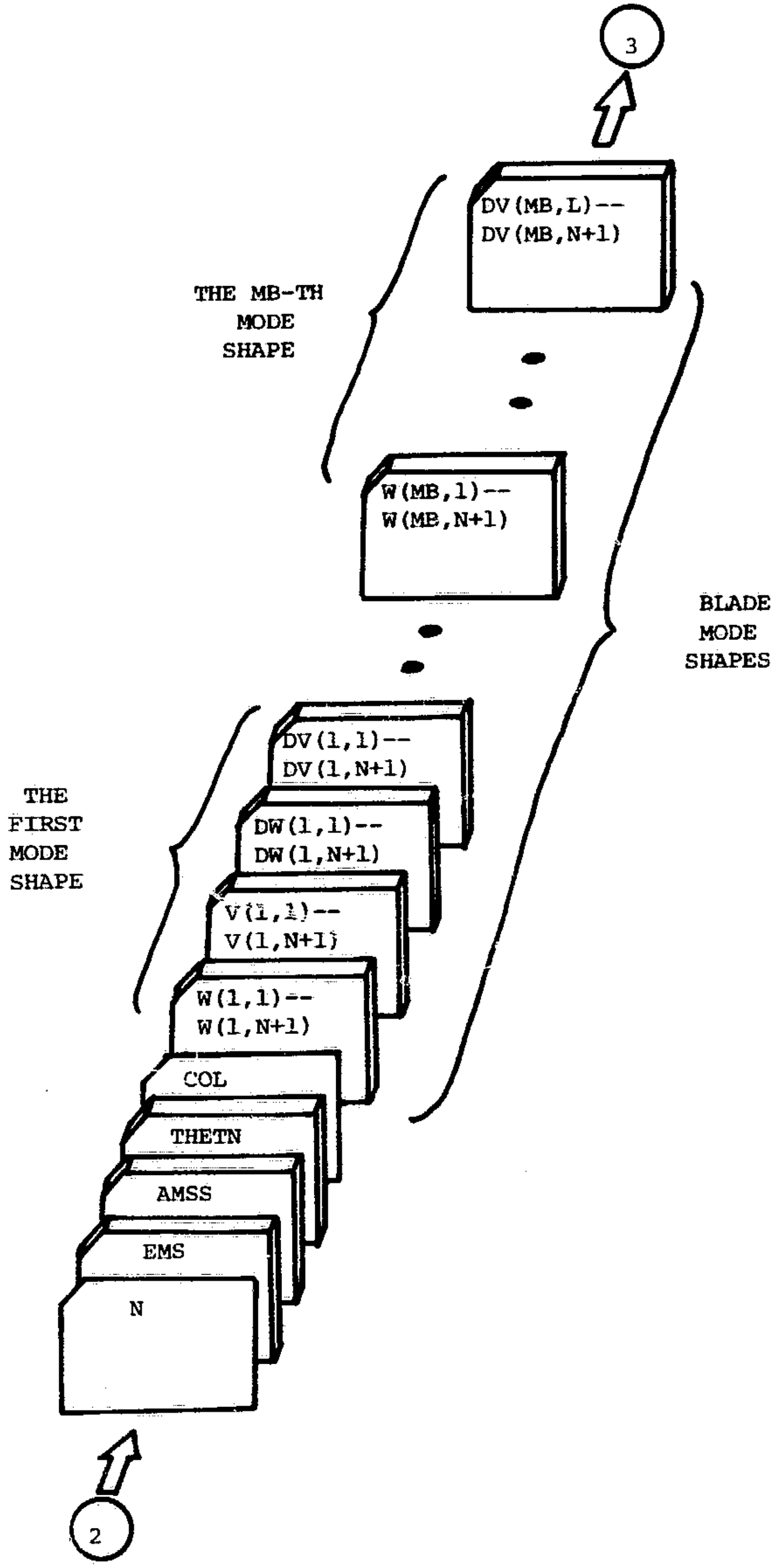
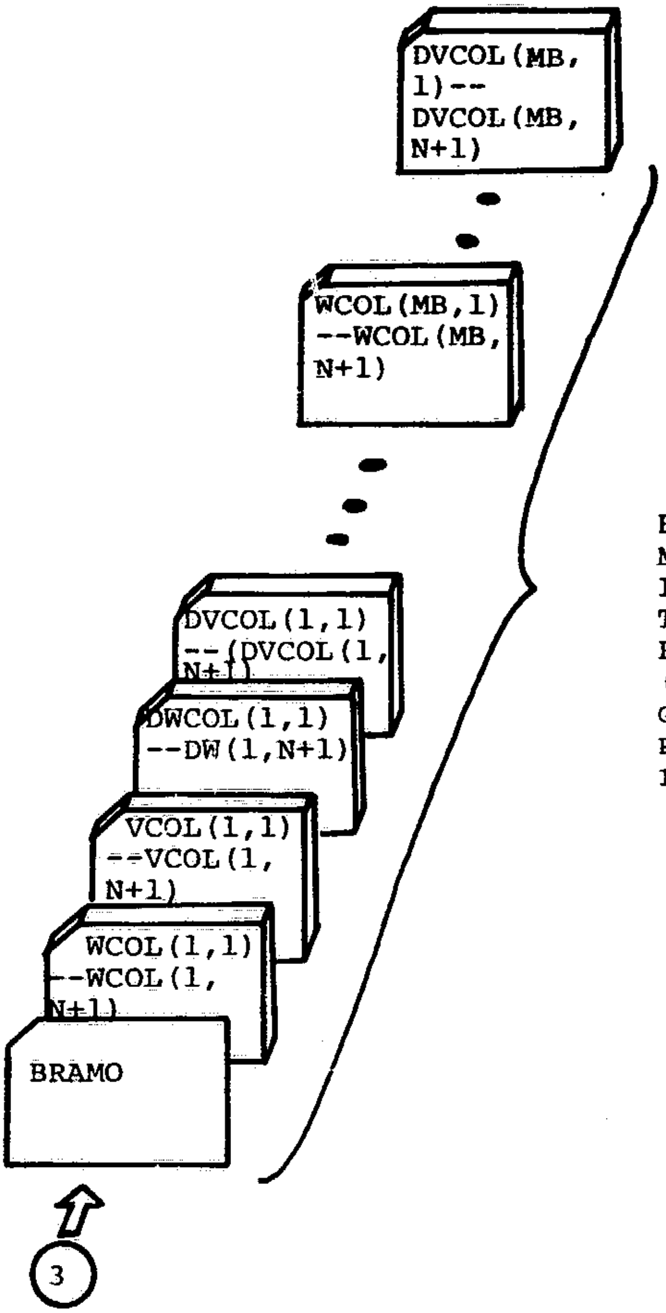


FIG. 8 CONTINUED



BLADE COLLECTIVE
MODE SHAPE DATA
IS FURNISHED IF
THE CASE IS AUTO-
ROTATION FLIGHT
(IFLT=1, ITYPE=1) OR
GIMBALLED ROTOR IN
POWERED FLIGHT (IFLT=0,
ITYPE=1)

FIG. 8 CONCLUDED

APPENDIX A

PROGRAM LISTING

A.1 The FREEVI Program Listing

```
C
C   *********************************************************
C
C             PROGRAM  ROTOR
C                  PART 1 ;   PROGRAM  FREEVI
C
C   *********************************************************
C
C      PURPOSE
C        TO OBTAIN THE NATURAL FREQUENCIES AND MODE SHAPES
C        OF THE ROTOR BLADE AND WING OF THE TILT-ROTOR AIRCRAFT
C
C        DEVELOPED BY MASAHIRO YASUE
C             OF AEROELASTIC AND STRUCTURES RESEARCH LABORATORY
C                  AUGUST 1974
C             ADDRESS ;  BLG 41-211
C                        MASSACHUSETTS INSTITUTE OF TECHNOLOGY
C                        CAMBRIDGE, MASS.  02139
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION NDPE(20), NNCDE(252), NOD(126), ICOL(126),INUM(126)
      DIMENSION   STK(1550), STM(1550),  EK(78),EM(78),XLR(1200),X(1200)
      DIMENSION Y(1200), U(126),EIG(9),LCH(9), RM(81), RK(81), RV(81)
      DIMENSION SRM(81),SRK(81)
 1    CALL TEIGEN(NDPE,                NOD,NNODE,ICOL,INUM,STK,STM,EK,EM,
     1    XLR,U,EIG,LCH,RM,RK,RV,SRM,SRK,X,Y)
      GO TO 1
      END
```

```
      SUBROUTINE TEIGEN(NDPE,                NOD,NNODE,ICOL,INUM,STK,STM,EK        TEIG0001
     @,EM,    XLR,U,EIG,LCH,RM,RK,RV,SRM,SRK,X,Y)                                  TEIG0002
C                                                                                  TEIG0003
C        TO CALCULATE THE NORMAL MODES AND FREQUENCIES                             TEIG0004
C                                                                                  TEIG0005
      IMPLICIT REAL*8(A-H,O-Z)                                                     TEIG0006
      COMMON /BW/       ICASE    ,   IGUEST                                        TEIG0007
      COMMON/HELP/ALPHAH                                                           TEIG0008
      DIMENSION NDPE(1),             NBCU(1),NOD(1),NNODE(1),ICOL(1)               TEIG0009
      DIMENSION INUM(1),STK(1),STM(1),EK(1),EM(1),      XLR(1)                     TEIG0010
      DIMENSION U(1),EIG(1),LCH(1)                                                 TEIG0011
      DIMENSION RM(1),RK(1),RV(1),SRK(1),SRM(1),X(1)                               TEIG0012
      DIMENSION Y(1)                                                               TEIG0013
      DIMENSION SQU(20),CYC(20)                                                    TEIG0014
      CALL INPUT(IEQ,NDPE,NET,NDT,MNC,NNODE,NBU,   INUM,ERR,NITR,M)                TEIG0015
      CALL MESH(NDPE,NET,NDT,NCDT,MNC,MN,      NOD,NNODE,ICOL,INUM,NBU             TEIG0016
     @  ,INDEX)                                                                    TEIG0017
      IF(INDEX .EQ. 0) GO TO 99                                                    TEIG0018
      CALL ASBV(STK,STM,IEQ,EK,EM,   NDPE,NDT,NCDT,NNODE,MN,NET,INUM)              TEIG0019
      MA=M                                                                         TEIG0020
      N=NCDT                                                                       TEIG0021
      NM=NCDT*M                                                                    TEIG0022
      MM=M*M                                                                       TEIG0023
      CALL FAC(STK,N,NNDG,ICOL,INUM,U)                                             TEIG0024
      IF(NNDG .LT. 0) GO TO 99                                                     TEIG0025
      DO 3 I=1,M                                                                   TEIG0026
    3 EIG(I)= .0                                                                   TEIG0027
      DO 74 K=1,M                                                                  TEIG0028
      II=(K-1)*N                                                                   TEIG0029
      IY=K*2-1                                                                     TEIG0030
      DO 74 I=1,N                                                                  TEIG0031
      IX=IY                                                                        TEIG0032
      IY=IX*65539                                                                  TEIG0033
      IF(IY) 75,76,76                                                              TEIG0034
   75 IY=IY+2147483647+1                                                           TEIG0035
   76 YFL=IY                                                                       TEIG0036
```

```
   74 XLR(II+I)=YFL*0.4656613D-9-0.5D+00                                TEIG0037
      IF(IGUEST.EQ.0) GO TO 200                                         TEIG0038
      GO TO (210,220,230,240,220),ICASE                                 TEIG0039
  210 GO TO (310,320,330),IGUEST                                        TEIG0040
C-----W ONLY                                                            TEIG0041
  310  IX=N/6                                                           TEIG0042
      DO 301 K=1,M                                                      TEIG0043
      II=NCDT*K-NCDT                                                    TEIG0044
      XLR(II+1)=0.0D+00                                                 TEIG0045
      DO 301 KK=1,IX                                                    TEIG0046
      JJ=KK*6-5                                                         TEIG0047
      XLR(II+JJ+2)=0.0D+00                                              TEIG0048
      XLR(II+JJ+4)=0.0D+00                                              TEIG0049
      XLR(II+JJ+5)=0.0D+00                                              TEIG0050
  301 XLR(II+JJ+6)=0.0D+00                                              TEIG0051
      GO TO 200                                                         TEIG0052
C-----V ONLY                                                            TEIG0053
  320 IX=N/6                                                            TEIG0054
      DO 302 K=1,M                                                      TEIG0055
      II=NCDT*K-NCDT                                                    TEIG0056
      XLR(II+1)=0.0D+00                                                 TEIG0057
      DO 302 KK=1,IX                                                    TEIG0058
      JJ=KK*6-5                                                         TEIG0059
      XLR(II+JJ+1)=0.0D+00                                              TEIG0060
      XLR(II+JJ+3)=0.0D+00                                              TEIG0061
      XLR(II+JJ+5)=0.0D+00                                              TEIG0062
  302 XLR(II+JJ+6)=0.0D+00                                              TEIG0063
      GO TO 200                                                         TEIG0064
C-----PHI ONLY                                                          TEIG0065
  330 IX=N/6                                                            TEIG0066
      DO 303 K=1,M                                                      TEIG0067
      II=NCDT*K-NCDT                                                    TEIG0068
      DO 303 KK=1,IX                                                    TEIG0069
      JJ=KK*6-5                                                         TEIG0070
      DO 303 KKK=1,4                                                    TEIG0071
  303 XLR(II+JJ+KKK)=0.0D+00                                            TEIG0072
```

```
      GO TO 200                                                         TEIG0073
  220 GO TO (221,222),IGUEST                                            TEIG0074
C-----W ONLY                                                            TEIG0075
  221 IX=N/2                                                            TEIG0076
      DO 201 K=1,M                                                      TEIG0077
      II=N*K-N                                                          TEIG0078
      DO 201 I=1,IX                                                     TEIG0079
  201 XLR(II+2*I)=0.0D0                                                 TEIG0080
      GO TO 200                                                         TEIG0081
C-----V ONLY                                                            TEIG0082
  222 IX=N/2                                                            TEIG0083
      DO 202 K=1,M                                                      TEIG0084
      II=K*N-N                                                          TEIG0085
      DO 202 I=1,IX                                                     TEIG0086
  202 XLR(II+2*I-1)=0.0D0                                               TEIG0087
      GO TO 200                                                         TEIG0088
  230 GO TO (231,232),IGUEST                                            TEIG0089
C-----W ONLY                                                            TEIG0090
  231 IX=N/2                                                            TEIG0091
      DO 233 K=1,M                                                      TEIG0092
      II=N*K-N                                                          TEIG0093
      XLR(II+1)=0.0D0                                                   TEIG0094
      DO 233 KK=1,IX                                                    TEIG0095
  233 XLR(II+2*KK+1)=0.000                                              TEIG0096
      GO TO 200                                                         TEIG0097
C-----V ONLY                                                            TEIG0098
  232 IX=N/2                                                            TEIG0099
      DO 234 K=1,M                                                      TEIG0100
      II=N*K-N                                                          TEIG0101
      DO 234 KK=1,IX                                                    TEIG0102
  234 XLR(II+2*KK)=0.0D0                                                TEIG0103
      GO TO 200                                                         TEIG0104
  240 GO TO (241,242),IGUEST                                            TEIG0105
C-----W ONLY                                                            TEIG0106
  241 IX=N/2                                                            TEIG0107
      DO 243 K=1,M                                                      TEIG0108
```

```
      II=N*K-N                                                          TEIG0109
      DO 243 KK=1,IX                                                    TEIG0110
  243 XLR(II+2*KK+1)=0.0D0                                              TEIG0111
      GO TO 200                                                         TEIG0112
C-----V ONLY                                                            TEIG0113
  242 IX=N/2                                                            TEIG0114
      DO 244 K=1,M                                                      TEIG0115
      II=N*K-N                                                          TEIG0116
      XLR(II+1)=0.0D0                                                   TEIG0117
      DO 244 KK=1,IX                                                    TEIG0118
  244 XLR(II+2*KK)=0.0D0                                                TEIG0119
  200 CONTINUE                                                          TEIG0120
      IST=1                                                             TEIG0121
      DO 21 KKK=1,NITR                                                  TEIG0122
      IT=(IST-1)*N+1                                                    TEIG0123
      DO 1 I=IT,NM                                                      TEIG0124
    1 Y(I)=XLR(I)                                                       TEIG0125
      CALL SOLZ(STK,Y,N,M-IST+1,ICOL,INUM,IST)                          TEIG0126
      DO 11 K=IST,M                                                     TEIG0127
      K1=K-1                                                            TEIG0128
      II=(K-1)*N                                                        TEIG0129
      XM=0.0                                                            TEIG0130
      LCH(K)=0                                                          TEIG0131
      DO 7 I=1,N                                                        TEIG0132
      D=DABS(Y(II+I))                                                   TEIG0133
      IF(D-XM)7,7,9                                                     TEIG0134
    9 XM=D                                                              TEIG0135
      LCH(K)=I                                                          TEIG0136
    7 CONTINUE                                                          TEIG0137
      IF(LCH(K) .EQ. 0) GO TO 99                                        TEIG0138
      E=Y(II+LCH(K))                                                    TEIG0139
      E=1.0/E                                                           TEIG0140
      DO 11 I=1,N                                                       TEIG0141
      III=I+II                                                          TEIG0142
      Y(III)=Y(III)*E                                                   TEIG0143
   11 XLR(III)=XLR(III)*E                                               TEIG0144
```

```
      CALL MTRTR(M,N,RK,Y,XLR,IST)                                      TEIG0145
      DO 31 I=IT,NM                                                     TEIG0146
   31 X(I)=Y(I)                                                         TEIG0147
      CALL MULTZ(STM,X,U,N,M-IST+1,ICOL,INUM,IST)                       TEIG0148
      CALL MTRTR(M.N,RM,Y,X,IST)                                        TEIG0149
      DO 39 I=1,MM                                                      TEIG0150
      SRM(I)=RM(I)                                                      TEIG0151
   39 SRK(I)=RK(I)                                                      TEIG0152
      CALL DNROOT(M,RM,RK,U,RV)                                         TEIG0153
      DO 40 I=1,M                                                       TEIG0154
   40 U(I)=1./U(I)-ALPHAH                                               TEIG0155
      WRITE(6,41) (U(I),I=1,M)                                          TEIG0156
   41 FORMAT(/2X,'EIGENVALUES=',/,(2X,10D13.5))                         TEIG0157
      DO 22 I=IST,MA                                                    TEIG0158
      IST=I                                                             TEIG0159
      IF(DABS(EIG(I)/U(I)-1.0) .GT. ERR) GO TO 23                       TEIG0160
   22 CONTINUE                                                          TEIG0161
      DO 504 I=1,M                                                      TEIG0162
      SQU(I)=DSQRT(DABS(U(I)))                                          TEIG0163
      CYC(I)=SQU(I)*0.5D0/3.141592D0                                    TEIG0164
  504 CONTINUE                                                          TEIG0165
      WRITE(6,505)(SQU(I),I=1,M)                                        TEIG0166
  505 FORMAT(/5X,'RADIAN/SEC',/,(2X,10D13.5))                           TEIG0167
  506 FORMAT(/5X,'HERTZ     ',/,(2X,10D13.5))                           TEIG0168
      WRITE(6,506)(CYC(I),I=1,M)                                        TEIG0169
      DO 10 I=1,NCOT                                                    TEIG0170
      DO 10 J=1,M                                                       TEIG0171
      JJ=(J-1)*NCOT                                                     TEIG0172
      XLR(I+JJ)=0.0                                                     TEIG0173
      DO 10 K=1,M                                                       TEIG0174
   10 XLR(I+JJ)=XLR(I+JJ)+Y(I+(K-1)*N)*RV((J-1)*M+K)                    TEIG0175
      DO 15 I=1,M                                                       TEIG0176
      II=(I-1)*M                                                        TEIG0177
      DO 15 J=1,M                                                       TEIG0178
      JJ=(J-1)*M                                                        TEIG0179
      IJ=JJ+I                                                           TEIG0180
```

```
      RM(IJ)=0.0                                                        TEIG0181
      RK(IJ)=0.0                                                        TEIG0182
      DO 15 K=1,M                                                       TEIG0183
      KK=(K-1)*M                                                        TEIG0184
      RM(IJ)=RM(IJ)+SRM(KK+I)*RV(JJ+K)                                  TEIG0185
   15 RK(IJ)=RK(IJ)+SRK(KK+I)*RV(JJ+K)                                  TEIG0186
      CALL MTRTR(M,M,SRM,RV,RM,1)                                       TEIG0187
      CALL MTRTR(M,M,SRK,RV,RK,1)                                       TEIG0188
      DO 42 I=1,M                                                       TEIG0189
      KK=M*I-M+I                                                        TEIG0190
      Y(I)=1./DSQRT(SRM(KK))                                            TEIG0191
      II=I*NCDT-NCDT                                                    TEIG0192
      DO 42 J=1,NCDT                                                    TEIG0193
 42   XLR(II+J)=XLR(II+J)*Y(I)                                          TEIG0194
      DO 17 I=1,M                                                       TEIG0195
      KK=I*M-M                                                          TEIG0196
      DO 17 J=1,I                                                       TEIG0197
      E=Y(I)*Y(J)                                                       TEIG0198
      IJ=KK+J                                                           TEIG0199
      SRM(IJ)= E*SRM(IJ)                                                TEIG0200
 17   SRK(IJ) = E*SRK(IJ)                                               TEIG0201
      CALL       OUTPUT(KKK,M,NCDT,NDT,NOD,NNODE,ERR,XLR,U,SRM,SRK)     TEIG0202
      GO TO 99                                                          TEIG0203
   23 DO 25 I=1,M                                                       TEIG0204
 25   EIG(I)= U(I)                                                      TEIG0205
      IF(IST.EQ.1) GO TO 45                                             TEIG0206
      IQ=IST-1                                                          TEIG0207
      DO 44 I=1,IQ                                                      TEIG0208
      II=(I-1)*M                                                        TEIG0209
      DO 44 J=1,IQ                                                      TEIG0210
      IJI=II+J                                                          TEIG0211
      RM(IJI)=SRM(IJI)                                                  TEIG0212
   44 RK(IJI)=SRK(IJI)                                                  TEIG0213
   45 CONTINUE                                                          TEIG0214
      DO 34 I=1,N                                                       TEIG0215
      DO 34 J=IST,M                                                     TEIG0216
```

```
      JJ=(J-1)*N                                                         TEIG0217
      XLR(I+JJ)=0.0                                                      TEIG0218
      DO 34 K=1,M                                                        TEIG0219
   34 XLR(I+JJ)=XLR(I+JJ)+X(I+(K-1)*N)*RV((J-1)*M+K)                     TEIG0220
   21 CONTINUE                                                           TEIG0221
      WRITE(6,26) KKK                                                    TEIG0222
   26 FORMAT(/2X,'NO. OF ITERATION=',I4,2X,'NOT CONVERGED')              TEIG0223
   99 RETURN                                                             TEIG0224
      END                                                                TEIG0225
```

```
      SUBROUTINE INPUT(IEQ,NDPE,NET,NDT,MNC,NODE,NBU,NBCU,ERR,NITR,M)
C
C        TO SUPPLY INPUT INFORMATION
C
C
C     DES             COMMENTS AND DESCRIPTIONS
C     ICASE=1         WING
C     ICASE=2         BLADE       B.C.      CANTILEVER+CANTILEVER
C     ICASE=3         BLADE       B.C.      CANTILEVER+HINGE
C     ICASE=4         BLADE       B.C.      HINGE+CANTILEVER
C     ICASE=5         BLADE       B.C.      HINGE+HINGE
C
C     IPUNCH=0        NO PUNCH OUTPUT
C     IPUNCH=1        PUNCH OUTPUT
C
C     IGUEST=0        COUPLED MODE SHAPES GENERATED
C     IGUEST=1        W    DEFLECTION ONLY
C     IGUEST=2        V    DEFLECTION ONLY
C     IGUEST=3        PHI  DEFLECTION ONLY
C --CAUTION--   TO DERIVE UNCOUPLED MODES, COUPLING TERMS SHOULD BE
C --CAUTION--   SET AT ZERO.
C --CAUTION--   PB IN THE WING, RAMDA,COL,THETAF IN THE BLADE
C
C     NET NO OF ELEMENTS
C     NITR = NO OF ITERATION ALLOWED
C     M= NO OF MODES WANTED
C
C     ERR  ERROR TOLERANCE FOR ITERATION
C     OMEG  ROTATION FREQ IN RAD / SEC
C     RAMDA=INFLOW RATIO
C     COL=COLLECTIVE PITCH ANGLE DETERMINED BY PERFOMANCE ANALYSIS
C     SPKB,SPKC =SPRING CONSTANT OF THE HINGED BLADE (BEAMWISE
C                & CHORDWISE)
C     ALPHAH=HELPER TO AVOID THE SINGULARITY OF THE STIFFNESS  MATRIX
C
C     EIBE= EI FOR SPANWISE BENDING
```

```
C     EICE  EI FOR CORDWISE BENDING                                       INPU0037
C     THETAE=AMGLE OF TWIST                                               INPU0038
C     AMASE=MASS/UNIT LENGTH                                              INPU0039
C     ESE=ELEMENT SIZE                                                    INPU0040
C                                                                         INPU0041
C     AMN =MASS OF NACELLE AND ALL BLADES OR TIP MASS IN CASE OF BLADES   INPU0042
C     PIR =ROLLING MOMENT  OF INERTIA OF NACELLE                          INPU0043
C     PIY=YAWING   MOMENT  OF INERTIA OF NACELLE,                         INPU0044
C         HALF OF THAT OF ALL BLADES AND (H**2)*MASS OF ALL BLADES        INPU0045
C     PIP=PITCHING MOMENT  OF INERTIA OF NACELLE,                         INPU0046
C         HALF OF THAT OF ALL BLADES AND (H**2)*MASS OF ALL BLADES        INPU0047
C     PBW=MASS COUPING OF WING DUE TO BLADES H*MASS    OF BLADES          INPU0048
C                                                                         INPU0049
C                                                                         INPU0050
C     IF ICASE=1, NEXT THREE DATA GJ,PI, AND PI12 SHOULD BE ADDED.        INPU0051
C                                                                         INPU0052
C     GJ =TORSIONAL RIGIDITY                                              INPU0053
C     PI=MOMENT OF INERTIA FOR TORSION                                    INPU0054
C     PI12=MASS COUPLING BETWEEN TORSION AND SPANWISE BENDING             INPU0055
C           POSITIVE IF C.G. IS AHEAD OF ELASTIC AXIS                     INPU0056
C                                                                         INPU0057
C                                                                         INPU0058
C     NDPN= NO OF DEGREES OF FREEDOM PER NODE                             INPU0059
C     NDPN = 4 FOR BLADE MODE                                             INPU0060
C     NDPN = 6 FOR WING MODE                                              INPU0061
C                                                                         INPU0062
      COMMON /PUNCH/    IPUNCH                                            INPU0063
      COMMON /BW/       ICASE    ,   IGUEST                               INPU0064
      COMMON/SPRING/SPKB,SPKC                                             INPU0065
      COMMON/HELP/ALPHAH                                                  INPU0066
      DOUBLE PRECISION ALPHAH                                             INPU0067
      DIMENSION NDPE(1),NODE(1), NBCU(1)                                  INPU0068
      DOUBLE PRECISION ERR                                                INPU0069
      DIMENSION EIBE(20),EICE(20), THETAE(20),AMASE(20),ESE(20),TN(21)    INPU0070
      DIMENSION GJ(20) , PI(20) ,PI12 (20)                                INPU0071
      COMMON /ELEMI/ OMEG,EIBE,EICE,THETAE,AMASE,ESE,TN,GJ,PI, PI12,AMN,  INPU0072
```

```
     1 PIR,PIY,PIP,PBW,BM,BI,R,NETT                                     INPU0073
      DIMENSION DES(20)                                                 INPU0074
 10   FORMAT (16I5)                                                     INPU0075
 1     FORMAT (8E10.6)                                                  INPU0076
    4 FORMAT(5E15.7)                                                    INPU0077
 149  FORMAT(20A4)                                                      INPU0078
 150  FORMAT(I1)                                                        INPU0079
      IEQ=0                                                             INPU0080
      READ(5,149)(DES(I),I=1,20)                                        INPU0081
      READ(5,150)ICASE                                                  INPU0082
      READ(5,150)IPUNCH                                                 INPU0083
      READ(5,150)        IGUEST                                         INPU0084
      READ (5,10)        NET,NITR,M                                     INPU0085
      IF(ICASE.EQ.1)  GO TO 192                                         INPU0086
      NDPN=4                                                            INPU0087
      GO TO 193                                                         INPU0088
 192   NDPN=6                                                           INPU0089
 193  CONTINUE                                                          INPU0090
      NETT=NET                                                          INPU0091
      READ (5,1) ERR                                                    INPU0092
      READ (5,1) OMEG                                                   INPU0093
      READ(5,1) RAMDA                                                   INPU0094
      READ(5,1) COL                                                     INPU0095
      READ(5,1)SPKR,SPKC                                                INPU0096
      READ(5,1)      ALPHAH                                             INPU0097
      READ (5,4) (   EIBE(I),I=1,NET)                                   INPU0098
      READ (5,4) (   EICE(I),I=1,NET)                                   INPU0099
      READ (5,1) (THETAE(I),I=1,NET)                                    INPU0100
      READ (5,1) ( AMASE(I),I=1,NET)                                    INPU0101
      READ (5,1) (    ESE(I),I=1,NET)                                   INPU0102
      READ (5,1)  AMN,PIR,PIY,PIP,PBW                                   INPU0103
      IF ( NDPN .EQ. 4) GO TO 100                                       INPU0104
      READ (5,4) (GJ(I),I=1,NET)                                        INPU0105
      READ (5,4) (PI(I),I=1,NET)                                        INPU0106
      READ  (5,4  ) ( PI12(I),I=1,NET)                                  INPU0107
 100  CONTINUE                                                          INPU0108
```

