LANGLEY GRANT
IN-61-CR
128068
110P.

SYRACUSE UNIVERSITY

DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING

SYRACUSE, NEW YORK 13244-1240

(NASA-CR-182537) COMPREHENSIVE ANALYSIS OF HELICOPTERS WITH BEARINGLESS ROTORS Status Report (Syracuse Univ.) 110 p CSCL 09B

N88-18366

Unclas
G3/61 0128068

LANGLEY GRANT
IN-61-CR

COMPREHENSIVE ANALYSIS OF HELICOPTERS WITH BEARINGLESS ROTORS

BY

V. R. MURTHY
PRINCIPAL INVESTIGATOR

STATUS REPORT

submitted to

NASA, Langley Research Center

under

NASA Research Grant NAG-1-759

MARCH 1988

ABSTRACT

A modified Galerkin's method is developed to analyze the dynamic problems of multiple-load-path bearingless rotor blades. The development and selection of functions are quite parallel to CAMRAD procedures. This greatly facilitates the implementation of the method into the CAMRAD program. A software is developed implementing the modified Galerkin's method to determine free vibration characteristics of multiple-load-path rotor blades undergoing coupled flapwise bending, chordwise bending, twisting, and extensional motions. Results are in the process of being obtained by debugging the software.

TABLE OF CONTENTS

LIST OF FIGURES

NOMENCLATURE

A	Area of cross-section
e	Mass centroid offset from elastic axis
E	Young's modulus
G	Shear modulus
I_R	Reference moment of inertia
I_1, I_2	Bending moments of inertia about major and minor neutral axes
k_m	Mass polar radius of gyration of blade cross section about elastic axis
k_{m_1}, k_{m_2}	Mass radii of gyration about major neutral axis and about an axis perpendicular to chord
ℓ	Length of the load paths
m	Mass per unit length
m_ℓ	Reference mass
M_x	Twisting moment about x-axis
M_y, M_z	Bending moments about y and z axes, respectively
n	Number of load path
N	Number of selected functions
R	Radius of the rotor
T	Tension
u, v, w	Elastic displacements in the x,y,z directions, respectively
V_x	Axial force
V_y, V_z	Shear forces along y and z directions, respectively
x, y, z	Mutually perpendicular axis system with x - along the blade elastic axis y - towards the leading edge

β	Pretwist angle
ϕ	Elastic twist about the elastic axis
ω	Frequencies of vibration
Ω	Blade rotational speed

1. INTRODUCTION

The bearingless rotorcraft offers reduced weight, less complexity, and superior flying qualities. Bearingless rotor technology was successfully applied by Sikorsky in the tail rotors of the Blackhawk and S-76 helicopters (Ref. 1). Boeing Vertol built the first successful bearingless main rotor (BMR) and flew it in 1978 (Refs. 2 and 3). During a recent study for concept definition of an Integrated Technology Rotor/Flight Research Rotor (ITR/FRR), thirty-three hub concepts were proposed and twenty-one of these were bearingless designs (Ref. 4). This suggests that the next generation of rotorcraft will most likely be equipped with bearingless rotors. Boeing Vertol is currently designing and advanced bearingless rotor hub for MBB-105 blade (ABRS Project).

All practical designs of bearingless rotors include multiple load paths and one that was flight tested by Boeing Vertol has three load paths. The existence of multiple load paths change the dynamic behavior of the helicopter significantly, and also provides a variety of options to the designer. Therefore, it is important to develop a capability to analyze the helicopters with multiple-load-path bearingless hubs. CAMRAD (Ref. 5) is a comprehensive rotorcraft analysis program developed at NASA-Ames. This program utilizes recently developed technology to analyze the following problems:

1. Trim solution
2. Performance, loads and noise
3. Stability derivatives and handling qualities

4. Aeroelastic stability
5. Vibration and gust response

CAMRAD does not model multiple load paths and many users need this capability in CAMRAD.

The modification of CAMRAD program to include the multiple load path is undertaken under the present program. This modification is a major effort requiring several years of concentrated study. The modification is undertaken basically in two major phases:

Phase 1: Extension and validation of CAMRAD procedures to include multiple load paths.