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR,

```
      WRITE(6,181)                                                      INPU0109
  181 FORMAT(///5X,45(1H*)/)                                            INPU0110
      GO TO (151,152,153,154,155),ICASE                                 INPU0111
  151 WRITE(6,171)                                                      INPU0112
  171 FORMAT(9X,'WING')                                                 INPU0113
      GO TO 180                                                         INPU0114
  152 WRITE(6,172)                                                      INPU0115
  172 FORMAT(9X,'BLADE',3X,'BEAM*CANTI',3X,'CHORD*CANTI')               INPU0116
      GO TO 180                                                         INPU0117
  153 WRITE(6,173)                                                      INPU0118
  173 FORMAT(9X,'BLADE',3X,'BEAM*CANTI',3X,'CHORD*HINGE')               INPU0119
      GO TO 180                                                         INPU0120
  154 WRITE(6,174)                                                      INPU0121
  174 FORMAT(9X,'BLADE',3X,'BEAM*HINGE',3X,'CHORD*CANTI')               INPU0122
      GO TO 180                                                         INPU0123
  155 WRITE(6,175)                                                      INPU0124
  175 FORMAT(9X,'BLADE',3X,'BEAM*HINGE',3X,'CHORD*HINGE')               INPU0125
  180 CONTINUE                                                          INPU0126
      WRITE(6,182)(DES(I),I=1,20)                                       INPU0127
  182 FORMAT(/5X,45(1H*)///15X,20A4///)                                 INPU0128
      WRITE(6,9)                                                        INPU0129
      WRITE(6,600)IPUNCH       ,IGUEST                                  INPU0130
  600 FORMAT(1X,'IPUNCH=',I1                    ,5X,'IGUEST=',I1)       INPU0131
      WRITE (6,16) NDPN, NET,NITR,M,EPR                                 INPU0132
      WRITE(6,114) OMEG                                                 INPU0133
  114 FORMAT(6H OMEG=,F15.5)                                            INPU0134
      WRITE(6,300) RAMDA                                                INPU0135
  300 FORMAT(8H LAMBDA=,F15.5)                                          INPU0136
      WRITE(6,301) COL                                                  INPU0137
  301 FORMAT(18H COLLECTIVE PITCH=,F15.5)                               INPU0138
      WRITE(6,299)SPKB,SPKC                                             INPU0139
  299 FORMAT(1X,'SPRING=    ',F15.5,F15.5)                              INPU0140
      WRITE(6,298) ALPHAH                                               INPU0141
  298 FORMAT(1X,'ALPHAH=',D15.5)                                        INPU0142
      WRITE(6,183)                                                      INPU0143
  183 FORMAT(1X,'--FLAPPING BENDING STIFFNESS--')                       INPU0144
```

```
      WRITE(6,105) (  EIBE(I),I=1,NET)                                  INPU0145
      WRITE(6,184)                                                      INPU0146
  184 FORMAT(1X,'--CHORDWISE BENDING STIFFNESS--')                      INPU0147
      WRITE(6,105) (  EICE(I),I=1,NET)                                  INPU0148
      WRITE(6,185)                                                      INPU0149
  185 FORMAT(1X,'--ANGLE OF TWIST--')                                   INPU0150
      WRITE(6,104) (THETAE(I),I=1,NET)                                  INPU0151
      THE=ATAN(RAMDA*4.0/3.0)+COL*SIGN(1.0,OMEG)                        INPU0152
      DO 302 I=1,NET                                                    INPU0153
  302 THETAE(I)=THETAE(I)*SIGN(1.0,OMEG)+THE                            INPU0154
      WRITE(6,186)                                                      INPU0155
  186 FORMAT(1X,'--MASS DISTRIBUTION--')                                INPU0156
      WRITE(6,104) ( AMASE(I),I=1,NET)                                  INPU0157
      WRITE(6,188)                                                      INPU0158
  188 FORMAT(1X,'--ELEMENT SIZE--')                                     INPU0159
      WRITE(6,104) (    ESE(I),I=1,NET)                                 INPU0160
      WRITE(6,187)                                                      INPU0161
  187 FORMAT(8X,'TIP MASS',T19 ,'ROLL INERTIA',T34,  'YAW INERTIA',     INPU0162
     &          T49,'PITCH INERTIA',T64,'MASS COUPLING')                INPU0163
      WRITE(6,104) AMN,PIR,PIY,PIP,PBW                                  INPU0164
      NN=NET+1                                                          INPU0165
  104 FORMAT (8F15.5)                                                   INPU0166
  105 FORMAT(5X,7E15.7)                                                 INPU0167
      IF (NOPN .EQ. 4)  GO TO 14                                        INPU0168
      WRITE(6,189)                                                      INPU0169
  189 FORMAT(1X,'--TORSIONAL RIGIDITY--')                               INPU0170
      WRITE(6,105) (GJ(I),I=1,NET)                                      INPU0171
      WRITE(6,190)                                                      INPU0172
  190 FORMAT(1X,'--MOMENT OF INERTIA--')                                INPU0173
      WRITE(6,105) (PI(I),I=1,NET)                                      INPU0174
      WRITE(6,191)                                                      INPU0175
  191 FORMAT(1X,'--MASS COUPLING ALONG SPAN--')                         INPU0176
      WRITE (6,105) ( PI12(I),I=1,NET)                                  INPU0177
    9 FORMAT (/8H   *    *  /12H  INPUT DATA ///)                       INPU0178
   16     FORMAT (25H NO OF DEGRE PER NODE=          , I3/              INPU0179
     1          17H  NO OF ELEMENTS=,I3/24H NO OF MAX ITER ALLOEWD=,I3  INPU0180
```

```
     2 /14H  NO OF MODES=, I3/ 6H  ERR=, F15.5)                         INPU0181
 14   NDE=NDPN+NDPN                                                     INPU0182
      NDT=NET*NDPN+NDPN                                                 INPU0183
      MNC=NDT*NDE-(NDE*NDE-NDE)/2                                       INPU0184
      DO 5 I=1,NET                                                      INPU0185
      NDPE(I)=NDE                                                       INPU0186
      NII=NDE*I-NDE                                                     INPU0187
      NDD=NDPN*I-NDPN                                                   INPU0188
      NODE(NII  +1)=NDD+1                                               INPU0189
      NODE(NII  +5)=NDD+2                                               INPU0190
      NODE(NII  +2)=NDD+3                                               INPU0191
      NODE(NII  +6)=NDD+4                                               INPU0192
      NODE(NII  +3)=NDD+NDPN+1                                          INPU0193
      NODE(NII  +7)=NDD+NDPN+2                                          INPU0194
      NODE(NII  +4)=NDD+NDPN+3                                          INPU0195
      NODE(NII  +8)=NDD+NDPN+4                                          INPU0196
      IF (NDPN .EQ. 4) GO TO  5                                         INPU0197
      NODE(NII  +9)=NDD+5                                               INPU0198
      NODE(NII +10)=NDD+6                                               INPU0199
      NODE(NII +11)=NDD+NDPN+5                                          INPU0200
      NODE(NII +12)=NDD+NDPN+6                                          INPU0201
 5    CONTINUE                                                          INPU0202
      GO TO (501,502,503,504,505),ICASE                                 INPU0203
501   NBU=NDPN-1                                                        INPU0204
      GO TO 506                                                         INPU0205
502   NBU=4                                                             INPU0206
506   DO 6 I=1,NBU                                                      INPU0207
6     NBCU(I)=I                                                         INPU0208
      GO TO 507                                                         INPU0209
503   NBU=3                                                             INPU0210
      NBCU(1)=1                                                         INPU0211
      NBCU(2)=2                                                         INPU0212
      NBCU(3)=3                                                         INPU0213
      GO TO 507                                                         INPU0214
504   NBU=3                                                             INPU0215
      NBCU(1)=1                                                         INPU0216
```

```
      NBCU(2)=2                                                         INPU0217
      NBCU(3)=4                                                         INPU0218
      GO TO 507                                                         INPU0219
  505 NBU=2                                                             INPU0220
      NBCU(1)=1                                                         INPU0221
      NBCU(2)=2                                                         INPU0222
  507 CONTINUE                                                          INPU0223
      R=0                                                               INPU0224
      DO 2 I=1,NET                                                      INPU0225
    2  R=R+ESE(I)                                                       INPU0226
      TN(NET+1)=AMN*R*OMEG*OMEG                                         INPU0227
      DO 3 I=1,NET                                                      INPU0228
      II=NET-I+1                                                        INPU0229
      R=R-ESE(II)                                                       INPU0230
    3 TN(II)= TN(II+1) +AMASE(II)*(R+.5*ESE(II) )*ESE(II)*OMEG*OMEG     INPU0231
      WRITE (6,15)                                                      INPU0232
      WRITE(6,104) (     TN(I),I=1,NN)                                  INPU0233
   15  FORMAT(///'  TENSION DUE TO CENTRIF FORCE')                      INPU0234
      R=0.                                                              INPU0235
      BI=0.                                                             INPU0236
      BM=0                                                              INPU0237
      DO 11 I=1,NET                                                     INPU0238
      BM=BM+ AMASE(I)*ESE(I)                                            INPU0239
      RR=R+ESE(I)                                                       INPU0240
      BI= BI+ AMASE(I)*(RR**3-R**3)*.33333333                           INPU0241
   11 R=RR                                                              INPU0242
      BM=BM+AMN                                                         INPU0243
      BI=BI+AMN*R*R                                                     INPU0244
      WRITE (6,12) BM,BI,R                                              INPU0245
   12 FORMAT(/'  MASS =',E13.5/ '  MOMENT OF INERTIA AT ROOT=',E13.5/   INPU0246
     1 '  TOTAL LENGTH OF THE BEAM=',E13.5)                             INPU0247
      RETURN                                                            INPU0248
      END                                                               INPU0249
```

```
      SUBROUTINE ELEMK  (EKT,EMT ,NDE,NE)                               ELEM0001
C                                                                       ELEM0002
C        TO CONTROL THE GENERATION OF ELEMENT STIFFNESS AND MASS        ELEM0003
C        MATRICES                                                       ELEM0004
C                                                                       ELEM0005
C     NDE=8 FOR BLADE MODE                                              ELEM0006
C     NDE=12 FOR WING MODE                                              ELEM0007
      COMMON/SPRING/SPKB,SPKC                                           ELEM0008
      COMMON/HELP/ALPHAH                                                ELEM0009
      DOUBLE PRECISION ALPHAH                                           ELEM0010
      DOUBLE PRECISION EKT,EMT                                          ELEM0011
      DIMENSION EKC(4,4),ETC(4,4) ,EMC(4,4),EKT(1),EMT(1)               ELEM0012
      DIMENSION EIBE(20),EICE(20), THETAE(20),AMASE(20),ESE(20),TN(21)  ELEM0013
      DIMENSION GJ(20) , PI(20) ,PI12 (20)                              ELEM0014
      COMMON /ELEMI/ OMEG,EIBE,EICE,THETAE,AMASE,ESE,TN,GJ,PI, PI12,AMN, ELEM0015
     1 PIR,PIY,PIP,PBW,BM,BI,R,NETT                                     ELEM0016
      NET=NETT                                                          ELEM0017
      EIB=EIBE(NE)                                                      ELEM0018
      EIC=EICE(NE)                                                      ELEM0019
      THETA=THETAE(NE)                                                  ELEM0020
      AMAS=AMASE(NE)                                                    ELEM0021
      ES=ESE(NE)                                                        ELEM0022
      T= (TN(NE)+ TN(NE+1) )*.5                                         ELEM0023
      ESS=ES*ES                                                         ELEM0024
      IF (NDE .EQ.8) GO TO 9                                            ELEM0025
      GJE=GJ(NE)/ES                                                     ELEM0026
      PIE=PI(NE) *ES                                                    ELEM0027
      PI12E=PI12(NE)  *ES                                               ELEM0028
    9 EKC(1,1)=12.                                                      ELEM0029
      EKC(2,1)=6.*ES                                                    ELEM0030
      EKC(3,1)=-12.                                                     ELEM0031
      EKC(4,1)=6.*ES                                                    ELEM0032
      EKC(2,2)=4.*ESS                                                   ELEM0033
      EKC(3,2)=-6.*ES                                                   ELEM0034
      EKC(4,2)=2.*ESS                                                   ELEM0035
      EKC(3,3)=12.                                                      ELEM0036
```

```
      EKC(4,3)=-6.*ES                                                   ELEM0037
      EKC(4,4)=4.*ESS                                                   ELEM0038
      ETC(1,1)=1.2                                                      ELEM0039
      ETC(2,1)=.1*ES                                                    ELEM0040
      ETC(3,1)=-1.2                                                     ELEM0041
      ETC(4,1)=.1*ES                                                    ELEM0042
      ETC(2,2)=.13333333*ESS                                            ELEM0043
      ETC(3,2)=-.1*ES                                                   ELEM0044
      ETC(4,2)=-.03333333*ESS                                           ELEM0045
      ETC(3,3)=1.2                                                      ELEM0046
      ETC(4,3)=-.1*ES                                                   ELEM0047
      ETC(4,4)=.13333333*ESS                                            ELEM0048
      EMC(1,1)=156./420.                                                ELEM0049
      EMC(2,1)=22./420.*ES                                              ELEM0050
      EMC(3,1)=54./420.                                                 ELEM0051
      EMC(4,1)=-13./420.*ES                                             ELEM0052
      EMC(2,2)=4./420.* ESS                                             ELEM0053
      EMC(3,2)=-EMC(4,1)                                                ELEM0054
      EMC(4,2)=-3./420.*ESS                                             ELEM0055
      EMC(3,3)=EMC(1,1)                                                 ELEM0056
      EMC(4,3)=-EMC(2,1)                                                ELEM0057
      EMC(4,4)=EMC(2,2)                                                 ELEM0058
      DO 10 I=1,4                                                       ELEM0059
      DO 10 J=1,I                                                       ELEM0060
      EMC(J,I)= EMC(I,J)                                                ELEM0061
   10 EKC(J,I)= EKC(I,J)                                                ELEM0062
      ESSS=ESS*ES                                                       ELEM0063
      SI=SIN(THETA)                                                     ELEM0064
      CO=COS(THETA)                                                     ELEM0065
      B=(EIB*CO*CO +EIC*SI*SI)/ESSS                                     ELEM0066
      C=(EIC*CO*CO + EIB*SI*SI)/ESSS                                    ELEM0067
      BC=(EIC-EIB)*SI*CO/ESSS                                           ELEM0068
      TE=T/ES                                                           ELEM0069
      AM=OMEG*OMEG*AMAS*ES                                              ELEM0070
      AMAE =AMAS*ES                                                     ELEM0071
      DO  5 I=1,4                                                       ELEM0072
```

```
      II=(I*I-I)/2                                                      ELEM0073
      I4=(I+4)*(I+3)/2                                                  ELEM0074
      DO 1 J=1,I                                                        ELEM0075
      EKT(II+J)= B*EKC(I,J)+TE*ETC(I,J)                                 ELEM0076
      EMT(II+J)= AMAE*EMC(I,J)                                          ELEM0077
      EKT(I4+J+4)= C*EKC(I,J) + TE*ETC(I,J) -AM*EMC(I,J)                ELEM0078
    1 EMT(I4+J+4)= AMAE*EMC(I,J)                                        ELEM0079
      DO 2 J=1,4                                                        ELEM0080
      EMT(I4+J)=0.                                                      ELEM0081
    2 EKT(I4+J)=BC*EKC(I,J)                                             ELEM0082
      IF (NDE  .EQ. 8) GO TO 5                                          ELEM0083
      I8=(I+8)*(I+7)/2                                                  ELEM0084
      DO 3 J=1,I                                                        ELEM0085
      EKT(I8+J+8)=GJE*ETC(I,J)                                          ELEM0086
    3 EMT(I8+J+8)= PIE *EMC(I,J)                                        ELEM0087
      DO 4 J=1,4                                                        ELEM0088
      EKT(I8+J  )=0.                                                    ELEM0089
      EKT(I8+J+4)=0.                                                    ELEM0090
      EMT(I8+J+4)=0.                                                    ELEM0091
    4 EMT(I8+J)=PI12E*EMC(I,J)                                          ELEM0092
    5 CONTINUE                                                          ELEM0093
      IF(NE.LT.NET) GO TO 100                                           ELEM0094
      EMT(6)= EMT(6)+AMN                                                ELEM0095
      EMT(28) =EMT(28)+AMN                                              ELEM0096
      IF(NDE.EQ.8) GO TO 100                                            ELEM0097
      EMT(10) =EMT(10)+ PIR                                             ELEM0098
      EMT(36) =EMT(36)+ PIY                                             ELEM0099
      EMT(66)= EMT(66)+PIP                                              ELEM0100
      EMT(58)= EMT(58)+ PBW                                             ELEM0101
  100 IF(NE.GT.1) GO TO 101                                             ELEM0102
      EKT(3)=EKT(3)+SPKB                                                ELEM0103
      EKT(21)=EKT(21)+SPKC                                              ELEM0104
  101 DO 102 I=1,36                                                     ELEM0105
  102 EKT(I)=EKT(I)+EMT(I)*ALPHAH                                       ELEM0106
      RETURN                                                            ELEM0107
      END                                                               ELEM0108
```

```
      SUBROUTINE MESH(NDPE,NET,NDT,NCDT,MNC,MN,      NOD,NNODE,ICOL,INUM,
     @NBU,      INDEX)
C
C        TO CALCULATE MESH INFORMATION OF THE FINITE ELEMENT
C
      DIMENSION          NOD(1),NNODE(1),ICOL(1),INUM(1),        NDPE(1)
      IF(NBU .GT. 0) GO TO 100
      DO 99 I=1,NDT
   99 NOD(I)=I
      NCDT=NDT
      GO TO 98
  100 DO 22 I=1,NBU
      IF (I .GE. NBU) GO TO 101
      I1=I+1
      DO 21 J=I1,NBU
      IF ( INUM(I).NE.INUM(J)) GO TO 21
      IF (J.GE.NBU) GO TO 20
      J1=J+1
      DO 19 K=J1,NBU
 19   INUM(K-1)= INUM(K)
 20   NBU=NBU-1
      J=J-1
 21   CONTINUE
 22   CONTINUE
 101  DO 1 I=1,NDT
    1 NOD(I)=1
      DO 2 I=1,NBU
      I1=INUM(I)
    2 NOD(I1)=0
      DO 3 I=2,NDT
    3 NOD(I)=NOD(I)+NOD(I-1)
      NCDT=NDT-NBU
      DO 4 I=1,NBU
      I1=INUM(I)
    4 NOD(I1)=NCDT+I
   98 CONTINUE
```

```
      II=0                                                               MESH0037
      DO 5 I=1,NET                                                       MESH0038
      JJ=NDPE(I)                                                         MESH0039
      DO 54 J=1,JJ                                                       MESH0040
   54 ICOL(J)=NNODE(J+II)                                                MESH0041
      DO 55 J=1,JJ                                                       MESH0042
      J1=ICOL(J)                                                         MESH0043
   55 NNODE(J+II)=NJC(J1)                                                MESH0044
    5 II=II+JJ                                                           MESH0045
      DO 6 I=1,NCDT                                                      MESH0046
    6 ICOL(I)=I                                                          MESH0047
      II=0                                                               MESH0048
      DO 88 I=1,NET                                                      MESH0049
      JJ=NDPE(I)                                                         MESH0050
      IMIN=NDT                                                           MESH0051
      DO 7 J=1,JJ                                                        MESH0052
      IN=NNODE(J+II)                                                     MESH0053
    7 IF(IN .LT. IMIN) IMIN=IN                                           MESH0054
      DO 8 J=1,JJ                                                        MESH0055
      ID=NNODE(J+II)                                                     MESH0056
    8 IF(ICOL(ID) .GT. IMIN) ICOL(ID)=IMIN                               MESH0057
   88 II=II+JJ                                                           MESH0058
      INUM(1)=0                                                          MESH0059
      DO 9 I=2,NCDT                                                      MESH0060
    9 INUM(I)=INUM(I-1)+I-ICOL(I)                                        MESH0061
      MN=INUM(NCDT)+NCDT                                                 MESH0062
      WRITE(6,10) MN,MNC                                                 MESH0063
   10 FORMAT(/2X,'MAX. SIZE OF STF IS',I6,5X,'SPECIFIED SIZE IS ',I6)    MESH0064
      INDEX=1                                                            MESH0065
      IF(MNC .LT. MN) INDEX=0                                            MESH0066
      RETURN                                                             MESH0067
      END                                                                MESH0068
```

```
      SUBROUTINE ASBV(STK,STM,IEQ,EK,EM,    NDPE,NDT,NCDT,NNODE,MN,NET,IN   ASBV0001
     @UM)                                                                   ASBV0002
C                                                                           ASBV0003
C        TO ASSEMBLE AND CONSTRAIN BOUNDARY CONDITIONS                      ASBV0004
C                                                                           ASBV0005
      IMPLICIT REAL*8(A-H,O-Z)                                              ASBV0006
      DIMENSION STK(1),STM(1),EK(1),EM(1),      NDPE(1),NNODE(1),INUM(1)    ASBV0007
      ID=NCDT+1                                                             ASBV0008
      DO 1 I=1,MN                                                           ASBV0009
      STK(I)=0.0                                                            ASBV0010
    1 STM(I)=0.0                                                            ASBV0011
      II=0                                                                  ASBV0012
      DO 15 N=1,NET                                                         ASBV0013
      JJ=NDPE(N)                                                            ASBV0014
      IF(IEQ .EQ. 0) GO TO 3                                                ASBV0015
      IF(N .GT. 1) GO TO 8                                                  ASBV0016
    3 CALL ELEMK(EK,EM,JJ,N)                                                ASBV0017
    8 DO 14 I=1,JJ                                                          ASBV0018
      KI=(I*I-I)/2                                                          ASBV0019
      KMA=NNODE(II+I)                                                       ASBV0020
      IF(KMA .GE. ID) GO TO 14                                              ASBV0021
      DO 13 J=1,I                                                           ASBV0022
      KNA=NNODE(II+J)                                                       ASBV0023
      IF(KNA .GE. ID) GO TO 13                                              ASBV0024
      KA=INUM(KMA)+KNA                                                      ASBV0025
      IF(KNA .GT. KMA) KA=INUM(KNA)+KMA                                     ASBV0026
      STK(KA)=STK(KA)+EK(KI+J)                                              ASBV0027
      STM(KA)=STM(KA)+EM(KI+J)                                              ASBV0028
   13 CONTINUE                                                              ASBV0029
   14 CONTINUE                                                              ASBV0030
   15 II=JJ+II                                                              ASBV0031
      RETURN                                                                ASBV0032
      END                                                                   ASBV0033
```

```
      SUBROUTINE FAC(STF,NDT,NNDG,ICOL,INUM,U)                          FACT0001
C                                                                       FACT0002
C     FACTORING A SYSMETRIC MATRIX INTO LDL*                            FACT0003
C                                                                       FACT0004
      IMPLICIT REAL*8(A-H,O-Z)                                          FACT0005
      DIMENSION STF(1),ICOL(1),INUM(1),U(1)                             FACT0006
      NNDG=0                                                            FACT0007
      IF(STF(1)) 2,1,3                                                  FACT0008
    1 IZR=1                                                             FACT0009
   99 WRITE(6,100) IZR                                                  FACT0010
  100 FORMAT(/2X,'THE',I4,'TH DIAG. AFTER FACT.=0.0,INCOMPLETE FACT.')  FACT0011
      NNDG=-1                                                           FACT0012
      RETURN                                                            FACT0013
    2 NNDG=1                                                            FACT0014
    3 IF(NDT .LT. 2) GO TO 11                                           FACT0015
      DO 10 IR=2,NDT                                                    FACT0016
      II=ICOL(IR)                                                       FACT0017
      IF(II .EQ. IR) GO TO 10                                           FACT0018
      JR=INUM(IR)                                                       FACT0019
      IE=IR-1                                                           FACT0020
      DO 5 IC=II,IE                                                     FACT0021
      IMAX=II                                                           FACT0022
      IF(II .LT. ICOL(IC)) IMAX=ICOL(IC)                                FACT0023
      JE=IC-1                                                           FACT0024
      SUM=STF(JR+IC)                                                    FACT0025
      IF(JE .LT. IMAX) GO TO 55                                         FACT0026
      JC=INUM(IC)                                                       FACT0027
      DO 4 J=IMAX,JE                                                    FACT0028
    4 SUM=SUM-U(J)*STF(JC+J)                                            FACT0029
   55 U(IC)=SUM                                                         FACT0030
    5 STF(JR+IC)=SUM/STF(INUM(IC)+IC)                                   FACT0031
      JJ=JR+IR                                                          FACT0032
      DO 6 J=II,IE                                                      FACT0033
    6 STF(JJ)=STF(JJ)-U(J)*STF(JR+J)                                    FACT0034
      IF(STF(JJ)) 8,7,10                                                FACT0035
    7 IZR=IR                                                            FACT0036
```

```
      GO TO 99                                                          FACT0037
    8 NNDG=NNDG+1                                                       FACT0038
   10 CONTINUE                                                          FACT0039
   11 WRITE(6,101) NNDG                                                 FACT0040
  101 FORMAT(/2X,'NO. OF NEGATIVE DIAGS.=',I4,5X,'FACT. COMPLETED')     FACT0041
      RETURN                                                            FACT0042
      END                                                               FACT0043
```

```
      SUBROUTINE MTRTR(M,N,RM,XLR,X,IST)                                MTRT0001
C                                                                       MTRT0002
C        MATRIX MULTIPLICATION                                          MTRT0003
C                                                                       MTRT0004
      IMPLICIT REAL*8(A-H,O-Z)                                          MTRT0005
      DIMENSION RM(1),XLR(1),X(1)                                       MTRT0006
      DO 2 I=IST,M                                                      MTRT0007
      II=(I-1)*N                                                        MTRT0008
      IJ=(I-1)*M                                                        MTRT0009
      DO 2 J=1,I                                                        MTRT0010
      JJ=(J-1)*N                                                        MTRT0011
      RM(IJ+J)=0.0                                                      MTRT0012
      DO 2 K=1,N                                                        MTRT0013
    2 RM(IJ+J)=RM(IJ+J)+XLR(II+K)*X(JJ+K)                               MTRT0014
      DO 3 I=IST,M                                                      MTRT0015
      JJ=(I-1)*M                                                        MTRT0016
      DO 3 J=1,I                                                        MTRT0017
    3 RM((J-1)*M+I)=RM(JJ+J)                                            MTRT0018
      RETURN                                                            MTRT0019
      END                                                               MTRT0020
```

```
      SUBROUTINE MULTZ(STF,X,Y,NDT,M,ICOL,INUM,MM)
C
C        MATRIX MULTIPLICATICN
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION STF(1),X(1),Y(1),ICCL(1),INUM(1)
      DO 1 I=1,NCT
    1 Y(I)=0.0
      MMM=M+MM-1
      DO 4 I=MM,MMM
      II=(I-1)*NDT
      DO 3 IR=1,NDT
      IS=INUM(IR)
      IC=ICCL(IR)
      IE=IR-1
      IF(IC .GT. IE) GC TC 3
      DO 2 J=IC,IE
      S=STF(IS+J)
      Y(IR)=Y(IR)+S*X(II+J)
    2 Y(J)=Y(J)+S*X(II+IR)
    3 Y(IR)=Y(IR)+STF(IS+IR)*X(II+IR)
      DO 4 J=1,NDT
      X(II+J)=Y(J)
    4 Y(J)=0.0
      RETURN
      END
```

```
      SUBROUTINE SOLZ(STF,U,NDT,M,ICOL,INUM,MM)
C
C     SOLVE (LDL*)(U)=U FOR GIVEN U OF M VECTORS OF LENGTH NDT
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION STF(1),U(1),ICOL(1),INUM(1)
      MMM=M+MM-1
      IF(NDT .LT. 2) GO TO 3
      DO 2 IR=2,NDT
      JI=ICOL(IR)
      JE=IR-1
      IF(JI .GT. JE) GO TO 2
      DO 1 I=MM,MMM
      II=(I-1)*NDT
      IS=II+IR
      DO 1 J=JI,JE
    1 U(IS)=U(IS)-STF(INUM(IR)+J)*U(II+J)
    2 CONTINUE
    3 DO 4 I=MM,MMM
      II=(I-1)*NDT
      DO 4 IR=1,NDT
    4 U(II+IR)=U(II+IR)/STF(INUM(IR)+IR)
      IF(NDT .LT. 2) GO TO 7
      DO 6 IK=2,NDT
      IR=NDT-IK+2
      JI=ICOL(IR)
      JE=IR-1
      IF(JI .GT. JE) GO TO 6
      DO 5 I=MM,MMM
      II=(I-1)*NDT
      IS=II+IR
      DO 5 J=JI,JE
    5 U(II+J)=U(II+J)-STF(INUM(IR)+J)*U(IS)
    6 CONTINUE
    7 RETURN
      END
```

```
      SUBROUTINE DNROOT(M,A,B,XL,X)
C
C        EIGENVALUE ANALYSIS ROUTINE
C
      DIMENSION A(1),B(1),XL(1),X(1)
      DOUBLE PRECISION A,B,XL,X,SUMV
      K=1
      DO 100 J=2,M
      L=M*(J-1)
      DO 100 I=1,J
      L=L+1
      K=K+1
  100 B(K)=B(L)
      MV=0
      CALL EIGEN (B,X,M,MV)
      L=0
      DO 110 J=1,M
      L=L+J
  110 XL(J)=1.0/DSQRT(DABS(B(L)))
      K=0
      DO 115 J=1,M
      DO 115 I=1,M
      K=K+1
  115 B(K)=X(K)*XL(J)
      DO 120 I=1,M
      N2=0
      DO 120 J=1,M
      N1=M*(I-1)
      L=M*(J-1)+I
      X(L)=0.0
      DO 120 K=1,M
      N1=N1+1
      N2=N2+1
  120 X(L)=X(L)+B(N1)*A(N2)
      L=0
      DO 130 J=1,M
```

```
      DO 130 I=1,J                                                      DNR00037
      N1=I-M                                                            DNR00038
      N2=M*(J-1)                                                        DNR00039
      L=L+1                                                             DNR00040
      A(L)=0.0                                                          DNR00041
      DO 130 K=1,M                                                      DNR00042
      N1=N1+M                                                           DNR00043
      N2=N2+1                                                           DNR00044
  130 A(L)=A(L)+X(N1)*B(N2)                                             DNR00045
      CALL EIGEN (A,X,M,MV)                                             DNR00046
      L=0                                                               DNR00047
      DO 140 I=1,M                                                      DNR00048
      L=L+I                                                             DNR00049
  140 XL(I)=A(L)                                                        DNR00050
      DO 150 I=1,M                                                      DNR00051
      N2=0                                                              DNR00052
      DO 150 J=1,M                                                      DNR00053
      N1=I-M                                                            DNR00054
      L=M*(J-1)+I                                                       DNR00055
      A(L)=0.0                                                          DNR00056
      DO 150 K=1,M                                                      DNR00057
      N1=N1+M                                                           DNR00058
      N2=N2+1                                                           DNR00059
  150 A(L)=A(L)+B(N1)*X(N2)                                             DNR00060
      L=0                                                               DNR00061
      K=0                                                               DNR00062
      DO 180 J=1,M                                                      DNR00063
      SUMV=0.0                                                          DNR00064
      DO 170 I=1,M                                                      DNR00065
      L=L+1                                                             DNR00066
  170 SUMV=SUMV+A(L)*A(L)                                               DNR00067
  175 SUMV=DSQRT(SUMV)                                                  DNR00068
      DO 180 I=1,M                                                      DNR00069
      K=K+1                                                             DNR00070
  180 X(K)=A(K)/SUMV                                                    DNR00071
      RETURN                                                            DNR00072
      END                                                               DNR00073
```

```
      SUBROUTINE EIGEN(A,R,N,MV)                                        EIGE0001
C                                                                       EIGE0002
C        EIGENVALUE ANALYSIS ROUTINE NEEDED IN DNROOT                   EIGE0003
C                                                                       EIGE0004
      DIMENSION A(1),R(1)                                               EIGE0005
      DOUBLE PRECISION A,R,ANORM,ANRMX,THR,X,Y,SINX,SINX2,COSX,         EIGE0006
     1                 COSX2,SINCS,RANGE                                EIGE0007
    5 RANGE=1.0D-12                                                     EIGE0008
      IF(MV-1) 10,25,10                                                 EIGE0009
   10 IQ=-N                                                             EIGE0010
      DO 20 J=1,N                                                       EIGE0011
      IQ=IQ+N                                                           EIGE0012
      DO 20 I=1,N                                                       EIGE0013
      IJ=IQ+I                                                           EIGE0014
      R(IJ)=0.0                                                         EIGE0015
      IF(I-J) 20,15,20                                                  EIGE0016
   15 R(IJ)=1.0                                                         EIGE0017
   20 CONTINUE                                                          EIGE0018
   25 ANORM=0.0                                                         EIGE0019
      DO 35 I=1,N                                                       EIGE0020
      DO 35 J=I,N                                                       EIGE0021
      IF(I-J) 30,35,30                                                  EIGE0022
   30 IA=I+(J*J-J)/2                                                    EIGE0023
      ANORM=ANORM+A(IA)*A(IA)                                           EIGE0024
   35 CONTINUE                                                          EIGE0025
      IF(ANORM) 165,165,40                                              EIGE0026
   40 ANORM=1.414*DSQRT(ANORM)                                          EIGE0027
      ANRMX=ANORM*RANGE/FLOAT(N)                                        EIGE0028
      IND=0                                                             EIGE0029
      THR=ANORM                                                         EIGE0030
   45 THR=THR/FLOAT(N)                                                  EIGE0031
   50 L=1                                                               EIGE0032
   55 M=L+1                                                             EIGE0033
   60 MQ=(M*M-M)/2                                                      EIGE0034
      LQ=(L*L-L)/2                                                      EIGE0035
      LM=L+MQ                                                           EIGE0036
```

```
 62 IF(DABS(A(LM))-THR) 130,65,65                                      EIGE0037
 65 IND=1                                                              EIGE0038
    LL=L+LQ                                                            EIGE0039
    MM=M+MQ                                                            EIGE0040
    X=0.5*(A(LL)-A(MM))                                                EIGE0041
 68 Y=-A(LM)/DSQRT(A(LM)*A(LM)+X*X)                                    EIGE0042
    IF(X) 70,75,75                                                     EIGE0043
 70 Y=-Y                                                               EIGE0044
 75 SINX=Y/DSQRT(2.0*(1.0+(DSQRT(1.0-Y*Y))))                           EIGE0045
    SINX2=SINX*SINX                                                    EIGE0046
 78 COSX=DSQRT(1.0-SINX2)                                              EIGE0047
    COSX2=COSX*COSX                                                    EIGE0048
    SINCS =SINX*COSX                                                   EIGE0049
    ILQ=N*(L-1)                                                        EIGE0050
    IMQ=N*(M-1)                                                        EIGE0051
    DO 125 I=1,N                                                       EIGE0052
    IQ=(I*I-I)/2                                                       EIGE0053
    IF(I-L) 80,115,80                                                  EIGE0054
 80 IF(I-M) 85,115,90                                                  EIGE0055
 85 IM=I+MQ                                                            EIGE0056
    GO TO 95                                                           EIGE0057
 90 IM=M+IQ                                                            EIGE0058
 95 IF(I-L) 100,105,105                                                EIGE0059
100 IL=I+LQ                                                            EIGE0060
    GO TO 110                                                          EIGE0061
105 IL=L+IQ                                                            EIGE0062
110 X=A(IL)*COSX-A(IM)*SINX                                            EIGE0063
    A(IM)=A(IL)*SINX+A(IM)*COSX                                        EIGE0064
    A(IL)=X                                                            EIGE0065
115 IF(MV-1) 120,125,120                                               EIGE0066
120 ILR=ILQ+I                                                          EIGE0067
    IMR=IMQ+I                                                          EIGE0068
    X=R(ILR)*COSX-R(IMR)*SINX                                          EIGE0069
    R(IMR)=R(ILR)*SINX+R(IMR)*COSX                                     EIGE0070
    R(ILR)=X                                                           EIGE0071
125 CONTINUE                                                           EIGE0072
```

```
      X=2.0*A(LM)*SINCS                                                 EIGE0073
      Y=A(LL)*COSX2+A(MM)*SINX2-X                                       EIGE0074
      X=A(LL)*SINX2+A(MM)*COSX2+X                                       EIGE0075
      A(LM)=(A(LL)-A(MM))*SINCS+A(LM)*(COSX2-SINX2)                     EIGE0076
      A(LL)=Y                                                           EIGE0077
      A(MM)=X                                                           EIGE0078
  130 IF(M-N) 135,140,135                                               EIGE0079
  135 M=M+1                                                             EIGE0080
      GO TO 60                                                          EIGE0081
  140 IF(L-(N-1)) 145,150,145                                           EIGE0082
  145 L=L+1                                                             EIGE0083
      GO TO 55                                                          EIGE0084
  150 IF(IND-1) 160,155,160                                             EIGE0085
  155 IND=0                                                             EIGE0086
      GO TO 50                                                          EIGE0087
  160 IF(THR-ANRMX) 165,165,45                                          EIGE0088
  165 IQ=-N                                                             EIGE0089
      DO 185 I=1,N                                                      EIGE0090
      IQ=IQ+N                                                           EIGE0091
      LL=I+(I*I-I)/2                                                    EIGE0092
      JQ=N*(I-2)                                                        EIGE0093
      DO 185 J=I,N                                                      EIGE0094
      JQ=JQ+N                                                           EIGE0095
      MM=J+(J*J-J)/2                                                    EIGE0096
      IF(A(LL)-A(MM)) 170,185,185                                       EIGE0097
  170 X=A(LL)                                                           EIGE0098
      A(LL)=A(MM)                                                       EIGE0099
      A(MM)=X                                                           EIGE0100
      IF(MV-1) 175,185,175                                              EIGE0101
  175 DO 180 K=1,N                                                      EIGE0102
      ILR=IQ+K                                                          EIGE0103
      IMR=JQ+K                                                          EIGE0104
      X=R(ILR)                                                          EIGE0105
      R(ILR)=R(IMR)                                                     EIGE0106
  180 R(IMR)=X                                                          EIGE0107
  185 CONTINUE                                                          EIGE0108
      RETURN                                                            EIGE0109
      END                                                               EIGE0110
```

```
      SUBROUTINE OUTPUT(KKK,M,NCDT,NDT,NOD,NNODE,ERR,XLR,U,SRM,SRK)         OUTP0001
C                                                                           OUTP0002
C        OUTPUT ROUTINE                                                     OUTP0003
C                                                                           OUTP0004
      COMMON /PUNCH/    IPUNCH                                              OUTP0005
      COMMON /BW/       ICASE    ,   IGUEST                                 OUTP0006
      DOUBLE PRECISION XLR,SRM,SRK ,U          ,ERR                         OUTP0007
      DIMENSION XLR(1),U(1),                         SRM(1), SRK(1)         OUTP0008
      DIMENSION  NOD(1),NNODE(1)                                            OUTP0009
      DIMENSION  XXLF(15,120)                                               OUTP0010
      IDEBUG=0                                                              OUTP0011
      WRITE(6,24) KKK,ERR                                                   OUTP0012
   24 FORMAT(/2X,'NO. OF ITERATION=',I4,2X,'CONVERGED WITHIN',D13.5)        OUTP0013
      IF(IDEBUG.EQ.0) GO TO 15                                              OUTP0014
      WRITE(6,12)                                                           OUTP0015
   12 FORMAT(/2X,'EIGENVECTORS=',/)                                         OUTP0016
      DO 14 I=1,M                                                           OUTP0017
      II=(I-1)*NCDT                                                         OUTP0018
   14 WRITE(6,13) (XLR(II+J),J=1,NCDT)                                      OUTP0019
   13 FORMAT(/,(2X,10D13.5))                                                OUTP0020
 15   CONTINUE                                                              OUTP0021
      WRITE(6,20)                                                           OUTP0022
      DO 18 I=1,M                                                           OUTP0023
      KK=I*M-M                                                              OUTP0024
  18  WRITE (6,16) (SRM(J+KK) , J=1,I)                                      OUTP0025
      WRITE(6,21)                                                           OUTP0026
      DO 19 I=1,M                                                           OUTP0027
      KK=I*M-M                                                              OUTP0028
  19  WRITE (6,16) (SRK(J+KK) ,J=1,I)                                       OUTP0029
   16 FORMAT                    (2X,10D13.5)                                OUTP0030
  20  FORMAT (/ 23H  REDUCED MASS MATRIX      /)                             OUTP0031
  21  FORMAT (/ 23H  REDUCED STIF MATRIX      /)                             OUTP0032
      NBU=NDT-NCDT                                                          OUTP0033
      GO TO (101,101,101,102,101),ICASE                                     OUTP0034
 101  DO 100 I=1,M                                                          OUTP0035
      DO 201 J=1,NBU                                                        OUTP0036
```

```
      XXLR(I,J)=0.0                                                     OUTP0037
  201 CONTINUE                                                          OUTP0038
      DO 202 J=1,NCDT                                                   OUTP0039
  202 XXLR(I,NBU+J)=XLR((I-1)*NCDT+J)                                   OUTP0040
  100 CONTINUE                                                          OUTP0041
      GO TO 205                                                         OUTP0042
  102 DO 203 I=1,M                                                      OUTP0043
      DO 206 J=1,NBU                                                    OUTP0044
  206 XXLR(I,J)=0.0                                                     OUTP0045
      XXLR(I,3)=XLR((I-1)*NCDT+1)                                       OUTP0046
      DO204 J=2,NCDT                                                    OUTP0047
  204 XXLR(I,J+NBU)=XLR((I-1)*NCDT+J)                                   OUTP0048
  203 CONTINUE                                                          OUTP0049
  205 CONTINUE                                                          OUTP0050
      IF(ICASE.EQ.1) GO TO 333                                          OUTP0051
      NOM=4                                                             OUTP0052
      NODE=NDT/NOM                                                      OUTP0053
      WRITE(6,350)                                                      OUTP0054
  350 FORMAT(///1X,'****  BLADE MODE SHAPES  ****')                     OUTP0055
      DO 351 I=1,M                                                      OUTP0056
      WRITE(6,4500)I                                                    OUTP0057
 4500 FORMAT(///1X,'I=',I2//)                                           OUTP0058
      WRITE(6,4900)                                                     OUTP0059
 4900 FORMAT(T5,'K',T13,'W(I,J)',T33,'V(I,J)',   T53,'DW(I,J)',  T73,   OUTP0060
     &          'DV(I,J)')                                              OUTP0061
      DO351 K=1,NODE                                                    OUTP0062
      WRITE(6,352) K,(XXLR(I,NOM*(K-1)+J),J=1,NOM)                      OUTP0063
  352 FORMAT((1X,I4,6(5X,E15.7)))                                       OUTP0064
  351 CONTINUE                                                          OUTP0065
      GO TO 360                                                         OUTP0066
  333 CONTINUE                                                          OUTP0067
      NOM=6                                                             OUTP0068
      NODE=NDT/NOM                                                      OUTP0069
      WRITE(6,353)                                                      OUTP0070
  353 FORMAT(///1X,'****  WING MODE SHAPES  ****')                      OUTP0071
      DO 354 I=1,M                                                      OUTP0072
```

```
      WRITE(6,4500) I                                                   OUTP0073
      WRITE(6,4901)                                                     OUTP0074
4901  FORMAT(T5,'K',T13,'W(I,J)',T33,'V(I,J)',    T53,'DW(I,J)',  T73,  OUTP0075
     &         'DV(I,J)',T93,'PHI(I,J)',T113,'DPHI(I,J)')               OUTP0076
      DO 355 K=1,NODE                                                   OUTP0077
      WRITE(6,352) K,(XXLR(I,NOM*(K-1)+J),J=1,NOM)                      OUTP0078
355   CONTINUE                                                          OUTP0079
354   CONTINUE                                                          OUTP0080
360   CONTINUE                                                          OUTP0081
      IF(IPUNCH.EQ.0) GO TO 370                                         OUTP0082
      DO 359 I=1,M                                                      OUTP0083
      DO 358 K=1,NOM                                                    OUTP0084
      WRITE(7,357)(XXLR(I,NOM*(J-1)+K),J=1,NODE)                        OUTP0085
357   FORMAT(6E13.5)                                                    OUTP0086
358   CONTINUE                                                          OUTP0087
359   CONTINUE                                                          OUTP0088
370   CONTINUE                                                          OUTP0089
      WRITE(6,371)                                                      OUTP0090
371   FORMAT(1H1)                                                       OUTP0091
      RETURN                                                            OUTP0092
      END                                                               OUTP0093
```

A.2 The TILDYN Program Listing

```
C
C     ************************************************************
C
C               PROGRAM  ROTOR
C                    PART 2 ;   PROGRAM  TILDYN
C
C     ************************************************************
C
C       PURPOSE
C         TO ANALYZE THE TILT-ROTOR DYNAMIC SYSTEM BY MEANS
C         OF FREQUENCY RESPONSE AND EIGENVALUES IN POWERED
C         AND AUTOROTATION FLIGHT
C
C         DEVELOPED BY MASAHIRO YASUE
C              OF AEROELASTIC AND STRUCTURES RESEARCH LABORATORY
C                   AUGUST 1974
C              ADDRESS ;  BLG 41-211
C                         MASSACHUSETTS INSTITUTE OF TECHNOLOGY
C                         CAMBRIDGE, MASS.  02139
C
C ................................................................................
C
C         MAIN PROGRAM
C
C         TO DEFINE THE SEQUENCE OF THE PROGRAM
C
      COMMON/DOF18/AAY(19,19),BBY(19,19),CCY(19,19),DDY(19,6)
      DIMENSION CGUST(6)   ,DDZ(19)
      COMMON /PARMT/     ITYPE ,IFLT
1     CONTINUE
5001  FORMAT(1H1)
      CALL INITIL
      CALL COEFF
      CALL INPUT(CGUST,IDOF,IRES,IEIGEN)
      CALL INTPL
      WRITE(6,5001)
```

```
      CALL AERODT                                                       MAIN0037
      CALL ORDINT                                                       MAIN0038
      WRITE(6,5001)                                                     MAIN0039
      CALL       AINER                                                  MAIN0040
      CALL       AEROMT                                                 MAIN0041
      CALL EQMTX(IDOF)                                                  MAIN0042
      IF(IFLT.EQ.0) GO TO 400                                           MAIN0043
      CALL AUTO(IDOF)                                                   MAIN0044
  400 CONTINUE                                                          MAIN0045
      IDIM=IDOF+IFLT                                                    MAIN0046
      IF (IRES.EQ.0) GO TO 200                                          MAIN0047
      WRITE(6,5001)                                                     MAIN0048
      CALL GUSTCO(CGUST,DDY,IDIM,DDZ)                                   MAIN0049
      CALL FRQRES(IDIM,AAY,BBY,CCY,DDZ,IFLT,IDOF)                       MAIN0050
  200 CONTINUE                                                          MAIN0051
      IF (IEIGEN.EQ.0) GO TO 1000                                       MAIN0052
      WRITE(6,5001)                                                     MAIN0053
      CALL EIGEN(IDIM,AAY,BBY,CCY,DDY,IDOF)                             MAIN0054
 1000 CONTINUE                                                          MAIN0055
      WRITE(6,5001)                                                     MAIN0056
      GO TO 1                                                           MAIN0057
      END                                                               MAIN0058
```

```
      BLOCK DATA
C
C        TO INITIALIZE THE COEFFICIENTS OF GAUSSIAN QUADRATURE
C
      COMMON /AREA2/NPT,XXX(20),A(20)
      DATA NPT/11/
      DATA A(1),A(2),A(3),A(4),A(5),A(6)/0.055668,
     @    0.125580  ,0.186290, 0.233193 ,0.262804 , 0.272925   /
      END
```

```
      SUBROUTINE INITIL
C
C        INITIALIZATION OF THE MATRICES
C
      COMMON/DOF19/AAY(19,19),BBY(19,19),CCY(19,19),DDY(19,6)
      COMMON/INERTI/TTMT(4,6,3),TTCTJ(4,6,3),AMJT(4,3,6),CJT(4,3,6)
      COMMON /ARR/WGUST(6,6),DAMX(6,6),AMX(6,6),DQ(4,6,3),Q(4,6,3)
     @          ,DHMAX(4,3,6),HMAX(4,3,6)
      COMMON/WINGAR/TSCS(20,6,6),TSAS(20,6,6),TSAG(20,6,3)
      DO 10 I=1,4
      DO 10 J=1,6
      DO 10 K=1,3
      TTMT(I,J,K)=0.0
      TTCTJ(I,J,K)=0.0
      AMJT(I,K,J)=0.0
      CJT(I,K,J)=0.0
      DQ(I,J,K)=0.0
      Q(I,J,K)=0.0
      DHMAX(I,K,J)=0.0
      HMAX(I,K,J)=0.0
10    CONTINUE
      DO 11 I=1,6
      DO 11 J=1,6
      WGUST(I,J)=0.0
      DAMX(I,J)=0.0
      AMX(I,J)=0.0
      DO 11 K=1,20
      TSDS(K,I,J)=0.0
      TSAS(K,I,J)=0.0
11    CONTINUE
      DO 12 I=1,20
      DO 12 J=1,6
      DO 12 K=1,3
      TSAG(I,J,K)=0.0
12    CONTINUE
      DO 13 I=1,19
```

```
      DO 14 J=1,19                                                      INIT0037
      AAY(I,J)=0.0                                                      INIT0038
      BBY(I,J)=0.0                                                      INIT0039
      CCY(I,J)=0.0                                                      INIT0040
14    CONTINUE                                                          INIT0041
      DO 15 K=1,6                                                       INIT0042
      DDY(I,K)=0.0                                                      INIT0043
15    CONTINUE                                                          INIT0044
13    CONTINUE                                                          INIT0045
      RETURN                                                            INIT0046
      END                                                               INIT0047
```

```
      SUBROUTINE COEFF                                                  COEF0001
C                                                                       COEF0002
C        TO DEFINE THE POINTS AND COEFFICIENTS OF GAUSSIAN QUADRATURE   COEF0003
C                                                                       COEF0004
      DIMENSION Y(20),YY(20)                                            COEF0005
      COMMON /AREA2/NPT,XXX(20),A(20)                                   COEF0006
      NPTH=NPT/2                                                        COEF0007
      IF((FLOAT(NPTH)-NPT/2.0).NE.0.0)GO TO 100                         COEF0008
      READ(5,5000)(Y(I),I=1,NPTH),(A(J),J=1,NPTH)                       COEF0009
 5000 FORMAT(8F10.5)                                                    COEF0010
      DO 10 II=1,NPTH                                                   COEF0011
      Y(NPTH+II)=-Y(NPTH-II+1)                                          COEF0012
      A(NPTH+II)=A(NPTH-II+1)                                           COEF0013
   10 CONTINUE                                                          COEF0014
      GO TO 200                                                         COEF0015
  100 NPTH1=NPTH+1                                                      COEF0016
      DATA Y(1),Y(2),Y(3),Y(4),Y(5),Y(6)/0.978228,                      COEF0017
     %      0.887062  , 0.730152 , 0.519096 , 0.269543 , 0.0      /     COEF0018
      DO 20 MM=1,NPTH                                                   COEF0019
      A(NPTH+MM+1)=A(NPTH-MM+1)                                         COEF0020
      Y(NPTH+MM+1)=-Y(NPTH-MM+1)                                        COEF0021
   20 CONTINUE                                                          COEF0022
  200 DO 50 KK=1,NPT                                                    COEF0023
      YY(KK)=Y(KK)                                                      COEF0024
   50 CONTINUE                                                          COEF0025
      DO 60 IN=1,NPT                                                    COEF0026
      Y(IN)=YY(NPT-IN+1)                                                COEF0027
   60 CONTINUE                                                          COEF0028
      DO 30 JJJ=1,NPT                                                   COEF0029
      XXX(JJJ)=(Y(JJJ)+1.0)/2.0                                         COEF0030
   30 CONTINUE                                                          COEF0031
      RETURN                                                            COEF0032
      END                                                               COEF0033
```

```
      SUBROUTINE INPUT(CGUST,IDOF,IRES,IEIGEN)
C
C        TO SUPPLY INPUT INFORMATION
C
C
C        DES   =IDENTIFYING INFORMATION
C        ITYPE=0: HINGELESS ROTOR IN POWERED FLIGHT
C        ITYPE=1: HINGELESS ROTOR IN AUTOROTATIONAL FLIGHT
C                 GIMBALLED ROTOR IN BOTH FLIGHTS
C        IFLT =0 : POWERED FLIGHT      IFLT=1 : AUTOROATION FLIGHT
C        IDOF =9 : BASIC DEGREES OF  FREEDOM IS 9     IDOF=18 : DOF IS 18
C        IRES =0 : FREQUENCY RESPONSE OFF        IRES=1 : RESPONSE ON
C        IFRMAG=0 : MODE NOMALIZED ROTOR RADIUS AND WING SEMISPAN
C        IFRMAG=1 : NORMAL MODES
C        IEIGEN=0 : EIGENANALYSIS OFF       IEIGEN=1 : EIGENANALYSIS ON
C
C        NOBLD= NUMBER OF BLADES
C        ROH= AIR DENSITY
C        OMEGA= ROTATIONAL SPEED (RAD/SEC)
C        RAMDA= INFLOW RATIO
C        VEL= CRUISING FLIGHT SPEED
C        R= ROTOR RADIUS
C        AIB= BLADE FLAPPING MOMENT OF INERTIA
C        CHOD= BLADE CHORD
C        CL= BLADE LIFT CURVE SLOPE
C        CD= BLADE DRAG COEFFICIENT
C        HMAST= MAST HEIGHT
C        DEL3= PITCH-FLAP COUPLING COEFFICIENT ( RADIAN )
C
C        WL= WING SEMISPAN
C        WCOD= WING CHORD
C        WCL= WING LIFT CURVE SLOPE
C        WCD= WING DRAG COEFFICIENT
C        WCMO= WING PITCHING MOMENT COEFFICIENT
C        WCMA= WING PITCHING MOMENT CURVE SLOPE
C        EDIS=DISTANCE BETWEEN AERODYNAMIC CENTER AND ELASTIC AXIS
```

```
C              ( NONDIMENSIONALIZED BY WING CHORD,POSITIVE AERODYMAMIC          INPU0037
C              CENTER AHEAD )                                                   INPU0038
C        WTHET= WING TRIM ANGLE OF ATTACK (RADIAN )                             INPU0039
C                                                                               INPU0040
C        CGUST= EXCITING FORCE COMPONENTS                                       INPU0041
C                                                                               INPU0042
C        BRAM= (BLADE NATURAL FREQUENCY )**2     (RADIAN/SEC)**2                INPU0043
C        WRAM= (WING NATURAL FREQUENCY )**2      (RADIAN/SEC)**2                INPU0044
C                                                                               INPU0045
C        NW= WING ELEMENT NUMBER                                                INPU0046
C        EMSW= WING ELEMENT SIZE NORMALIZED BY THE SEMISPAN                     INPU0047
C        G= VERTICAL BENDING MODE COMPONENT AT THE NODE OF THE WING             INPU0048
C        Z= CHORDWISE BENDING MODE COMPONENT AT THE NODE OF THE WING            INPU0049
C        DG= VERTICAL BENDING MODE SLOPE AT THE NODE OF THE WING                INPU0050
C        DZ= CHORDWISW BENDING MODE SLOPE AT THE NODE OF THE WING               INPU0051
C        WPHI= TOSION MODE COMPONENT AT THE NODE OF THE WING                    INPU0052
C       DWPHI=TORSION MODE SLOPE AT THE NODE OF THE WING                        INPU0053
C                                                                               INPU0054
C        N= BLADE ELEMENT NUMBER                                                INPU0055
C        EMS=BLADE ELEMENT SIZE NORMALIZED BY THE ROTOR RADIUS                  INPU0056
C        AMASS= MASS DISTRIBUTION AT THE NODE OF THE BLADE                      INPU0057
C        THTEN= ANGLE OF TWIST AT THE NODE OF THE BLADE                         INPU0058
C        COL= COLLECTIVE PITCH ANGLE DETERMINED BY THE PERFORMANCE (RADIAN)     INPU0059
C        W=OUT-OF-PLANE MODE COMPONENT AT THE NODE OF THE WING                  INPU0060
C        V= INPLANE MODE COMPONEBT AT THE NODE OF THE BLADE                     INPU0061
C        DW= OUT-OF- PLANE MODE SLOPE COMPONENT AT THE NODE OF TNE BLADE        INPU0062
C        DV= INPLANE MODE SLOPE COMPONENT AT THE NODE OF THE BLADE              INPU0063
C                                                                               INPU0064
C        BRAMO= (COLLECTIVE MODE  NATURAL FREQUENCY OF THE BLADE)**2            INPU0065
C               (RADIAN/SEC)**2                                                 INPU0066
C        WCOL= OUT-OF-PLANE COMPONENT OF THE BLADE COLLECTIVE MODE AT THE NODE  INPU0067
C        VCOL= INPLANE COMPONENT OF THE BLADE COLLECTIVE MODE AT THE NODE       INPU0068
C        DWCOL= OUT-OF-PLANE SLOPE COMPONENT OF THE BLADE COLLECTIVE MODE       INPU0069
C               AT THE NODE                                                     INPU0070
C        DVCOL=INPLANE SLOPE COMPONENT OF THE BLADE COLLECTIVE MODE AT THE NODE INPU0071
C                                                                               INPU0072
```

```
C
      DIMENSION BMAX(4),BOMAX(4),WMAX(6),BIG(6)
      DIMENSION CGUST(6) ,SBMO(4),SBM(4),SWG(6),DES(80)
      COMMON /PARMT/    ITYPE ,IFLT
      COMMON/AMATIC/TT(6,5),C(6,6)     ,T(5,6)
      COMMON /AREA1/OMEGA,R,VEL,CL,CD,RAMDA,SNOMEG
      COMMON /AREA6/NOBLD,ROH,CHOD,AIB,CK,HMAST    ,ALOCK ,ANO,HR
      COMMON/AREA8/BLAM(4),WLAM(6),BRAM(4),WRAM(6),BLAMO(4),BRAMO(4)
      COMMON/GIN/      V(4,21),W(4,21),DV(4,21),DW(4,21),THETA(21),
     %EMS(20),AMASS (20),N,NDP
      COMMON /GINO/     VCOL(4,21),WCOL(4,21),DVCOL(4,21),DWCOL(4,21)
      COMMON/WING/ NW,NDPW,EMSW(20),G(6,21),DG(6,21),Z(6,21),DZ(6,21),
     *WPHI(6,21),DWPHI(6,21)
      COMMON/WICH/WL,WCOD,WCL,WCD,WCMO,WCMA,EDIS,WTHET,VV
      COMMON /COUPL/    AKPC(4),AKPO(4)
      COMMON/FRMAG/     FRB(4),FRBO(4),FRW(6) ,IFRMAG
      COMMON/THE/THETN(21) ,MB,MW
      IDEBUG=0
      MB=2
      MW=3
      READ(5,5003)(DES(I),I=1,80)
      READ(5,5001) ITYPE
      READ(5,5001) IFLT
      READ(5,2034)IDOF
      IF(IDOF.EQ.9) GO TO 62
      MB=4
      MW=6
 62   CONTINUE
      READ(5,5001)IRES
      READ(5,5001)IFRMAG
      READ(5,5001)IEIGEN
      READ(5,5001)NOBLD
      READ(5,5000)ROH,OMEGA,RAMDA,VEL
      READ(5,5000)R,AIB,CHOD,CL,CD,HMAST,DEL3
      READ(5,5000)WL,WCOD,WCL,WCD,WCMO,WCMA,EDIS,WTHET
      READ(5,5000)CGUST
```

```
      READ(5,5000)BRAM                                                  INPU0109
      READ(5,5000)WRAM                                                  INPU0110
      READ(5,2034)NW                                                    INPU0111
      NDPW=NW+1                                                         INPU0112
      READ(5,5000)(EMSW(K),K=1,NW)                                      INPU0113
      DO 60 I=1,MW                                                      INPU0114
      READ(5,1001)(     G(I,J),J=1,NDPW)                                INPU0115
      READ(5,1001)(     Z(I,J),J=1,NDPW)                                INPU0116
      READ(5,1001)(    DG(I,J),J=1,NDPW)                                INPU0117
      READ(5,1001)(    DZ(I,J),J=1,NDPW)                                INPU0118
      READ(5,1001)( WPHI(I,J),J=1,NDPW)                                 INPU0119
      READ(5,1001)(DWPHI(I,J),J=1,NDPW)                                 INPU0120
   60 CONTINUE                                                          INPU0121
      READ(5,2034)N                                                     INPU0122
      NDP=N+1                                                           INPU0123
      READ(5,5000)(EMS(K),K=1,N)                                        INPU0124
      READ(5,5000)(AMASS(J),J=1,NDP)                                    INPU0125
      READ(5,5000)(THETN(J),J=1,NDP)                                    INPU0126
      READ(5,5000)COL                                                   INPU0127
      DO 50 I=1,MB                                                      INPU0128
      READ(5,1001)(     W(I,J),J=1,NDP )                                INPU0129
      READ(5,1001)(     V(I,J),J=1,NDP )                                INPU0130
      READ(5,1001)(    DW(I,J),J=1,NDP )                                INPU0131
      READ(5,1001)(    DV(I,J),J=1,NDP )                                INPU0132
      AKPC(I)=DW(I,1)*TAN(DEL3)*(-1.0)                                  INPU0133
   50 CONTINUE                                                          INPU0134
      IF(ITYPE.EQ.0) GO TO 100                                          INPU0135
      READ(5,5000)PRAMO                                                 INPU0136
      DO 52 I=1,MB                                                      INPU0137
      READ(5,1001)( WCOL(I,J),J=1,NDP )                                 INPU0138
      READ(5,1001)( VCOL(I,J),J=1,NDP )                                 INPU0139
      READ(5,1001)(DWCOL(I,J),J=1,NDP )                                 INPU0140
      READ(5,1001)(DVCOL(I,J),J=1,NDP )                                 INPU0141
      AKPO(I)=DWCOL(I,1)*TAN(DEL3)*(-1.0)                               INPU0142
   52 CONTINUE                                                          INPU0143
      DO 200 I=1,4                                                      INPU0144
```

```
      BLAMO(I)=BRAMO(I)/OMEGA**2                                        INPU0145
      SBMO(I)=SQRT(BLAMO(I))                                            INPU0146
200   CONTINUE                                                          INPU0147
100   CONTINUE                                                          INPU0148
      IF(IDOF.EQ.18) GO TO 101                                          INPU0149
      DO 305 I=4,6                                                      INPU0150
      DO 305 J=1,NDPW                                                   INPU0151
      G(I,J)=0.0                                                        INPU0152
      DG(I,J)=0.0                                                       INPU0153
      Z(I,J)=0.0                                                        INPU0154
      DZ(I,J)=0.0                                                       INPU0155
      WPHI(I,J)=0.0                                                     INPU0156
      DWPHI(I,J)=0.0                                                    INPU0157
305   CONTINUE                                                          INPU0158
      DO 307 I=3,4                                                      INPU0159
      DO 306 J=1,NDP                                                    INPU0160
      W(I,J)=0.0                                                        INPU0161
      DW(I,J)=0.0                                                       INPU0162
      V(I,J)=0.0                                                        INPU0163
306   DV(I,J)=0.0                                                       INPU0164
307   AKPC(I)=0.0                                                       INPU0165
      IF(ITYPE.EQ.0) GO TO 101                                          INPU0166
      DO 308 I=3,4                                                      INPU0167
      DO 309 J=1,NDP                                                    INPU0168
      WCOL(I,J)=0.0                                                     INPU0169
      DWCOL(I,J)=0.0                                                    INPU0170
      VCOL(I,J)=0.0                                                     INPU0171
309   DVCOL(I,J)=0.0                                                    INPU0172
308   AKPO(I)=0.0                                                       INPU0173
101   CONTINUE                                                          INPU0174
      DO 22 I=1,NDP                                                     INPU0175
      THETA(I)=THETN(I)+COL                                             INPU0176
22    CONTINUE                                                          INPU0177
      DO 6 I=1,4                                                        INPU0178
      BLAM(I)=BRAM(I)/OMEGA**2                                          INPU0179
6     SBM(I)=SQRT(BLAM(I))                                              INPU0180
```

```
      DO 5 I=1,6                                                        INPU0181
      WLAM(I)=WRAM(I)/OMEGA**2                                          INPU0182
5     SWG(I)=SQRT(WLAM(I))                                              INPU0183
      DO 30 I=1,6                                                       INPU0184
      TT(I,1)=G(I,NDPW)                                                 INPU0185
      TT(I,2)=Z(I,NDPW)                                                 INPU0186
      TT(I,3)=DZ(I,NDPW)                                                INPU0187
      TT(I,4)=WPHI(I,NDPW)                                              INPU0188
      TT(I,5)=-DG(I,NDPW)                                               INPU0189
30    CONTINUE                                                          INPU0190
      IF(IFLT.EQ.0) GO TO 23                                            INPU0191
      DO 24 I=1,6                                                       INPU0192
24    TT(I,5)=0.0                                                       INPU0193
23    CONTINUE                                                          INPU0194
      DO 40 I=1,6                                                       INPU0195
      DO 40 J=1,6                                                       INPU0196
40    C(I,J)=DZ(I,NDPW)*WPHI(J,NDPW)-DZ(J,NDPW)*WPHI(I,NDPW)            INPU0197
      ALOCK=ROH*CL*CHOD*R**4/AIB                                        INPU0198
      ANO=FLOAT(NOBLD)                                                  INPU0199
      HR=HMAST/R                                                        INPU0200
      SNOMEG=SIGN(1.0,OMEGA)                                            INPU0201
2034  FORMAT(I2)                                                        INPU0202
5003  FORMAT(80A1)                                                      INPU0203
5001  FORMAT(I1)                                                        INPU0204
5000  FORMAT(8E10.0)                                                    INPU0205
1001  FORMAT(6F13.5)                                                    INPU0206
C  ***************  PRINT OUT OF INPUT DATA     ***************          INPU0207
      WRITE(6,5002) (DES(I),I=1,80)                                     INPU0208
5002  FORMAT(///10X,100(1H*),//20X,80A1,//10X,100(1H*)//)               INPU0209
      WRITE(6,5004) ITYPE,IFLT,       IDOF,IRES,IEIGEN ,IFRMAG          INPU0210
5004  FORMAT(//10X,'ITYPE=',I2,3X,'IFLT=',I2,                3X,'IDOF=',INPU0211
     &     I2,3X,'IRES=',I2,3X,'IEIGEN=',I2,3X,'IFRMAG=',I2)            INPU0212
      WRITE(6,1)NOBLD,ROH,CHOD,AIB ,HMAST,ALOCK                         INPU0213
1     FORMAT(/// T4,'NO OF BLADES',T25,'ROH',T41,'CHORD',T59,'IB'       INPU0214
     $,T74,'HMAST',T94,'LOCK NO',                                       INPU0215
     <///     T7,I2,T18,5(1PE15.7,2X))                                  INPU0216
```

```
      WRITE(6,2)OMEGA,R,VEL,CL,CD,RAMDA                                 INPU0217
 2    FORMAT(/// T6,'OMEGA',T25,'R',T41,'VEL',T59,'CL',T74,'CD'         INPU0218
     #,T94,'RAMDA'        ///1X, 6(1PE15.7,2X))                         INPU0219
      WRITE(6,61) COL    ,DEL3                                          INPU0220
 61   FORMAT(///T6,'COLLECTIVE PITCH',T25,'DEL3'    ///1X,2(1PE15.7,2X))INPU0221
      WRITE(6,3)WL,WCOD,WCL,WCD,WCMO,WCMA                               INPU0222
 3    FORMAT(/// T6,'WING L',T25,'WING CHOD',T41,'WING CL',T59,'WING CD'INPU0223
     %,T74,'WING CMO',T94,'WING CMA'                                    INPU0224
     *///1X ,6(1PE15.7,2X))                                             INPU0225
      WRITE(6,4)EDIS,WTHET                                              INPU0226
 4    FORMAT(/// T6,'DISTANCE AC EA',T25,'WING ALPHAH'                  INPU0227
     @///1X ,2(1PE15.7,2X))                                             INPU0228
      WRITE(6,250)                                                      INPU0229
 250  FORMAT(///1X,35(1H-)/1X,'EIGENVALUES ( NATURAL FREQUENCIES )'     INPU0230
     &        /1X,35(1H-)//2X,'--( RAD/SEC )**2--')                     INPU0231
      IF(ITYPE.EQ.0) GO TO 254                                          INPU0232
      WRITE(6,251)(BRAMO(I),I=1,MB)                                     INPU0233
 251  FORMAT(/1X,'  **BLADE COLLECTIVE**'/4X,4(F12.3,3X))               INPU0234
      WRITE(6,252)                                                      INPU0235
 252  FORMAT(/1X,'  **BLADE CYCLIC**')                                  INPU0236
      GO TO 255                                                         INPU0237
 254  WRITE(6,253)                                                      INPU0238
 253  FORMAT(/1X,'  **BLADE**')                                         INPU0239
 255  WRITE(6,256)(BRAM(I),I=1,MB)                                      INPU0240
 256  FORMAT(4X,4(F12.3,3X))                                            INPU0241
      WRITE(6,257)(WRAM(I),I=1,MW)                                      INPU0242
 257  FORMAT(/1X,'  **WING**'/4X,6(F12.3,3X))                           INPU0243
      WRITE(6,408)                                                      INPU0244
 408  FORMAT(/2X,'-- RAD/SEC/OMEGA --')                                 INPU0245
      IF(ITYPE.EQ.0) GO TO 354                                          INPU0246
      WRITE(6,351)(SBMO(I),I=1,MB)                                      INPU0247
 351  FORMAT(/1X,'  **BLADE COLLECTIVE/OMEGA**'/4X,4(F12.3,3X))         INPU0248
      WRITE(6,352)                                                      INPU0249
 352  FORMAT(/1X,'  **BLADE CYCLIC/OMEGA**')                            INPU0250
      GO TO 355                                                         INPU0251
 354  WRITE(6,353)                                                      INPU0252
```

```
353   FORMAT(/1X,'  **BLADE/OMEGA**')                                     INPU0253
355   WRITE(6,356)(SBM(I),I=1,MB)                                         INPU0254
356   FORMAT(4X,4(F12.3,3X))                                              INPU0255
      WRITE(6,357)(SWG(I),I=1,MW)                                         INPU0256
357   FORMAT(/1X,'  **WING/OMEGA**'/4X,6(F12.3,3X))                       INPU0257
      WRITE(6,409)                                                        INPU0258
409   FORMAT(///1X,'EXCITING FORCE COMPONENTS')                           INPU0259
      WRITE(6,358)CGUST                                                   INPU0260
358   FORMAT( //T6,'U GUST',T25,'V GUST', T41,'W GUST', T59,              INPU0261
     *        'THETA 0',T74,'THETA 1C', T94,'THETA 1S',/ /1X,             INPU0262
     *        6(1PE15.7,2X))                                              INPU0263
      DO 430 I=1,MB                                                       INPU0264
      DO 431 J=1,NDP                                                      INPU0265
      BIG(I)=0.0                                                          INPU0266
      PA=ABS(W(I,J))                                                      INPU0267
      IF(PA    -BIG(I)) 460,460,461                                       INPU0268
461   BIG(I)=PA                                                           INPU0269
460   PA=ABS(V(I,J))                                                      INPU0270
      IF(PA    -BIG(I)) 431,431,462                                       INPU0271
462   BIG(I)=PA                                                           INPU0272
431   CONTINUE                                                            INPU0273
430   BMAX(I)=BIG(I)                                                      INPU0274
      DO 435 I=1,MW                                                       INPU0275
      DO 436 J=1,NDPW                                                     INPU0276
      BIG(I)=0.0                                                          INPU0277
      PA=ABS(G(I,J))                                                      INPU0278
      IF(PA    -BIG(I)) 465,465,466                                       INPU0279
466   BIG(I)=PA                                                           INPU0280
465   PA=(Z(I,J))                                                         INPU0281
      IF(PA    -BIG(I)) 436,436,467                                       INPU0282
467   BIG(I)=PA                                                           INPU0283
436   CONTINUE                                                            INPU0284
435   WMAX(I)=BIG(I)                                                      INPU0285
      IF(ITYPE.EQ.0) GO TO 480                                            INPU0286
      DO 440 I=1,MB                                                       INPU0287
      DO 441 J=1,NDP                                                      INPU0288
```

```
      BIG(I)=0.0                                                        INPU0289
      PA=ABS(WCOOL(I,J))                                                INPU0290
      IF(PA         -BIG(I)) 470,470,471                                INPU0291
  471 BIG(I)=PA                                                         INPU0292
  470 PA=ABS(VCOOL(I,J))                                                INPU0293
      IF(PA         -BIG(I)) 441,441,472                                INPU0294
  472 BIG(I)=PA                                                         INPU0295
  441 CONTINUE                                                          INPU0296
  440 BOMAX(I)=BIG(I)                                                   INPU0297
      GO TO 487                                                         INPU0298
  480 DO 481 I=1,MB                                                     INPU0299
  481 BOMAX(I)=BMAX(I)                                                  INPU0300
  487 CONTINUE                                                          INPU0301
      DO 475 I=1,MB                                                     INPU0302
      FRB(I)=BMAX(I)     /R                                             INPU0303
      FRBO(I)=BOMAX(I)     /R                                           INPU0304
  475 CONTINUE                                                          INPU0305
      DO 476 I=1,MW                                                     INPU0306
      FRW(I)=WMAX(I)     /WL                                            INPU0307
  476 CONTINUE                                                          INPU0308
      FRW(3)=ABS(WPHI(3,NDPW))                                          INPU0309
      IDEBUG=0                                                          INPU0310
      IF( IDEBUG.EQ.0) GO TO 477                                        INPU0311
      WRITE(6,5005)(BMAX(I),I=1,MB),(BOMAX(I),I=1,MB),(WMAX(I),I=1,MW)  INPU0312
 5005 FORMAT(///(10X,E15.7/))                                           INPU0313
  477 CONTINUE                                                          INPU0314
      RETURN                                                            INPU0315
      END                                                               INPU0316
```

```
      SUBROUTINE INTPL
C
C        INTERPOLATION FOR THE NUMERICAL INTEGRATION
C
C INTERPOLATION FUNCTION-------HERMIT  INTERPOLATION(2 POINTS)
C INTERPOLATION FUNCTION----LAGRANGIAN INTERPOLATION FOR THE ANGLE OF TWIST
      COMMON/THE/THETN(21) ,MB,MW
      COMMON /PARMT/    ITYPE ,IFLT
      COMMON /AREA1/OMEGA,R,VEL,CL,CD,RAMDA,SNOMEG
      COMMON/WICH/WL,WCOD,WCL,WCD,WCMO,WCMA,EDIS,WTHET,VV
      DIMENSION WX(21)
      COMMON/WING/ NW,NDPW,EMSW(20),G(6,21),DG(6,21),Z(6,21),DZ(6,21),
     *WPHI(6,21),DWPHI(6,21)
      DIMENSION GI(6,20),ZI(6,20),WPHII(6,20)
      COMMON/WIMOD/STR(20,3,6),TSTR(20,6,3)
      DIMENSION XX(21)
      COMMON/GIN/      V(4,21),W(4,21),DV(4,21),DW(4,21),THETA(21),
     %EMS(20),AMASS (20),N,NDP
      COMMON /GINO/     VCOL(4,21),WCOL(4,21),DVCOL(4,21),DWCOL(4,21)
      COMMON /AREA3/VI(4,20),WI(4,20),THETAI(20)  ,AMASSI(20)
     *           ,VICOL(4,20),WICOL(4,20)
      COMMON /AREA2/NPT,XXX(20),A(20)
      IDEBUG=0
      WRITE(6,50)
50    FORMAT(///1X,'*****  BLADE MODE SHAPES  *****')
      XX(1)=0.0
      DO 80 I=1,N
      XX(I+1)=XX(I)+EMS(I)
   80 CONTINUE
      IF(ITYPE.EQ. ) GOTO 100
      WRITE(6,51)
51    FORMAT(/1X,'---  COLLECTIVE MODES  ---')
      DO 36 I=1,MB
      WRITE(6,4500)I
      WRITE(6,4999)
      WRITE(6,3)(J,XX(J),VCOL(I,J),DVCOL(I,J),WCOL(I,J),DWCOL(I,J),
```

```
     *     J=1,NDP)                                                     INTP0037
36    CONTINUE                                                          INTP0038
      WRITE(6,52)                                                       INTP0039
52    FORMAT(//1X,'---  CYCLIC MODES  ---')                             INTP0040
100   CONTINUE                                                          INTP0041
      DO 35 I=1,MB                                                      INTP0042
      WRITE(6,4500)I                                                    INTP0043
4500  FORMAT(/// 1X,'I=',I1)                                            INTP0044
      WRITE(6,4999)                                                     INTP0045
4999  FORMAT(    T5,'J',T13,'XX(J)',T33,'V(I,J)',T53,'DV(I,J)',T73,'W(IINTP0046
     %,J)',T93,'DW(I,J)')                                               INTP0047
      WRITE(6,3)(J,XX(J),V(I,J),DV(I,J),W(I,J),DW(I,J),J=1,NDP)         INTP0048
3     FORMAT((1X,       I4,5(5X,E15.7)))                                INTP0049
35    CONTINUE                                                          INTP0050
      WRITE(6,5999)                                                     INTP0051
5999  FORMAT(/// T5,'J',T13,'XX(J)',T33,'THETN(J)',T53,'AMASS(J)')      INTP0052
      WRITE(6,4)(J,XX(J),THETN(J),AMASS(J),J=1,NDP)                     INTP0053
4     FORMAT((1X,       I4,3(5X,E15.7)))                                INTP0054
      DO 70 II=1,NPT                                                    INTP0055
      DO 60 I=1,NDP                                                     INTP0056
      IF(XX(I).GE.XXX(II)) GO TO 110                                    INTP0057
   60 CONTINUE                                                          INTP0058
  110 EA=XX(I)-XX(I-1)                                                  INTP0059
      EB=XX(I)+XX(I-1)                                                  INTP0060
      XKSI=2.0/EA*XXX(II)-EB/EA                                         INTP0061
      F1=(XKSI+2.0)*(XKSI-1.0)**2/4.0                                   INTP0062
      F2=(2.0-XKSI)*(XKSI+1.0)**2/4.0                                   INTP0063
      G1=(XKSI+1.0)*(XKSI-1.0)**2/4.0                                   INTP0064
      G2=(XKSI-1.0)*(XKSI+1.0)**2/4.0                                   INTP0065
      F1L=(1.0-XKSI)/2.0                                                INTP0066
      F2L=(1.0+XKSI)/2.0                                                INTP0067
      DO 90 JJ=1,4                                                      INTP0068
      VI(JJ,II)=V(JJ,I-1)*F1+V(JJ,I)*F2+(DV(JJ,I-1)*G1+DV(JJ,I)*G2)     INTP0069
     &*EA/2.0*R                                                         INTP0070
      WI(JJ,II)=W(JJ,I-1)*F1+W(JJ,I)*F2+(DW(JJ,I-1)*G1+DW(JJ,I)*G2)     INTP0071
     &*EA/2.0*R                                                         INTP0072
```

```
      IF(ITYPE.EQ.0)GO TO 90
      VICOL(JJ,II)=VCOL(JJ,I-1)*F1+VCOL(JJ,I)*F2+(DVCOL(JJ,I-1)*G1
     *   +DVCOL(JJ,I)*G2)*EA/2.0*R
      WICOL(JJ,II)=WCOL(JJ,I-1)*F1+WCOL(JJ,I)*F2+(DWCOL(JJ,I-1)*G1
     *   +DWCOL(JJ,I)*G2)*EA/2.0*R
   90 CONTINUE
      AMASSI(II)=AMASS(I-1)*F1L+AMASS(I)*F2L
      THETAI(II)=THETA( I-1)*F1L+THETA(I )*F2L
   70 CONTINUE
      IF(IDEBUG.EQ.0) GO TO 400
      WRITE(6,5048)
5048  FORMAT(/// T5,'J',T13,'XXX(J)',T33,'VI(1,J)',T53,'WI(1,J)',T73,
     @'VI(2,J)',T93,'WI(2,J)')
      WRITE(6,3)(JJ,XXX(JJ),VI(1,JJ),WI(1,JJ),VI(2,JJ),WI(2,JJ),JJ=1,
     %NPT)
      WRITE(6,5047)
5047  FORMAT(/// T5,'J',T13,'XXX(J)',T33,'VI(3,J)',T53,'WI(3,J)',T73,
     @'VI(4,J)',T93,'WI(4,J)')
      WRITE(6,3)(JJ,XXX(JJ),VI(3,JJ),WI(3,JJ),VI(4,JJ),WI(4,JJ),JJ=1,
     %NPT)
      WRITE(6,5046)
5046  FORMAT(/// T5,'J',T13,'XXX(J)',T33, 'AMASSI(J)',T53,'THETAI(J)')
      WRITE(6,4)(JJ,XXX(JJ),AMASSI(JJ),THETAI(JJ),JJ=1,NPT)
400   CONTINUE
      WRITE(6,53)
53    FORMAT(///1X,'*****  WING MODE SHAPES  *****')
      WX(1)=0.0
      DO 81 I=1,NW
      WX(I+1)=WX(I)+EMSW(I)
81    CONTINUE
      DO 38 II=1,MW
      WRITE(6,7000)II
7000  FORMAT(/// 1X,'II=',I1)
      WRITE(6,7001)
7001  FORMAT(      T5,'J',T9,'WX(J)',T25,'G(II,J)',T41,'DG(II,J)',T57,
     *'Z(II,J)',T73,'DZ(II,J)',T89,'WPHI(II,J)',T105,'DWPHI(II,J)')
```

```
      WRITE(6,5)(J,WX(J),G(II,J),DG(II,J),Z(II,J),DZ(II,J),WPHI(II,J),
     %DWPHI(II,J),J=1,NDPW)
5     FORMAT((1X,     I4,7(1X,E15.7)))
38    CONTINUE
      DO 10 II=1,NPT
      DO 20 I=1,NDPW
      IF(WX(I).GE.XXX(II)) GO TO 220
20    CONTINUE
220   WEA=WX(I)-WX(I-1)
      WEB=WX(I)+WX(I-1)
      XKSI=2.0/WEA*XXX(II)-WEB/WEA
      F1=(XKSI+2.0)*(XKSI-1.0)**2/4.0
      F2=(2.0-XKSI)*(XKSI+1.0)**2/4.0
      G1=(XKSI+1.0)*(XKSI-1.0)**2/4.0
      G2=(XKSI-1.0)*(XKSI+1.0)**2/4.0
      DO 37 IJ=1,6
      GI(IJ,II)=G(IJ,I-1)*F1+G(IJ,I)*F2+(DG(IJ,I-1)*G1+DG(IJ,I)*G2)*
     *WEA/2.0 *WL
      ZI(IJ,II)=Z(IJ,I-1)*F1+Z(IJ,I)*F2+(DZ(IJ,I-1)*G1+DZ(IJ,I)*G2)*
     *WEA/2.0 *WL
      WPHII(IJ,II)=WPHI(IJ,I-1)*F1+WPHI(IJ,I)*F2+(DWPHI(IJ,I-1)*G1+
     @DWPHI(IJ,I)*G2)*WEA/2.0*WL
      STR(II,1,IJ)=GI(IJ,II)
      STR(II,2,IJ)=ZI(IJ,II)
      STR(II,3,IJ)=WPHII(IJ,II)
      TSTR(II,IJ,1)=GI(IJ,II)
      TSTR(II,IJ,2)=ZI(IJ,II)
      TSTR(II,IJ,3)=WPHII(IJ,II)
37    CONTINUE
10    CONTINUE
      IF(IDEBUG.EQ.0) GO TO 401
      WRITE(6,7003)
7003  FORMAT(/// T5,'J',T9,'XXX(J)',T25,'GI(1,J)',T41, 'HI(1,J)',T57,
     %'WPHII(1,J)',T73,'GI(2,J)',T89,'H(2,J)',T105,'WPHII(2,J)')
      WRITE(6,5)(J,XXX(J),GI(1,J),ZI(1,J),WPHII(1,J),GI(2,J),ZI(2,J),
     @WPHII(2,J),J=1,NPT)
```

```
      WRITE(6,7004)                                                     INTP0145
7004  FORMAT(/// T5,'J',T9,'XXX(J)',T25,'GI(3,J)',T41, 'HI(3,J)',T57,   INTP0146
     %'WPHII(3,J)',T73,'GI(4,J)',T89,'H(4,J)',T105,'WPHII(4,J)')        INTP0147
      WRITE(6,5)(J,XXX(J),GI(3,J),ZI(3,J),WPHII(3,J),GI(4,J),ZI(4,J),   INTP0148
     @WPHII(4,J),J=1,NPT)                                               INTP0149
      WRITE(6,7005)                                                     INTP0150
7005  FORMAT(/// T5,'J',T9,'XXX(J)',T25,'GI(5,J)',T41, 'HI(5,J)',T57,   INTP0151
     %'WPHII(5,J)',T73,'GI(6,J)',T89,'H(6,J)',T105,'WPHII(6,J)')        INTP0152
      WRITE(6,5)(J,XXX(J),GI(5,J),ZI(5,J),WPHII(5,J),GI(6,J),ZI(6,J),   INTP0153
     @WPHII(6,J),J=1,NPT)                                               INTP0154
401   CONTINUE                                                          INTP0155
      RETURN                                                            INTP0156
      END                                                               INTP0157
```

```
      SUBROUTINE AERODT                                                 AER00001
C                                                                       AER00002
C        TO DEFINE THE AERODYNAMIC COEFFICIENTS AT THE POINTS OF GAUSSIAN AER00003
C        QUADRATURE                                                     AER00004
C                                                                       AER00005
      COMMON /PARMT/    ITYPE ,IFLT                                     AER00006
      COMMON /AREA1/OMEGA,R,VEL,CL,CD,RAMDA,SNOMEG                      AER00007
      COMMON /AREA6/NOBLD,ROH,CHOD,AIB,CK,HMAST    ,ALOCK ,ANO,HR       AER00008
      COMMON /AREA2/NPT,XXX(20),A(20)                                   AER00009
      COMMON /AREA3/VI(4,20),WI(4,20),THETAI(20) ,AMASSI(20)            AER00010
     *                   ,VICOL(4,20),WICOL(4,20)                       AER00011
      COMMON/AREA4/H(4,4,20),HI(4,20),HII(4,20),HRZ(4,20),HNR(4,20),    AER00012
     %WO(4,20),VO(4,20),W1(4,20),V1(4,20)                               AER00013
     @        ,OH(4,4,20),OWO(4,20),OV1(4,20),OHIII(4,20),CHIII(4,20)   AER00014
     @         ,CHV(4,20),OHIV(4,20)                                    AER00015
      COMMON/AKKH/FT0P1(20),FT1P0(20),FT1P2(20),FT2P1(20), FZ2P0(20),   AER00016
     /FZ0P0(20),FZ1P1(20),FZ2P2(20)                                     AER00017
     @        ,FT3P0(20),FT3P1(20),FZ3P0(20),FZ3P1(20)                  AER00018
      COMMON/ADFE/HIII(4,20),HIV(4,20),HV(4,20),HVI(4,20),HVII(4,20)    AER00019
      COMMON/WIMOD/STR(20,3,6),TSTR(20,6,3)                             AER00020
      COMMON/WICH/WL,WCOD,WCL,WCD,WCMO,WCMA,EDIS,WTHET,VV               AER00021
      COMMON/WINGAR/TSDS(20,6,6),TSAS(20,6,6),TSAG(20,6,3)              AER00022
      DIMENSION DAWA(3,3),AWA(3,3),AWG(3,3)                             AER00023
      DIMENSION TSDA(20,6,3),TSA(20,6,3)                                AER00024
      DO 1 I=1,3                                                        AER00025
      DO 1 J=1,3                                                        AER00026
      DAWA(I,J)=0.0                                                     AER00027
      AWA(I,J)=0.0                                                      AER00028
1     AWG(I,J)=0.0                                                      AER00029
      DO 2 I=1,20                                                       AER00030
      DO 2 J=1,6                                                        AER00031
      DO 2 K=1,3                                                        AER00032
      TSDA(I,J,K)=0.0                                                   AER00033
2     TSA(I,J,K)=0.0                                                    AER00034
      CK=-0.5*ROH*CL*CHOD*R**4                                          AER00035
      CA=CD/CL                                                          AER00036
```

```
      CA1=1.0+CA                                                        AER00037
      CA2=1.0-CA                                                        AER00038
      ARAMDA=RAMDA                                                      AER00039
      RAMDA=ABS(RAMDA)                                                  AER00040
      DO 11 JJ=1,NPT                                                    AER00041
      XSQ=SQRT(RAMDA**2+XXX(JJ)**2)                                     AER00042
      TAU0=1.0/XSQ                                                      AER00043
      TAU1=XXX(JJ)/XSQ                                                  AER00044
      TAU2=XXX(JJ)**2/XSQ                                               AER00045
      TAU3=XXX(JJ)**3/XSQ                                               AER00046
      ALPHA=THETAI(JJ)-ATAN( RAMDA/XXX(JJ)  )+ATAN(      RAMDA*4.0/3.0) AER00047
      FTH0=RAMDA**3*ALPHA*TAU0+RAMDA**2   *CA*TAU1+RAMDA*ALPHA*TAU2     AER00048
     /+CA*TAU3                                                          AER00049
      FTH1=RAMDA**2*CA1*TAU0+RAMDA*ALPHA*TAU1+2.0*CA*TAU2               AER00050
      FTH2=2.0*RAMDA**2*ALPHA*TAU0-RAMDA*CA2*TAU1+ALPHA*TAU2            AER00051
      FTH3=RAMDA**3*TAU0+RAMDA*TAU2                                     AER00052
      FTH0=-FTH0*SNOMEG                                                 AER00053
      FTH1=-FTH1                                                        AER00054
      FTH2=-FTH2*SNOMEG                                                 AER00055
      FTH3=-FTH3*SNOMEG                                                 AER00056
      FZ0=-RAMDA**3*CA*TAU0+RAMDA**2*ALPHA*TAU1   -RAMDA*CA*TAU2        AER00057
     /+ALPHA*TAU3                                                       AER00058
      FZ1=RAMDA**2*ALPHA*TAU0+RAMDA*CA2*TAU1+2.0*ALPHA*TAU2             AER00059
      FZ2=-2.0*RAMDA**2*CA*TAU0+RAMDA*ALPHA*TAU1-CA1*TAU2               AER00060
      FZ3=RAMDA**2*TAU1+TAU3                                            AER00061
      FZ1=FZ1*SNOMEG                                                    AER00062
      DO 200 J=1,4                                                      AER00063
      DO 100 I=1,4                                                      AER00064
      H(J,I,JJ)=FTH1*VI(J,JJ)*VI(I,JJ)+FZ1*WI(J,JJ)*VI(I,JJ)+FTH2*      AER00065
     @VI(J,JJ)*WI(I,JJ)+FZ2*WI(J,JJ)*WI(I,JJ)                           AER00066
       IF(ITYPE.EQ.0) GO TO 100                                         AER00067
      CH(J,I,JJ)=FTHI*VICOL(J,JJ)*VICOL(I,JJ)+FZ1*WICOL(J,JJ)*          AER00068
     @           VICOL(I,JJ)+FTH2*VICOL(J,JJ)*WICOL(I,JJ)+FZ2*WICOL     AER00069
     %           (J,JJ) *WICOL(I,JJ)                                    AER00070
  100 CONTINUE                                                          AER00071
      HI(J,JJ)=FTHI*VI(J,JJ)+FZ1*WI(J,JJ)                               AER00072
```

```
      HII(J,JJ)=(FTH2*VI(J,JJ)+FZ2*WI(J,JJ))*XXX(JJ)                    AERO0073
      HRZ(J,JJ)=FTH2*VI(J,JJ)+FZ2*WI(J,JJ)                              AERO0074
      HNR(J,JJ)=(FTH1*VI(J,JJ)+FZ1*WI(J,JJ))*XXX(JJ)                    AERO0075
      W0(J,JJ)=AMASSI(JJ)*WI(J,JJ)                                      AERO0076
      V0(J,JJ)=AMASSI(JJ)*VI(J,JJ)                                      AERO0077
      W1(J,JJ)=AMASSI(JJ)*WI(J,JJ)*XXX(JJ)                              AERO0078
      V1(J,JJ)=AMASSI(JJ)*VI(J,JJ)*XXX(JJ)                              AERO0079
      HIII(J,JJ)=FTH1*VI(J,JJ)+FTH2*WI(J,JJ)                            AERO0080
      HIV(J,JJ)=HIII(J,JJ)*XXX(JJ)                                      AERO0081
      HV(J,JJ)=FZ1*VI(J,JJ)+FZ2*WI(J,JJ)                                AERO0082
      HVI(J,JJ)=HV(J,JJ)*XXX(JJ)                                        AERO0083
      HVII(J,JJ)=FZ0*VI(J,JJ)-FTH0*WI(J,JJ)                             AERO0084
      CHIII(J,JJ)=FZ3*WI(J,JJ)+FTH3*VI(J,JJ)                            AERO0085
      IF(ITYPE.EQ.0) GO TO 200                                          AERO0086
      OW0(J,JJ)=AMASSI(JJ)*WICOL(J,JJ)                                  AERO0087
      OV1(J,JJ)=AMASSI(JJ)*VICOL(J,JJ)*XXX(JJ)                          AERO0088
      HRZ(J,JJ)=FTH2*VICOL(J,JJ)+FZ2*WICOL(J,JJ)                        AERO0089
      HNR(J,JJ)=(FTH1*VICOL(J,JJ)+FZ1*WICOL(J,JJ))*XXX(JJ)              AERO0090
      OHIII(J,JJ)=FZ3*WICOL(J,JJ)+FTH3*VICOL(J,JJ)                      AERO0091
      OHV(J,JJ)=FZ1*VICOL(J,JJ)+FZ2*WICOL(J,JJ)                         AERO0092
      OHIV(J,JJ)=(FTH1*VICOL(J,JJ)+FTH2*WICOL(J,JJ))*XXX(JJ)            AERO0093
  200 CONTINUE                                                          AERO0094
C    AERO FOR WING DUE TO BLADES                                        AERO0095
      FT0P1(JJ)   =FTH0*XXX(JJ)                                         AERO0096
      FT1P0(JJ)=FTH1                                                    AERO0097
      FT1P2(JJ)=FTH1*XXX(JJ)**2                                         AERO0098
      FT2P1(JJ)=FTH2*XXX(JJ)                                            AERO0099
      FZ0P0(JJ)=FZ0                                                     AERO0100
      FZ1P1(JJ)=FZ1*XXX(JJ)                                             AERO0101
      FZ2P0(JJ)=FZ2                                                     AERO0102
      FZ2P2(JJ)=FZ2*XXX(JJ)**2                                          AERO0103
      FT3P0(JJ)=FTH3                                                    AERO0104
      FT3P1(JJ)=FTH3*XXX(JJ)                                            AERO0105
      FZ3P0(JJ)=FZ3                                                     AERO0106
      FZ3P1(JJ)=FZ3*XXX(JJ)                                             AERO0107
   11 CONTINUE                                                          AERO0108
```

```
C     AERO FOR WING DUE TO ITSELF                                        AERO0109
      RAMDA=ARAMDA                                                       AERO0110
      RC=ROH*WCOD                                                        AERO0111
      DAWA(1,1)=-0.5*RC*VEL*(WCL+WCD)                                    AERO0112
      DAWA(1,2)=RC*WCL*WTHET*VEL                                         AERO0113
      DAWA(2,1)=-0.5*RC*WCL*WTHET*VEL                                    AERO0114
      DAWA(2,2)=-RC*WCD*VEL                                              AERO0115
      DAWA(3,1)=-0.5*RC*WCOD*VEL*(WCL*EDIS+WCMA)                         AERO0116
      DAWA(3,2)=RC*WCOD*(WCMO+WCMA*WTHET+WCL*WTHET*EDIS)*VEL             AERO0117
      AWA(1,3)=0.5*RC*WCL*VEL**2                                         AERO0118
      AWA(3,3)=0.5*RC*WCOD*VEL**2*(WCMA+WCL*EDIS)                        AERO0119
      AWG(1,1)=DAWA(1,1)*VEL                                             AERO0120
      AWG(1,3)=DAWA(1,2)*VEL                                             AERO0121
      AWG(2,1)=DAWA(2,1)*VEL                                             AERO0122
      AWG(2,3)=DAWA(2,2)*VEL                                             AERO0123
      AWG(3,1)=DAWA(3,1)*VEL                                             AERO0124
      AWG(3,3)=DAWA(3,2)*VEL                                             AERO0125
      DO 501 II=1,NET                                                    AERO0126
      DO 502 I=1,6                                                       AERO0127
      DO 502 J=1,3                                                       AERO0128
      DO 502 K=1,3                                                       AERO0129
      TSDA(II,I,J)=TSTR(II,I,K)*DAWA(K,J)+TSDA(II,I,J)                   AERO0130
      TSA(II,I,J)=TSTR(II,I,K)*AWA(K,J)+TSA(II,I,J)                      AERO0131
      TSAG(II,I,J)=TSTR(II,I,K)*AWG(K,J)+TSAG(II,I,J)                    AERO0132
  502 CONTINUE                                                           AERO0133
      DO 503 I=1,6                                                       AERO0134
      DO 503 J=1,6                                                       AERO0135
      DO 503 K=1,3                                                       AERO0136
      TSDS(II,I,J)=TSDA(II,I,K)*STR(II,K,J)+TSDS(II,I,J)                 AERO0137
      TSAS(II,I,J)=TSA(II,I,K)*STR(II,K,J)+TSAS(II,I,J)                  AERO0138
  503 CONTINUE                                                           AERO0139
  501 CONTINUE                                                           AERO0140
      RETURN                                                             AERO0141
      END                                                                AERO0142
```

```
      SUBROUTINE ORDINT                                                 ORDI0001
C                                                                       ORDI0002
C         TO DEFINE THE ORDER OF    NUMERICAL INTEGRATION               ORDI0003
C                                                                       ORDI0004
      COMMON /PARMT/    ITYPE ,IFLT                                     ORDI0005
      COMMON /AREA1/OMEGA,R,VEL,CL,CD,RAMDA,SNOMEG                      ORDI0006
      COMMON /AREA6/NOBLD,ROH,CHOD,AIB,CK,HMAST    ,ALOCK ,ANO,HR       ORDI0007
      COMMON/AREA5/AH(4,4),AHI(4),AHII(4),AHRZ(4),AHNR(4),AWO(4),AVO(4) ORDI0008
     @,AW1(4),AV1(4),OAH(4,4),OAWO(4),OAV1(4),OAHIII(4),CAHIII(4)       ORDI0009
     @         ,OAHV(4),OAHIV(4)                                        ORDI0010
      COMMON/AERO/AFT0P1,AFT1P0,AFT1P2,AFT2P1,AFZ0P0,AFZ1P1,AFZ2P0,     ORDI0011