Phase 2: Incorporation of procedures developed under phase 1 into the CAMRAD code under consultation with Dr. Wayne Johnson.

The first two years will be spent for Phase 1 studies. As a part of this phase, a systematic study is undertaken to extend the CAMRAD technologies. CAMRAD uses Galerkin's method (Ref. 5) following reference 6. The limitations of this method as exists in CAMRAD are:

1. Single-load-path modelling can only be done
2. Axial degree-of-freedom is not included in the analysis, and this degree-of-freedom is important for multiple-load-path modelling
3. There is no coupling between bending and torsion in the blade modes

Therefore, a modified Galerkin's method as employed in CAMRAD following reference 6 is extended for bearingless multiple-load-path blade. A computer program is developed to validate the formulation. The details of the formulation and the listing of the computer program are included in this status report. The results are in the process of being obtained.

2. BASIC EQUATIONS

The linear, homogeneous, undamped equations of motion for simple harmonic free vibrations with frequency ω can be written as (Ref. 7)

$$-(EAu')' - \Omega^2 mu - \omega^2 mu = 0 \tag{1}$$

$$\begin{aligned} &-(Tv')' + \{(EI_1 \sin^2\beta + EI_2 \cos^2\beta)v'' \\ &+ (EI_2 - EI_1)\cos\beta\sin\beta w''\}'' - \Omega^2 mv - \omega^2 mv \\ &+ \Omega^2 me\sin\beta\phi + \omega^2 me\sin\beta\phi + (\Omega^2 mex\sin\beta\phi)' = 0 \end{aligned} \tag{2}$$

$$\begin{aligned} &-(Tw')' + \{(EI_1 \cos^2\beta + EI_2 \sin^2\beta)w'' \\ &+ (EI_2 - EI_1)\cos\beta\sin\beta v''\}'' - \omega^2 mw - \omega^2 me\cos\beta\phi \\ &- (\Omega^2 mex\cos\beta\phi)' = 0 \end{aligned} \tag{3}$$

$$\begin{aligned} &-(GJ\phi')' - \omega^2 mk_m^2 + \Omega^2 m(k_{m_2}^2 - k_{m_1}^2)\cos_2\beta\phi \\ &- \Omega^2 mex(-\sin\beta v' + \cos\beta w') \\ &+ \Omega^2 me\sin\beta v + \omega^2 me\sin\beta v - \omega^2 me\cos\beta w = 0 \end{aligned} \tag{4}$$