     @AFZ2P2,AHIII(4),AHIV(4),AHV(4),AHVI(4),AHVII(4)                   ORDI0012
     @         ,AFT3P0,AFT3P1,AFZ3P0,AFZ3P1                             ORDI0013
      COMMON/WICH/WL,WCDD,WCL,WCD,WCMO,WCMA,EDIS,WTHET,VV               ORDI0014
      COMMON/ARWNG/DARWA(6,6),ARWA(6,6),ARWG(6,3)                       ORDI0015
      IDEBUG=0                                                          ORDI0016
      NN=1                                                              ORDI0017
      DO 100 JQ=1,4                                                     ORDI0018
      DO 100 IQ=1,4                                                     ORDI0019
      CALL INTEG(FSUM,NN,JQ,IQ)                                         ORDI0020
      AH(JQ,IQ)=CK*FSUM/R**2                                            ORDI0021
100   CONTINUE                                                          ORDI0022
      NK=NN                                                             ORDI0023
1000  NN=NN+1                                                           ORDI0024
      DO 200 JQ=1,4                                                     ORDI0025
      CALL INTEG(FSUM,NN,JQ,IQ)                                         ORDI0026
      FSUM=FSUM*CK/R                                                    ORDI0027
      NM=NN-NK                                                          ORDI0028
      GO TO (2,3,4,5),NM                                                ORDI0029
2     AHI(JQ)=FSUM                                                      ORDI0030
      GO TO 200                                                         ORDI0031
3     AHII(JQ)=FSUM                                                     ORDI0032
      GO TO 200                                                         ORDI0033
4     AHRZ(JQ)=FSUM                                                     ORDI0034
      GO TO 200                                                         ORDI0035
5     AHNR(JQ)=FSUM                                                     ORDI0036
```

```
200    CONTINUE
       IF(NN.LT.5) GO TO 1000
       NK=NN
2000   NN=NN+1
       DO 300 JQ=1,4
       CALL INTEG(FSUM,NN,JQ,IQ)
       FSUM=FSUM*R**2
       NM=NN-NK
       GO TO (6,7,8,9),NM
6      AWO(JQ)=FSUM
       GO TO 300
7      AVO(JQ)=FSUM
       GO TO 300
8      AW1(JQ)=FSUM
       GO TO 300
9      AV1(JQ)=FSUM
300    CONTINUE
       IF(NN.LT.9) GO TO 2000
       NK=NN
3000   NN=NN+1
       CALL INTEG(FSUM,NN,JQ,IQ)
       FSUM=FSUM*CK*ANO
       NM=NN-NK
       GO TO (  10,11,12,13,14,15,16,17),NM
10     AFTOP1=FSUM
       GO TO 400
11     AFT1P0=FSUM
       GO TO 400
12     AFT1P2=FSUM
       GO TO 400
13     AFT2P1=FSUM
       GO TO 400
14     AFZOP0=FSUM
       GO TO 400
15     AFZ1P1=FSUM
       GO TO 400
```

```
16    AFZ2PO=FSUM                                    ORDI0073
      GO TO 400                                      ORDI0074
17    AFZ2P2=FSUM                                    ORDI0075
400   IF(NN.LT.17) GO TO 3000                        ORDI0076
      NK=NN                                          ORDI0077
4000  NN=NN+1                                        ORDI0078
      DO 500 JQ=1,4                                  ORDI0079
      CALL INTEG(FSUM,NN,JQ,IQ)                      ORDI0080
      FSUM=FSUM/R*CK*ANO                             ORDI0081
      NM=NN-NK                                       ORDI0082
      GO TO (    18,19,20,21,22),NM                  ORDI0083
18    AHIII(JQ)=FSUM                                 ORDI0084
      GO TO 500                                      ORDI0085
19    AHIV(JQ)=FSUM                                  ORDI0086
      GO TO 500                                      ORDI0087
20    AHV(JQ)=FSUM                                   ORDI0088
      GO TO 500                                      ORDI0089
21    AHVI(JQ)=FSUM                                  ORDI0090
      GO TO 500                                      ORDI0091
22    AHVII(JQ)=FSUM                                 ORDI0092
500   CONTINUE                                       ORDI0093
      IF(NN.LT.22) GO TO 4000                        ORDI0094
      NK=NN                                          ORDI0095
5000  NN=NN+1                                        ORDI0096
      DO 600 JQ=1,6                                  ORDI0097
      DO 600 IQ=1,6                                  ORDI0098
      CALL INTEG(FSUM,NN,JQ,IQ)                      ORDI0099
      NM=NN-NK                                       ORDI0100
      GO TO (23,24),NM                               ORDI0101
23    FSUM=FSUM*WL/ABS(OMEGA)                        ORDI0102
      CARWA(JQ,IQ)=FSUM                              ORDI0103
      GO TO 600                                      ORDI0104
24    FSUM=FSUM*WL/OMEGA**2                          ORDI0105
      ARWA(JQ,IQ)=FSUM                               ORDI0106
600   CONTINUE                                       ORDI0107
      IF(NN.LT.24) GO TO 5000                        ORDI0108
```

```
      NN=25                                                             ORDI0109
      DO 700 JQ=1,6                                                     ORDI0110
      DO 700 IQ=1,3                                                     ORDI0111
      CALL INTEG(FSUM,NN,JQ,IQ)                                         ORDI0112
      FSUM=FSUM*WL/OMEGA**2                                             ORDI0113
      ARWG(JQ,IQ)=FSUM                                                  ORDI0114
700   CONTINUE                                                          ORDI0115
      NN=28                                                             ORDI0116
      DO 801 JQ=1,4                                                     ORDI0117
      CALL INTEG(FSUM,NN,JQ,IQ)                                         ORDI0118
      FSUM=FSUM*CK/R                                                    ORDI0119
      CAHIII(JQ)=FSUM                                                   ORDI0120
801   CONTINUE                                                          ORDI0121
      NK=NN                                                             ORDI0122
2003  NN=NN+1                                                           ORDI0123
      CALL INTEG(FSUM,NN,JQ,IQ)                                         ORDI0124
      FSUM=FSUM*CK*ANO                                                  ORDI0125
      NM=NN-NK                                                          ORDI0126
      GOTO (27,28,29,30),NM                                             ORDI0127
27    AFT3P0=FSUM                                                       ORDI0128
      GO TO 2003                                                        ORDI0129
28    AFT3P1=FSUM                                                       ORDI0130
      GO TO 2003                                                        ORDI0131
29    AFZ3P0=FSUM                                                       ORDI0132
      GO TO 2003                                                        ORDI0133
30    AFZ3P1=FSUM                                                       ORDI0134
      IF(IDEBUG.EQ.0) GO TO 450                                         ORDI0135
      WRITE(6,50)((AH(I,J),J=1,4),I=1,4)                                ORDI0136
50    FORMAT(////1X,2X, 'AH',4(T10,4(E15.7,2X)/1X ))                    ORDI0137
      WRITE(6,51)AHI,AHII,AHRZ,AHNR,AHIII,AHIV,AHV,AHVI,AHVII,CAHIII    ORDI0138
51    FORMAT(2X,'AHI',T10,4(E15.7,2X)                                   ORDI0139
     %       /2X,'AHII',       T10,4(E15.7,2X)                          ORDI0140
     %       /2X,'AHRZ',       T10,4(E15.7,2X)                          ORDI0141
     %       /2X,'AHNR',       T10,4(E15.7,2X)                          ORDI0142
     *       /2X,'AHIII',      T10,4(E15.7,2X)                          ORDI0143
     %       /2X,'AHIV',       T10,4(E15.7,2X)                          ORDI0144
```

```
     @         /2X,'AHV',         T10,4(E15.7,2X)                           ORDI0145
     *         /2X,'AHVI',        T10,4(E15.7,2X)                           ORDI0146
     %         /2X,'AHVII',       T10,4(E15.7,2X)                           ORDI0147
     @         /2X,'CAHIII',      T10,4(E15.7,2X)          )                ORDI0148
      WRITE(6,82) AFT3P0,AFT3P1,AFZ3P0,AFZ3P1                               ORDI0149
82    FORMAT(2X,'AFT3P0'     ,T10,E15.7                                     ORDI0150
     *         /2X,'AFT3P1'     ,T10,E15.7                                  ORDI0151
     *         /2X,'AFZ3P0'     ,T10,E15.7                                  ORDI0152
     *         /2X,'AFZ3P1'     ,T10,E15.7                )                 ORDI0153
450   CONTINUE                                                              ORDI0154
      IF(ITYPE.EQ.0) GO TO 851                                              ORDI0155
      NN=NN+1                                                               ORDI0156
      DO 101 JQ=1,4                                                         ORDI0157
      DO 101 IQ=1,4                                                         ORDI0158
      CALL INTEG(FSUM,NN,JQ,IQ)                                             ORDI0159
      OAH(JQ,IQ)=CK*FSUM/R**2                                               ORDI0160
101   CONTINUE                                                              ORDI0161
      NK=NN                                                                 ORDI0162
2001  NN=NN+1                                                               ORDI0163
      DO 301 JQ=1,4                                                         ORDI0164
      CALL INTEG(FSUM,NN,JQ,IQ)                                             ORDI0165
      FSUM=FSUM*R**2                                                        ORDI0166
      NM=NN-NK                                                              ORDI0167
      GC TO (32,33),NM                                                      ORDI0168
32    OAWO(JQ)=FSUM                                                         ORDI0169
      GO TO 301                                                             ORDI0170
33    OAV1(JQ)=FSUM                                                         ORDI0171
301   CONTINUE                                                              ORDI0172
      IF(NN.LT.33) GO TO 2001                                               ORDI0173
      NN=NN+1                                                               ORDI0174
      DO 302 JQ=1,4                                                         ORDI0175
      CALL INTEG(FSUM,NN,JQ,IQ)                                             ORDI0176
      FSUM=FSUM*CK/R                                                        ORDI0177
      OAHIII(JQ)=FSUM                                                       ORDI0178
302   CONTINUE                                                              ORDI0179
      NN=NN+1                                                               ORDI0180
```

```
      DO 780 JQ=1,4                                                     ORDI0181
      CALL INTEG(FSUM,NN,JQ,IQ)                                         ORDI0182
      FSUM=FSUM*CK/R*ANO                                                ORDI0183
      OAHV(JQ)=FSUM                                                     ORDI0184
780   CONTINUE                                                          ORDI0185
      NN=NN+1                                                           ORDI0186
      DO 781 JQ=1,4                                                     ORDI0187
      CALL INTEG(FSUM,NN,JQ,IQ)                                         ORDI0188
      FSUM=FSUM*CK/R*ANO                                                ORDI0189
      OAHIV(JQ)=FSUM                                                    ORDI0190
781   CONTINUE                                                          ORDI0191
      IF (IDEBUG.EQ.0) GO TO 851                                        ORDI0192
      WRITE(6,80) ((OAH(I,J),J=1,4),I=1,4)                              ORDI0193
80    FORMAT(2X,'OAH',      4(T10,4(E15.7,2X)/1X))                      ORDI0194
      WRITE(6,81) OAWO,OAV1,OAHIII ,OAHV,OAHIV                          ORDI0195
81    FORMAT(2X,'OAWO'       ,T10,4(E15.7,2X)                           ORDI0196
     *      /2X,'OAV1'       ,T10,4(E15.7,2X)                           ORDI0197
     *      /2X,'OAHIII'     ,T10,4(E15.7,2X)                           ORDI0198
     *      /2X,'OAHV'       ,T10,4(E15.7,2X)                           ORDI0199
     *      /2X,'OAHIV'      ,T10,4(E15.7,2X)      )                    ORDI0200
851   RETURN                                                            ORDI0201
      END                                                               ORDI0202
```

```
      SUBROUTINE INTEG(FSUM,NN,JQ,IQ)
C
C NUMERICAL INTEGRATION----GAUSSIAN QUADRATURE
C
      COMMON /AREA2/NPT,XXX(20),A(20)
      SUM=0.0
      DO 40 JJJ=1,NPT
      X=XXX(JJJ)
      SUM=SUM+A(JJJ)*F(X,NN,JJJ,JQ,IQ)
   40 CONTINUE
      FSUM=0.5*SUM
      RETURN
      END
```

```
      FUNCTION F(X,NN,JJJ,JQ,IQ)
C
C         TO DEFINE THE INTEGRAND FUNCTIONS
C
      COMMON/AREA4/H(4,4,20),HI(4,20),HII(4,20),HRZ(4,20),HNR(4,20),
     %W0(4,20),V0(4,20),W1(4,20),V1(4,20)
     @        ,OH(4,4,20),OW0(4,20),OV1(4,20),OHIII(4,20),CHIII(4,20)
     @         ,OHV(4,20),OHIV(4,20)
      COMMON/AKKH/FT0P1(20),FT1P0(20),FT1P2(20),FT2P1(20), FZ2P0(20),
     /FZ0P0(20),FZ1P1(20),FZ2P2(20)
     @        ,FT3P0(20),FT3P1(20),FZ3P0(20),FZ3P1(20)
      COMMON/ADFB/HIII(4,20),HIV(4,20),HV(4,20),HVI(4,20),HVII(4,20)
      COMMON/WINGAR/TSDS(20,6,6),TSAS(20,6,6),TSAG(20,6,3)
      GO TO(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,
     @21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36),NN
 1    F=H(JQ,IQ,JJJ)
      RETURN
 2    F=HI(JQ,JJJ)
      RETURN
 3    F=HII(JQ,JJJ)
      RETURN
 4    F=HRZ(JQ,JJJ)
      RETURN
 5    F=HNR(JQ,JJJ)
      RETURN
 6    F= W0(JQ,JJJ)
      RETURN
 7    F= V0(JQ,JJJ)
      RETURN
 8    F= W1(JQ,JJJ)
      RETURN
 9    F= V1(JQ,JJJ)
      RETURN
 10   F=FT0P1(JJJ)
      RETURN
 11   F=FT1P0(JJJ)
```

```
      RETURN                                                            F    0037
   12 F=FT1P2(JJJ)                                                      F    0038
      RETURN                                                            F    0039
   13 F=FT2P1(JJJ)                                                      F    0040
      RETURN                                                            F    0041
   14 F=FZ0P0(JJJ)                                                      F    0042
      RETURN                                                            F    0043
   15 F=FZ1P1(JJJ)                                                      F    0044
      RETURN                                                            F    0045
   16 F=FZ2P0(JJJ)                                                      F    0046
      RETURN                                                            F    0047
   17 F=FZ2P2(JJJ)                                                      F    0048
      RETURN                                                            F    0049
   18 F=HIII(JQ,JJJ)                                                    F    0050
      RETURN                                                            F    0051
   19 F=HIV(JQ,JJJ)                                                     F    0052
      RETURN                                                            F    0053
   20 F=HV(JQ,JJJ)                                                      F    0054
      RETURN                                                            F    0055
   21 F=HVI(JQ,JJJ)                                                     F    0056
      RETURN                                                            F    0057
   22 F=HVII(JQ,JJJ)                                                    F    0058
      RETURN                                                            F    0059
   23 F=TSDS(JJJ,JQ,IQ)                                                 F    0060
      RETURN                                                            F    0061
   24 F=TSAS(JJJ,JQ,IQ)                                                 F    0062
      RETURN                                                            F    0063
   25 F=TSAG(JJJ,JQ,IQ)                                                 F    0064
      RETURN                                                            F    0065
   26 F=CHIII(JQ,JJJ)                                                   F    0066
      RETURN                                                            F    0067
   27 F=FT3P0(JJJ)                                                      F    0068
      RETURN                                                            F    0069
   28 F=FT3P1(JJJ)                                                      F    0070
      RETURN                                                            F    0071
   29 F=FZ3P0(JJJ)                                                      F    0072
```

```
      RETURN
30    F=FZ3P1(JJJ)
      RETURN
31    F=OH(JQ,IG,JJJ)
      RETURN
32    F=OWO(JQ,JJJ)
      RETURN
33    F=OV1(JQ,JJJ)
      RETURN
34    F=OHIII(JQ,JJJ)
      RETURN
35    F=OHV(JQ,JJJ)
      RETURN
36    F=OHIV(JQ,JJJ)
      RETURN
      END
```

```
      SUBROUTINE AINER                                                  AINE0001
C                                                                       AINE0002
C        TO DEFINE THE EQUATION'S COEFFICIENTS IN MATRIX FORM RELATING TO AINE0003
C        INERTIA TERMS                                                  AINE0004
C                                                                       AINE0005
      COMMON /PARMT/    ITYPE ,IFLT                                     AINE0006
      COMMON /AREA1/OMEGA,R,VEL,CL,CD,RAMDA,SNOMEG                      AINE0007
      COMMON /AREA6/NOBLD,ROH,CHOD,AIB,CK,HMAST    ,ALOCK ,ANO,HR       AINE0008
      COMMON/AREA5/AH(4,4),AHI(4),AHII(4),AHRZ(4),AHNR(4),AWO(4),AVO(4) AINE0009
     @,AW1(4),AV1(4),OAH(4,4),OAWO(4),OAV1(4),OAHIII(4),CAHIII(4)       AINE0010
     @          ,OAHV(4),OAHIV(4)                                       AINE0011
      COMMON/INERTI/TTMT(4,6,3),TTCTJ(4,6,3),AMJT(4,3,6),CJT(4,3,6)     AINE0012
      COMMON/AMATIC/TT(6,5),C(6,6)     ,T(5,6)                          AINE0013
      DIMENSION AMT(4,5,3),CJ(4,3,5),AM(4,3,5),TCJ(4,5,3)               AINE0014
      DO 50 I=1,4                                                       AINE0015
      DO 50 J=1,5                                                       AINE0016
      DO 50 K=1,3                                                       AINE0017
      AMT(I,J,K)=0.0                                                    AINE0018
   50 CJ(I,K,J)=0.0                                                     AINE0019
      DO 210 NM=1,4                                                     AINE0020
      AMT(NM,1,3)=-AVO(NM)/R                                            AINE0021
      AMT(NM,2,1)=AWO(NM)/R                                             AINE0022
      AMT(NM,3,2)=-AVO(NM)*HR                                           AINE0023
      AMT(NM,3,3)=AW1(NM)                                               AINE0024
      AMT(NM,4,2)=-AW1(NM)                                              AINE0025
      AMT(NM,4,3)=-AVO(NM)*HR                                           AINE0026
      AMT(NM,5,1)=AV1(NM)                                               AINE0027
      IF(ITYPE.EQ.0) GO TO 300                                          AINE0028
      AMT(NM,2,1)=OAWO(NM)/R                                            AINE0029
      AMT(NM,5,1)=OAV1(NM)                                              AINE0030
  300 CONTINUE                                                          AINE0031
      DO 2 I=1,6                                                        AINE0032
      DO 2 J=1,3                                                        AINE0033
      DO 2 K=1,5                                                        AINE0034
    2 TTMT(NM,I,J)=TT(I,K)*AMT(NM,K,J)+TTMT(NM,I,J)                     AINE0035
      DO 3 I=1,6                                                        AINE0036
```

```
      DO 3 J=2,3                                                        AINE0037
3     TTMT(NM,I,J)=0.5*TTMT(NM,I,J)                                     AINE0038
      CJ(NM,2,3)=2.0*AW1(NM)                                            AINE0039
      CJ(NM,3,4)=2.0*AW1(NM)                                            AINE0040
      DO 5 I=1,5                                                        AINE0041
      DO 5 J=1,3                                                        AINE0042
5     TCJ(NM,I,J)=CJ(NM,J,I)                                            AINE0043
      DO 6 I=1,6                                                        AINE0044
      DO 6 J=1,3                                                        AINE0045
      DO 6 K=1,5                                                        AINE0046
6     TTCTJ(NM,I,J)=TT(I,K)*TCJ(NM,K,J)+TTCTJ(NM,I,J)                   AINE0047
      DO 100 I=1,6                                                      AINE0048
      DO 100 J=2,3                                                      AINE0049
100   TTCTJ(NM,I,J)=0.5*TTCTJ(NM,I,J)                                   AINE0050
      DO 7 I=1,5                                                        AINE0051
      DO 7 J=1,6                                                        AINE0052
7     T(I,J)=TT(J,I)                                                    AINE0053
      DO 8 I=1,3                                                        AINE0054
      DO 8 J=1,5                                                        AINE0055
8     AM(NM,I,J)=AMT(NM,J,I)                                            AINE0056
      DO 9 I=1,3                                                        AINE0057
      DO 9 J=1,6                                                        AINE0058
      DO 9 K=1,5                                                        AINE0059
9     AMJT(NM,I,J)=AM(NM,I,K)*T(K,J)+AMJT(NM,I,J)                       AINE0060
      DO 10 I=1,3                                                       AINE0061
      DO 10 J=1,6                                                       AINE0062
      DO 10 K=1,5                                                       AINE0063
10    CJT(NM,I,J)=CJ(NM,I,K)*T(K,J)+CJT(NM,I,J)                         AINE0064
210   CONTINUE                                                          AINE0065
      RETURN                                                            AINE0066
      END                                                               AINE0067
```

```
      SUBROUTINE AEROMT                                                 AER00001
C                                                                       AER00002
C        TO DEFINE THE EQUATION'S COEFFICIENTS IN MATRIX FORM RELATING TO AER00003
C        AERODYNAMIC TERMS                                              AER00004
C                                                                       AER00005
      COMMON /PARMT/    ITYPE ,IFLT                                     AER00006
      COMMON /AREA6/NOBLD,RCH,CHOD,AIB,CK,HMAST    ,ALOCK ,ANO,HR       AER00007
      COMMON/AREA5/AH(4,4),AHI(4),AHII(4),AHRZ(4),AHNR(4),AWO(4),AVO(4) AER00008
     @,AW1(4),AV1(4),OAH(4,4),CAWO(4),OAV1(4),OAHIII(4),CAHIII(4)       AER00009
     @     ,CAHV(4),OAHIV(4)                                            AER00010
      COMMON /AREA1/OMEGA,R,VEL,CL,CD,RAMDA,SNOMEG                      AER00011
      COMMON/AERO/AFT0P1,AFT1P0,AFT1P2,AFT2P1,AFZ0P0,AFZ1P1,AFZ2P0,     AER00012
     @AFZ2P2,AHIII(4),AHIV(4),AHV(4),AHVI(4),AHVII(4)                   AER00013
     @     ,AFT3P0,AFT3P1,AFZ3P0,AFZ3P1                                 AER00014
      COMMON/AMATIC/TT(6,5),C(6,6)    ,T(5,6)                           AER00015
      COMMON /ARR/WGUST(6,6),CAMX(6,6),AMX(6,6),DQ(4,6,3),Q(4,6,3)      AER00016
     @      ,DHMAX(4,3,6),HMAX(4,3,6)                                   AER00017
      COMMON /COUPL/    AKPC(4),AKPO(4)                                 AER00018
      DIMENSION                         CDHMX(4,3,5),CHMX(4,3,5)        AER00019
      DIMENSION GUST(5,6),CDAMX(5,5),CCDAMX(6,5),CAMX(5,5),CCAMX(6,5)   AER00020
     %,CDQ(4,5,3),CQ(4,5,3)                                             AER00021
      DO 100 I=1,4                                                      AER00022
      DO 100 J=1,3                                                      AER00023
      DO 100 K=1,5                                                      AER00024
      CDHMX(I,J,K)=0.0                                                  AER00025
      CHMX(I,J,K)=0.0                                                   AER00026
      CDQ(I,K,J)=0.0                                                    AER00027
100   CQ(I,K,J)=0.0                                                     AER00028
      DO 101 I=1,5                                                      AER00029
      DO 102 J=1,5                                                      AER00030
      CAMX(I,J)=0.0                                                     AER00031
102   CDAMX(I,J)=0.0                                                    AER00032
      DO 103 K=1,6                                                      AER00033
      GUST(I,K)=0.0                                                     AER00034
      CCDAMX(K,I)=0.0                                                   AER00035
103   CCAMX(K,I)=0.0                                                    AER00036
```

```
101   CONTINUE                                                          AER00037
      DO 1 I=1,2                                                        AER00038
      DO 1 J=1,6                                                        AER00039
1     T(I,J)=T(I,J)/R                                                   AER00040
      DO 2 I=1,6                                                        AER00041
      DO 2 J=1,2                                                        AER00042
2     TT(I,J)=TT(I,J)/R                                                 AER00043
      GUST(1,1)=0.5*AFT1P0                                              AER00044
      GUST(2,3)=AFZ2P0                                                  AER00045
      GUST(3,1)=-0.5*AFZ1P1                                             AER00046
      GUST(3,2)=-HR*AFT1P0  *0.5                                        AER00047
      GUST(4,1)=0.5*HR*AFT1P0                                           AER00048
      GUST(4,2)=-AFZ1P1 *0.5                                            AER00049
      GUST(5,3)=AFT2P1                                                  AER00050
      GUST(1,6)=0.5*AFT3P0                                              AER00051
      GUST(2,4)=AFZ3P0                                                  AER00052
      GUST(3,5)=-0.5*HR*AFT3P0                                          AER00053
      GUST(3,6)=0.5*AFZ3P1                                              AER00054
      GUST(4,5)=-0.5*AFZ3P1                                             AER00055
      GUST(4,6)=-0.5*HR*AFT3P0                                          AER00056
      GUST(5,4)=AFT3P1                                                  AER00057
      DO 5 I=1,6                                                        AER00058
      DO 5 J=1,6                                                        AER00059
      DO 5 K=1,5                                                        AER00060
5     WGUST(I,J)=TT(I,K)*GUST(K,J)+WGUST(I,J)                           AER00061
      DO 6 I=1,6                                                        AER00062
      DO 6 J=1,3                                                        AER00063
6     WGUST(I,J)=WGUST(I,J)*ABS(RAMDA)                                  AER00064
      CDAMX(1,1)=0.5*AFT1P0                                             AER00065
      CDAMX(1,3)=-0.5*AFT2P1                                            AER00066
      CDAMX(1,4)=0.5*HR*AFT1P0                                          AER00067
      CDAMX(2,2)=AFZ2P0                                                 AER00068
      CDAMX(2,5)=AFZ1P1                                                 AER00069
      CDAMX(3,1)=-0.5*AFZ1P1                                            AER00070
      CDAMX(3,3)=0.5*(HR**2*AFT1P0+AFZ2P2)                              AER00071
      CDAMX(3,4)=0.5*HR*(AFT2P1-AFZ1P1)                                 AER00072
```

```
      CDAMX(4,1)=0.5*HR*AFT1P0
      CDAMX(4,3)=HR*(     -AFT2P1+AFZ1P1)     *0.5
      CDAMX(4,4)=0.5*(HR**2*AFT1P0+AFZ2P2)
      CDAMX(5,2)=AFT2P1
      CDAMX(5,5)=AFT1P2
        DO 9 I=1,6
      DO 9 J=1,5
      DO 9 K=1,5
    9 CCDAMX(I,J)=TT(I,K)*CDAMX(K,J)+CCDAMX(I,J)
      DO 11 I=1,6
      DO 11 J=1,6
      DO 11 K=1,5
   11 DAMX(I,J)=CCDAMX(I,K)*T(K,J)+DAMX(I,J)
      AMDA=ABS(RAMDA)
      CAMX(1,4)=-0.5* AMDA*AFT1P0+AFZ0P0
      CAMX(3,3)=HR*(AFZ0P0-0.5* AMDA*AFT1P0)
      CAMX(3,4)=0.5* AMDA*AFZ1P1+AFT0P1
      CAMX(4,3)=-0.5* AMDA*AFZ1P1
      CAMX(4,4)=HR*(-0.5* AMDA*AFT1P0+AFZ0P0)
      DO 12 I=1,6
      DO 12 J=1,5
      DO 12 K=1,5
   12 CCAMX(I,J)=TT(I,K)*CAMX(K,J)+CCAMX(I,J)
      DO 13 I=1,6
      DO 13 J=1,6
      DO 13 K=1,5
   13 AMX(I.J)=CCAMX(I,K)*T(K,J)+AMX(I,J)
      DO 40 NM=1,4
      CDQ(NM,1,3)=-0.5*AHIII(NM)
      CDQ(NM,2,1)=AHV(NM)
      CDQ(NM,3,2)=-0.5*HR*AHIII(NM)
      CDQ(NM,3,3)=0.5*AHVI(NM)
      CDQ(NM,4,2)=-0.5*AHVI(NM)
      CDQ(NM,4,3)=-0.5*HR*AHIII(NM)
      CDQ(NM,5,1)=AHIV(NM)
      CQ(NM,1,2)=0.5*AHIII(NM) *SNOMEG
```

```
      CQ(NM,3,2)=0.5*(AHVII(NM)-AHVI(NM)*SNOMEG)                        AERO0109
      CQ(NM,3,3)=-0.5*HR*AHIII(NM)*SNOMEG                               AERO0110
      CQ(NM,4,2)=0.5*HR*AHIII(NM)    *SNOMEG                            AERO0111
      CQ(NM,4,3)=0.5*(AHVII(NM)-    AHVI(NM)   *SNOMEG)                 AERO0112
      IF(ITYPE.EQ.0) GO TO 110                                          AERO0113
      CDQ(NM,2,1)=OAHV(NM)                                              AERO0114
      CDQ(NM,5,1)=OAHIV(NM)                                             AERO0115
      CQ(NM,1,3)=0.5*AKPC(NM)*AFT3P0                                    AERO0116
      CQ(NM,2,1)=AKPO(NM)*AFZ3P0                                        AERO0117
      CQ(NM,3,2)=CQ(NM,3,2)-0.5*HR*AKPC(NM)*AFT3P0                      AERO0118
      CQ(NM,3,3)=CQ(NM,3,3)+0.5*AKPC(NM)*AFZ3P1                         AERO0119
      CQ(NM,4,2)=CQ(NM,4,2)-0.5*AKPC(NM)*AFZ3P1                         AERO0120
      CQ(NM,4,3)=CQ(NM,4,3)-0.5*HR*AKPC(NM)*AFT3P0                      AERO0121
      CQ(NM,5,1)=AKPO(NM)*AFT3P1                                        AERO0122
 110  CONTINUE                                                          AERO0123
      DO 16 I=1,6                                                       AERO0124
      DO 16 J=1,3                                                       AERO0125
      DO 16 K=1,5                                                       AERO0126
      Q(NM,I,J)=TT(I,K)*CQ(NM,K,J)+Q(NM,I,J)                            AERO0127
 16   DQ(NM,I,J)=TT(I,K)*CDQ(NM,K,J)+DQ(NM,I,J)                         AERO0128
C   AERO FOR BLADES DUE TO WING MOTION                                  AERO0129
      CDHMX(NM,1,2)=AHRZ(NM)                                            AERO0130
      CDHMX(NM,1,5)=AHNR(NM)                                            AERO0131
      CDHMX(NM,2,3)=-HR*AHI(NM)                                         AERO0132
      CDHMX(NM,2,4)=-AHII(NM)                                           AERO0133
      CDHMX(NM,3,1)=-AHI(NM)                                            AERO0134
      CDHMX(NM,3,3)=AHII(NM)                                            AERO0135
      CDHMX(NM,3,4)=-HR*AHI(NM)                                         AERO0136
       CHMX(NM,2,3)= AMDA*AHI(NM)                                       AERO0137
       CHMX(NM,3,4)= AMDA*AHI(NM)                                       AERO0138
      DO 20 I=1,3                                                       AERO0139
      DO 20 J=1,6                                                       AERO0140
      DO 20 K=1,5                                                       AERO0141
      DHMAX(NM,I,J)=CDHMX(NM,I,K)*T(K,J)+DHMAX(NM,I,J)                  AERO0142
 20   HMAX(NM,I,J)=CHMX(NM,I,K)*T(K,J)+HMAX(NM,I,J)                     AERO0143
 40   CONTINUE                                                          AERO0144
```

```
      RETURN                                                            AERO0145
      END                                                               AERO0146
```

```
      SUBROUTINE EQMTX(IDOF)                                            EQMT0001
C                                                                       EQMT0002
C        TO DEFINE THE COEFFICIENT  MATRICES A,B,C AND D IN EQ. 2.3      EQMT0003
C                                                                       EQMT0004
      COMMON /PARMT/     ITYPE  ,IFLT                                   EQMT0005
      COMMON/AREA8/BLAM(4),WLAM(6),BRAM(4),WRAM(6),BLAMO(4),BRAMO(4)    EQMT0006
      COMMON /AREA1/OMEGA,R,VEL,CL,CD,RAMDA,SNOMEG                      EQMT0007
      COMMON /AREA6/NOBLD,RCH,CHOD,AIB,CK,HMAST    ,ALOCK ,ANO,HR       EQMT0008
      COMMON/AMATIC/TT(6,5),C(6,6)     ,T(5,6)                          EQMT0009
      COMMON/AREA5/AH(4,4),AHI(4),AHII(4),AHRZ(4),AHNR(4),AWO(4),AVO(4) EQMT0010
     @,AW1(4),AV1(4),CAH(4,4),DAWO(4),DAV1(4),DAHIII(4),CAHIII(4)       EQMT0011
     @          ,CAHV(4),DAHIV(4)                                       EQMT0012
      COMMON/INERTI/TTMT(4,6,3),TTCTJ(4,6,3),AMJT(4,3,6),CJT(4,3,6)     EQMT0013
      COMMON /ARR/WGUST(6,6),DAMX(6,6),AMX(6,6),DQ(4,6,3),Q(4,6,3)      EQMT0014
     @          ,DHMAX(4,3,6),HMAX(4,3,6)                               EQMT0015
      COMMON/ARWNG/DARWA(6,6),ARWA(6,6),ARWG(6,3)                       EQMT0016
      COMMON/DOF18/AAY(19,19),BBY(19,19),CCY(19,19),DDY(19,6)           EQMT0017
      COMMON /COUPL/     AKPC(4),AKPD(4)                                EQMT0018
      WRITE(6,450)                                                      EQMT0019
450   FORMAT(///1X,75(1H-)//15X,44HEQUATIONS  OF  MOTION  ;  A*X"+B*X'+C EQMT0020
     &*X=D*E  ,//1X,75(1H-)////)                                        EQMT0021
      DO 801 I=1,18                                                     EQMT0022
801   AAY(I,I)=1.0                                                      EQMT0023
      DO 802 NM=1,4                                                     EQMT0024
      DO 802 I=1,3                                                      EQMT0025
      DO 802 J=1,6                                                      EQMT0026
      AAY(3*(NM-1)+I,J+12)=AMJT(NM,I,J)                                 EQMT0027
802   AAY(J+12,3*(NM-1)+I)=ANO*TTMT(NM,J,I)                             EQMT0028
      DO 804 I=1,4                                                      EQMT0029
      BBY(3*I-1,3*I)=2.0*SNOMEG                                         EQMT0030
804   BBY(3*I,3*I-1)=-2.0*SNOMEG                                        EQMT0031
      DO 805 J=1,4                                                      EQMT0032
      DO 805 I=1,4                                                      EQMT0033
      DO 805 K=1,3                                                      EQMT0034
805   BBY(3*(J-1)+K,3*(I-1)+K)=AH(J,I)                                  EQMT0035
      DO 806 NM=1,4                                                     EQMT0036
```

```
      DO 806 I=1,3                                                      EQMT0037
      DO 806 J=1,6                                                      EQMT0038
      BBY(3*(NM-1)+I,J+12)=DHMAX(NM,I,J)+CJT(NM,I,J)*SNOMEG             EQMT0039
  806 BBY(J+12,3*(NM -1)+I)=DG(NM,J,I)-ANO*TTCTJ(NM,J,I)*SNOMEG         EQMT0040
      DO 808 I=1,6                                                      EQMT0041
      DO 808 J=1,6                                                      EQMT0042
  808 BBY(I+12,J+12)=DAMX(I,J)+C(I,J)*SNOMEG-DARWA(I,J)                 EQMT0043
      DO 809 I=1,4                                                      EQMT0044
      CCY(3*(I-1)+1,3*(I-1)+1)=BLAM(I)                                  EQMT0045
      CCY(3*(I-1)+2,3*(I-1)+2)=BLAM(I)-1.0                              EQMT0046
  809 CCY(3*I,3*I)=BLAM(I)-1.0                                          EQMT0047
      DO 810 J=1,4                                                      EQMT0048
      DO 810 I=1,4                                                      EQMT0049
      CCY(3*J-1,3*I)=AH(J,I)*SNOMEG                                     EQMT0050
  810 CCY(3*J,3*I-1)=-AH(J,I)*SNOMEG                                    EQMT0051
      DO 811 NM=1,4                                                     EQMT0052
      DO 811 I=1,3                                                      EQMT0053
      DO 811 J=1,6                                                      EQMT0054
      CCY(3*(NM-1)+I,J+12)=HMAX(NM,I,J)                                 EQMT0055
  811 CCY(J+12,3*(NM-1)+I)= G(NM,J,I)                                   EQMT0056
      DO 812 I=1,6                                                      EQMT0057
      DO 812 J=1,6                                                      EQMT0058
  812 CCY(I+12,J+12)=AMX(I,J)         -ARWA(I,J)                        EQMT0059
      DO 813 I=1, 6                                                     EQMT0060
  813 CCY(I+12,I+12)=CCY(I+12,I+12)+WLAM(I)                             EQMT0061
      DO 814 NM=1,4                                                     EQMT0062
      DDY(3*(NM-1)+1,4)=-CAHIII(NM)                                     EQMT0063
      DDY(3*(NM-1)+2,5)=-CAHIII(NM)                                     EQMT0064
      DDY(3*(NM-1)+3,6)=-CAHIII(NM)                                     EQMT0065
      DDY(3*NM-1,2)=AHI(NM) *ABS(RAMDA)      *(-1.0)                    EQMT0066
      DDY(3*NM-2,3)=AHRZ(NM)*ABS(RAMDA)      *(-1.0)                    EQMT0067
  814 DDY(3*NM,1)=-AHI(NM)  *ABS(RAMDA)      *(-1.0)                    EQMT0068
      DO 815 I=1,6                                                      EQMT0069
      DO 816 J=1,3                                                      EQMT0070
  816 DDY(I+12,J)=-WGUST(I,J)         +ARWG(I,J)                        EQMT0071
      DO 817 J=4,6                                                      EQMT0072
```

```
817    DDY(I+12,J)=-WGUST(I,J)                                       EQMT0073
815    CONTINUE                                                      EQMT0074
       IF(ITYPE.EQ.0)GO TO 300                                       EQMT0075
       DO 500 J=1,4                                                  EQMT0076
       DO 500 I=1,4                                                  EQMT0077
500    BBY(3*(J-1)+1,3*(I-1)+1)=OAH(J,I)                             EQMT0078
       DO 550 I=1,4                                                  EQMT0079
       CCY(3*(I-1)+1,3*(I-1)+1)=BLAMO(I)                             EQMT0080
       CCY(3*(I-1)+2,3*(I-1)+2)=BLAM(I)-1.0                          EQMT0081
550    CCY(3*I,3*I)=BLAM(I)-1.0                                      EQMT0082
       DO 502 J=1,4                                                  EQMT0083
       DO 502 I=1,4                                                  EQMT0084
       CCY(3*J-2,3*I-2)=CCY(3*J-2,3*I-2)+AKPO(I)*OAHIII(J)           EQMT0085
       CCY(3*J-1,3*I-1)=CCY(3*J-1,3*I-1)+AKPO(I)*OAHIII(J)           EQMT0086
502    CCY(3*J  ,3*I  )=CCY(3*J  ,3*I  )+AKPO(I)*OAHIII(J)           EQMT0087
       DO 501 NM=1,4                                                 EQMT0088
501    DDY(3*(NM-1)+1,4)=-OAHIII(NM)                                 EQMT0089
300    CONTINUE                                                      EQMT0090
       IF(IDOF.EQ.9) GO TO 100                                       EQMT0091
       IF(IFLT.NE.0) GO TO 205                                       EQMT0092
       WRITE(6,451)                                                  EQMT0093
       WRITE(6,6)(( AAY(I,J),J=1,9),I=1,18)                          EQMT0094
       WRITE(6,6)((AAY(I,J),J=10,18),I=1,18)                         EQMT0095
       WRITE(6,5)                                                    EQMT0096
       WRITE(6,452)                                                  EQMT0097
       WRITE(6,6)(( BBY(I,J),J=1,9),I=1,18)                          EQMT0098
       WRITE(6,6)((BBY(I,J),J=10,18),I=1,18)                         EQMT0099
       WRITE(6,5)                                                    EQMT0100
       WRITE(6,453)                                                  EQMT0101
       WRITE(6,6)(( CCY(I,J),J=1,9),I=1,18)                          EQMT0102
       WRITE(6,6)((CCY(I,J),J=10,18),I=1,18)                         EQMT0103
       WRITE(6,5)                                                    EQMT0104
       WRITE(6,454)                                                  EQMT0105
       WRITE(6,7)((DDY(I,J),J=1,6),I=1,18)                           EQMT0106
5      FORMAT(1H1)                                                   EQMT0107
6      FORMAT(////1X,18(/1X,9(E12.5,1X)))                            EQMT0108
```

```
7       FORMAT(////1X,18(/1X,6(E12.5,1X)))                                       EQMT0109
451     FORMAT(20X,'A MATRIX=')                                                  EQMT0110
452      FORMAT(//20X,'B MATRIX=')                                               EQMT0111
453      FORMAT(//20X,'C MATRIX=')                                               EQMT0112
454      FORMAT(//20X,'D MATRIX=')                                               EQMT0113
        RETURN                                                                   EQMT0114
100     CONTINUE                                                                 EQMT0115
        DO 201 I=1,6                                                             EQMT0116
        DO 201 J=7,9                                                             EQMT0117
        AAY(I,J)=AAY(I,J+6)                                                      EQMT0118
        BBY(I,J)=BBY(I,J+6)                                                      EQMT0119
        CCY(I,J)=CCY(I,J+6)                                                      EQMT0120
        AAY(J,I)=AAY(J+6,I)                                                      EQMT0121
        BBY(J,I)=BBY(J+6,I)                                                      EQMT0122
201     CCY(J,I)=CCY(J+6,I)                                                      EQMT0123
        DO 202 I=7,9                                                             EQMT0124
        DO 202 J=7,9                                                             EQMT0125
        AAY(I,J)=AAY(I+6,J+6)                                                    EQMT0126
        BBY(I,J)=BBY(I+6,J+6)                                                    EQMT0127
202     CCY(I,J)=CCY(I+6,J+6)                                                    EQMT0128
        DO 204 I=7,9                                                             EQMT0129
        DO 204 J=1,6                                                             EQMT0130
204     DDY(I,J)=DDY(I+6,J)                                                      EQMT0131
        IF(IFLT.NE.0) GO TO 205                                                  EQMT0132
        WRITE(6,451)                                                             EQMT0133
        WRITE(6,850)((AAY(I,J),J=1,9),I=1,9)                                     EQMT0134
        WRITE(6,452)                                                             EQMT0135
        WRITE(6,850)((BBY(I,J),J=1,9),I=1,9)                                     EQMT0136
        WRITE(6,453)                                                             EQMT0137
        WRITE(6,850)((CCY(I,J),J=1,9),I=1,9)                                     EQMT0138
        WRITE(6,454)                                                             EQMT0139
        WRITE(6,950)((DDY(I,J),J=1,6),I=1,9)                                     EQMT0140
850     FORMAT(///1X,9(/1X,9(E12.5,1X)))                                         EQMT0141
950     FORMAT(///1X,9(/1X,6(E12.5,1X)))                                         EQMT0142
205     CONTINUE                                                                 EQMT0143
        RETURN                                                                   EQMT0144
```

```
      END                                                               EQMT0145
```

```
      SUBROUTINE AUTO(IDOF)
C
C        IN AUTOROTION FLIGHT ANOTHER DEGREE OF FREEDOM IS ADDED
C
      COMMON /PARMT/     ITYPE ,IFLT
      COMMON/AMATIC/TT(6,5),C(6,6)      ,T(5,6)
      COMMON /AREA6/NOBLD,ROH,CHOD,AIB,CK,HMAST    ,ALOCK ,ANO,HR
      COMMON/AREA5/AH(4,4),AHI(4),AHII(4),AHRZ(4),AHNR(4),AWO(4),AVO(4)
     @,AW1(4),AV1(4),OAH(4,4),OAWO(4),OAV1(4),OAHII(4),OAHIII(4)
     @          ,OAHV(4),OAHIV(4)
      COMMON/DOF18/AAY(19,19),BBY(19,19),CCY(19,19),DDY(19,6)
      COMMON /COUPL/     AKPC(4),AKPO(4)
      COMMON/AERO/AFTOP1,AFT1P0,AFT1P2,AFT2P1,AFZ0P0,AFZ1P1,AFZ2P0,
     @AFZ2P2,AHIII(4),AHIV(4),AHV(4),AHVI(4),AHVII(4)
     @          ,AFT3P0,AFT3P1,AFZ3P0,AFZ3P1
      NB=4
      NW=6
      NR=19
      DO 11 I=1,NB
      AAY(3*I-2,NR)=AV1(I)
      AAY(NR,3*I-2)=AV1(I)*ANO
      BBY(3*I-2,NR)=AHNR(I)
      BBY(NR,3*I-2)=AHIV(I)
   11 CONTINUE
      DO 12 I=1,NW
      BBY(NR,3*NB+I)=AFT2P1*T(2,I)
      BBY(3*NB+I,NR)=AFZ1P1*T(2,I)
   12 CONTINUE
      AAY(NR,NR)=AIB*ANO
      BBY(NR,NR)=AFT1P2
      CCY(NR,3)=-AFT2P1
      DDY(NR,4)=-AFT3P1
      IF(ITYPE.EQ.0) GO TO 13
      DO 14 I=1,NB
      AAY(3*I-2,NR)=OAV1(I)
      AAY(NR,3*I-2)=OAV1(I)*ANO
```

```
      BBY(NR,3*I-2)=DAHIV(I)                                            AUTO0037
      CCY(NR,3*I-2)=AKPO(I)*AFT3P1                                      AUTO0038
 14   CONTINUE                                                          AUTO0039
 13   CONTINUE                                                          AUTO0040
      IF(IDOF.EQ.9) GO TO 100                                           AUTO0041
      WRITE(6,451)                                                      AUTO0042
      WRITE(6,6)(( AAY(I,J),J=1,9),I=1,19)                              AUTO0043
      WRITE(6,8)((AAY(I,J),J=10,19),I=1,19)                             AUTO0044
      WRITE(6,5)                                                        AUTO0045
      WRITE(6,452)                                                      AUTO0046
      WRITE(6,6)(( BBY(I,J),J=1,9),I=1,19)                              AUTO0047
      WRITE(6,8)((BBY(I,J),J=10,19),I=1,19)                             AUTO0048
      WRITE(6,5)                                                        AUTO0049
      WRITE(6,453)                                                      AUTO0050
      WRITE(6,6)(( CCY(I,J),J=1,9),I=1,19)                              AUTO0051
      WRITE(6,8)((CCY(I,J),J=10,19),I=1,19)                             AUTO0052
      WRITE(6,5)                                                        AUTO0053
      WRITE(6,454)                                                      AUTO0054
      WRITE(6,7)((DDY(I,J),J=1,6),I=1,19)                               AUTO0055
 5    FORMAT(1H1)                                                       AUTO0056
 6    FORMAT(////1X,19(/1X,9(E12.5,1X)))                                AUTO0057
 7    FORMAT(////1X,19(/1X,6(E12.5,1X)))                                AUTO0058
 8    FORMAT(////1X,19(/1X,10(E12.5,1X)))                               AUTO0059
      RETURN                                                            AUTO0060
 100  CONTINUE                                                          AUTO0061
      DO 15 I=1,6                                                       AUTO0062
      AAY(I,10)=AAY(I,19)                                               AUTO0063
      BBY(I,10)=BBY(I,19)                                               AUTO0064
      CCY(I,10)=CCY(I,19)                                               AUTO0065
      AAY(10,I)=AAY(19,I)                                               AUTO0066
      BBY(10,I)=BBY(19,I)                                               AUTO0067
      CCY(10,I)=CCY(19,I)                                               AUTO0068
      DDY(10,I)=DDY(19,I)                                               AUTO0069
 15   CONTINUE                                                          AUTO0070
      DO 16 I=1,3                                                       AUTO0071
      AAY(I+6,10)=AAY(I+12,19)                                          AUTO0072
```

```
      BBY(I+6,10)=BBY(I+12,19)                                          AUT00073
      CCY(I+6,10)=CCY(I+12,19)                                          AUT00074
      AAY(10,I+6)=AAY(19,I+12)                                          AUT00075
      BBY(10,I+6)=BBY(19,I+12)                                          AUT00076
      CCY(10,I+6)=CCY(19,I+12)                                          AUT00077
   16 CONTINUE                                                          AUT00078
      AAY(10,10)=AAY(19,19)                                             AUT00079
      BBY(10,10)=BBY(19,19)                                             AUT00080
      CCY(10,10)=CCY(19,19)                                             AUT00081
      WRITE(6,451)                                                      AUT00082
      WRITE(6,850)((AAY(I,J),J=1,10),I=1,10)                            AUT00083
      WRITE(6,452)                                                      AUT00084
      WRITE(6,850)((BBY(I,J),J=1,10),I=1,10)                            AUT00085
      WRITE(6,453)                                                      AUT00086
      WRITE(6,850)((CCY(I,J),J=1,10),I=1,10)                            AUT00087
      WRITE(6,454)                                                      AUT00088
      WRITE(6,950)((DDY(I,J),J=1,6),I=1,10)                             AUT00089
  451 FORMAT(20X,'A MATRIX=')                                           AUT00090
  452  FORMAT(//20X,'B MATRIX=')                                        AUT00091
  453  FORMAT(//20X,'C MATRIX=')                                        AUT00092
  454  FORMAT(//20X,'D MATRIX=')                                        AUT00093
  850 FORMAT(///1X,10(/1X,10(E12.5,1X)))                                AUT00094
  950 FORMAT(///1X,10(/1X,6(E12.5,1X)))                                 AUT00095
      RETURN                                                            AUT00096
      END                                                               AUT00097
```

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR,

```
      SUBROUTINE GUSTCC(CGUST,DDY,L,DDZ)                                        GUST0001
C                                                                               GUST0002
C        TO DEFINE GUST AND BLADE PITCH CONTROL COMPONENTS                      GUST0003
C                                                                               GUST0004
      DIMENSION CGUST(6),DDY(19,6) ,DDZ(19)                                     GUST0005
      DO 1 I=1,L                                                                GUST0006
1     DDZ(I)=0.0                                                                GUST0007
      DO 2 I=1,L                                                                GUST0008
      DO 2 J=1,6                                                                GUST0009
2     DDZ(I)=DDY(I,J)*CGUST(J)+DDZ(I)                                           GUST0010
      RETURN                                                                    GUST0011
      END                                                                       GUST0012
```

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

```
      SUBROUTINE FRQRES(L,AAA,BBB,CCC,DDD,IFLT,IDOF)                    FROR0001
C                                                                       FROR0002
C        TO CALCULATE THE FREQUENCY RESPONSE                            FROR0003
C                                                                       FROR0004
      DIMENSION AAA(19,19),BBB(19,19),CCC(19,19),DDD(19)                FROR0005
      DOUBLE PRECISION  FREQ,DPA(19,19),DPB(19,19),DPC(19,19),DPD(19)   FROR0006
      COMPLEX*16 COMA(19,19),COME(19), DCMPLX                           FROR0007
      WRITE(6,1000)                                                     FROR0008
1000  FORMAT(///10X,40(1H-),//20X,'FREQUENCY RESPONSE',//10X,40(1H-)    FROR0009
     *     /////4X,'--FREQUENCY/OMEGA--')                               FROR0010
      IF(IDOF.EQ.18) GO TO 1001                                         FROR0011
      WRITE(6,1002)                                                     FROR0012
1002  FORMAT(//10X,'Q1C',T22,'Q1C',T34,'Q1S',T46,'Q2C',T58,'Q2C',T70,   FROR0013
     *          'Q2S',T82,'WING 1', T94,'WING 2',T106,'WING 3')         FROR0014
      GO TO 1003                                                        FROR0015
1001  WRITE(6,1004)                                                     FROR0016
1004  FORMAT(//10X,'Q1C',T22,'Q1C',T34,'Q1S',T46,'Q2C',T58,'Q2C',T70,   FROR0017
     *          'Q2S',T82,'Q3C',T94, 'Q3C', T106,'Q3S' //10X,'Q4C',T22, FROR0018
     %          'Q4C',T34,'Q4S',T46,'WING 1' ,T58,'WING 2', T70,'WING 3',FROR0019
     *          T82,'WING 4', T94,'WING 5',T106,'WING 6')               FROR0020
1003  CONTINUE                                                          FROR0021
      IF(IFLT.EQ.0) GO TO 1007                                          FROR0022
      WRITE(6,1005)                                                     FROR0023
1005  FORMAT(//10X,'D(NU R)/DT')                                        FROR0024
1007  CONTINUE                                                          FROR0025
      DO 180 I=1,L                                                      FROR0026
      DO 180 J=1,L                                                      FROR0027
      DPA(I,J)=AAA(I,J)                                                 FROR0028
      DPB(I,J)=BBB(I,J)                                                 FROR0029
      DPC(I,J)=CCC(I,J)                                                 FROR0030
180   CONTINUE                                                          FROR0031
      DO 181 I=1,L                                                      FROR0032
      DPD(I)  =DDD(I)                                                   FROR0033
181   CONTINUE                                                          FROR0034
      IK=0                                                              FROR0035
      FREQ=0.0D0                                                        FROR0036
```