$$T = \int_x^R \Omega^2 mx\,dx \tag{5}$$

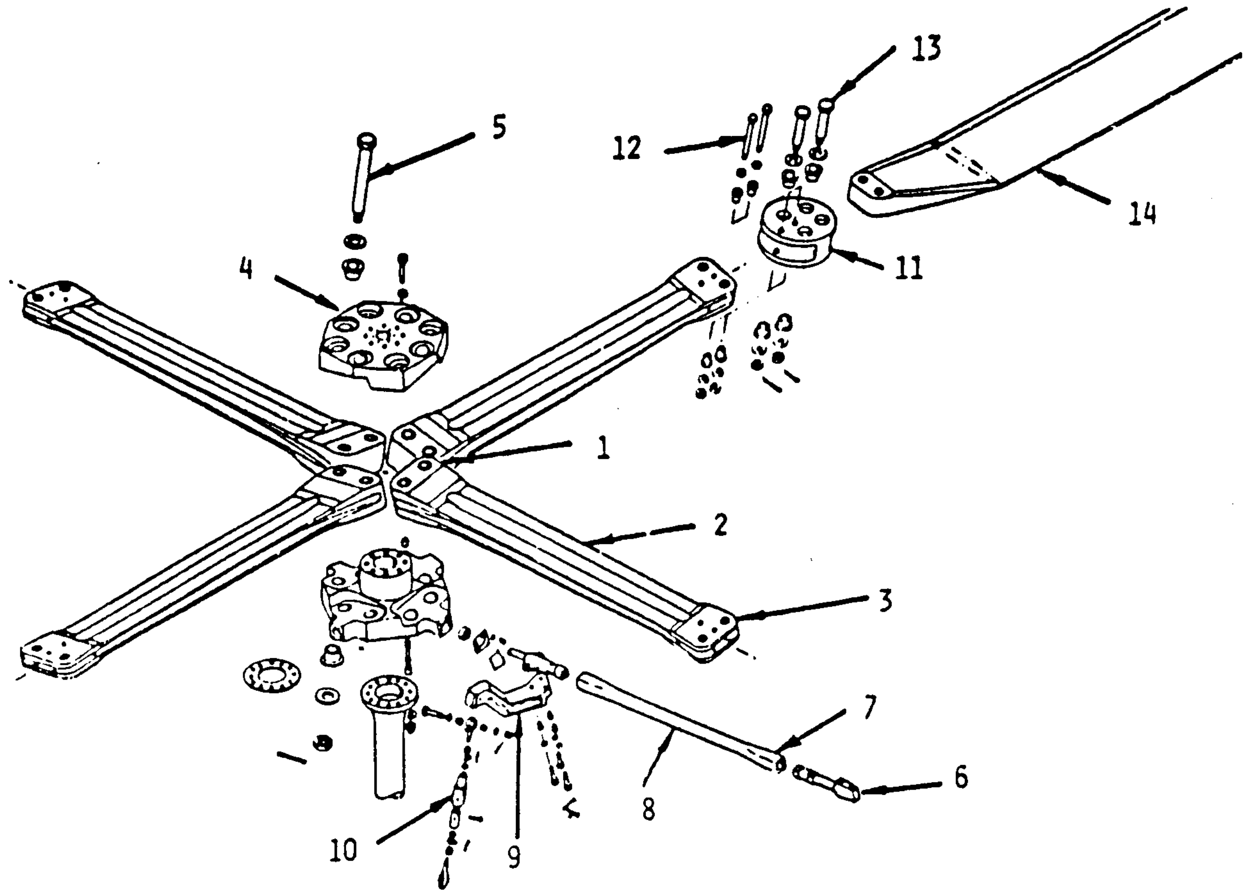
For algebraic simplicity, the coefficients B_1, B_2, e_A, k_A^2 are assumed to be zero in Eqs. (2) to (4). Also, Eq. (1) governing the stretching motion is added to introduce the axial coupling associated with the multiple-load-path blades.

3. BEARINGLESS ROTOR BLADES

The bearingless main rotor that was flight tested by Boeing Vertol has three load paths, viz., two fiberglass flexbeams and a filament wound torque tube (Ref. 8). The components of this bearingless main rotor are shown in Fig. 1. The outboard-ends of the flexbeams and torque tube are connected to the blade through a rigid clevis. The inboard-ends of the flexbeams are rigidly connected to the hub. The inboard-end of the torque tube is restrained in torsion by the control system stiffness and is pinned to the hub by a rod-end bearing (translational deflections and bending moments are equal to zero). The idealization of such a blade is shown in Fig. 2. Locations 1, 2, 3 in this figure correspond to the root of the load paths, clevis and tip of the blade respectively. The bearingless main rotor that is currently being designed by Boeing-Vertol has two load paths similar to the one shown in Fig. 3. The formulation presented here is quite general and the computer program is developed for two load paths and can be modified easily for increased number of load paths. For the analysis of multiple-load-path blades, it is important to establish the equilibrium and the compatibility relations across the clevis.

3.1 Equilibrium Across the Clevis

Consider the free-body diagram for the clevis as shown in Fig. 4 and let h_{y_i} and h_{z_i} be the y and z coordinates of the ith load path with reference to a coordinate system located at the blade (Point 'O').