OF THE ORIGINAL PAGE IS POOR,

```
211   FREQ=FREQ+0.01D0                                                   FRQR0037
      IK=IK+1                                                            FRQR0038
      GO TO 511                                                          FRQR0039
311   FREQ=FREQ+0.02D0                                                   FRQR0040
      IK=IK+1                                                            FRQR0041
      GO TO 511                                                          FRQR0042
611   FREQ=FREQ+0.05D0                                                   FRQR0043
      IK=IK+1                                                            FRQR0044
      GO TO 511                                                          FRQR0045
711   FREQ=FREQ+0.1D0                                                    FRQR0046
      IK=IK+1                                                            FRQR0047
      GO TO 511                                                          FRQR0048
811   FREQ=FREQ+0.5D0                                                    FRQR0049
      IK=IK+1                                                            FRQR0050
      GO TO 511                                                          FRQR0051
511   DO 100 I=1,L                                                       FRQR0052
      DO 100 J=1,L                                                       FRQR0053
100   CCMA(I,J)=DCMPLX(DPC(I,J)-FREQ**2* DPA(I,J),FREQ*DPB(I,J))         FRQR0054
      DO 301 I=1,L                                                       FRQR0055
301   CCMB(I)=DCMPLX( DPD(I),0.0D0)                                      FRQR0056
      CALL GAELI(CCMA,CCMB,L   ,FREQ,IDOF,IFLT)                          FRQR0057
      IF(IK.LT.10) GO TO 211                                             FRQR0058
      IF(IK.LT.25) GO TO 311                                             FRQR0059
      IF(IK.LT.37) GO TO 611                                             FRQR0060
      IF(IK.LT.57) GO TO 711                                             FRQR0061
      IF(IK.LT.71) GO TO 811                                             FRQR0062
      RETURN                                                             FRQR0063
      END                                                                FRQR0064
```

```
      SUBROUTINE GAEL1(A,Y,N,FREQ,IDOF,IFLT)                             GAEL0001
C                                                                        GAEL0002
C        THE GAUSS-JORDAN REDUCTION                                      GAEL0003
C                                                                        GAEL0004
      COMPLEX*16 A(19,19),Y(19),X(19)                                    GAEL0005
      DOUBLE PRECISION FREQ ,CDABS                                       GAEL0006
      DIMENSION ABX(19)                                                  GAEL0007
      COMMON/FRMAG/    FRB(4),FRBC(4),FRW(6) ,IFRMAG                     GAEL0008
      M=N-1                                                              GAEL0009
      DO 10 I=1,M                                                        GAEL0010
      L=I+1                                                              GAEL0011
      DO 10 J=L,N                                                        GAEL0012
      IF(CDABS(A(I,J)).EQ.0.0D0)GO TO 10                                 GAEL0013
      DO 8 K=L,N                                                         GAEL0014
      A(J,K)=A(J,K)-A(I,K)*A(J,I)/A(I,I)                                 GAEL0015
  8   CONTINUE                                                           GAEL0016
      Y(J)=Y(J)-Y(I)*A(J,I)/A(I,I)                                       GAEL0017
 10   CONTINUE                                                           GAEL0018
      X(N)=Y(N)/A(N,N)                                                   GAEL0019
      DO 30 I=1,M                                                        GAEL0020
      K=N-I                                                              GAEL0021
      L=K+1                                                              GAEL0022
      DO 20 J=L,N                                                        GAEL0023
 20   Y(K)=Y(K)-X(J)*A(K,J)                                              GAEL0024
 30   X(K)=Y(K)/A(K,K)                                                   GAEL0025
      DO 40 I=1,N                                                        GAEL0026
 40   ABX(I)=CDABS(X(I))                                                 GAEL0027
      IF(IFRMAG.EQ.1) GO TO 50                                           GAEL0028
      IF(IDOF.EQ.9) GO TO 51                                             GAEL0029
      LT=4                                                               GAEL0030
      LTT=6                                                              GAEL0031
      GO TO 54                                                           GAEL0032
 51   LT=2                                                               GAEL0033
      LTT=3                                                              GAEL0034
 54   DO 52 I=1,LT                                                       GAEL0035
      DO 52 J=1,3                                                        GAEL0036
```

```
 52    ABX(3*(I-1)+J)=ABX(3*(I-1)+J)*FRB(I)                                  GAEL0037
       DO 55 I=1,LT                                                          GAEL0038
 55    ABX(3*(I-1)+1)=ABX(3*(I-1)+1)*FRBO(I)/FRB(I)                          GAEL0039
       DO 53 I=1,LTT                                                         GAEL0040
 53    ABX(I+3*LT)=ABX(I+3*LT)*FRW(I)                                        GAEL0041
 50    CONTINUE                                                              GAEL0042
       IF (IFLT.EQ.0) GO TO 56                                               GAEL0043
       ABX(N)=FREQ*ABX(N)                                                    GAEL0044
 56    CONTINUE                                                              GAEL0045
       WRITE(6,100) FREQ                                                     GAEL0046
 100   FORMAT(//3X,'--',F 6.2,'--')                                          GAEL0047
       WRITE(6,200)(ABX(I),I=1,N)                                            GAEL0048
 200   FORMAT(/(8X,9(E10.3,2X)/))                                            GAEL0049
       RETURN                                                                GAEL0050
       END                                                                   GAEL0051
```

```
      SUBROUTINE EIGEN(N,AAA,BBB,CCC,DDD,IDOF)
C
C        ROUTINE TO FORM AN EIGENVALUE PROBLEM AND TO CALL EIPACK SUBROUTINE
C
      DIMENSION AAA(19,19),BBB(19,19),CCC(19,19),DDD(19,6)
      DIMENSION A(361),L(19),M(19),AINV(19,19)
      DIMENSION AAN(19,19),BBN(19,19),CCN(19,19),DDN(19,6)
      REAL*8 AEIG(38,38),WR(38),WI(38),ZR(38,38)
      REAL*8 SCALE(38)
      INTEGER INT(38)
      DIMENSION    BIG(25), ABMOD(25,25) ,DAMP(38)
      COMPLEX      AMOD(25,25) ,BIGCOM(25)
      COMMON/FRMAG/      FRB(4),FRBD(4),FRW(6) ,IFRMAG
      IDEBUG=0
      WRITE(6,153)
      DO 3003 I=1,N
      DO 3004 J=1,N
      AAN(I,J)=0.0
      BBN(I,J)=0.0
 3004 CCN(I,J)=0.0
      DO 3005 K=1,6
 3005 DDN(I,K)=0.0
 3003 CONTINUE
      LL=0
      DO 1000 J=1,N
      DO 1000 I=1,N
      LL=LL+1
 1000 A(LL)=AAA(I,J)
      CALL MINV(A,N,D,L,M)
      LL=0
      DO 2000 J=1,N
      DO 2000 I=1,N
      LL=LL+1
 2000 AINV(I,J)=A(LL)
      DO 3000 I=1,N
      DO 3001 J=1,N
```

```
      DO 3001 K=1,N                                                     EIGE0037
      AAN(I,J)=AINV(I,K)*AAA(K,J)+AAN(I,J)                              EIGE0038
      BBN(I,J)=AINV(I,K)*BBB(K,J)+BBN(I,J)                              EIGE0039
 3001 CCN(I,J)=AINV(I,K)*CCC(K,J)+CCN(I,J)                              EIGE0040
      DO 3002 J=1,6                                                     EIGE0041
      DO 3002 K=1,N                                                     EIGE0042
 3002 DDN(I,J)=AINV(I,K)*DDD(K,J)+DDN(I,J)                              EIGE0043
 3000 CONTINUE                                                          EIGE0044
 6000 N2=2*N                                                            EIGE0045
      DO 300 I=1,N                                                      EIGE0046
      DO 300 J=1,N                                                      EIGE0047
      AEIG(I,J)=-BBN(I,J)                                               EIGE0048
  300 AEIG(I,J+N)=-CCN(I,J)                                             EIGE0049
      DO 301 I=1,N                                                      EIGE0050
      DO 301 J=1,N2                                                     EIGE0051
  301 AEIG(I+N,J)=0.0D0                                                 EIGE0052
      DO 302 I=1,N                                                      EIGE0053
  302 AEIG(I+N,I)=1.0D0                                                 EIGE0054
      CALL EIPACK(38,N2,AEIG,WR, WI, ZR, IERROR , SCALE , INT )         EIGE0055
      IF(IERROR.EQ.0) GO TO 152                                         EIGE0056
      WRITE(6,150)IERROR                                                EIGE0057
  150 FORMAT(15X,'IERROR=',I5)                                          EIGE0058
  152 CONTINUE                                                          EIGE0059
      IF(IDEBUG.EQ.0) GO TO 61                                          EIGE0060
      WRITE(6,67)(WR(I),WI(I),I=1,N2)                                   EIGE0061
   67 FORMAT(/10X,D15.7,2X,D15.7)                                       EIGE0062
      N3=N/3                                                            EIGE0063
      DO 400 IL=1,N3                                                    EIGE0064
      IJ=6*(IL-1)+1                                                     EIGE0065
      KL=6*(IL-1)+6                                                     EIGE0066
  400 WRITE(6,251)((ZR(I,J),J=IJ,KL),I=1,N2)                            EIGE0067
  251 FORMAT(///1X,(/1X,(2D15.7,4X,2D15.7,4X,2D15.7)))                  EIGE0068
      IF(3*N3.EQ.N) GO TO 61                                            EIGE0069
      K=2*(N-3*N3)                                                      EIGE0070
      IJ=KL+1                                                           EIGE0071
      KL=KL+K                                                           EIGE0072
```

```
      WRITE(6,751)((ZP(I,J),J=IJ,KL),I=1,N2)                            EIGE0073
751   FORMAT(///1X,(/1X,(2D15.7)))                                      EIGE0074
61    CONTINUE                                                          EIGE0075
      DO 140 I=1,N2                                                     EIGE0076
      XX=SNGL(WR(I)**2+WI(I)**2)                                        EIGE0077
      IF(XX.EQ.0.0) GO TO 141                                           EIGE0078
      DAMP(I)=-SNGL(WR(I))/SQRT(XX)                                     EIGE0079
      GO TO 140                                                         EIGE0080
141   DAMP(I)=0.0                                                       EIGE0081
140   CONTINUE                                                          EIGE0082
      N1=N+1                                                            EIGE0083
      LK=0                                                              EIGE0084
      LKK=0                                                             EIGE0085
      NTOT=0                                                            EIGE0086
      I=1                                                               EIGE0087
64    CONTINUE                                                          EIGE0088
      IF(I.GE.N2+1) GO TO 63                                            EIGE0089
      NTOT=NTOT+1                                                       EIGE0090
      K=NTOT                                                            EIGE0091
      IF(WI(I).EQ.0.0D0)  GO TO 65                                      EIGE0092
         INT(I)=K                                                       EIGE0093
         INT(I+1)=K                                                     EIGE0094
      LK=LKK+1                                                          EIGE0095
      LKK=LK+1                                                          EIGE0096
      IF(IDEBUG.EQ.0) GO TO 68                                          EIGE0097
      WRITE(6,69)I,LK,LKK,K                                             EIGE0098
69    FORMAT(1X,4I5)                                                    EIGE0099
68    CONTINUE                                                          EIGE0100
      DO 50 J=N1,N2                                                     EIGE0101
      IF(IDEBUG.EQ.0) GO TO 71                                          EIGE0102
      WRITE(6,72)ZP(J,LK),ZP(J,LKK)                                     EIGE0103
72    FORMAT(1X,2D15.7)                                                 EIGE0104
71    CONTINUE                                                          EIGE0105
      AMOD(K,J-N1+1)=CMPLX(SNGL(ZP(J,LK)),SNGL(ZP(J,LKK)))              EIGE0106
50    CONTINUE                                                          EIGE0107
      I=I+2                                                             EIGE0108
```

```
      GO TO 64                                                          EIGE0109
   65 CONTINUE                                                          EIGE0110
         INT(I)=K                                                       EIGE0111
      LK=LKK+1                                                          EIGE0112
      LKK=LK                                                            EIGE0113
      IF(IDEBUG.EQ.0) GO TO 73                                          EIGE0114
      WRITE(6,69)I,LK,LKK,K                                             EIGE0115
   73 CONTINUE                                                          EIGE0116
      DO 66 J=N1,N2                                                     EIGE0117
      IF(IDEBUG.EQ.0) GO TO 74                                          EIGE0118
      WRITE(6,72)ZP(J,LK),ZP(J,LKK)                                     EIGE0119
   74 CONTINUE                                                          EIGE0120
      AMOD(K,J-N1+1)=CMPLX(SNGL(ZP(J,LK)),0.0)                          EIGE0121
   66 CONTINUE                                                          EIGE0122
      I=I+1                                                             EIGE0123
      GO TO 64                                                          EIGE0124
   63 CONTINUE                                                          EIGE0125
      IF(IFRMAG.EQ.1) GO TO 130                                         EIGE0126
      IF(IDOF.EQ.9) GO TO 131                                           EIGE0127
      LT=4                                                              EIGE0128
      LTT=6                                                             EIGE0129
      GO TO 134                                                         EIGE0130
  131 LT=2                                                              EIGE0131
      LTT=3                                                             EIGE0132
  134 CONTINUE                                                          EIGE0133
      DO 136 IJ=1,NTOT                                                  EIGE0134
      DO 132 I=1,LT                                                     EIGE0135
      DO 132J=1,3                                                       EIGE0136
  132 AMOD(IJ,3*(I-1)+J)=     AMOD(IJ,3*(I-1)+J)*FRB(I)                 EIGE0137
      DO 135I=1,LT                                                      EIGE0138
  135 AMOD(IJ,3*(I-1)+1)=     AMOD(IJ,3*(I-1)+1)*FRBO(I)/FRB(I)         EIGE0139
      DO 133I=1,LTT                                                     EIGE0140
  133 AMOD(IJ,I+3*LT)=AMOD(IJ,I+3*LT)*FRW(I)                            EIGE0141
  136 CONTINUE                                                          EIGE0142
  130 CONTINUE                                                          EIGE0143
      DO 51 I=1,NTOT                                                    EIGE0144
```

```
      DO 51 J=1,N
 51   ABMOD(I,J)=CABS(AMOD(I,J))
      DO 52 I=1,NTOT
      BIG(I)=0.0
      DO 53 J=1,N
      IF(ABMOD(I,J) -  BIG(I))  53,53,54
 54   BIG(I)=ABMOD(I,J)
       BIGCOM(I)=AMOD(I,J)
 53   CONTINUE
 52   CONTINUE
      DO 60 I=1,NTOT
      DO 60 J=1,N
      ABMOD(I,J)=ABMOD(I,J)/BIG(I)
       AMOD(I,J)=AMOD(I,J)/BIGCOM(I)
 60   CONTINUE
 153  FORMAT(///15X,20(1H*)//20X,'EIGENVALUES',//15X,  20(1H*),
     &     ///20X,'** REAL PART **', 3X,'** IMAGINARY PART ***'
     &          ,10X,'** DAMPING RATIO **'/)
      WRITE(6,151)(INT(I),WR(I),WI(I),DAMP(I),I=1,N2)
 151  FORMAT(/ 9X,'NO.',I2,5X,D15.7,5X,D15.7,15X,F15.7)
      WRITE(6,59)
 59   FORMAT(1H1,///5X,  20 (1H*),//10X,  'EIGENVECTORS',//5X, 20(1H*))
      IP=0
      DO 55 I=1,NTOT
      IP=IP+1
      WRITE(6,154)I       ,WR(IP),WI(IP)
 154  FORMAT(///2X,'-- CORRESPONDING TO NO.',I2,1X,'EIGENVALUE --',
     &     5X,'(',D10.3,')+IMAG(',D10.3,')',
     &      ///8X,'** ABSOLUTE VALUE **',20X,'** REAL PART **',2X,
     &      '** IMAGINARY PART **',//)
      IF((IP+1).GT.N2)  GO TO 752
      IF(INT(IP).EQ.INT(IP+1))    IP=IP+1
 752  CONTINUE
      DO 56 J=1,N
      WRITE(6,57)ABMOD(I,J), AMOD(I,J)
 57   FORMAT(10X,F15.7,22X,E15.7,3X,E15.7)
```

```
 56   CONTINUE                                                          EIGE0181
      WRITE(6,58)                                                       EIGE0182
 58   FORMAT(///1X)                                                     EIGE0183
 55   CONTINUE                                                          EIGE0184
      RETURN                                                            EIGE0185
      END                                                               EIGE0186
```

```
      SUBROUTINE EIPACK( NM, N, A, WR, WI, Z , IERR,SCALE,INT )         EIPA0001
C                                                                       EIPA0002
C        AN EIGENSYSTEM PROBLEM SOLVER FOR THE GENERAL MATRIX           EIPA0003
C                                                                       EIPA0004
      REAL*8 A(NM,N),Z(NM,N), WR(N), WI(N) ,SCALE(N)                    EIPA0005
      INTEGER INT(N)                                                    EIPA0006
      CALL BALANC(NM,N,A,LOW,IGH,SCALE)                                 EIPA0007
      CALL ELMHES(NM,N,LOW,IGH,A,INT)                                   EIPA0008
      CALL ELTRAN(NM,N,LOW,IGH,A,INT,Z)                                 EIPA0009
      CALL HQR2(NM,N,LOW,IGH,A,WR,WI,Z,IERR)                            EIPA0010
      CALL BALBAK(NM,N,LOW,IGH,SCALE,N,Z)                               EIPA0011
      RETURN                                                            EIPA0012
      END                                                               EIPA0013
```

```
C                                                                       69210001
C     ------------------------------------------------------------------69210002
C                                                                       69210003
      SUBROUTINE BALANC(NM,N,A,LOW,IGH,SCALE)                           69210004
C                                                                       69210005
      INTEGER I,J,K,L,M,N,JJ,NM,IGH,LOW,IEXC                            69210006
      REAL*8 A(NM,N),SCALE(N)                                           69210007
      REAL*8 C,F,G,R,S,B2,RADIX                                         69210008
      REAL*8 DABS                                                       69210009
      LOGICAL NOCONV                                                    69210010
C                                                                       69210011
C     THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE BALANCE,  69210012
C     NUM. MATH. 13, 293-304(1969) BY PARLETT AND REINSCH.              69210013
C     HANDBOOK FOR AUTO. COMP., VOL.II-LINEAR ALGEBRA, 315-326(1971).   69210014
C                                                                       69210015
C     THIS SUBROUTINE BALANCES A REAL MATRIX AND ISOLATES               69210016
C     EIGENVALUES WHENEVER POSSIBLE.                                    69210017
C                                                                       69210018
C     ON INPUT:                                                         69210019
C                                                                       69210020
C        NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL         69210021
C          ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM          69210022
C          DIMENSION STATEMENT;                                         69210023
C                                                                       69210024
C        N IS THE ORDER OF THE MATRIX;                                  69210025
C                                                                       69210026
C        A CONTAINS THE INPUT MATRIX TO BE BALANCED.                    69210027
C                                                                       69210028
C     ON OUTPUT:                                                        69210029
C                                                                       69210030
C        A CONTAINS THE BALANCED MATRIX;                                69210031
C                                                                       69210032
C        LOW AND IGH ARE TWO INTEGERS SUCH THAT A(I,J)                  69210033
C          IS EQUAL TO ZERO IF                                          69210034
C           (1) I IS GREATER THAN J AND                                 69210035
C           (2) J=1,...,LOW-1 OR I=IGH+1,...,N;                         69210036
```

```
C                                                                       69210037
C        SCALE CONTAINS INFORMATION DETERMINING THE                     69210038
C           PERMUTATIONS AND SCALING FACTORS USED.                      69210039
C                                                                       69210040
C     SUPPOSE THAT THE PRINCIPAL SUBMATRIX IN ROWS LOW THROUGH IGH      69210041
C     HAS BEEN BALANCED, THAT P(J) DENOTES THE INDEX INTERCHANGED       69210042
C     WITH J DURING THE PERMUTATION STEP, AND THAT THE ELEMENTS         69210043
C     OF THE DIAGONAL MATRIX USED ARE DENOTED BY D(I,J).  THEN          69210044
C        SCALE(J) = P(J),    FOR J = 1,...,LOW-1                        69210045
C                 = D(J,J),      J = LOW,...,IGH                        69210046
C                 = P(J)         J = IGH+1,...,N.                       69210047
C     THE ORDER IN WHICH THE INTERCHANGES ARE MADE IS N TO IGH+1,       69210048
C     THEN 1 TO LOW-1.                                                  69210049
C                                                                       69210050
C     NOTE THAT 1 IS RETURNED FOR IGH IF IGH IS ZERO FORMALLY.          69210051
C                                                                       69210052
C     THE ALGOL PROCEDURE EXC CONTAINED IN BALANCE APPEARS IN           69210053
C     BALANC  IN LINE.  (NOTE THAT THE ALGOL ROLES OF IDENTIFIERS       69210054
C     K,L HAVE BEEN REVERSED.)                                          69210055
C                                                                       69210056
C     QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,        69210057
C     APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY         69210058
C                                                                       69210059
C     ------------------------------------------------------------------69210060
C                                                                       69210061
C     :::::::::: RADIX IS A MACHINE DEPENDENT PARAMETER SPECIFYING      69210062
C                THE BASE OF THE MACHINE FLOATING POINT REPRESENTATION. 69210063
C                RADIX = 16.0D0 FOR LONG FORM ARITHMETIC                69210064
C                ON S360 ::::::::::                                     69210065
      DATA RADIX/Z4210000000000000/                                     69210066
C                                                                       69210067
      B2 = RADIX * RADIX                                                69210068
      K = 1                                                             69210069
      L = N                                                             69210070
      GO TO 100                                                         69210071
C     :::::::::: IN-LINE PROCEDURE FOR ROW AND                          69210072
```

```
C                COLUMN EXCHANGE ::::::::::                             69210073
   20 SCALE(M) = J                                                      69210074
      IF (J .EQ. M) GO TO 50                                            69210075
C                                                                       69210076
      DO 30 I = 1, L                                                    69210077
         F = A(I,J)                                                     69210078
         A(I,J) = A(I,M)                                                69210079
         A(I,M) = F                                                     69210080
   30 CONTINUE                                                          69210081
C                                                                       69210082
      DO 40 I = K, N                                                    69210083
         F = A(J,I)                                                     69210084
         A(J,I) = A(M,I)                                                69210085
         A(M,I) = F                                                     69210086
   40 CONTINUE                                                          69210087
C                                                                       69210088
   50 GO TO (80,130), IEXC                                              69210089
C     :::::::::: SEARCH FOR ROWS ISOLATING AN EIGENVALUE                69210090
C                AND PUSH THEM DOWN ::::::::::                          69210091
   80 IF (L .EQ. 1) GO TO 280                                           69210092
      L = L - 1                                                         69210093
C     :::::::::: FOR J=L STEP -1 UNTIL 1 DO -- ::::::::::               69210094
  100 DO 120 JJ = 1, L                                                  69210095
         J = L + 1 - JJ                                                 69210096
C                                                                       69210097
         DO 110 I = 1, L                                                69210098
            IF (I .EQ. J) GO TO 110                                     69210099
            IF (A(J,I) .NE. 0.0D0) GO TO 120                            69210100
  110    CONTINUE                                                       69210101
C                                                                       69210102
         M = L                                                          69210103
         IEXC = 1                                                       69210104
         GO TO 20                                                       69210105
  120 CONTINUE                                                          69210106
C                                                                       69210107
      GO TO 140                                                         69210108
```

```
C     :::::::::: SEARCH FOR COLUMNS ISOLATING AN EIGENVALUE             69210109
C                AND PUSH THEM LEFT ::::::::::                          69210110
  130 K = K + 1                                                         69210111
C                                                                       69210112
  140 DO 170 J = K, L                                                   69210113
C                                                                       69210114
         DO 150 I = K, L                                                69210115
            IF (I .EQ. J) GO TO 150                                     69210116
            IF (A(I,J) .NE. 0.0D0) GO TO 170                            69210117
  150    CONTINUE                                                       69210118
C                                                                       69210119
         M = K                                                          69210120
         IEXC = 2                                                       69210121
         GO TO 20                                                       69210122
  170 CONTINUE                                                          69210123
C     :::::::::: NOW BALANCE THE SUBMATRIX IN ROWS K TO L ::::::::::    69210124
      DO 180 I = K, L                                                   69210125
  180 SCALE(I) = 1.0D0                                                  69210126
C     :::::::::: ITERATIVE LOOP FOR NORM REDUCTION ::::::::::           69210127
  190 NOCONV = .FALSE.                                                  69210128
C                                                                       69210129
      DO 270 I = K, L                                                   69210130
         C = 0.0D0                                                      69210131
         R = 0.0D0                                                      69210132
C                                                                       69210133
         DO 200 J = K, L                                                69210134
            IF (J .EQ. I) GO TO 200                                     69210135
            C = C + DABS(A(J,I))                                        69210136
            R = R + DABS(A(I,J))                                        69210137
  200    CONTINUE                                                       69210138
C                                                                       69210139
         G = R / RADIX                                                  69210140
         F = 1.0D0                                                      69210141
         S = C + R                                                      69210142
  210    IF (C .GE. G) GO TO 220                                        69210143
         F = F * RADIX                                                  69210144
```

```
        C = C * B2                                                      69210145
        GO TO 210                                                       69210146
  220   G = R * RADIX                                                   69210147
  230   IF (C .LT. G) GO TO 240                                         69210148
        F = F / RADIX                                                   69210149
        C = C / B2                                                      69210150
        GO TO 230                                                       69210151
C     :::::::::: NOW BALANCE ::::::::::                                 69210152
  240   IF ((C + R) / F .GE. 0.95D0 * S) GO TO 270                      69210153
        G = 1.0D0 / F                                                   69210154
        SCALE(I) = SCALE(I) * F                                         69210155
        NOCONV = .TRUE.                                                 69210156
C                                                                       69210157
        DO 250 J = K, N                                                 69210158
  250   A(I,J) = A(I,J) * G                                             69210159
C                                                                       69210160
        DO 260 J = 1, L                                                 69210161
  260   A(J,I) = A(J,I) * F                                             69210162
C                                                                       69210163
  270 CONTINUE                                                          69210164
C                                                                       69210165
      IF (NOCONV) GO TO 190                                             69210166
C                                                                       69210167
  280 LOW = K                                                           69210168
      IGH = L                                                           69210169
      RETURN                                                            69210170
C     :::::::::: LAST CARD OF BALANC ::::::::::                         69210171
      END                                                               69210172
```

```
C                                                                       73210001
C     ------------------------------------------------------------------73210002
C                                                                       73210003
      SUBROUTINE ELMHES(NM,N,LOW,IGH,A,INT)                             73210004
C                                                                       73210005
      INTEGER I,J,M,N,LA,NM,IGH,KP1,LOW,MM1,MP1                         73210006
      REAL*8 A(NM,N)                                                    73210007
      REAL*8 X,Y                                                        73210008
      REAL*8 DABS                                                       73210009
      INTEGER INT(IGH)                                                  73210010
C                                                                       73210011
C     THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE ELMHES,   73210012
C     NUM. MATH. 12, 349-368(1968) BY MARTIN AND WILKINSON.             73210013
C     HANDBOOK FOR AUTO. COMP., VOL.II-LINEAR ALGEBRA, 339-358(1971).   73210014
C                                                                       73210015
C     GIVEN A REAL GENERAL MATRIX, THIS SUBROUTINE                      73210016
C     REDUCES A SUBMATRIX SITUATED IN ROWS AND COLUMNS                  73210017
C     LOW THROUGH IGH TO UPPER HESSENBERG FORM BY                       73210018
C     STABILIZED ELEMENTARY SIMILARITY TRANSFORMATIONS.                 73210019
C                                                                       73210020
C     ON INPUT:                                                         73210021
C                                                                       73210022
C        NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL         73210023
C          ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM          73210024
C          DIMENSION STATEMENT;                                         73210025
C                                                                       73210026
C        N IS THE ORDER OF THE MATRIX;                                  73210027
C                                                                       73210028
C        LOW AND IGH ARE INTEGERS DETERMINED BY THE BALANCING           73210029
C          SUBROUTINE  BALANC.  IF  BALANC  HAS NOT BEEN USED,          73210030
C          SET LOW=1, IGH=N;                                            73210031
C                                                                       73210032
C        A CONTAINS THE INPUT MATRIX.                                   73210033
C                                                                       73210034
C     ON OUTPUT:                                                        73210035
C                                                                       73210036
```

```
C        A CONTAINS THE HESSENBERG MATRIX.  THE MULTIPLIERS             73210037
C          WHICH WERE USED IN THE REDUCTION ARE STORED IN THE           73210038
C          REMAINING TRIANGLE UNDER THE HESSENBERG MATRIX;              73210039
C                                                                       73210040
C        INT CONTAINS INFORMATION ON THE ROWS AND COLUMNS               73210041
C          INTERCHANGED IN THE REDUCTION.                               73210042
C          ONLY ELEMENTS LOW THROUGH IGH ARE USED.                      73210043
C                                                                       73210044
C     QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,        73210045
C     APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY         73210046
C                                                                       73210047
C     ------------------------------------------------------------------73210048
C                                                                       73210049
      LA = IGH - 1                                                      73210050
      KP1 = LOW + 1                                                     73210051
      IF (LA .LT. KP1) GO TO 200                                        73210052
C                                                                       73210053
      DO 180 M = KP1, LA                                                73210054
         MM1 = M - 1                                                    73210055
         X = 0.0D0                                                      73210056
         I = M                                                          73210057
C                                                                       73210058
         DO 100 J = M, IGH                                              73210059
            IF (DABS(A(J,MM1)) .LE. DABS(X)) GO TO 100                  73210060
            X = A(J,MM1)                                                73210061
            I = J                                                       73210062
  100    CONTINUE                                                       73210063
C                                                                       73210064
         INT(M) = I                                                     73210065
         IF (I .EQ. M) GO TO 130                                        73210066
C    :::::::::: INTERCHANGE ROWS AND COLUMNS OF A ::::::::::            73210067
         DO 110 J = MM1, N                                              73210068
            Y = A(I,J)                                                  73210069
            A(I,J) = A(M,J)                                             73210070
            A(M,J) = Y                                                  73210071
  110    CONTINUE                                                       73210072
```

```
C
         DO 120 J = 1, IGH
            Y = A(J,I)
            A(J,I) = A(J,M)
            A(J,M) = Y
  120    CONTINUE
C    :::::::::: END INTERCHANGE ::::::::::
  130    IF (X .EQ. 0.0D0) GO TO 180
         MP1 = M + 1
C
         DO 160 I = MP1, IGH
            Y = A(I,MM1)
            IF (Y .EQ. 0.0D0) GO TO 160
            Y = Y / X
            A(I,MM1) = Y
C
            DO 140 J = M, N
  140       A(I,J) = A(I,J) - Y * A(M,J)
C
            DO 150 J = 1, IGH
  150       A(J,M) = A(J,M) + Y * A(J,I)
C
  160    CONTINUE
C
  180 CONTINUE
C
  200 RETURN
C    :::::::::: LAST CARD OF ELMHES ::::::::::
      END
```

```
C                                                                       20210001
C     ------------------------------------------------------------------20210002
C                                                                       20210003
      SUBROUTINE ELTRAN(NM,N,LOW,IGH,A,INT,Z)                           20210004
C                                                                       20210005
      INTEGER I,J,N,KL,MM,MP,NM,IGH,LOW,MP1                             20210006
      REAL*8 A(NM,IGH),Z(NM,N)                                          20210007
      INTEGER INT(IGH)                                                  20210008
C                                                                       20210009
C     THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE ELMTRANS, 20210010
C     NUM. MATH. 16, 181-204(1970) BY PETERS AND WILKINSON.             20210011
C     HANDBOOK FOR AUTO. COMP., VOL.II-LINEAR ALGEBRA, 372-395(1971).   20210012
C                                                                       20210013
C     THIS SUBROUTINE ACCUMULATES THE STABILIZED ELEMENTARY             20210014
C     SIMILARITY TRANSFORMATIONS USED IN THE REDUCTION OF A             20210015
C     REAL GENERAL MATRIX TO UPPER HESSENBERG FORM BY  ELMHES.           20210016
C                                                                       20210017
C     ON INPUT:                                                         20210018
C                                                                       20210019
C        NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL         20210020
C          ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM          20210021
C          DIMENSION STATEMENT;                                         20210022
C                                                                       20210023
C        N IS THE ORDER OF THE MATRIX;                                  20210024
C                                                                       20210025
C        LOW AND IGH ARE INTEGERS DETERMINED BY THE BALANCING           20210026
C          SUBROUTINE  BALANC.  IF  BALANC  HAS NOT BEEN USED,          20210027
C          SET LOW=1, IGH=N;                                            20210028
C                                                                       20210029
C        A CONTAINS THE MULTIPLIERS WHICH WERE USED IN THE              20210030
C          REDUCTION BY  ELMHES  IN ITS LOWER TRIANGLE                  20210031
C          BELOW THE SUBDIAGONAL;                                       20210032
C                                                                       20210033
C        INT CONTAINS INFORMATION ON THE ROWS AND COLUMNS               20210034
C          INTERCHANGED IN THE REDUCTION BY  ELMHES.                    20210035
C          ONLY ELEMENTS LOW THROUGH IGH ARE USED.                      20210036
```

```
C                                                                       20210037
C     ON OUTPUT:                                                        20210038
C                                                                       20210039
C        Z CONTAINS THE TRANSFORMATION MATRIX PRODUCED IN THE           20210040
C          REDUCTION BY  ELMHES.                                        20210041
C                                                                       20210042
C     QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,        20210043
C     APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY         20210044
C                                                                       20210045
C     ------------------------------------------------------------------20210046
C                                                                       20210047
C     :::::::::: INITIALIZE Z TO IDENTITY MATRIX ::::::::::             20210048
      DO 80 I = 1, N                                                    20210049
C                                                                       20210050
         DO 60 J = 1, N                                                 20210051
   60    Z(I,J) = 0.0D0                                                 20210052
C                                                                       20210053
         Z(I,I) = 1.0D0                                                 20210054
   80 CONTINUE                                                          20210055
C                                                                       20210056
      KL = IGH - LOW - 1                                                20210057
      IF (KL .LT. 1) GO TO 200                                          20210058
C     :::::::::: FOR MP=IGH-1 STEP -1 UNTIL LOW+1 DO -- ::::::::::      20210059
      DO 140 MM = 1, KL                                                 20210060
         MP = IGH - MM                                                  20210061
         MP1 = MP + 1                                                   20210062
C                                                                       20210063
         DO 100 I = MP1, IGH                                            20210064
  100    Z(I,MP) = A(I,MP-1)                                            20210065
C                                                                       20210066
         I = INT(MP)                                                    20210067
         IF (I .EQ. MP) GO TO 140                                       20210068
C                                                                       20210069
         DO 130 J = MP, IGH                                             20210070
            Z(MP,J) = Z(I,J)                                            20210071
            Z(I,J) = 0.0D0                                              20210072
```

```
  130    CONTINUE                                                       20210073
C                                                                       20210074
         Z(I,NP) = 1.000                                                20210075
  140 CONTINUE                                                          20210076
C                                                                       20210077
  200 RETURN                                                            20210078
C     :::::::::: LAST CARD OF ELTRAN ::::::::::                         20210079
      END                                                               20210080
```

```
C
C     ------------------------------------------------------------------
C
      SUBROUTINE HQR2(NM,N,LOW,IGH,H,WR,WI,Z,IERR)
C
      INTEGER I,J,K,L,M,N,EN,II,JJ,LL,MM,NA,NM,NN,
     X        IGH,ITS,LOW,MP2,ENM2,IERR
      REAL*8 H(NM,N),WR(N),WI(N),Z(NM,N)
      REAL*8 P,Q,R,S,T,W,X,Y,RA,SA,VI,VR,ZZ,NORM,MACHEP
      REAL*8 DSQRT,DABS,DSIGN
      INTEGER MIN0
      LOGICAL NOTLAS
      COMPLEX*16 Z3
      COMPLEX*16 DCMPLX
      REAL*8 T3(2)
      EQUIVALENCE (Z3,T3(1))
C
C     THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE HQR2,
C     NUM. MATH. 16, 181-204(1970) BY PETERS AND WILKINSON.
C     HANDBOOK FOR AUTO. COMP., VOL.II-LINEAR ALGEBRA, 372-395(1971).
C
C     THIS SUBROUTINE FINDS THE EIGENVALUES AND EIGENVECTORS
C     OF A REAL UPPER HESSENBERG MATRIX BY THE QR METHOD.  THE
C     EIGENVECTORS OF A REAL GENERAL MATRIX CAN ALSO BE FOUND
C     IF  ELMHES  AND  ELTRAN  OR  ORTHES  AND  ORTRAN  HAVE
C     BEEN USED TO REDUCE THIS GENERAL MATRIX TO HESSENBERG FORM
C     AND TO ACCUMULATE THE SIMILARITY TRANSFORMATIONS.
C
C     ON INPUT:
C
C        NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL
C          ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM
C          DIMENSION STATEMENT;
C
C        N IS THE ORDER OF THE MATRIX;
C
```

```
C        LOW AND IGH ARE INTEGERS DETERMINED BY THE BALANCING           87210037
C          SUBROUTINE  BALANC.  IF  BALANC  HAS NOT BEEN USED,          87210038
C          SET LOW=1, IGH=N;                                            87210039
C                                                                       87210040
C        H CONTAINS THE UPPER HESSENBERG MATRIX;                        87210041
C                                                                       87210042
C        Z CONTAINS THE TRANSFORMATION MATRIX PRODUCED BY  ELTRAN       87210043
C          AFTER THE REDUCTION BY  ELMHES, OR BY  ORTRAN  AFTER THE     87210044
C          REDUCTION BY  ORTHES, IF PERFORMED.  IF THE EIGENVECTORS     87210045
C          OF THE HESSENBERG MATRIX ARE DESIRED, Z MUST CONTAIN THE     87210046
C          IDENTITY MATRIX.                                             87210047
C                                                                       87210048
C     ON OUTPUT:                                                        87210049
C                                                                       87210050
C        H HAS BEEN DESTROYED;                                          87210051
C                                                                       87210052
C        WR AND WI CONTAIN THE REAL AND IMAGINARY PARTS,                87210053
C          RESPECTIVELY, OF THE EIGENVALUES.  THE EIGENVALUES           87210054
C          ARE UNORDERED EXCEPT THAT COMPLEX CONJUGATE PAIRS            87210055
C          OF VALUES APPEAR CONSECUTIVELY WITH THE EIGENVALUE           87210056
C          HAVING THE POSITIVE IMAGINARY PART FIRST.  IF AN             87210057
C          ERROR EXIT IS MADE, THE EIGENVALUES SHOULD BE CORRECT        87210058
C          FOR INDICES IERR+1,...,N;                                    87210059
C                                                                       87210060
C        Z CONTAINS THE REAL AND IMAGINARY PARTS OF THE EIGENVECTORS.   87210061
C          IF THE I-TH EIGENVALUE IS REAL, THE I-TH COLUMN OF Z         87210062
C          CONTAINS ITS EIGENVECTOR.  IF THE I-TH EIGENVALUE IS COMPLEX 87210063
C          WITH POSITIVE IMAGINARY PART, THE I-TH AND (I+1)-TH          87210064
C          COLUMNS OF Z CONTAIN THE REAL AND IMAGINARY PARTS OF ITS     87210065
C          EIGENVECTOR.  THE EIGENVECTORS ARE UNNORMALIZED.  IF AN      87210066
C          ERROR EXIT IS MADE, NONE OF THE EIGENVECTORS HAS BEEN FOUND; 87210067
C                                                                       87210068
C        IERR IS SET TO                                                 87210069
C          ZERO       FOR NORMAL RETURN,                                87210070
C          J          IF THE J-TH EIGENVALUE HAS NOT BEEN               87210071
C                     DETERMINED AFTER 30 ITERATIONS.                   87210072
```

```
C                                                                       87210073
C     ARITHMETIC IS REAL EXCEPT FOR THE REPLACEMENT OF THE ALGOL        87210074
C     PROCEDURE CDIV BY COMPLEX DIVISION.                               87210075
C                                                                       87210076
C     QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,        87210077
C     APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY         87210078
C                                                                       87210079
C     ------------------------------------------------------------------87210080
C                                                                       87210081
C     :::::::::: MACHEP IS A MACHINE DEPENDENT PARAMETER SPECIFYING     87210082
C                THE RELATIVE PRECISION OF FLOATING POINT ARITHMETIC.   87210083
C                MACHEP = 16.0D0**(-13) FOR LONG FORM ARITHMETIC        87210084
C                ON S360 ::::::::::                                     87210085
      DATA MACHEP/Z3410000000000000/                                    87210086
C                                                                       87210087
      IERR = 0                                                          87210088
C     :::::::::: STORE ROOTS ISOLATED BY BALANC ::::::::::              87210089
      DO 50 I = 1, N                                                    87210090
         IF (I .GE. LOW .AND. I .LE. IGH) GO TO 50                      87210091
         WR(I) = H(I,I)                                                 87210092
         WI(I) = 0.0D0                                                  87210093
   50 CONTINUE                                                          87210094
C                                                                       87210095
      EN = IGH                                                          87210096
      T = 0.0D0                                                         87210097
C     :::::::::: SEARCH FOR NEXT EIGENVALUES ::::::::::                 87210098
   60 IF (EN .LT. LOW) GO TO 340                                        87210099
      ITS = 0                                                           87210100
      NA = EN - 1                                                       87210101
      ENM2 = NA - 1                                                     87210102
C     :::::::::: LOOK FOR SINGLE SMALL SUB-DIAGONAL ELEMENT             87210103
C                FOR L=EN STEP -1 UNTIL LOW DO -- ::::::::::             87210104
   70 DO 80 LL = LOW, EN                                                87210105
         L = EN + LOW - LL                                              87210106
         IF (L .EQ. LOW) GO TO 100                                      87210107
         IF (DABS(H(L,L-1)) .LE. MACHEP * (DABS(H(L-1,L-1))             87210108
```

```
     X       + DABS(H(L,L)))) GO TO 100                                 87210109
   80 CONTINUE                                                          87210110
C     :::::::::: FORM SHIFT ::::::::::                                  87210111
  100 X = H(EN,EN)                                                      87210112
      IF (L .EQ. EN) GO TO 270                                          87210113
      Y = H(NA,NA)                                                      87210114
      W = H(EN,NA) * H(NA,EN)                                           87210115
      IF (L .EQ. NA) GO TO 280                                          87210116
      IF (ITS .EQ. 30) GO TO 1000                                       87210117
      IF (ITS .NE. 10 .AND. ITS .NE. 20) GO TO 130                      87210118
C     :::::::::: FORM EXCEPTIONAL SHIFT ::::::::::                      87210119
      T = T + X                                                         87210120
C                                                                       87210121
      DO 120 I = LOW, EN                                                87210122
  120 H(I,I) = H(I,I) - X                                               87210123
C                                                                       87210124
      S = DABS(H(EN,NA)) + DABS(H(NA,ENM2))                             87210125
      X = 0.75D0 * S                                                    87210126
      Y = X                                                             87210127
      W = -0.4375D0 * S * S                                             87210128
  130 ITS = ITS + 1                                                     87210129
C     :::::::::: LOOK FOR TWO CONSECUTIVE SMALL                         87210130
C                SUB-DIAGONAL ELEMENTS.                                 87210131
C                FOR M=EN-2 STEP -1 UNTIL L DO -- ::::::::::            87210132
      DO 140 MM = L, ENM2                                               87210133
         M = ENM2 + L - MM                                              87210134
         ZZ = H(M,M)                                                    87210135
         R = X - ZZ                                                     87210136
         S = Y - ZZ                                                     87210137
         P = (R * S - W) / H(M+1,M) + H(M,M+1)                          87210138
         Q = H(M+1,M+1) - ZZ - R - S                                    87210139
         R = H(M+2,M+1)                                                 87210140
         S = DABS(P) + DABS(Q) + DABS(R)                                87210141
         P = P / S                                                      87210142
         Q = Q / S                                                      87210143
         R = R / S                                                      87210144
```

```
         IF (M .EQ. L) GO TO 150                                        87210145
         IF (DABS(H(M,M-1)) * (DABS(Q) + DABS(R)) .LE. MACHEP * DABS(P) 87210146
     X    * (DABS(H(M-1,M-1)) + DABS(ZZ) + DABS(H(M+1,M+1)))) GO TO 150 87210147
 140  CONTINUE                                                          87210148
C                                                                       87210149
 150  MP2 = M + 2                                                       87210150
C                                                                       87210151
      DO 160 I = MP2, EN                                                87210152
         H(I,I-2) = 0.0D0                                               87210153
         IF (I .EQ. MP2) GO TO 160                                      87210154
         H(I,I-3) = 0.0D0                                               87210155
 160  CONTINUE                                                          87210156
C     :::::::::: DOUBLE QR STEP INVOLVING ROWS L TO EN AND              87210157
C                COLUMNS M TO EN ::::::::::                             87210158
      DO 260 K = M, NA                                                  87210159
         NOTLAS = K .NE. NA                                             87210160
         IF (K .EQ. M) GO TO 170                                        87210161
         P = H(K,K-1)                                                   87210162
         Q = H(K+1,K-1)                                                 87210163
         R = 0.0D0                                                      87210164
         IF (NOTLAS) R = H(K+2,K-1)                                     87210165
         X = DABS(P) + DABS(Q) + DABS(R)                                87210166
         IF (X .EQ. 0.0D0) GO TO 260                                    87210167
         P = P / X                                                      87210168
         Q = Q / X                                                      87210169
         R = R / X                                                      87210170
 170     S = DSIGN(DSQRT(P*P+Q*Q+R*R),P)                                87210171
         IF (K .EQ. M) GO TO 180                                        87210172
         H(K,K-1) = -S * X                                              87210173
         GO TO 190                                                      87210174
 180     IF (L .NE. M) H(K,K-1) = -H(K,K-1)                             87210175
 190     P = P + S                                                      87210176
         X = P / S                                                      87210177
         Y = Q / S                                                      87210178
         ZZ = R / S                                                     87210179
         Q = Q / P                                                      87210180
```

```
         R = R / P
C     :::::::::: ROW MODIFICATION ::::::::::
         DO 210 J = K, N
            P = H(K,J) + Q * H(K+1,J)
            IF (.NOT. NOTLAS) GO TO 200
            P = P + R * H(K+2,J)
            H(K+2,J) = H(K+2,J) - P * ZZ
  200       H(K+1,J) = H(K+1,J) - P * Y
            H(K,J) = H(K,J) - P * X
  210    CONTINUE
C
         J = MIN0(EN,K+3)
C     :::::::::: COLUMN MODIFICATION ::::::::::
         DO 230 I = 1, J
            P = X * H(I,K) + Y * H(I,K+1)
            IF (.NOT. NOTLAS) GO TO 220
            P = P + ZZ * H(I,K+2)
            H(I,K+2) = H(I,K+2) - P * R
  220       H(I,K+1) = H(I,K+1) - P * Q
            H(I,K) = H(I,K) - P
  230    CONTINUE
C     :::::::::: ACCUMULATE TRANSFORMATIONS ::::::::::
         DO 250 I = LOW, IGH
            P = X * Z(I,K) + Y * Z(I,K+1)
            IF (.NOT. NOTLAS) GO TO 240
            P = P + ZZ * Z(I,K+2)
            Z(I,K+2) = Z(I,K+2) - P * R
  240       Z(I,K+1) = Z(I,K+1) - P * Q
            Z(I,K) = Z(I,K) - P
  250    CONTINUE
C
  260 CONTINUE
C
      GO TO 70
C     :::::::::: ONE ROOT FOUND ::::::::::
  270 H(EN,EN) = X + T
```

```
      WR(EN) = H(EN,EN)                                                 87210217
      WI(EN) = 0.0D0                                                    87210218
      EN = NA                                                           87210219
      GO TO 60                                                          87210220
C     :::::::::: TWO ROOTS FOUND ::::::::::                             87210221
  280 P = (Y - X) / 2.0D0                                               87210222
      Q = P * P + W                                                     87210223
      ZZ = DSQRT(DABS(Q))                                               87210224
      H(EN,EN) = X + T                                                  87210225
      X = H(EN,EN)                                                      87210226
      H(NA,NA) = Y + T                                                  87210227
      IF (Q .LT. 0.0D0) GO TO 320                                       87210228
C     :::::::::: REAL PAIR ::::::::::                                   87210229
      ZZ = P + DSIGN(ZZ,P)                                              87210230
      WR(NA) = X + ZZ                                                   87210231
      WR(EN) = WR(NA)                                                   87210232
      IF (ZZ .NE. 0.0D0) WR(EN) = X - W / ZZ                            87210233
      WI(NA) = 0.0D0                                                    87210234
      WI(EN) = 0.0D0                                                    87210235
      X = H(EN,NA)                                                      87210236
      R = DSQRT(X*X+ZZ*ZZ)                                              87210237
      P = X / R                                                         87210238
      Q = ZZ / R                                                        87210239
C     :::::::::: ROW MODIFICATION ::::::::::                            87210240
      DO 290 J = NA, N                                                  87210241
         ZZ = H(NA,J)                                                   87210242
         H(NA,J) = Q * ZZ + P * H(EN,J)                                 87210243
         H(EN,J) = Q * H(EN,J) - P * ZZ                                 87210244
  290 CONTINUE                                                          87210245
C     :::::::::: COLUMN MODIFICATION ::::::::::                         87210246
      DO 300 I = 1, EN                                                  87210247
         ZZ = H(I,NA)                                                   87210248
         H(I,NA) = Q * ZZ + P * H(I,EN)                                 87210249
         H(I,EN) = Q * H(I,EN) - P * ZZ                                 87210250
  300 CONTINUE                                                          87210251
C     :::::::::: ACCUMULATE TRANSFORMATIONS ::::::::::                  87210252
```

```
      DO 310 I = LOW, IGH
         ZZ = Z(I,NA)
         Z(I,NA) = Q * ZZ + P * Z(I,EN)
         Z(I,EN) = Q * Z(I,EN) - P * ZZ
  310 CONTINUE
C
      GO TO 330
C     :::::::::: COMPLEX PAIR ::::::::::
  320 WR(NA) = X + P
      WR(EN) = X + P
      WI(NA) = ZZ
      WI(EN) = -ZZ
  330 EN = ENM2
      GO TO 60
C     :::::::::: ALL ROOTS FOUND.  BACKSUBSTITUTE TO FIND
C                VECTORS OF UPPER TRIANGULAR FORM ::::::::::
  340 NORM = 0.0D0
      K = 1
C
      DO 360 I = 1, N
C
         DO 350 J = K, N
  350    NORM = NORM + DABS(H(I,J))
C
         K = I
  360 CONTINUE
C
      IF (NORM .EQ. 0.0D0) GO TO 1001
C     :::::::::: FOR EN=N STEP -1 UNTIL 1 DO -- ::::::::::
      DO 800 NN = 1, N
         EN = N + 1 - NN
         P = WR(EN)
         Q = WI(EN)
         NA = EN - 1
         IF (Q) 710, 600, 800
C     :::::::::: REAL VECTOR ::::::::::
```

```
  600 M = EN
      H(EN,EN) = 1.0D0
      IF (NA .EQ. 0) GO TO 800
C     :::::::::: FOR I=EN-1 STEP -1 UNTIL 1 DO -- ::::::::::
      DO 700 II = 1, NA
         I = EN - II
         W = H(I,I) - P
         R = H(I,EN)
         IF (M .GT. NA) GO TO 620
C
         DO 610 J = M, NA
  610    R = R + H(I,J) * H(J,EN)
C
  620    IF (WI(I) .GE. 0.0D0) GO TO 630
         ZZ = W
         S = R
         GO TO 700
  630    M = I
         IF (WI(I) .NE. 0.0D0) GO TO 640
         T = W
         IF (W .EQ. 0.0D0) T = MACHEP * NORM
         H(I,EN) = -R / T
         GO TO 700
C     :::::::::: SOLVE REAL EQUATIONS ::::::::::
  640    X = H(I,I+1)
         Y = H(I+1,I)
         Q = (WR(I) - P) * (WR(I) - P) + WI(I) * WI(I)
         T = (X * S - ZZ * R) / Q
         H(I,EN) = T
         IF (DABS(X) .LE. DABS(ZZ)) GO TO 650
         H(I+1,EN) = (-R - W * T) / X
         GO TO 700
  650    H(I+1,EN) = (-S - Y * T) / ZZ
  700 CONTINUE
C     :::::::::: END REAL VECTOR ::::::::::
      GO TO 800
```

```
C     :::::::::: COMPLEX VECTOR ::::::::::                              87210325
  710 M = NA                                                            87210326
C     :::::::::: LAST VECTOR COMPONENT CHOSEN IMAGINARY SO THAT         87210327
C                EIGENVECTOR MATRIX IS TRIANGULAR ::::::::::            87210328
      IF (DABS(H(EN,NA)) .LE. DABS(H(NA,EN))) GO TO 720                 87210329
      H(NA,NA) = Q / H(EN,NA)                                           87210330
      H(NA,EN) = -(H(EN,EN) - P) / H(EN,NA)                             87210331
      GO TO 730                                                         87210332
  720 Z3 = DCMPLX(0.0D0,-H(NA,EN)) / DCMPLX(H(NA,NA)-P,Q)               87210333
      H(NA,NA) = T3(1)                                                  87210334
      H(NA,EN) = T3(2)                                                  87210335
  730 H(EN,NA) = 0.0D0                                                  87210336
      H(EN,EN) = 1.0D0                                                  87210337
      ENM2 = NA - 1                                                     87210338
      IF (ENM2 .EQ. 0) GO TO 800                                        87210339
C                                                                       87210340
      DO 790 II = 1, ENM2                                               87210341
         I = NA - II                                                    87210342
         W = H(I,I) - P                                                 87210343
         RA = 0.0D0                                                     87210344
         SA = H(I,EN)                                                   87210345
C                                                                       87210346
         DO 760 J = M, NA                                               87210347
            RA = RA + H(I,J) * H(J,NA)                                  87210348
            SA = SA + H(I,J) * H(J,EN)                                  87210349
  760    CONTINUE                                                       87210350
C                                                                       87210351
         IF (WI(I) .GE. 0.0D0) GO TO 770                                87210352
         ZZ = W                                                         87210353
         R = RA                                                         87210354
         S = SA                                                         87210355
         GO TO 790                                                      87210356
  770    M = I                                                          87210357
         IF (WI(I) .NE. 0.0D0) GO TO 780                                87210358
         Z3 = DCMPLX(-RA,-SA) / DCMPLX(W,Q)                             87210359
         H(I,NA) = T3(1)                                                87210360
```

```
               H(I,EN) = T3(2)                                          87210361
               GO TO 790                                                87210362
C    :::::::::: SOLVE COMPLEX EQUATIONS ::::::::::                      87210363
  780          X = H(I,I+1)                                             87210364
               Y = H(I+1,I)                                             87210365
               VR = (WR(I) - P) * (WR(I) - P) + WI(I) * WI(I) - Q * Q   87210366
               VI = (WR(I) - P) * 2.0D0 * Q                             87210367
               IF (VR .EQ. 0.0D0 .AND. VI .EQ. 0.0D0) VR = MACHEP * NORM 87210368
     X          * (DABS(W) + DABS(Q) + DABS(X) + DABS(Y) + DABS(ZZ))    87210369
               Z3 = DCMPLX(X*R-ZZ*RA+Q*SA,X*S-ZZ*SA-Q*RA) / DCMPLX(VR,VI) 87210370
               H(I,NA) = T3(1)                                          87210371
               H(I,EN) = T3(2)                                          87210372
               IF (DABS(X) .LE. DABS(ZZ) + DABS(Q)) GO TO 785           87210373
               H(I+1,NA) = (-RA - W * H(I,NA) + Q * H(I,EN)) / X        87210374
               H(I+1,EN) = (-SA - W * H(I,EN) - Q * H(I,NA)) / X        87210375
               GO TO 790                                                87210376
  785          Z3 = DCMPLX(-R-Y*H(I,NA),-S-Y*H(I,EN)) / DCMPLX(ZZ,Q)    87210377
               H(I+1,NA) = T3(1)                                        87210378
               H(I+1,EN) = T3(2)                                        87210379
  790       CONTINUE                                                    87210380
C    :::::::::: END COMPLEX VECTOR ::::::::::                           87210381
  800 CONTINUE                                                          87210382
C    :::::::::: END BACK SUBSTITUTION.                                  87210383
C               VECTORS OF ISOLATED ROOTS ::::::::::                    87210384
      DO 840 I = 1, N                                                   87210385
         IF (I .GE. LOW .AND. I .LE. IGH) GO TO 840                     87210386
C                                                                       87210387
         DO 820 J = I, N                                                87210388
  820    Z(I,J) = H(I,J)                                                87210389
C                                                                       87210390
  840 CONTINUE                                                          87210391
C    :::::::::: MULTIPLY BY TRANSFORMATION MATRIX TO GIVE               87210392
C               VECTORS OF ORIGINAL FULL MATRIX.                        87210393
C               FOR J=N STEP -1 UNTIL LOW DO -- ::::::::::              87210394
      DO 880 JJ = LOW, N                                                87210395
         J = N + LOW - JJ                                               87210396
```

```
         M = MIN0(J,IGH)                                                87210397
C                                                                       87210398
         DO 880 I = LOW, IGH                                            87210399
            ZZ = 0.0D0                                                  87210400
C                                                                       87210401
            DO 860 K = LOW, M                                           87210402
  860       ZZ = ZZ + Z(I,K) * H(K,J)                                   87210403
C                                                                       87210404
            Z(I,J) = ZZ                                                 87210405
  880 CONTINUE                                                          87210406
C                                                                       87210407
      GO TO 1001                                                        87210408
C     :::::::::: SET ERROR -- NO CONVERGENCE TO AN                      87210409
C                EIGENVALUE AFTER 30 ITERATIONS ::::::::::              87210410
 1000 IERR = EN                                                         87210411
 1001 RETURN                                                            87210412
C     :::::::::: LAST CARD OF HQR2 ::::::::::                           87210413
      END                                                               87210414
```

```
C
C     ------------------------------------------------------------------
C
      SUBROUTINE BALBAK(NM,N,LOW,IGH,SCALE,M,Z)
C
      INTEGER I,J,K,M,N,II,NM,IGH,LOW
      REAL*8 SCALE(N),Z(NM,M)
      REAL*8 S
C
C     THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE BALBAK,
C     NUM. MATH. 13, 293-304(1969) BY PARLETT AND REINSCH.
C     HANDBOOK FOR AUTO. COMP., VOL.II-LINEAR ALGEBRA, 315-326(1971).
C
C     THIS SUBROUTINE FORMS THE EIGENVECTORS OF A REAL GENERAL
C     MATRIX BY BACK TRANSFORMING THOSE OF THE CORRESPONDING
C     BALANCED MATRIX DETERMINED BY  BALANC.
C
C     ON INPUT:
C
C        NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL
C          ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM
C          DIMENSION STATEMENT;
C
C        N IS THE ORDER OF THE MATRIX;
C
C        LOW AND IGH ARE INTEGERS DETERMINED BY  BALANC;
C
C        SCALE CONTAINS INFORMATION DETERMINING THE PERMUTATIONS
C          AND SCALING FACTORS USED BY  BALANC;
C
C        M IS THE NUMBER OF COLUMNS OF Z TO BE BACK TRANSFORMED;
C
C        Z CONTAINS THE REAL AND IMAGINARY PARTS OF THE EIGEN-
C          VECTORS TO BE BACK TRANSFORMED IN ITS FIRST M COLUMNS.
C
C     ON OUTPUT:
```

```
C                                                                       70210037
C        Z CONTAINS THE REAL AND IMAGINARY PARTS OF THE                 70210038
C          TRANSFORMED EIGENVECTORS IN ITS FIRST M COLUMNS.             70210039
C                                                                       70210040
C     QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,        70210041
C     APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY         70210042
C                                                                       70210043
C     ------------------------------------------------------------------70210044
C                                                                       70210045
      IF (IGH .EQ. LOW) GO TO 120                                       70210046
C                                                                       70210047
      DO 110 I = LOW, IGH                                               70210048
         S = SCALE(I)                                                   70210049
C     :::::::::: LEFT HAND EIGENVECTORS ARE BACK TRANSFORMED            70210050
C                IF THE FOREGOING STATEMENT IS REPLACED BY              70210051
C                S=1.0D0/SCALE(I). ::::::::::                           70210052
         DO 100 J = 1, M                                                70210053
  100    Z(I,J) = Z(I,J) * S                                            70210054
C                                                                       70210055
  110 CONTINUE                                                          70210056
C     ::::::::: FOR I=LOW-1 STEP -1 UNTIL 1,                            70210057
C               IGH+1 STEP 1 UNTIL N DO -- ::::::::::                   70210058
  120 DO 140 II = 1, N                                                  70210059
         I = II                                                         70210060
         IF (I .GE. LOW .AND. I .LE. IGH) GO TO 140                     70210061
         IF (I .LT. LOW) I = LOW - II                                   70210062
         K = SCALE(I)                                                   70210063
         IF (K .EQ. I) GO TO 140                                        70210064
C                                                                       70210065
         DO 130 J = 1, M                                                70210066
            S = Z(I,J)                                                  70210067
            Z(I,J) = Z(K,J)                                             70210068
            Z(K,J) = S                                                  70210069
  130    CONTINUE                                                       70210070
C                                                                       70210071
  140 CONTINUE                                                          70210072
```

```
C                                                                       70210073
      RETURN                                                            70210074
C     ::::::::::: LAST CARD OF BALBAK ::::::::::                        70210075
      END                                                               70210076
```

```
C
C     ..........................................................
C                                                                       MINV  30
C        SUBROUTINE MINV                                                MINV  40
C                                                                       MINV  50
C        PURPOSE                                                        MINV  60
C           INVERT A MATRIX                                             MINV  70
C                                                                       MINV  80
C        USAGE                                                          MINV  90
C           CALL MINV(A,N,D,L,M)                                        MINV 100
C                                                                       MINV 110
C        DESCRIPTION OF PARAMETERS                                      MINV 120
C           A - INPUT MATRIX, DESTROYED IN COMPUTATION AND REPLACED BY  MINV 130
C               RESULTANT INVERSE.                                      MINV 140
C           N - ORDER OF MATRIX A                                       MINV 150
C           D - RESULTANT DETERMINANT                                   MINV 160
C           L - WORK VECTOR OF LENGTH N                                 MINV 170
C           M - WORK VECTOR OF LENGTH N                                 MINV 180
C                                                                       MINV 190
C        REMARKS                                                        MINV 200
C           MATRIX A MUST BE A GENERAL MATRIX                           MINV 210
C                                                                       MINV 220
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                  MINV 230
C           NONE                                                        MINV 240
C                                                                       MINV 250
C        METHOD                                                         MINV 260
C           THE STANDARD GAUSS-JORDAN METHOD IS USED. THE DETERMINANT   MINV 270
C           IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT    MINV 280
C           THE MATRIX IS SINGULAR.                                     MINV 290
C                                                                       MINV 300
C     ..................................................................MINV 310
C                                                                       MINV 320
      SUBROUTINE MINV(A,N,D,L,M)                                        MINV 330
      DIMENSION A(1),L(1),M(1)                                          MINV 340
C                                                                       MINV 350
C        ...............................................................MINV 360
```

```
C                                                                       MINV 370
C        IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE   MINV 380
C        C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION       MINV 390
C        STATEMENT WHICH FOLLOWS.                                        MINV 400
C                                                                       MINV 410
C     DOUBLE PRECISION A,D,BIGA,HOLD                                     MINV 420
C                                                                       MINV 430
C        THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS     MINV 440
C        APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS       MINV 450
C        ROUTINE.                                                        MINV 460
C                                                                       MINV 470
C        THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO       MINV 480
C        CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS.  ABS IN STATEMENT   MINV 490
C        10 MUST BE CHANGED TO DABS.                                     MINV 500
C                                                                       MINV 510
C        ...............................................................MINV 520
C                                                                       MINV 530
C        SEARCH FOR LARGEST ELEMENT                                      MINV 540
C                                                                       MINV 550
      D=1.0                                                             MINV 560
      NK=-N                                                             MINV 570
      DO 80 K=1,N                                                       MINV 580
      NK=NK+N                                                           MINV 590
      L(K)=K                                                            MINV 600
      M(K)=K                                                            MINV 610
      KK=NK+K                                                           MINV 620
      BIGA=A(KK)                                                        MINV 630
      DO 20 J=K,N                                                       MINV 640
      IZ=N*(J-1)                                                        MINV 650
      DO 20 I=K,N                                                       MINV 660
      IJ=IZ+I                                                           MINV 670
   10 IF( ABS(BIGA)- ABS(A(IJ))) 15,20,20                               MINV 680
   15 BIGA=A(IJ)                                                        MINV 690
      L(K)=I                                                            MINV 700
      M(K)=J                                                            MINV 710
   20 CONTINUE                                                          MINV 720
```

```
C                                                                       MINV 730
C        INTERCHANGE ROWS                                               MINV 740
C                                                                       MINV 750
      J=L(K)                                                            MINV 760
      IF(J-K) 35,35,25                                                  MINV 770
   25 KI=K-N                                                            MINV 780
      DO 30 I=1,N                                                       MINV 790
      KI=KI+N                                                           MINV 800
      HOLD=-A(KI)                                                       MINV 810
      JI=KI-K+J                                                         MINV 820
      A(KI)=A(JI)                                                       MINV 830
   30 A(JI) =HOLD                                                       MINV 840
C                                                                       MINV 850
C        INTERCHANGE COLUMNS                                            MINV 860
C                                                                       MINV 870
   35 I=M(K)                                                            MINV 880
      IF(I-K) 45,45,38                                                  MINV 890
   38 JP=N*(I-1)                                                        MINV 900
      DO 40 J=1,N                                                       MINV 910
      JK=NK+J                                                           MINV 920
      JI=JP+J                                                           MINV 930
      HOLD=-A(JK)                                                       MINV 940
      A(JK)=A(JI)                                                       MINV 950
   40 A(JI) =HOLD                                                       MINV 960
C                                                                       MINV 970
C        DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS        MINV 980
C        CONTAINED IN BIGA)                                             MINV 990
C                                                                       MINV1000
   45 IF(BIGA) 48,46,48                                                 MINV1010
   46 D=0.0                                                             MINV1020
      RETURN                                                            MINV1030
   48 DO 55 I=1,N                                                       MINV1040
      IF(I-K) 50,55,50                                                  MINV1050
   50 IK=NK+I                                                           MINV1060
      A(IK)=A(IK)/(-BIGA)                                               MINV1070
   55 CONTINUE                                                          MINV1080
```

```
C                                                                       MINV1090
C        REDUCE MATRIX                                                  MINV1100
C                                                                       MINV1110
      DO 65 I=1,N                                                       MINV1120
      IK=NK+I                                                           MINV1130
      HOLD=A(IK)                                                        MINV1140
      IJ=I-N                                                            MINV1150
      DO 65 J=1,N                                                       MINV1160
      IJ=IJ+N                                                           MINV1170
      IF(I-K) 60,65,60                                                  MINV1180
   60 IF(J-K) 62,65,62                                                  MINV1190
   62 KJ=IJ-I+K                                                         MINV1200
      A(IJ)=HOLD*A(KJ)+A(IJ)                                            MINV1210
   65 CONTINUE                                                          MINV1220
C                                                                       MINV1230
C        DIVIDE ROW BY PIVOT                                            MINV1240
C                                                                       MINV1250
      KJ=K-N                                                            MINV1260
      DO 75 J=1,N                                                       MINV1270
      KJ=KJ+N                                                           MINV1280
      IF(J-K) 70,75,70                                                  MINV1290
   70 A(KJ)=A(KJ)/BIGA                                                  MINV1300
   75 CONTINUE                                                          MINV1310
C                                                                       MINV1320
C        PRODUCT OF PIVOTS                                              MINV1330
C                                                                       MINV1340
      D=D*BIGA                                                          MINV1350
C                                                                       MINV1360
C        REPLACE PIVOT BY RECIPROCAL                                    MINV1370
C                                                                       MINV1380
      A(KK)=1.0/BIGA                                                    MINV1390
   80 CONTINUE                                                          MINV1400
C                                                                       MINV1410
C        FINAL ROW AND COLUMN INTERCHANGE                               MINV1420
C                                                                       MINV1430
      K=N                                                               MINV1440
```

```
  100 K=(K-1)                                                          MINV1450
      IF(K) 150,150,105                                                MINV1460
  105 I=L(K)                                                           MINV1470
      IF(I-K) 120,120,108                                              MINV1480
  108 JQ=N*(K-1)                                                       MINV1490
      JR=N*(I-1)                                                       MINV1500
      DO 110 J=1,N                                                     MINV1510
      JK=JQ+J                                                          MINV1520
      HOLD=A(JK)                                                       MINV1530
      JI=JR+J                                                          MINV1540
      A(JK)=-A(JI)                                                     MINV1550
  110 A(JI) =HOLD                                                      MINV1560
  120 J=M(K)                                                           MINV1570
      IF(J-K) 100,100,125                                              MINV1580
  125 KI=K-N                                                           MINV1590
      DO 130 I=1,N                                                     MINV1600
      KI=KI+N                                                          MINV1610
      HOLD=A(KI)                                                       MINV1620
      JI=KI-K+J                                                        MINV1630
      A(KI)=-A(JI)                                                     MINV1640
  130 A(JI) =HOLD                                                      MINV1650
      GO TO 100                                                        MINV1660
  150 RETURN                                                           MINV1670
      END                                                              MINV1680
```

APPENDIX B

INPUT DATA AND OUTPUT LISTING OF THE SAMPLE PROBLEMS

B.1 Application of the FREEVI Program to the Wing

B.1.1 Input Data Listing for the Bell Wing

The FREEVI program input data for the wing are illustrated in this section. The structural data are shown in Fig. 6 and in Table 3.

```
          BELL  WING                                                                  8/17/74    DATA0001
 1                                                                                      ICASE    DATA0002
 0                                                                                      IPUNCH   DATA0003
 0                                                                                      IGUEST   DATA0004
     9    20    6                                                                 NET,NITP,M     DATA0005
 0.001                                                                                  ERR      DATA0006
 0.0                                                                                    OMEG     DATA0007
 0.0                                                                                    RAMDA    DATA0008
 0.0                                                                                    COL      DATA0009
 0.0        0.0                                                                     SPKB,SPKC    DATA0010
 0.0                                                                                    ALPHAH   DATA0011
 9202.9        E62653.0        E62653.0        E62653.0        E62653.0        E6                DATA0012
 2653.0        E62653.0        E62653.0        E62653.0        E6                                DATA0013
 18794.0       E610410.0       E610410.0       E610410.0       E610410.0       E6                DATA0014
 10410.0       E610410.0       E610410.0       E610410.0       E6                                DATA0015
 0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0                DATA0016
 0.0                                                                                             DATA0017
 0.00705    0.00705    0.00488    0.00488    0.00488    0.00488    0.00508    0.03248            DATA0018
 0.00001                                                                                         DATA0019
 41.0       21.0       21.0       21.0       21.0       21.0       21.0       18.4               DATA0020
 14.6                                                                                            DATA0021
 4.71       4288.8     6591.6     6570.0     53.076                                              DATA0022
 16868.0       E62696.0        E62696.0        E62696.0        E62696.0        E6                DATA0023
 2696.0        E62696.0        E62696.0        E62696.0        E6                                DATA0024
 0.389         E00.389         E00.389         E00.389         E00.389         E0                DATA0025
 0.389         E00.389         E00.389         E00.389         E0                                DATA0026
 0.0           E00.0           E00.0           E00.0           E00.0           E0                DATA0027
 0.0           E00.0           E00.0           E00.0           E0                                DATA0028
```

B.1.2 The FEEVI program output data for the Bell Wing.

The example output of the Bell wing is shown in this subsection.

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

          WING

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


                              BELL   WING                                                                   8/17/74


   *    *
  INPUT DATA


 IPUNCH=0       IGUEST=0
 NO OF DEGRE PER NODE=       6
  NO OF ELEMENTS=  9
 NO OF MAX ITER ALLOEWD= 20
  NO OF MODES=  6
  ERR=         0.00100
 OMEG=         0.0
 LAMBDA=            0.0
 COLLECTIVE PITCH=            0.0
 SPRING=                0.0              0.0
 ALPHAH=     0.0
 --FLAPPING BENDING STIFFNESS--
       0.9202901E+10  0.2653000E+10  0.2653000E+10  0.2653000E+10  0.2653000E+10  0.2653000E+10   .2653000E+10
       0.2653000E+10  0.2653000E+10
 --CHORDWISE BENDING STIFFNESS--
       0.1879400E+11  0.1041000E+11  0.1041000E+11  0.1041000E+11  0.1041000E+11  0.1041000E+11  0.1041000E+11
       0.1041000E+11  0.1041000E+11
 --ANGLE OF TWIST--
        0.0            0.0            0.0            0.0            0.0            0.0            0.0            0.
        0.0
 --MASS DISTRIBUTION--
        0.00705        0.00705        0.00488        0.00488        0.00488        0.00488        0.01508        .03248
        0.00061
 --ELEMENT SIZE--
       41.00000       21.00000       21.00000       21.00000       21.00000       21.00000       21.00000       19.39999
       14.60000
        TIP MASS   ROLL INERTIA   YAW INERTIA    PITCH INERTIA  MASS COUPLING
        4.71000      4288.80078     6591.60156     6570.00000         53.07600
 --TORSIONAL RIGIDITY--
       0.1696830E+11  0.2696000E+10  0.2696000E+10  0.2696000E+10  0.2696000E+10  0.2696000E+10  0.2696000E+10
       0.2696000E+10  0.2696000E+10
 --MOMENT OF INERTIA--
       0.3890000E+00  0.3890000E+00  0.3890000E+00  0.3890000E+00  0.3890000E+00  0.3890000E+00   .3890000E+00
       0.3890000E+00  0.3890000E+00
 --MASS COUPLING ALONG SPAN--
       0.0            0.0            0.0            0.0            0.0            0.0            0.
       0.0            0.0
```

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.

```
TENSION DUE TO CENTRIF FORCE
      0.0            0.0            0.0            0.0            0.0            0.0            0.0            0.0
      0.0            0.0

MASS =  0.62615E+01
MOMENT OF INERTIA AT ROOT=  0.21483E+06
TOTAL LENGTH OF THE BEAM=  0.20000E+03

MAX. SIZE OF STF IS   484     SPECIFIED SIZE IS    654

NO. OF NEGATIVE DIAGS.=   0     FACT. COMPLETED

EIGENVALUES=
  0.28091D+03  0.89998D+03  0.16245D+05  0.34651D+05  0.26504D+06  0.53303D+06

EIGENVALUES=
  0.27768D+03  0.89084D+03  0.27303D+04  0.12970D+05  0.32610D+05  0.27656D+06

EIGENVALUES=
  0.27768D+03  0.89084D+03  0.27300D+04  0.12968D+05  0.32494D+05  0.26217D+06

EIGENVALUES=
  0.27768D+03  0.89084D+03  0.27300D+04  0.12968D+05  0.32494D+05  0.26097D+06

EIGENVALUES=
  0.27768D+03  0.89084D+03  0.27300D+04  0.12968D+05  0.32494D+05  0.26086D+06

    RADIAN/SEC
  0.16664D+02  0.29847D+02  0.52250D+02  0.11388D+03  0.18026D+03  0.51074D+03

    HERTZ
  0.26521D+01  0.47503D+01  0.83158D+01  0.18124D+02  0.28689D+02  0.81287D+02

NO. OF ITERATION=   5  CONVERGED WITHIN  0.10000D-02

REDUCED MASS MATRIX

  0.10000D+01
  0.11699D-22  0.10000D+01
 -0.51511D-20 -0.33025D-21  0.10000D+01
 -0.11609D-19 -0.13470D-17 -0.92236D-19  0.10000D+01
  0.88490D-24 -0.13376D-20  0.12102D-18  0.10999D-19  0.10000D+01
 -0.51707D-20  0.76081D-20  0.13915D-17  0.86677D-18 -0.20838D-19  0.10000D+01

REDUCED STIF MATRIX

  0.27768D+03
  0.12056D-18  0.89084D+03
 -0.59669D-13  0.38420D-17  0.27300D+04
 -0.36503D-17 -0.11888D-16  0.93751D-16  0.12968D+05
 -0.27794D-16  0.63537D-13 -0.98895D-18 -0.58898D-14  0.32494D+05
 -0.44715D-16 -0.12874D-12 -0.11159D-15 -0.89091D-14  0.16965D-15  0.26086D+06
```

```
****  WING MODE SHAPES  ****


I= 1

   K       W(I,J)              V(I,J)              DW(I,J)             DV(I,J)             PHI(I,J)            DPHI(I,J)
   1       0.0                 0.0                 0.0                 0.0                 0.0                 0.4732315E-06
   2       0.1096412E-01      -0.4118297E-09       0.5154577E-03      -0.1607660E-10       0.1781619E-04       0.5906245E-05
   3       0.2995049E-01      -0.7596179E-09       0.1275148E-02      -0.1562293E-10       0.6824774E-04       0.2253[illegible]5E-05
   4       0.5378090E-01      -0.1020656E-08       0.1929206E-02      -0.8398275E-11       0.1220887E-03       0.2538390E-05
   5       0.1102392E+00      -0.1089811E-08       0.2477859E-02       0.2030671E-11       0.1765127E-03       0.2587192E-05
   6       0.1671160E+00      -0.9396308E-09       0.2921520E-02       0.1186243E-10       0.2310340E-03       0.2595412E-05
   7       0.2322127E+00      -0.6202676E-09       0.3260845E-02       0.1763036E-10       0.2855684E-03       0.2596636E-05
   8       0.3033477E+00      -0.2455758E-09       0.3496786E-02       0.1680664E-10       0.3401066E-03       0.2596617E-05
   9       0.3689417E+00       0.5387467E-11       0.3620509E-02       0.9286009E-11       0.3979768E-03       0.2596393E-05
  10       0.4221950E+00       0.6504591E-10       0.3666978E-02      -0.1889718E-11       0.4257825E-03       0.2596161E-05


I= 2

   K       W(I,J)              V(I,J)              DW(I,J)             DV(I,J)             PHI(I,J)            DPHI(I,J)
   1       0.0                 0.0                 0.0                 0.0                 0.0                 0.1334282E-13
   2      -0.1165887E-08       0.1734024E-01      -0.4099619E-10       0.8156588E-03       0.4967241E-12       0.1593391E-13
   3      -0.1780436E-08       0.4120878E-01      -0.1078385E-10       0.1443248E-02       0.1793459E-11       0.5271469E-13
   4      -0.1508654E-08       0.7735723E-01       0.3577688E-10       0.1985208E-02       0.2899238E-11       0.4211952E-13
   5      -0.4287495E-09       0.1239857E+00       0.6140244E-10       0.2441776E-02       0.3564264E-11       0.1910476E-13
   6       0.7976879E-09       0.1793168E+00       0.4901710E-10       0.2813344E-02       0.3702648E-11      -0.5992711E-14
   7       0.1419738E-08       0.2415592E+00       0.7250959E-11       0.3100497E-02       0.3337820E-11      -0.2797991E-13
   8       0.1085335E-08       0.3089525E+00      -0.3688012E-10       0.3304057E-02       0.2568981E-11      -0.4423885E-13
   9       0.2516420E-09       0.3708636E+00      -0.4400065E-10       0.3415339E-02       0.1653754E-11      -0.5336117E-13
  10      -0.1154645E-09       0.4211117E+00       0.2109291E-11       0.3461931E-02       0.8527415E-12      -0.5725260E-13


I= 3

   K       W(I,J)              V(I,J)              DW(I,J)             DV(I,J)             PHI(I,J)            DPHI(I,J)
   1       0.0                 0.0                 0.0                 0.0                 0.0                -0.1426321E-04
   2       0.3132011E-02       0.1067552E-05       0.1483440E-03       0.4164287E-07      -0.5369098E-03      -0.1776540E-04
   3       0.8682542E-02       0.1967319E-05       0.3762855E-03       0.4036162E-07      -0.2056599E-02      -0.4788566E-04
   4       0.1876992E-01       0.2640408E-05       0.5815080E-03       0.2158016E-07      -0.3678715E-02      -0.7646301E-04
   5       0.3290243E-01       0.2815990E-05       0.7615838E-03      -0.5406847E-08      -0.5317826E-02      -0.7790148E-04
   6       0.5060757E-01       0.2425011E-05       0.9210168E-03      -0.3075634E-07      -0.6959036E-02      -0.7810385E-04
   7       0.7144558E-01       0.1598726E-05       0.1060446E-02      -0.4554926E-07      -0.8599557E-02      -0.7908290E-04
   8       0.9502667E-01       0.6317913E-06       0.1182828E-02      -0.4332351E-07      -0.1023867E-01      -0.7901231E-04
   9       0.1177033E+00      -0.1453627E-07       0.1282109E-02      -0.2388719E-07      -0.1167341E-01      -0.7793421E-04
  10       0.1370279E+00      -0.1678002E-06       0.1366351E-02       0.4879347E-08      -0.1281075E-01      -0.7786400E-04
```

```
I= 4

 K       W(I,J)              V(I,J)              DW(I,J)             DV(I,J)             PHI(I,J)            DPHI(I,J)
 1       0.0                 0.0                 0.0                 0.0                 0.0                -0.8024932E-06
 2       0.5561896E-01       0.1292796E-04       0.2441053E-02       0.5067096E-06      -0.3020681E-04      -0.9993528E-06
 3       0.1320716E+00       0.2396124E-04       0.4596125E-02       0.4992868E-06      -0.1156774E-03      -0.3916693E-05
 4       0.2385222E+00       0.3238695E-04       0.5303770E-02       0.2753594E-06      -0.2068353E-03      -0.4294215E-05
 5       0.3449906E+00       0.3479325E-04       0.4606590E-02      -0.5469537E-07      -0.2989146E-03      -0.4367300E-05
 6       0.4226109E+00       0.3018290E-04       0.2568961E-02      -0.3721857E-06      -0.3907180E-03      -0.4368017E-05
 7       0.4440947E+00       0.2005251E-04      -0.7243240E-03      -0.5633744E-06      -0.4823287E-03      -0.4353352E-05
 8       0.3840748E+00       0.8010285E-05      -0.5176779E-02      -0.5431260E-06      -0.5735478E-03      -0.4333073E-05
 9       0.2462974E+00      -0.1363861E-06      -0.9906098E-02      -0.3031313E-06      -0.6537981E-03      -0.4312068E-05
10       0.7180148E-01      -0.2095530E-05      -0.1404954E-01       0.6066438E-07      -0.7159102E-03      -0.4293397E-05


I= 5

 K       W(I,J)              V(I,J)              DW(I,J)             DV(I,J)             PHI(I,J)            DPHI(I,J)
 1       0.0                 0.0                 0.0                 0.0                 0.0                -0.1459670E-10
 2       0.2360992E-05       0.6912452E-01       0.8368607E-07       0.2999669E-02      -0.5442764E-09      -0.1754913E-10
 3       0.3658400E-05       0.1467586E+00       0.2591068E-07       0.4220549E-02      -0.1985294E-08      -0.5971562E-10
 4        0.3202141E-05       0.2391850E+00      -0.6824558E-07       0.4412588E-02      -0.3275932E-08      -0.5190257E-10
 5        0.1072496E-05       0.3250826E+00      -0.1235147E-06       0.3604392E-02      -0.4168427E-08      -0.3356966E-10
 6       -0.1439281E-05       0.3838446E+00      -0.1026097E-06       0.1835852E-02      -0.4545821E-08      -0.4846506E-11
 7       -0.2793039E-05       0.3958154E+00      -0.1964723E-07      -0.8429433E-03      -0.4376279E-08       0.2069658E-10
 8       -0.2200011E-05       0.3424572E+00       0.7231574E-07      -0.4376560E-02      -0.3732741E-08       0.4262971E-10
 9       -0.5198935E-06       0.2284233E+00       0.9011211E-07      -0.8104231E-02      -0.2779422E-08       0.5684986E-10
10        0.2345656E-06       0.8644497E-01      -0.4498673E-08      -0.1138999E-01      -0.1892494E-08       0.6401363E-10


I= 6

 K       W(I,J)              V(I,J)              DW(I,J)             DV(I,J)             PHI(I,J)            DPHI(I,J)
 1       0.0                 0.0                 0.0                 0.0                 0.0                 0.1009331E-05
 2       0.4376729E+00      -0.2575393E-02       0.1791070E-01      -0.1012992E-03       0.4022557E-04       0.1326101E-05
 3       0.9022505E+00      -0.4794151E-02       0.2376941E-01      -0.1110866E-03       0.1530777E-03       0.4997642E-05
 4       0.1344618E+01      -0.6516449E-02       0.1658912E-01      -0.5735150E-04       0.2711348E-03       0.5473314E-05
 5       0.1544214E+01      -0.7044502E-02       0.1633530E-02       0.8875477E-05       0.3860518E-03       0.5325627E-05
 6       0.1403761E+01      -0.6153475E-02      -0.1466982E-01       0.7372542E-04       0.4943394E-03       0.4998302E-05
 7       0.9648077E+00      -0.4121412E-02      -0.2577179E-01       0.1139862E-03       0.5955135E-03       0.4572572E-05
 8       0.3952152E+00      -0.1666956E-02      -0.2639399E-01       0.1115181E-03       0.6963750E-03       0.4082211E-05
 9      -0.6146964E-02       0.1676549E-04      -0.1504268E-01       0.6317307E-04       0.7567226E-03       0.3549906E-05
10      -0.3912896E-01       0.4287334E-03       0.3793120E-02      -0.1231682E-04       0.8057512E-03       0.3141465E-05
```

B.2 Application of the FREEVI Program to the Blade

B.2.1 The FREEVI Program Input Data Listing for the Boeing Rotor (Hingeless Rotor)

The input data deck setup is illustrated in this subsection for the hingeless rotor. Structural data are shown in Fig. 7.

```
BOEING BLADE FOR POWERED FLIGHT  ---  CLAMPED & CLAMPED  B.C               8/18/74    DATA0001
2                                                                            ICASE    DATA0002
1                                                                           IPUNCH    DATA0003
0                                                                           IGUEST    DATA0004
   10     20     4                                                      NET,NITR,M    DATA0005
0.001                                                                          ERR    DATA0006
-40.422                                                                       OMEG    DATA0007
-0.7                                                                         RAMDA    DATA0008
0.017453                                                                       COL    DATA0009
0.0            0.0                                                       SPKB,SPKC    DATA0010
8525.0                                                                      ALPHAH    DATA0011
106.0         E619.0         E68.5          E67.5          E65.5             E6       DATA0012
5.0           E64.50         E64.5          E64.5          E64.0             E6       DATA0013
110.0         E640.0         E640.0         E6200.0        E6500.0           E6       DATA0014
520.0         E6510.0        E6515.0        E6530.0        E6535.0           E6       DATA0015
0.6981     0.4712     0.3403     0.2531     0.1833     0.1272     0.0611     0.0      DATA0016
-0.0611    -0.1134                                                                    DATA0017
0.016583   0.001036   0.000777   0.000906530.001036   0.001036   0.001036   0.0012303 DATA0018
0.0012303  0.001036                                                                   DATA0019
15.6       15.6       15.6       15.6       15.6       15.6       15.6       15.6     DATA0020
15.6       15.6                                                                       DATA0021
0.0106064                                                                             DATA0022
```

B.2.2 The FREEVI Program Output Data for the Boeing Blade

The example output of the Boeing blade is shown in this subsection.

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

```
          BLADE    BEAM*CANTI    CHORD*CANTI

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

                  BOEING BLADE FOR POWERED FLIGHT  ---  CLAMPED & CLAMPED  B.C.          8/19/74

 INPUT DATA

IPUNCH=1      IGUEST=0
NO OF DEGRE PER NODE=       4
 NO OF ELEMENTS= 10
NO OF MAX ITER ALLOEWD= 20
 NO OF MODES=  4
 ERR=          0.00100
OMEG=        -40.42200
LAMBDA=          -0.70000
COLLECTIVE PITCH=          0.01745
SPRING=              0.0              0.0
ALPHAH=     0.85250D+04
--FLAPPING BENDING STIFFNESS--
      0.1060000E+09  0.1900000E+08  0.8500000E+07  0.7500000E+07  0.5500000E+07  0.5000000E+07  0.4500000E+07
      0.4500000E+07  0.4500000E+07  0.4000000E+07
--CHORDWISE BENDING STIFFNESS--
      0.1100000E+09  0.4000000E+08  0.4000000E+08  0.2000000E+09  0.5000000E+09  0.5200000E+09  0.5100001E+09
      0.5150001E+09  0.5300001E+09  0.5350001E+09
--ANGLE OF TWIST--
      0.69810        0.47120        0.34030        0.25310        0.18330        0.12220        0.06110        0.0
     -0.06110       -0.11340
--MASS DISTRIBUTION--
       0.01658        0.00104        0.00078        0.00091        0.00104        0.00104        0.00104        0.00123
       0.00123        0.00104
--ELEMENT SIZE--
      15.60000       15.60000       15.60000       15.60000       15.60000       15.60000       15.60000       15.6000
      15.60000       15.60000
       TIP MASS  ROLL INERTIA   YAW INERTIA   PITCH INERTIA  MASS COUPLING
       0.01061        0.0           0.0            0.0            0.0


 TENSION DUE TO CENTRIF FORCE
   27190.4219     23893.4883     23275.5703     22503.1680     21241.5352     19387.7656     17122.3469     14444.178
   10775.3086      6617.02734     2703.50903

 MASS =  0.41476E+00
 MOMENT OF INERTIA AT ROOT=  0.16714E+04
 TOTAL LENGTH OF THE BEAM=  0.15600E+03

 MAX. SIZE OF STF IS    244       SPECIFIED SIZE IS     324

 NO. OF NEGATIVE DIAGS.=    0       FACT. COMPLETED
```

```
EIGENVALUES=
  0.20880D+04   0.12950D+05   0.27569D+05   0.42059D+06

EIGENVALUES=
  0.11180D+04   0.28840D+04   0.19005D+05   0.90243D+05

EIGENVALUES=
  0.11172D+04   0.28520D+04   0.18887D+05   0.75550D+05

EIGENVALUES=
  0.11172D+04   0.28519D+04   0.18886D+05   0.74979D+05

EIGENVALUES=
  0.11172D+04   0.28519D+04   0.18886D+05   0.74934D+05

    RADIAN/SEC
  0.33424D+02   0.53403D+02   0.13743D+03   0.27374D+03

    HERTZ
  0.53197D+01   0.84993D+01   0.21872D+02   0.43567D+02

NO. OF ITERATION=   5  CONVERGED WITHIN  0.1000D-02

REDUCED MASS MATRIX

  0.10000D+01
  0.92346D-19   0.10000D+01
  0.21682D-16   0.54358D-20   0.10000D+01
  0.92709D-20  -0.19679D-21  -0.12149D-19   0.10000D+01

REDUCED STIF MATRIX

  0.96422D+04
  0.59174D-14   0.11377D+05
 -0.77329D-15   0.45209D-16   0.27411D+05
  0.99997D-15  -0.51587D-16   0.30254D-14   0.83459D+05

****  BLADE MODE SHAPES  ****

I= 1

 K        W(I,J)              V(I,J)              DW(I,J)             DV(I,J)
 1        0.0                 0.0                 0.0                 0.0
 2       -0.3071345E-02       0.3320938E-01      -0.4138651E-03       0.3957305E-02
 3       -0.4010361E-02       0.1970022E+00       0.3891176E-04       0.1585099E-01
 4        0.1771531E-01       0.5429144E+00       0.2249592E-02       0.2720761E-01
 5        0.7252330E-01       0.1007425E+01       0.4471347E-02       0.3180430E-01
 6        0.1537749E+00       0.1521890E+01       0.5777858E-02       0.3390279E-01
 7        0.2471325E+00       0.2056107E+01       0.6131638E-02       0.3450109E-01
 8        0.3411351E+00       0.2593284E+01       0.5890395E-02       0.3432557E-01
 9        0.4277322E+00       0.3123945E+01       0.5184587E-02       0.3367477E-01
10        0.5013666E+00       0.3643093E+01       0.4255518E-02       0.3287940E-01
11        0.5611690E+00       0.4150885E+01       0.3513225E-02       0.3230092E-01
```

```
I= 2

  K       W(I,J)            V(I,J)            DW(I,J)           DV(I,J)
  1       0.0               0.0               0.0               0.0
  2       0.4854569E-01    -0.1528439E-01     0.5902052E-02    -0.1884356E-02
  3       0.2357861E+00    -0.7704520E-01     0.1740308E-01    -0.6000815E-02
  4       0.5839912E+00    -0.1807868E+00     0.2659480E-01    -0.7669512E-02
  5       0.1021941E+01    -0.2842678E+00     0.2921298E-01    -0.6042998E-02
  6       0.1496003E+01    -0.3587406E+00     0.3124226E-01    -0.3921565E-02
  7       0.1999373E+01    -0.4039884E+00     0.3303983E-01    -0.2163647E-02
  8       0.2529638E+01    -0.4239655E+00     0.3474081E-01    -0.5953177E-03
  9       0.3083344E+01    -0.4232069E+00     0.3608362E-01     0.5515825E-03
 10       0.3653433E+01    -0.4089971E+00     0.3684258E-01     0.1142329E-02
 11       0.4229228E+01    -0.3904818E+00     0.3681497E-01     0.1111808E-02


I= 3

  K       W(I,J)            V(I,J)            DW(I,J)           DV(I,J)
  1       0.0               0.0               0.0               0.0
  2      -0.7415771E-01    -0.6834286E-01    -0.8815017E-02    -0.7958513E-02
  3      -0.3768928E+00    -0.4091305E+00    -0.2786428E-01    -0.3210701E-01
  4      -0.9542273E+00    -0.1096154E+01    -0.4312693E-01    -0.5081352E-01
  5      -0.1634645E+01    -0.1890661E+01    -0.4194136E-01    -0.4759142E-01
  6      -0.2171334E+01    -0.2465169E+01    -0.2487231E-01    -0.2328279E-01
  7      -0.2331151E+01    -0.2544001E+01     0.5753405E-02     0.1486159E-01
  8      -0.1917689E+01    -0.1956378E+01     0.4791136E-01     0.6118270E-01
  9      -0.8257181E+00    -0.6669863E+00     0.9105462E-01     0.1031131E+00
 10       0.8585268E+00     0.1168359E+01     0.1214661E+00     0.1292541E+00
 11       0.2855342E+01     0.3263458E+01     0.1302493E+00     0.1361389E+00


I= 4

  K       W(I,J)            V(I,J)            DW(I,J)           DV(I,J)
  1       0.0               0.0               0.0               0.0
  2       0.5017383E-01     0.2488598E+00     0.6019752E-02     0.2334813E-01
  3       0.3397898E+00     0.1251101E+01     0.2845969E-01     0.8762511E-01
  4       0.9262699E+00     0.2709785E+01     0.4089152E-01     0.8272529E-01
  5       0.1313488E+01     0.3515694E+01     0.4193374E-02     0.1209554E-01
  6       0.9240110E+00     0.2383393E+01    -0.6620183E-01    -0.9232397E-01
  7      -0.5951878E+00     0.9211577E+00    -0.1073777E+00    -0.1433128E+00
  8      -0.2063768E+01    -0.1215673E+01    -0.6792390E-01    -0.1167410E+00
  9      -0.2215355E+01    -0.2093108E+01     0.5533580E-01     0.5103759E-02
 10      -0.2487896E+01    -0.1983831E+01     0.1872610E+00     0.1163710E+00
 11       0.3242643E+01     0.1165999E+01     0.2399881E+00     0.1555466E+00
```

B.3 The TILDYN Program Examples

B.3.1 Input Data Listing for the TILDYN Program

In this subsection, the sample problem data deck setup is illustrated. The flight condition is powered flight for the Boeing model and autorotation flight for the Bell model. The data for the computation is shown in Table 3 in detail.

```
BOEING ROTOR --POWERED FLT-- 9DOF, U-GUST,FREQ RES & EIGEN ANALYSIS     8/20/74  DATA0001
0                                                                        ITYPE    DATA0002
0                                                                        IFLT     DATA0003
 9                                                                       DOF      DATA0004
1                                                                        FREQRE   DATA0005
0                                                                        IFRMAG   DATA0006
1                                                                        IEIGEN   DATA0007
3                                                                                 DATA0008
1.1468 E-7-40.422   -0.7        4414.082                                          DATA0009
156.0      1800.0      18.8496     5.7         0.0065      55.224                 DATA0010
200.5      62.18       5.7         0.004       -0.05       0.0         0.01        0.031       DATA0011
1.0        0.0         0.0         0.0         0.0         0.0                   U GUST  DATA0012
1117.2     2851.9      18886.0     74927.0                                        DATA0013
217.68     696.97      2002.2      5988.4      25466.0     253710.0               DATA0014
 9                                                                                DATA0015
0.205      0.105       0.105       0.105       0.105       0.105       0.105       0.092       DATA0016
0.073                                                                             DATA0017
  0.0          0.96884E-02  0.26486E-01  0.56422E-01  0.97570E-01  0.14798E+00   DATA0018
  0.20573E+00  0.26889E+00  0.32718E+00  0.37453E+00                              DATA0019
  0.0         -0.19216E-09 -0.35539E-09 -0.47946E-09 -0.51490E-09 -0.44798E-09   DATA0020
 -0.30091E-09 -0.12576E-09 -0.61370E-11  0.24547E-10                              DATA0021
  0.0          0.45561E-03  0.11282E-02  0.17080E-02  0.21954E-02  0.25905E-02   DATA0022
  0.28939E-02  0.31061E-02  0.32185E-02  0.32613E-02                              DATA0023
  0.0         -0.75167E-11 -0.73673E-11 -0.40539E-11  0.78800E-12  0.54042E-11   DATA0024
  0.81805E-11  0.79227E-11  0.45167E-11 -0.68893E-12                              DATA0025
  0.0          0.12465E-04  0.47748E-04  0.85418E-04  0.12350E-03  0.16164E-03   DATA0026
  0.19980E-03  0.23795E-03  0.27138E-03  0.29790E-03                              DATA0027
  0.33113E-06  0.41245E-06  0.15763E-05  0.17759E-05  0.18101E-05  0.18159E-05   DATA0028
  0.18168E-05  0.18168E-05  0.18167E-05  0.18166E-05                              DATA0029
  0.0         -0.69305E-09 -0.10621E-08 -0.90808E-09 -0.27148E-09  0.46049E-09   DATA0030
  0.84255E-09  0.65858E-09  0.16850E-09 -0.52234E-10                              DATA0031
  0.0          0.15288E-01  0.36346E-01  0.66257E-01  0.10945E+00  0.15835E+00   DATA0032
  0.21341E+00  0.27306E+00  0.32790E+00  0.37243E+00                              DATA0033
  0.0         -0.24413E-10 -0.67135E-11  0.20905E-10  0.36406E-10  0.29522E-10   DATA0034
  0.50409E-11 -0.21349E-10 -0.26157E-10  0.96106E-12                              DATA0035
  0.0          0.71933E-03  0.12737E-02  0.17530E-02  0.21576E-02  0.24876E-02   DATA0036
```

BILITY OF THE ORIGINAL PAGE IS POOR

```
   0.27435E-02  0.29258E-02  0.30262E-02  0.30684E-02                            DATA0037
   0.0          0.26198E-12  0.54384E-12  0.15192E-11  0.18532E-11  0.19004E-11  DATA0038
   0.16760E-11  0.12374E-11  0.73507E-12  0.29310E-12                            DATA0039
   0.70398E-14  0.83981E-14  0.27607E-13  0.21644E-13  0.90527E-14 -0.45720E-14  DATA0040
  -0.16357E-13 -0.24832E-13 -0.29312E-13 -0.30956E-13                            DATA0041
   0.0          0.20401E-02  0.56893E-02  0.12371E-01  0.21795E-01  0.33683E-01  DATA0042
   0.47769E-01  0.63816E-01  0.79330E-01  0.92582E-01                            DATA0043
   0.0          0.51587E-06  0.95302E-06  0.12840E-05  0.13769E-05  0.11962E-05  DATA0044
   0.80224E-06  0.33481E-06  0.16006E-07 -0.65507E-07                            DATA0045
   0.0          0.96922E-04  0.24829E-03  0.38574E-03  0.50960E-03  0.62045E-03  DATA0046
   0.71923E-03  0.80735E-03  0.87878E-03  0.93691E-03                            DATA0047
   0.0          0.20161E-07  0.19697E-07  0.10769E-07 -0.22027E-08 -0.14514E-07  DATA0048
  -0.21872E-07 -0.21124E-07 -0.12014E-07  0.18404E-08                            DATA0049
   0.0         -0.46056E-03 -0.17642E-02 -0.31557E-02 -0.45620E-02 -0.59703E-02  DATA0050
  -0.73783E-02 -0.87854E-02 -0.10017E-01 -0.10994E-01                            DATA0051
  -0.12235E-04 -0.15239E-04 -0.58235E-04 -0.65598E-04 -0.66841E-04 -0.67026E-04  DATA0052
  -0.67023E-04 -0.66980E-04 -0.66931E-04 -0.66887E-04                            DATA0053
10                                                                               DATA0054
0.1       0.1       0.1       0.1       0.1       0.1       0.1       0.1        DATA0055
0.1       0.1                                                                    DATA0056
0.0165    0.00877   0.00091   0.00085   0.00098   0.00104   0.00104   0.00116    DATA0057
0.00123   0.00116   0.00104                                                      DATA0058
0.6545    0.5236    0.3927    0.2793    0.2007    0.1484    0.0873    0.0291     DATA0059
-0.0297   -0.0873   -0.1396                                                      DATA0060
0.0175                                                                           DATA0061
   0.0         -0.30837E-02 -0.40433E-02  0.17669E-01  0.72429E-01  0.15358E+00  DATA0062
   0.24691E+00  0.34107E+00  0.42791E+00  0.50157E+00  0.56107E+00               DATA0063
   0.0          0.33251E-01  0.19717E+00  0.54318E+00  0.10076E+01  0.15219E+01  DATA0064
   0.20560E+01  0.25932E+01  0.31241E+01  0.36432E+01  0.41507E+01               DATA0065
   0.0         -0.41524E-03  0.37868E-04  0.22488E-02  0.44661E-02  0.57715E-02  DATA0066
   0.61346E-02  0.59060E-02  0.51967E-02  0.42450E-02  0.34889E-02               DATA0067
   0.0          0.39618E-02  0.15860E-01  0.27209E-01  0.31795E-01  0.33890E-01  DATA0068
   0.34499E-01  0.34336E-01  0.33681E-01  0.32866E-01  0.32277E-01               DATA0069
   0.0          0.48522E-01  0.23571E+00  0.58387E+00  0.10218E+01  0.14959E+01  DATA0070
   0.19993E+01  0.25296E+01  0.30834E+01  0.36535E+01  0.42293E+01               DATA0071
   0.0         -0.15271E-01 -0.76997E-01 -0.18072E+00 -0.28422E+00 -0.35870E+00  DATA0072
```

```
-0.40395E+00 -0.42391E+00 -0.42317E+00 -0.40902E+00 -0.39059E+00                DATA0073
 0.0          0.58995E-02  0.17399E-01  0.26594E-01  0.29213E-01  0.31244E-01    DATA0074
 0.33043E-01  0.34745E-01  0.36086E-01  0.36842E-01  0.36814E-01                 DATA0075
 0.0         -0.18829E-02 -0.60045E-02 -0.76701E-02 -0.60446E-02 -0.39221E-02    DATA0076
-0.21631E-02 -0.59548E-03  0.54903E-03  0.11376E-02  0.11063E-02                 DATA0077
```

```
BELL ROTOR  AUTOROTATION FLIGHT  9-DOF  U-GUST FREQ ANALYSIS & EIGEN     8/18/74   DATA0001
1                                                                        ITYPE      DATA0002
1                                                                        IFLT       DATA0003
 9                                                                       DOF        DATA0004
1                                                                        FREQRE     DATA0005
0                                                                        IFRMAG     DATA0006
1                                                                        EIGEN      DATA0007
3                                                                                   DATA0008
1.1468 E-748.0      0.7        5040.0                                               DATA0009
150.0     1260.0    14.0       5.7        0.0065    51.3      -0.2618               DATA0010
200.5     62.18     9.7        0.004      -0.05     0.0       0.01      0.031       DATA0011
1.0       0.0       0.0        0.0        0.0       0.0                   U GUST    DATA0012
2380.1    4143.4    43503.0    236090.0                                             DATA0013
277.68    890.84    2730.0     12968.0    32494.0   260860.0                        DATA0014
 9                                                                                  DATA0015
0.205     0.105     0.105      0.105      0.105     0.105     0.105     0.092       DATA0016
0.073                                                                               DATA0017
  0.0            0.10964E-01  0.29950E-01  0.63781E-01  0.11024E+00  0.16712E+00    DATA0018
  0.23221E+00    0.30335E+00  0.36894E+00  0.42219E+00                              DATA0019
  0.0           -0.41183E-09 -0.75962E-09 -0.10207E-08 -0.10898E-08 -0.93963E-09    DATA0020
 -0.62027E-09   -0.24558E-09  0.53875E-11  0.65046E-10                              DATA0021
  0.0            0.51546E-03  0.12751E-02  0.19292E-02  0.24779E-02  0.29215E-02    DATA0022
  0.32608E-02    0.34988E-02  0.36209E-02  0.36670E-02                              DATA0023
  0.0           -0.16077E-10 -0.15623E-10 -0.83983E-11  0.20307E-11  0.11862E-10    DATA0024
  0.17630E-10    0.16807E-10  0.92860E-11 -0.18897E-11                              DATA0025
  0.0            0.17816E-04  0.68248E-04  0.12209E-03  0.17651E-03  0.23103E-03    DATA0026
  0.28557E-03    0.34010E-03  0.38788E-03  0.42578E-03                              DATA0027
  0.47329E-06    0.58952E-06  0.22530E-05  0.25383E-05  0.25872E-05  0.25954E-05    DATA0028
  0.25966E-05    0.25966E-05  0.25964E-05  0.25962E-05                              DATA0029
  0.0           -0.11659E-08 -0.17804E-08 -0.15087E-08 -0.42875E-09  0.79769E-09    DATA0030
  0.14197E-08    0.10853E-08  0.25164E-09 -0.11546E-09                              DATA0031
  0.0            0.17340E-01  0.41209E-01  0.77357E-01  0.12399E+00  0.17932E+00    DATA0032
  0.24156E+00    0.30895E+00  0.37086E+00  0.42111E+00                              DATA0033
  0.0           -0.40996E-10 -0.10784E-10  0.35777E-10  0.61402E-10  0.49017E-10    DATA0034
  0.72510E-11   -0.36880E-10 -0.44000E-10  0.21093E-11                              DATA0035
  0.0            0.81566E-03  0.14432E-02  0.19852E-02  0.24418E-02  0.28133E-02    DATA0036
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
  0.31005E-02  0.33041E-02  0.34153E-02  0.34619E-02                              DATA0037
  0.0          0.49672E-12  0.17935E-11  0.28992E-11  0.35643E-11  0.37026E-11    DATA0038
  0.33378E-11  0.25690E-11  0.16638E-11  0.89274E-12                              DATA0039
  0.13343E-13  0.15940E-13  0.52715E-13  0.42119E-13  0.19105E-13 -0.59927E-14    DATA0040
 -0.27980E-13 -0.44239E-13 -0.53361E-13 -0.57253E-13                              DATA0041
  0.0          0.31320E-02  0.86825E-02  0.18770E-01  0.32902E-01  0.50608E-01    DATA0042
  0.71446E-01  0.95027E-01  0.11770E+00  0.13703E+00                              DATA0043
  0.0          0.10676E-05  0.19673E-05  0.26404E-05  0.28160E-05  0.24250E-05    DATA0044
  0.15987E-05  0.63179E-06 -0.14536E-07 -0.16780E-06                              DATA0045
  0.0          0.14834E-03  0.37629E-03  0.58051E-03  0.76168E-03  0.92102E-03    DATA0046
  0.10604E-02  0.11828E-02  0.12821E-02  0.13664E-02                              DATA0047
  0.0          0.41643E-07  0.40362E-07  0.21580E-07 -0.54068E-08 -0.30756E-07    DATA0048
 -0.45549E-07 -0.43324E-07 -0.23867E-07  0.48793E-08                              DATA0049
  0.0         -0.53691E-03 -0.20566E-02 -0.36787E-02 -0.53178E-02 -0.69590E-02    DATA0050
 -0.85996E-02 -0.10239E-01 -0.11673E-01 -0.12811E-01                              DATA0051
 -0.14263E-04 -0.17765E-04 -0.67886E-04 -0.76463E-04 -0.77901E-04 -0.78104E-04    DATA0052
 -0.78083E-04 -0.78012E-04 -0.77934E-04 -0.77864E-04                              DATA0053
10                                                                                DATA0054
0.1       0.1       0.1       0.1       0.1       0.1       0.1       0.1         DATA0055
0.1       0.1                                                                     DATA0056
0.006993  0.005439  0.003108  0.001295  0.001336  0.001465  0.001165  0.001036    DATA0057
0.0009712 0.0008417 0.000712                                                      DATA0058
0.6807    0.5585    0.4363    0.3054    0.2094    0.1222    0.0698    0.0262      DATA0059
-0.0175   -0.0611   -0.1047                                                       DATA0060
0.0218                                                                            DATA0061
  0.0          0.43406E+00  0.87001E+00  0.13071E+01  0.17459E+01  0.21867E+01    DATA0062
  0.26295E+01  0.30737E+01  0.35166E+01  0.39633E+01  0.44072E+01                 DATA0063
  0.0         -0.42327E-03 -0.16251E-02 -0.42110E-02 -0.91843E-02 -0.16809E-01    DATA0064
 -0.26659E-01 -0.37990E-01 -0.49970E-01 -0.61872E-01 -0.73142E-01                 DATA0065
  0.28838E-01  0.29028E-01  0.29095E-01  0.29179E-01  0.29315E-01  0.29452E-01    DATA0066
  0.29572E-01  0.29644E-01  0.29661E-01  0.29630E-01  0.29554E-01                 DATA0067
  0.0         -0.53766E-04 -0.16352E-03 -0.23308E-03 -0.41699E-03 -0.58510E-03    DATA0068
 -0.71476E-03 -0.78626E-03 -0.80295E-03 -0.77788E-03 -0.72129E-03                 DATA0069
  0.0          0.13814E+00  0.27332E+00  0.37629E+00  0.40968E+00  0.35495E+00    DATA0070
  0.21551E+00  0.12193E-01 -0.23014E+00 -0.49140E+00 -0.75705E+00                 DATA0071
  0.0          0.63863E-01  0.22856E+00  0.50534E+00  0.93759E+00  0.15260E+01    DATA0072
```

```
  0.22382E+01  0.30330E+01  0.38752E+01  0.47389E+01  0.56075E+01                  DATA0073
  0.90778E-02  0.93612E-02  0.87107E-02  0.52225E-02 -0.39969E-03 -0.64671E-02     DATA0074
 -0.11715E-01 -0.15082E-01 -0.16982E-01 -0.17683E-01 -0.17663E-01                  DATA0075
  0.0          0.81963E-02  0.13524E-01  0.22872E-01  0.34064E-01  0.43729E-01     DATA0076
  0.50693E-01  0.54895E-01  0.57104E-01  0.57871E-01  0.57870E-01                  DATA0077
7988.699   37453.0    136740.0   376680.0                                          DATA0078
  0.0          0.94671E-01  0.31163E+00  0.59558E+00  0.97291E+00  0.14786E+01     DATA0079
  0.21260E+01  0.28973E+01  0.37551E+01  0.46611E+01  0.55816E+01                  DATA0080
  0.0          0.22730E+00  0.44648E+00  0.61473E+00  0.67209E+00  0.58087E+00     DATA0081
  0.33649E+00 -0.31189E-01 -0.47888E+00 -0.96805E+00 -0.14689E+01                  DATA0082
  0.0          0.12196E-01  0.16565E-01  0.21075E-01  0.28798E-01  0.38043E-01     DATA0083
  0.47582E-01  0.54627E-01  0.59182E-01  0.61171E-01  0.61353E-01                  DATA0084
  0.15072E-01  0.15215E-01  0.14058E-01  0.86605E-02 -0.47470E-03 -0.11016E-01     DATA0085
 -0.20858E-01 -0.27575E-01 -0.31613E-01 -0.33246E-01 -0.33377E-01                  DATA0086
  0.0          0.20944E+00  0.67206E+00  0.12009E+01  0.16929E+01  0.19969E+01     DATA0087
  0.19371E+01  0.14438E+01  0.56983E+00 -0.55484E+00 -0.17710E+01                  DATA0088
  0.0         -0.66471E+00 -0.12762E+01 -0.17966E+01 -0.21002E+01 -0.20027E+01     DATA0089
 -0.13642E+01 -0.18150E+00  0.14329E+01  0.33066E+01  0.52677E+01                  DATA0090
  0.0          0.26660E-01  0.34476E-01  0.35341E-01  0.29105E-01  0.10407E-01     DATA0091
 -0.18427E-01 -0.46323E-01 -0.68326E-01 -0.79558E-01 -0.81443E-01                  DATA0092
 -0.45442E-01 -0.42335E-01 -0.38601E-01 -0.29705E-01 -0.92970E-02  0.23359E-01     DATA0093
  0.61542E-01  0.94766E-01  0.11840E+00  0.12938E+00  0.13108E+00                  DATA0094
```

B.3.2 The Output Printout for the TILDYN Program

The output printout is illustrated for autorotation flight of the Bell model in this subsection.

```
****************************************************************************************************
          BELL ROTOR  AUTOROTATION FLIGHT  9-DOF  U-GUST FREQ ANALYSIS & EIGEN     8/18/74
****************************************************************************************************
     ITYPE= 1    IFLT= 1    IDOF= 9    IRES= 1    IEIGEN= 1    IFRMAG= 0


NO OF BLADES         ROH             CHORD             IB              HMAST              LOCK NO

     3          1.1468001E-07   1.4000000E+01   1.2600000E+03   5.1300003E+01   3.6769247E+00


  OMEGA               R              VEL               CL              CD                 RAMDA

4.8000000E+01   1.5000000E+02   5.0400000E+03   5.6999998E+00   6.4999983E-03   6.9999999E-01


  COLLECTIVE PITCH   DEL3

2.1800000E-02   -2.6179999E-01


  WING L             WING CHOD        WING CL           WING CD          WING CMD            WING CMA

2.0050000E+02   6.2179993E+01   5.6999998E+00   4.0000007E-03   -5.0000001E-02   0.0


  DISTANCE AC EA     WING ALPHAH

1.0000002E-02   3.0999999E-02
-----------------------------------
EIGENVALUES ( NATURAL FREQUENCIES )
-----------------------------------

 --( RAD/SEC )**2--

  **BLADE COLLECTIVE**
      7988.699      37453.000

  **BLADE CYCLIC**
      2389.100       4143.398
```

```
  **WING**
       277.680          890.840          2730.000

 -- RAD/SEC/OMEGA --

  **BLADE COLLECTIVE/OMEGA**
         1.862            4.032

  **BLADE CYCLIC/OMEGA**
         1.016            1.341

  **WING/OMEGA**
         0.347            0.622            1.089


EXCITING FORCE COMPONENTS


   U GUST                 V GUST               W GUST               THETA 0         THETA 1C             THETA 1S

 1.0000000E+00   0.0               0.0               0.0               0.0               0.0


*****  BLADE MODE SHAPES  *****

 ---  COLLECTIVE MODES  ---

I=1
   J          XX(J)                 V(I,J)                DV(I,J)               W(I ,J)               DW(I,J)
   1          0.0                   0.0                   0.1507200E-01         0.0                   0.0
   2          0.1000000E+00         0.2273000E+00         0.1521500E-01         0.9467101E-01         0.1219600E-01
   3          0.2000000E+00         0.4464800E+00         0.1405800E-01         0.3116300E+00         0.1656500E-01
   4          0.3000001E+00         0.6147300E+00         0.8660499E-02         0.5955800E+00         0.2107500E-01
   5          0.4000001E+00         0.6720900E+00        -0.4747000E-03         0.9729100E+00         0.2879800E-01
   6          0.5000001E+00         0.5808700E+00        -0.1101600E-01         0.1478600E+01         0.3804300E-01
   7          0.6000001E+00         0.3364900E+00        -0.2085800E-01         0.2126000E+01         0.4758200E-01
   8          0.7000002E+00        -0.3118900E-01        -0.2757500E-01         0.2897300E+01         0.5462700E-01
   9          0.8000002E+00        -0.4788800E+00        -0.3161300E-01         0.3755100E+01         0.5918200E-01
  10          0.9000002E+00        -0.9680500E+00        -0.3324600E-01         0.4661100E+01         0.6117100E-01
  11          0.1000000E+01        -0.1468900E+01        -0.3337700E-01         0.5581600E+01         0.6135300E-01

I=2
   J          XX(J)                 V(I,J)                DV(I,J)               W(I ,J)               DW(I,J)
   1          0.0                   0.0                  -0.4544200E-01         0.0                   0.0
   2          0.1000000E+00        -0.6647100E+00        -0.4233500E-01         0.2094400E+00         0.2666300E-01
   3          0.2000000E+00        -0.1276200E+01        -0.3860100E-01         0.6720600E+00         0.3447600E-01
   4          0.3000001E+00        -0.1796600E+01        -0.2970500E-01         0.1200900E+01         0.3534100E-01
   5          0.4000001E+00        -0.2100200E+01        -0.9296998E-02         0.1692900E+01         0.2910500E-01
   6          0.5000001E+00        -0.2002700E+01         0.2335900E-01         0.1996900E+01         0.1040700E-01
   7          0.6000001E+00        -0.1364200E+01         0.6154200E-01         0.1937100E+01        -0.1842700E-01
   8          0.7000002E+00        -0.1815000E+00         0.9476602E-01         0.1443800E+01        -0.4632300E-01
   9          0.8000002E+00         0.1432900E+01         0.1184000E+00         0.5698300E+00        -0.6832600E-01
  10          0.9000002E+00         0.3306600E+01         0.1293800E+00        -0.5548400E+00        -0.7955801E-01
  11          0.1000000E+01         0.5267700E+01         0.1310800E+00        -0.1771000E+01        -0.8144301E-01
```

```
---  CYCLIC MODES  ---


I=1
   J        XX(J)              V(I,J)             DV(I,J)             W(I ,J)            DW(I,J)
   1        0.0                0.0                0.2930400E-01       0.0                0.2883800E-01
   2        0.1000000E+00     -0.4232700E-03     -0.5376599E-04       0.4340600E+00      0.2902800E-01
   3        0.2000000E+00     -0.1625100E-02     -0.1035200E-03       0.8700100E+00      0.2909500E-01
   4        0.3000001E+00     -0.4211001E-02     -0.2330800E-03       0.1307100E+01      0.2917900E-01
   5        0.4000001E+00     -0.9184301E-02     -0.4169899E-03       0.1745900E+01      0.2931500E-01
   6        0.5000001E+00     -0.1680900E-01     -0.5850999E-03       0.2186700E+01      0.2945200E-01
   7        0.6000001E+00     -0.2665900E-01     -0.7147600E-03       0.2629500E+01      0.2957200E-01
   8        0.7000002E+00     -0.3799000E-01     -0.7862600E-03       0.3073700E+01      0.2964400E-01
   9        0.8000002E+00     -0.4997000E-01     -0.8029500E-03       0.3518600E+01      0.2966100E-01
  10        0.9000002E+00     -0.6187200E-01     -0.7778800E-03       0.3963300E+01      0.2963000E-01
  11        0.1000000E+01     -0.7314199E-01     -0.7212900E-03       0.4407200E+01      0.2955400E-01


I=2
   J        XX(J)              V(I,J)             DV(I,J)             W(I ,J)            DW(I,J)
   1        0.0                0.0                0.1507200E-01       0.0                0.9077799E-02
   2        0.1000000E+00      0.6386298E-01      0.8196302E-02       0.1381400E+00      0.9361200E-02
   3        0.2000000E+00      0.2285600E+00      0.1352400E-01       0.2733200E+00      0.8710701E-02
   4        0.3000001E+00      0.5053400E+00      0.2287200E-01       0.3762900E+00      0.5222499E-02
   5        0.4000001E+00      0.9375900E+00      0.3406400E-01       0.4096800E+00     -0.3996899E-03
   6        0.5000001E+00      0.1526000E+01      0.4372900E-01       0.3549500E+00     -0.6467100E-02
   7        0.6000001E+00      0.2238200E+01      0.5069300E-01       0.2155100E+00     -0.1171500E-01
   8        0.7000002E+00      0.3033000E+01      0.5489500E-01       0.1219300E-01     -0.1508200E-01
   9        0.8000002E+00      0.3875200E+01      0.5710400E-01      -0.2301400E+00     -0.1698200E-01
  10        0.9000002E+00      0.4738900E+01      0.5787100E-01      -0.4914000E+00     -0.1768300E-01
  11        0.1000000E+01      0.5607500E+01      0.5787000E-01      -0.7570500E+00     -0.1766300E-01


   J        XX(J)              THETN(J)           AMASS(J)
   1        0.0                0.6807000E+00      0.6992999E-02
   2        0.1000000E+00      0.5585000E+00      0.5438998E-02
   3        0.2000000E+00      0.4363000E+00      0.3108000E-02
   4        0.3000001E+00      0.3054000E+00      0.1295000E-02
   5        0.4000001E+00      0.2094000E+00      0.1336000E-02
   6        0.5000001E+00      0.1222000E+00      0.1465000E-02
   7        0.6000001E+00      0.6980002E-01      0.1165000E-02
   8        0.7000002E+00      0.2620000E-01      0.1036000E-02
   9        0.8000002E+00     -0.1750000E-01      0.9711999E-03
  10        0.9000002E+00     -0.6110000E-01      0.8417000E-03
  11        0.1000000E+01     -0.1047000E+00      0.7120001E-03
```

```
*****  WING MODE SHAPES  *****


II=1
  J   WX(J)            G(II,J)          DG(II,J)         Z(II,J)          DZ(II,J)         WPHI(II,J)       DWPHI(II,J)
  1   0.0              0.0              0.0              0.0              0.0              0.0              0.4732900E-06
  2   0.2050000E+00    0.1096400E-01    0.5154600E-03   -0.4118299E-09   -0.1607700E-10    0.1781600E-04    0.5895200E-06
  3   0.3100000E+00    0.2995000E-01    0.1275100E-02   -0.7596199E-09   -0.1562300E-10    0.6824800E-04    0.2253000E-05
  4   0.4150000E+00    0.6378102E-01    0.1929200E-02   -0.1020700E-08   -0.8398300E-11    0.1220900E-03    0.2538300E-05
  5   0.5200000E+00    0.1102400E+00    0.2477900E-02   -0.1089800E-08    0.2030700E-11    0.1765100E-03    0.2587200E-05
  6   0.6250001E+00    0.1671200E+00    0.2921500E-02   -0.9396299E-09    0.1186200E-10    0.2310300E-03    0.2595400E-05
  7   0.7300001E+00    0.2322100E+00    0.3260800E-02   -0.6202701E-09    0.1762999E-10    0.2855700E-03    0.2596600E-05
  8   0.8350001E+00    0.3033500E+00    0.3496800E-02   -0.2455800E-09    0.1680700E-10    0.3400999E-03    0.2596600E-05
  9   0.9270001E+00    0.3689400E+00    0.3620500E-02    0.5387500E-11    0.9286000E-11    0.3878800E-03    0.2596400E-05
 10   0.1000000E+01    0.4221900E+00    0.3667000E-02    0.6504600E-10   -0.1889700E-11    0.4257800E-03    0.2596200E-05


II=2
  J   WX(J)            G(II,J)          DG(II,J)         Z(II,J)          DZ(II,J)         WPHI(II,J)       DWPHI(II,J)
  1   0.0              0.0              0.0              0.0              0.0              0.0              0.1334300E-13
  2   0.2050000E+00   -0.1165900E-08   -0.4099600E-10    0.1734000E-01    0.8156600E-03    0.4967200E-12    0.1594000E-13
  3   0.3100000E+00   -0.1780400E-08   -0.1078400E-10    0.4120900E-01    0.1443200E-02    0.1793500E-11    0.5271500E-13
  4   0.4150000E+00   -0.1508700E-08    0.3577701E-10    0.7735699E-01    0.1985200E-02    0.2899200E-11    0.4211900E-13
  5   0.5200000E+00   -0.4287499E-09    0.6140199E-10    0.1239900E+00    0.2441800E-02    0.3564300E-11    0.1910500E-13
  6   0.6250001E+00    0.7976899E-09    0.4901700E-10    0.1793200E+00    0.2813300E-02    0.3702600E-11   -0.5992701E-14
  7   0.7300001E+00    0.1419700E-08    0.7251000E-11    0.2415600E+00    0.3100500E-02    0.3337800E-11   -0.2798000E-13
  8   0.8350001E+00    0.1085300E-08   -0.3688000E-10    0.3089500E+00    0.3304100E-02    0.2569000E-11   -0.4423900E-13
  9   0.9270001E+00    0.2516400E-09   -0.4400000E-10    0.3708600E+00    0.3415300E-02    0.1663800E-11   -0.5336100E-13
 10   0.1000000E+01   -0.1154600E-09    0.2109300E-11    0.4211100E+00    0.3461900E-02    0.8527400E-12   -0.5725298E-13


II=3
  J   WX(J)            G(II,J)          DG(II,J)         Z(II,J)          DZ(II,J)         WPHI(II,J)       DWPHI(II,J)
  1   0.0              0.0              0.0              0.0              0.0              0.0             -0.1426300E-04
  2   0.2050000E+00    0.3132000E-02    0.1483400E-03    0.1067600E-05    0.4164300E-07   -0.5369100E-03   -0.1776501E-04
  3   0.3100000E+00    0.8682501E-02    0.3762899E-03    0.1967300E-05    0.4036200E-07   -0.2056600E-02   -0.6788599E-04
  4   0.4150000E+00    0.1877000E-01    0.5805099E-03    0.2640400E-05    0.2158000E-07   -0.3678700E-02   -0.7646299E-04
  5   0.5200000E+00    0.3290200E-01    0.7616801E-03    0.2816000E-05   -0.5406800E-08   -0.5317800E-02   -0.7790100E-04
  6   0.6250001E+00    0.5060800E-01    0.9210201E-03    0.2425000E-05   -0.3075600E-07   -0.6958999E-02   -0.7810400E-04
  7   0.7300001E+00    0.7144600E-01    0.1060400E-02    0.1598700E-05   -0.4554900E-07   -0.8599602E-02   -0.7808300E-04
  8   0.8350001E+00    0.9502703E-01    0.1182800E-02    0.6317900E-06   -0.4332400E-07   -0.1023900E-01   -0.7801200E-04
  9   0.9270001E+00    0.1177000E+00    0.1282100E-02   -0.1453600E-07   -0.2388700E-07   -0.1167300E-01   -0.7793400E-04
 10   0.1000000E+01    0.1370300E+00    0.1366400E-02   -0.1678000E-06    0.4879301E-08   -0.1281100E-01   -0.7786400E-04
```

```
-----------------------------------------------------------------------------

              EQUATIONS  OF  MOTION  :  A*X"+B*X'+C*X=D*E

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


                  A MATRIX=

 0.10000E+01  0.0          0.0          0.0          0.0          0.0          0.20516E-10  0.13282E+00 -0.52925E-07  0.23858E+00
 0.0          0.10000E+01  0.0          0.0          0.0          0.0         -0.14719E-01 -0.98082E-05  0.44288E+00  0.0
 0.0          0.0          0.10000E+01  0.0          0.0          0.0         -0.24523E-04  0.11968E+00  0.28897E-04  0.0
 0.0          0.0          0.0          0.10000E+01  0.0          0.0          0.13386E-10  0.86664E-01 -0.34533E-07 -0.79480E+00
 0.0          0.0          0.0          0.0          0.10000E+01  0.0         -0.25539E-03 -0.56142E-01  0.76842E-02  0.0
 0.0          0.0          0.0          0.0          0.0          0.10000E+01 -0.14037E+00  0.20765E-02  0.16444E+00  0.0
 0.61547E-10 -0.22079E-01 -0.36785E-04  0.40159E-10 -0.38309E-03 -0.21055E+00  0.10000E+01  0.0          0.0          0.0
 0.39846E+00 -0.14712E-04  0.17952E+00  0.25999E+00 -0.84213E-01  0.31148E-02  0.0          0.10000E+01  0.0          0.0
-0.15877E-06  0.66432E+00  0.43345E-04 -0.10360E-06  0.11526E-01  0.24666E+00  0.0          0.0          0.10000E+01  0.0
 0.71574E+00  0.0          0.0         -0.23844E+01  0.0          0.0          0.0          0.0          0.0          0.37800E+04

                  B MATRIX=

 0.54017E+00  0.0          0.0         -0.26876E+00  0.0          0.0          0.69574E-11  0.45043E-01 -0.17948E-07 -0.13242E+02
 0.0          0.37042E+00  0.20000E+01  0.0         -0.41167E+00  0.0         -0.53078E-02  0.25859E+00  0.15970E+00  0.0
 0.0         -0.20000E+01  0.37042E+00  0.0          0.0         -0.41167E+00  0.77536E-01  0.43156E-01 -0.94210E+00  0.0
-0.19099E+00  0.0          0.0          0.43642E+00  0.0          0.0          0.17233E-11  0.11157E-01 -0.44457E-08 -0.23007E+00
 0.0         -0.34603E+00  0.0          0.0          0.39483E+00  0.20000E+01  0.49567E-02 -0.14325E-01 -0.14914E+00  0.0
 0.0          0.0         -0.34603E+00  0.0         -0.20000E+01  0.39483E+00 -0.45688E-01 -0.40302E-01  0.38752E-01  0.0
 0.22544E-10 -0.79814E-02  0.25224E-01 -0.21837E-12  0.88643E-02 -0.66846E-01  0.32997E-01  0.67399E-02 -0.18662E-01 -0.71542E-09
 0.14595E+00 -0.33128E+00  0.64894E-01 -0.14137E-02 -0.32659E-01 -0.72073E-01  0.85543E-02  0.28552E-01  0.22520E-02 -0.46317E+01
-0.58157E-07  0.24015E+00  0.12474E+01  0.56333E-09 -0.26671E+00  0.10046E+00 -0.18944E-01  0.21209E-02  0.12854E+00  0.18456E-05
-0.33172E+02  0.0          0.0         -0.33280E+01  0.0          0.0         -0.68859E-09 -0.44580E+01  0.17764E-05  0.11459E+04

                  C MATRIX=

 0.34673E+01  0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
 0.0         -0.14993E+00  0.37042E+00  0.0         -0.57593E-01 -0.41167E+00  0.21493E-10 -0.39374E-01 -0.55495E-07  0.0
 0.0         -0.37042E+00 -0.14993E+00  0.0          0.41167E+00 -0.57593E-01 -0.48426E-02 -0.96987E-11  0.14571E+00  0.0
 0.0          0.0          0.0          0.16256E+02  0.0          0.0          0.0          0.0          0.0          0.0
 0.0          0.17862E+00 -0.34603E+00  0.0          0.85458E+00  0.39483E+00 -0.20644E-10  0.37820E-01  0.53304E-07  0.0
 0.0          0.34603E+00  0.17862E+00  0.0         -0.39483E+00  0.85458E+00  0.46514E-02  0.93158E-11 -0.13995E+00  0.0
 0.0         -0.65434E-01  0.36719E-01  0.0          0.67322E-01  0.22616E-01  0.11688E+00  0.85114E-03  0.10968E+00  0.0
 0.0         -0.85611E-01 -0.43438E-02  0.0          0.62958E-01 -0.36532E-01 -0.82583E-03  0.38097E+00  0.24848E-01  0.0
 0.0         -0.37486E-01  0.33214E+00  0.0         -0.11480E+00 -0.22815E+00  0.14950E-02 -0.25609E-01  0.11399E+01  0.0
 0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
```

```
                        D MATRIX=


 0.0           0.0          -0.11231E+02  0.24135E+02  0.0           0.0
 0.0           0.11374E+02   0.0          0.0          0.23678E+02   0.0
-0.11374E+02   0.0           0.0          0.0          0.0           0.23678E+02
 0.0           0.0          -0.27819E+01  0.37430E+01  0.0           0.0
 0.0          -0.10925E+02   0.0          0.0         -0.23117E+02   0.0
 0.10925E+02   0.0           0.0          0.0          0.0          -0.23117E+02
-0.10349E+02  -0.24586E+00   0.38116E+00  0.14824E-08 -0.51082E+00  -0.57982E+01
-0.22047E+01   0.16228E+01  -0.45530E+01  0.95973E+01  0.25721E+01   0.41534E+01
 0.29504E+01   0.73975E+01   0.37122E+00 -0.38242E-05  0.15370E+02  -0.11503E+02
 0.0           0.0           0.15879E+04 -0.23930E+04  0.0           0.0
```

```
                    FREQUENCY  RESPONSE

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



      --FREQUENCY/OMEGA--

            Q10         Q1C         Q1S         Q20         Q2C         Q2S         WING 1      WING 2      WING 3

            D(NU R)/DT

     --  0.01--
           0.188E-06   0.772E+00   0.304E+00   0.549E-07   0.476E-01   0.243E-01   0.162E+00   0.518E-03   0.527E-02
           0.959E-05

     --  0.02--
           0.473E-06   0.771E+00   0.304E+00   0.114E-06   0.486E-01   0.480E-01   0.162E+00   0.531E-03   0.584E-02
           0.196E-04

     --  0.03--
           0.921E-06   0.769E+00   0.305E+00   0.180E-06   0.505E-01   0.719E-01   0.164E+00   0.553E-03   0.667E-02
           0.306E-04

     --  0.04--
           0.159E-05   0.766E+00   0.307E+00   0.259E-06   0.534E-01   0.959E-01   0.165E+00   0.584E-03   0.767E-02
           0.429E-04

     --  0.05--
           0.252E-05   0.762E+00   0.308E+00   0.355E-06   0.576E-01   0.120E+00   0.167E+00   0.623E-03   0.878E-02
           0.570E-04

     --  0.06--
           0.379E-05   0.758E+00   0.311E+00   0.473E-06   0.633E-01   0.145E+00   0.170E+00   0.670E-03   0.994E-02
           0.732E-04
```

```
--  0.07--
      0.548E-05   0.753E+00   0.314E+00   0.619E-06   0.708E-01   0.170E+00   0.173E+00   0.726E-03   0.111E-01
      0.920E-04

--  0.08--
      0.766E-05   0.746E+00   0.317E+00   0.798E-06   0.802E-01   0.195E+00   0.177E+00   0.790E-03   0.123E-01
      0.114E-03

--  0.09--
      0.104E-04   0.739E+00   0.321E+00   0.102E-05   0.918E-01   0.220E+00   0.181E+00   0.862E-03   0.135E-01
      0.139E-03

--  0.10--
      0.140E-04   0.731E+00   0.326E+00   0.129E-05   0.106E+00   0.246E+00   0.186E+00   0.944E-03   0.147E-01
      0.167E-03

--  0.12--
      0.237E-04   0.710E+00   0.338E+00   0.203E-05   0.140E+00   0.300E+00   0.198E+00   0.114E-02   0.170E-01
      0.237E-03

--  0.14--
      0.384E-04   0.685E+00   0.354E+00   0.312E-05   0.186E+00   0.356E+00   0.213E+00   0.138E-02   0.192E-01
      0.329E-03

--  0.16--
      0.602E-04   0.652E+00   0.373E+00   0.474E-05   0.243E+00   0.414E+00   0.231E+00   0.169E-02   0.212E-01
      0.449E-03
```

```
--  0.18--
     0.921E-04   0.610E+00   0.396E+00   0.713E-05   0.312E+00   0.473E+00   0.253E+00   0.208E-02   0.229E-01
     0.608E-03

--  0.20--
     0.139E-03   0.558E+00   0.420E+00   0.107E-04   0.391E+00   0.530E+00   0.278E+00   0.259E-02   0.242E-01
     0.820E-03

--  0.22--
     0.206E-03   0.495E+00   0.444E+00   0.158E-04   0.476E+00   0.579E+00   0.309E+00   0.325E-02   0.253E-01
     0.110E-02

--  0.24--
     0.302E-03   0.425E+00   0.462E+00   0.232E-04   0.562E+00   0.617E+00   0.348E+00   0.407E-02   0.259E-01
     0.146E-02

--  0.26--
     0.437E-03   0.355E+00   0.475E+00   0.338E-04   0.644E+00   0.642E+00   0.401E+00   0.512E-02   0.263E-01
     0.194E-02

--  0.28--
     0.633E-03   0.292E+00   0.484E+00   0.494E-04   0.724E+00   0.662E+00   0.484E+00   0.651E-02   0.264E-01
     0.258E-02

--  0.30--
     0.944E-03   0.239E+00   0.500E+00   0.745E-04   0.822E+00   0.694E+00   0.635E+00   0.861E-02   0.242E-01
     0.355E-02

--  0.32--
     0.154E-02   0.183E+00   0.540E+00   0.123E-03   0.987E+00   0.784E+00   0.967E+00   0.125E-01   0.242E-01
     0.536E-02
```

```
--  0.34--
      0.268E-02   0.209E+00   0.569E+00   0.216E-03   0.120E+01   0.932E+00   0.169E+01   0.197E-01   0.105E-01
      0.869E-02

--  0.36--
      0.185E-02   0.375E+00   0.209E+00   0.151E-03   0.524E+00   0.438E+00   0.130E+01   0.123E-01   0.229E-01
      0.560E-02

--  0.38--
      0.777E-03   0.360E+00   0.140E+00   0.643E-04   0.161E+00   0.158E+00   0.703E+00   0.471E-02   0.306E-01
      0.220E-02

--  0.40--
      0.310E-03   0.346E+00   0.169E+00   0.260E-04   0.185E+00   0.116E+00   0.457E+00   0.172E-02   0.332E-01
      0.825E-03

--  0.45--
      0.795E-03   0.331E+00   0.190E+00   0.684E-04   0.261E+00   0.116E+00   0.234E+00   0.359E-02   0.377E-01
      0.182E-02

--  0.50--
      0.210E-02   0.323E+00   0.190E+00   0.185E-03   0.273E+00   0.107E+00   0.153E+00   0.785E-02   0.430E-01
      0.417E-02

--  0.55--
      0.569E-02   0.318E+00   0.191E+00   0.511E-03   0.269E+00   0.103E+00   0.113E+00   0.179E-01   0.524E-01
      0.990E-02

--  0.60--
      0.344E-01   0.237E+00   0.147E+00   0.314E-02   0.211E+00   0.197E+00   0.918E-01   0.920E-01   0.847E-01
      0.528E-01
```

```
--  0.65--
      0.114E-01   0.268E+00   0.138E+00   0.105E-02   0.242E+00   0.781E-01   0.673E-01   0.262E-01   0.252E-01
      0.156E-01

--  0.70--
      0.707E-02   0.274E+00   0.152E+00   0.658E-03   0.239E+00   0.694E-01   0.566E-01   0.140E-01   0.390E-01
      0.865E-02

--  0.75--
      0.586E-02   0.273E+00   0.155E+00   0.548E-03   0.234E+00   0.687E-01   0.483E-01   0.101E-01   0.466E-01
      0.645E-02

--  0.80--
      0.548E-02   0.271E+00   0.158E+00   0.512E-03   0.229E+00   0.704E-01   0.418E-01   0.825E-02   0.536E-01
      0.545E-02

--  0.85--
      0.550E-02   0.270E+00   0.161E+00   0.513E-03   0.226E+00   0.736E-01   0.367E-01   0.724E-02   0.613E-01
      0.496E-02

--  0.90--
      0.580E-02   0.271E+00   0.165E+00   0.537E-03   0.225E+00   0.781E-01   0.326E-01   0.670E-02   0.706E-01
      0.477E-02

--  0.95--
      0.639E-02   0.275E+00   0.170E+00   0.585E-03   0.226E+00   0.840E-01   0.293E-01   0.649E-02   0.826E-01
      0.480E-02
```

```
--  1.00--
      0.733E-02   0.282E+00   0.178E+00   0.663E-03   0.231E+00   0.915E-01   0.266E-01   0.656E-02   0.989E-01
      0.505E-02

--  1.10--
      0.113E-01   0.324E+00   0.203E+00   0.990E-03   0.260E+00   0.116E+00   0.230E-01   0.785E-02   0.160E+00
      0.662E-02

--  1.20--
      0.266E-01   0.510E+00   0.273E+00   0.222E-02   0.383E+00   0.187E+00   0.221E-01   0.143E-01   0.385E+00
      0.133E-01

--  1.30--
      0.360E-01   0.500E+00   0.475E-01   0.286E-02   0.225E+00   0.110E+00   0.785E-02   0.150E-01   0.529E+00
      0.157E-01

--  1.40--
      0.131E-01   0.243E+00   0.122E+00   0.987E-03   0.238E-01   0.248E-01   0.981E-02   0.417E-02   0.195E+00
      0.500E-02

--  1.50--
      0.783E-02   0.241E+00   0.178E+00   0.557E-03   0.722E-01   0.591E-01   0.947E-02   0.189E-02   0.119E+00
      0.263E-02

--  1.60--
      0.546E-02   0.271E+00   0.229E+00   0.372E-03   0.937E-01   0.808E-01   0.866E-02   0.997E-03   0.875E-01
      0.162E-02

--  1.70--
      0.411E-02   0.318E+00   0.291E+00   0.276E-03   0.104E+00   0.984E-01   0.773E-02   0.585E-03   0.692E-01
      0.109E-02
```

```
--  1.80--
     0.383E-02   0.381E+00   0.369E+00   0.266E-03   0.103E+00   0.110E+00   0.661E-02   0.463E-03   0.540E-01
     0.907E-03

--  1.90--
     0.429E-02   0.430E+00   0.438E+00   0.325E-03   0.742E-01   0.100E+00   0.525E-02   0.504E-03   0.342E-01
     0.913E-03

--  2.00--
     0.333E-02   0.400E+00   0.425E+00   0.289E-03   0.921E-02   0.551E-01   0.457E-02   0.422E-03   0.792E-02
     0.640E-03

--  2.10--
     0.116E-02   0.310E+00   0.341E+00   0.119E-03   0.670E-01   0.453E-01   0.511E-02   0.165E-03   0.148E-01
     0.202E-03

--  2.20--
     0.128E-02   0.225E+00   0.254E+00   0.156E-03   0.141E+00   0.115E+00   0.591E-02   0.204E-03   0.314E-01
     0.201E-03

--  2.30--
     0.340E-02   0.156E+00   0.175E+00   0.499E-03   0.226E+00   0.211E+00   0.675E-02   0.602E-03   0.466E-01
     0.485E-03

--  2.40--
     0.529E-02   0.845E-01   0.821E-01   0.930E-03   0.304E+00   0.310E+00   0.689E-02   0.102E-02   0.571E-01
     0.684E-03
```

```
--  2.50--
    0.501E-02   0.408E-01   0.389E-01   0.105E-02   0.273E+00   0.299E+00   0.496E-02   0.105E-02   0.468E-01
    0.587E-03

--  2.60--
    0.364E-02   0.463E-01   0.644E-01   0.913E-03   0.194E+00   0.225E+00   0.308E-02   0.811E-03   0.305E-01
    0.385E-03

--  2.70--
    0.262E-02   0.470E-01   0.694E-01   0.781E-03   0.139E+00   0.170E+00   0.219E-02   0.617E-03   0.200E-01
    0.249E-03

--  2.80--
    0.196E-02   0.436E-01   0.667E-01   0.694E-03   0.105E+00   0.134E+00   0.182E-02   0.485E-03   0.138E-01
    0.167E-03

--  2.90--
    0.153E-02   0.392E-01   0.619E-01   0.640E-03   0.820E-01   0.110E+00   0.164E-02   0.393E-03   0.992E-02
    0.115E-03

--  3.00--
    0.122E-02   0.350E-01   0.568E-01   0.608E-03   0.662E-01   0.923E-01   0.154E-02   0.326E-03   0.737E-02
    0.796E-04

--  3.50--
    0.539E-03   0.201E-01   0.375E-01   0.674E-03   0.298E-01   0.495E-01   0.123E-02   0.156E-03   0.219E-02
    0.758E-05
```

```
--  4.00--
     0.420E-03   0.125E-01   0.265E-01   0.135E-02   0.169E-01   0.324E-01   0.993E-03   0.999E-04   0.798E-03
     0.646E-04

--  4.50--
     0.227E-03   0.840E-02   0.199E-01   0.492E-03   0.107E-01   0.234E-01   0.807E-03   0.745E-04   0.319F-03
     0.331E-04

--  5.00--
     0.140E-03   0.592E-02   0.155E-01   0.203E-03   0.733E-02   0.178E-01   0.665E-03   0.503E-04   0.174E-03
     0.155E-04

--  5.50--
     0.979E-04   0.435E-02   0.125E-01   0.113E-03   0.526E-02   0.141E-01   0.556E-03   0.366E-04   0.156E-03
     0.928E-05

--  6.00--
     0.727E-04   0.329E-02   0.103E-01   0.731E-04   0.392E-02   0.115E-01   0.471E-03   0.278E-04   0.159E-03
     0.626E-05

--  6.50--
     0.561E-04   0.256E-02   0.861E-02   0.513E-04   0.301E-02   0.959E-02   0.404E-03   0.218E-04   0.159E-03
     0.456E-05

--  7.00--
     0.446E-04   0.203E-02   0.734E-02   0.380E-04   0.236E-02   0.812E-02   0.350E-03   0.176E-04   0.152E-03
     0.350E-05
```

```
--  7.50--
     0.362E-04   0.164E-02   0.633E-02   0.294E-04   0.189E-02   0.698E-02   0.306E-03   0.144E-04   0.144E-03
     0.279E-05

--  8.00--
     0.300E-04   0.134E-02   0.552E-02   0.234E-04   0.154E-02   0.606E-02   0.269E-03   0.121E-04   0.135E-03
     0.230E-05

--  8.50--
     0.252E-04   0.112E-02   0.485E-02   0.191E-04   0.127E-02   0.532E-02   0.239E-03   0.102E-04   0.125E-03
     0.194E-05

--  9.00--
     0.215E-04   0.938E-03   0.431E-02   0.159E-04   0.106E-02   0.471E-02   0.214E-03   0.876E-05   0.116E-03
     0.167E-05

--  9.50--
     0.185E-04   0.796E-03   0.385E-02   0.134E-04   0.895E-03   0.420E-02   0.192E-03   0.759E-05   0.107E-03
     0.146E-05

-- 10.00--
     0.162E-04   0.682E-03   0.346E-02   0.115E-04   0.763E-03   0.377E-02   0.174E-03   0.664E-05   0.992E-04
     0.130E-05
```

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

          EIGENVALUES

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

          ** REAL PART **   ** IMAGINARY PART ***          ** DAMPING RATIO **

NO. 1       0.0                  0.0                             0.0
NO. 2     -0.2092248D+00       0.4071266D+01                   0.5132288E-01
NO. 2     -0.2092248D+00      -0.4071266D+01                   0.5132288E-01
NO. 3     -0.1481056D+00       0.2421839D+01                   0.6104013E-01
NO. 3     -0.1481056D+00      -0.2421839D+01                   0.6104013E-01
NO. 4     -0.2021002D+00       0.1945624D+01                   0.1033183E+00
NO. 4     -0.2021002D+00      -0.1945624D+01                   0.1033183E+00
NO. 5     -0.2632103D+00       0.1875147D+01                   0.1390050E+00
NO. 5     -0.2632103D+00      -0.1875147D+01                   0.1390050E+00
NO. 6     -0.3342787D-01       0.1260777D+01                   0.2650439E-01
NO. 6     -0.3342787D-01      -0.1260777D+01                   0.2650439E-01
NO. 7     -0.1144592D-01       0.6041955D+00                   0.1894066E-01
NO. 7     -0.1144592D-01      -0.6041955D+00                   0.1894066E-01
NO. 8     -0.1537569D-01       0.3459382D+00                   0.4440252E-01
NO. 8     -0.1537569D-01      -0.3459382D+00                   0.4440252E-01
NO. 9     -0.1094665D+00       0.2452122D+00                   0.4076408E+00
NO. 9     -0.1094665D+00      -0.2452122D+00                   0.4076408E+00
NO.10     -0.2820982D+00       0.4124979D-01                   0.9894778E+00
NO.10     -0.2820982D+00      -0.4124979D-01                   0.9894778E+00
NO.11     -0.3179021D+00       0.0                             0.1000000E+01
```

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

     EIGENVECTORS

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


-- CORRESPONDING TO NO. 1 EIGENVALUE --      ( 0.0       )+IMAG( 0.0       )

     ** ABSOLUTE VALUE **                         ** REAL PART **  ** IMAGINARY PART **


        0.0                                         0.0              0.0
        0.0                                         0.0              0.0
        0.0                                         0.0              0.0
        0.0                                         0.0              0.0
        0.0                                         0.0              0.0
        0.0                                         0.0              0.0
        0.0                                         0.0              0.0
        0.0                                         0.0              0.0
        0.0                                         0.0              0.0
        0.1000000E+01                               0.1000000E+01    0.0




-- CORRESPONDING TO NO. 2 EIGENVALUE --      (-0.209D+00)+IMAG( 0.407D+01)

     ** ABSOLUTE VALUE **                         ** REAL PART **  ** IMAGINARY PART **


        0.1107064E+00                               0.5960987E-01   -0.9328753E-01
        0.4401162E-02                               0.4983202E-03   -0.4372858E-02
        0.3052047E-01                               0.3021199E-01   -0.4328273E-02
        0.1000000E+01                               0.1000000E+01    0.0
        0.2562390E-01                              -0.2548093E-01    0.2702991E-02
        0.1051418E-01                               0.2121001E-02    0.1029802E-01
        0.1073238E-03                               0.3453433E-04    0.1016158E-03
        0.1779419E-01                              -0.1765273E-01    0.2239201E-02
        0.5262852E-02                               0.1084798E-02    0.5149834E-02
        0.1380952E-01                               0.1266621E-01   -0.5501822E-02
```

```
-- CORRESPONDING TO NO. 3 EIGENVALUE --     (-0.148D+00)+IMAG( 0.242D+01)

     ** ABSOLUTE VALUE **                        ** REAL PART **  ** IMAGINARY PART **


         0.1813177E-01                            0.5256832E-02    -0.1735301E-01
         0.4931911E+00                            0.4925372E+00    -0.2539496E-01
         0.3320159E+00                           -0.2398358E-01    -0.3311487E+00
         0.3046985E-02                            0.5288171E-03     0.3000749E-02
         0.9230196E+00                            0.5510997E-02     0.9230042E+00
         0.1000000E+01                            0.1000000E+01     0.0
         0.1226842E-01                            0.1224252E-01    -0.7968692E-03
         0.3355967E-02                           -0.1360433E-03     0.3353212E-02
         0.1654489E+00                           -0.1650685E+00     0.1121332E-01
         0.9685631E-03                           -0.8815918E-03    -0.4011374E-03
```

```
-- CORRESPONDING TO NO. 4 EIGENVALUE --     (-0.202D+00)+IMAG( 0.195D+01)

     ** ABSOLUTE VALUE **                        ** REAL PART **  ** IMAGINARY PART **


         0.2562918E-01                            0.2539718E-01     0.4704274E-02
         0.1000000E+01                            0.1000000E+01     0.0
         0.9137043E+00                           -0.5365834E-01    -0.9121275E+00
         0.1247240E-02                           -0.8799934E-03     0.8838666E-03
         0.3160684E+00                           -0.2004018E+00     0.2444144E+00
         0.3624439E+00                            0.2816455E+00     0.2281262E+00
         0.4554648E-02                            0.4206777E-02     0.1745808E-02
         0.1546631E-02                           -0.1443410E-02     0.5555502E-03
         0.1053016E+00                           -0.9349859E-01    -0.4844000E-01
         0.2736067E-02                            0.7346331E-03    -0.2635599E-02
```

```
-- CORRESPONDING TO NO. 5 EIGENVALUE --     (-0.263D+00)+IMAG( 0.188D+01)

    ** ABSOLUTE VALUE **                     ** REAL PART **  ** IMAGINARY PART **

        0.1000000E+01                         0.1000000E+01     0.0
        0.9817582E-01                        -0.7606643E-01    -0.6206755E-01
        0.5672661E-01                         0.2192894E-01     0.5231665E-01
        0.3115343E-01                        -0.1489975E-01     0.2735936E-01
        0.1848055E-01                         0.1686821E-01     0.7549502E-02
        0.5520776E-01                         0.1644658E-02    -0.5518328E-01
        0.6100570E-03                         0.7883347E-04    -0.6049422E-03
        0.2630625E-01                        -0.2571483E-01     0.5546831E-02
        0.4581391E-01                         0.2269105E-01     0.3979991E-01
        0.1181544E+00                        -0.8819941E-03    -0.1181511E+00
```

```
-- CORRESPONDING TO NO. 6 EIGENVALUE --     (-0.334D-01)+IMAG( 0.126D+01)

    ** ABSOLUTE VALUE **                     ** REAL PART **  ** IMAGINARY PART **

        0.5219237E-01                        -0.4088937E-01    -0.3243618E-01
        0.1000000E+01                         0.1000000E+01     0.0
        0.3072107E+00                         0.1548536E+00    -0.2653276E+00
        0.4124053E-02                        -0.1214229E-02    -0.3941245E-02
        0.5034070E+00                        -0.2209883E+00    -0.4523084E+00
        0.2649921E+00                        -0.9316647E-01     0.2480742E+00
        0.1200790E-01                        -0.9718515E-02     0.7052649E-02
        0.2355482E-01                        -0.9464018E-02    -0.2156994E-01
        0.7685637E+00                        -0.7237886E+00     0.2584966E+00
        0.1921409E-01                        -0.1733962E-01     0.8277591E-02
```

```
-- CORRESPONDING TO NO. 7 EIGENVALUE --     (-0.114D-01)+IMAG( 0.604D+00)

    ** ABSOLUTE VALUE **                         ** REAL PART **  ** IMAGINARY PART **

        0.1899540E+00                            -0.9512359E-01     0.1644202E+00
        0.7186074E+00                            -0.1414695E+00     0.7045444E+00
        0.1000000E+01                             0.1000000E+01     0.0
        0.1727831E-01                            -0.1256843E-01     0.1185641E-01
        0.3531273E+00                             0.1046424E+00    -0.3372667E+00
        0.6229418E+00                            -0.2424563E+00    -0.5738217E+00
        0.5332358E-01                            -0.2112583E-01    -0.4896023E-01
        0.5013081E+00                            -0.3956270E+00     0.3078783E+00
        0.3995003E+00                            -0.2640249E+00    -0.2998188E+00
        0.4824777E+00                             0.1262509E+00     0.4656667E+00
```

```
-- CORRESPONDING TO NO. 8 EIGENVALUE --     (-0.154D-01)+IMAG( 0.346D+00)

    ** ABSOLUTE VALUE **                         ** REAL PART **  ** IMAGINARY PART **

        0.1800445E-02                             0.6788953E-03     0.1667545E-02
        0.2911242E+00                             0.2866872E+00     0.5063349E-01
        0.4352518E+00                             0.4194663E+00    -0.1161560E+00
        0.1424049E-03                             0.4478541E-04     0.1351793E-03
        0.8451266E+00                             0.5476120E+00    -0.6437076E+00
        0.6462139E+00                             0.1607376E+00     0.6259041E+00
        0.1000000E+01                             0.1000000E+01     0.0
        0.1269024E-01                             0.1486531E-02     0.1260288E-01
        0.2067884E-01                            -0.9552781E-02     0.1834010E-01
        0.1675373E-01                             0.1418500E-01     0.8914784E-02
```

```
-- CORRESPONDING TO NO. 9 EIGENVALUE --     (-0.109D+00)+IMAG( 0.245D+00)

     ** ABSOLUTE VALUE **                        ** REAL PART **  ** IMAGINARY PART **

        0.2434947E-03                             0.8291703E-04    0.2289419E-03
        0.5692835E+00                            -0.5446492E+00   -0.1656526E+00
        0.5258981E+00                             0.9681088E-01   -0.5169105E+00
        0.1357527E-04                             0.4603972E-05    0.1277072E-04
        0.9552389E+00                            -0.1093801E-01   -0.9551762E+00
        0.1000000E+01                             0.1000000E+01    0.0
        0.8736736E-01                            -0.8715111E-01    0.6141596E-02
        0.2355762E-02                             0.1651092E-02    0.1680330E-02
        0.8463684E-02                             0.7221349E-02   -0.4414298E-02
        0.4402682E-02                             0.4384559E-02   -0.3990191E-03



-- CORRESPONDING TO NO.10 EIGENVALUE --     (-0.282D+00)+IMAG( 0.412D-01)

     ** ABSOLUTE VALUE **                        ** REAL PART **  ** IMAGINARY PART **

        0.2436394E-02                            -0.2435637E-02    0.6074352E-04
        0.1000000E+01                             0.1000000E+01    0.0
        0.9246252E+00                            -0.2559280E+00   -0.8885002E+00
        0.1798500E-04                            -0.9399982E-05   -0.1533299E-04
        0.6785136E+00                            -0.5765943E+00   -0.3576586E+00
        0.4797139E+00                            -0.2741120E+00    0.3936852E+00
        0.4522312E-01                             0.4180256E-01    0.1725332E-01
        0.5225778E-02                             0.2554759E-02    0.4558720E-02
        0.3273904E-01                            -0.2485351E-01    0.2131075E-01
        0.5858627E-01                             0.5792243E-01    0.8794524E-02
```

```
-- CORRESPONDING TO NO.11 EIGENVALUE --      (-0.318D+00)+IMAG( 0.0        )

      ** ABSOLUTE VALUE **                          ** REAL PART **   ** IMAGINARY PART **

         0.4641774E-01                             -0.4641774E-01     0.0
         0.1062824E+00                              0.1062824E+00     0.0
         0.5789774E-01                              0.5789774E-01     0.0
         0.2153117E-03                              0.2153117E-03     0.0
         0.3157100E-01                             -0.3157100E-01     0.0
         0.7271057E-01                             -0.7271057E-01     0.0
         0.3048246E-02                              0.3048246E-02     0.0
         0.6654162E-02                             -0.6654162E-02     0.0
         0.2990245E-02                             -0.2990245E-02     0.0
         0.1000000E+01                              0.1000000E+01     0.0
```