ITEM	DESCRIPTION
1	FLEXURE, INBOARD ATTACHMENT
2	FLEXURE
3	FLEXURE, OUTBOARD ATTACHMENT
4	STEEL HUB
5	FLEXURE/HUB ATTACHMENT BOLTS
6	TORQUE TUBE, OUTBOARD FITTING
7	TORQUE TUBE, OUTBOARD ATTACHMENT
8	TORQUE TUBE
9	PITCH ARM ASSEMBLY
10	PITCH LINK ASSEMBLY
11	CLEVIS
12	TORQUE TUBE/CLEVIS ATTACHMENT BOLTS
13	BLADE/CLEVIS ATTACHMENT BOLTS
14	ROTOR BLADE ASSEMBLY

Fig. 1. Components of a Bearingless Main Rotor

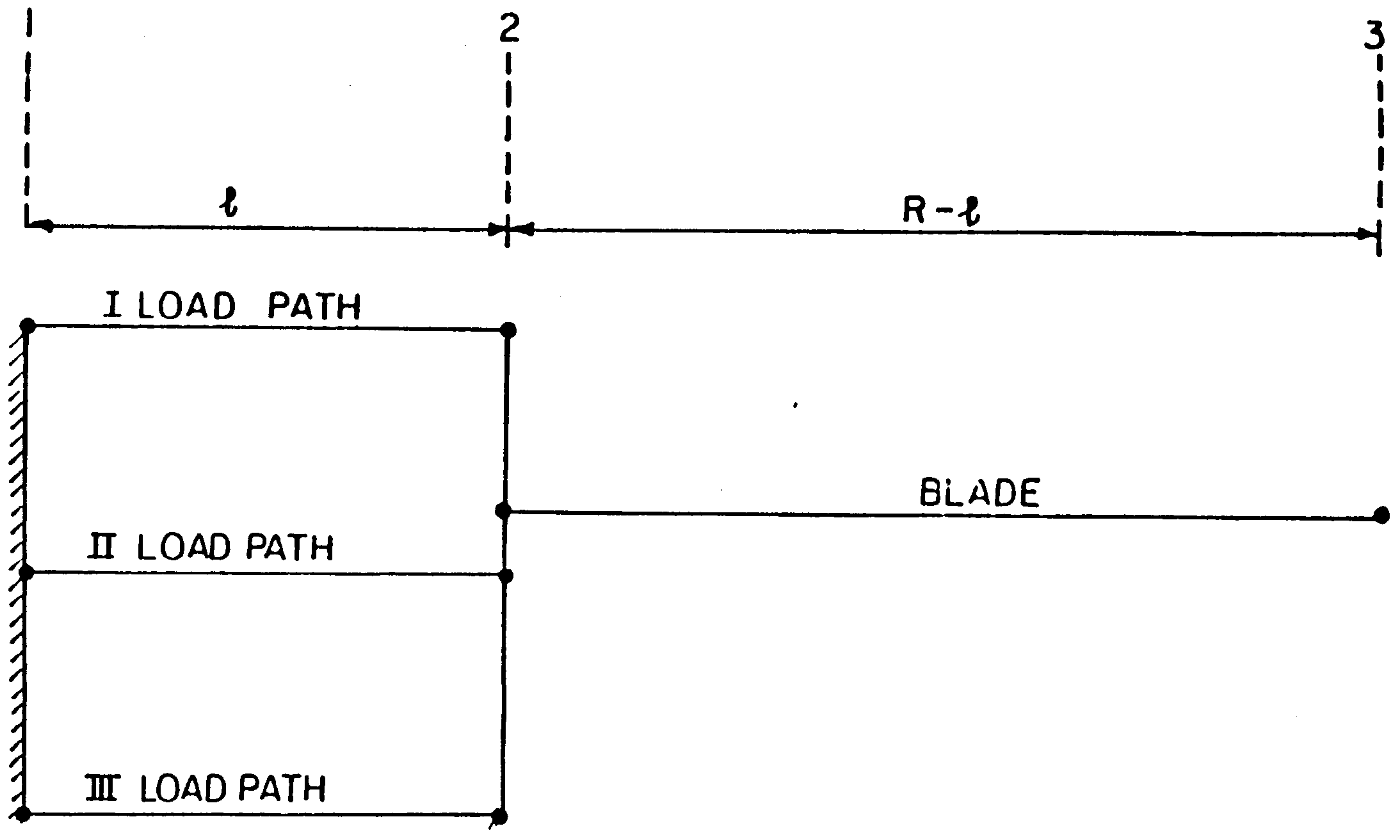


Fig. 2. Model for a Triple-Load-Path Blade

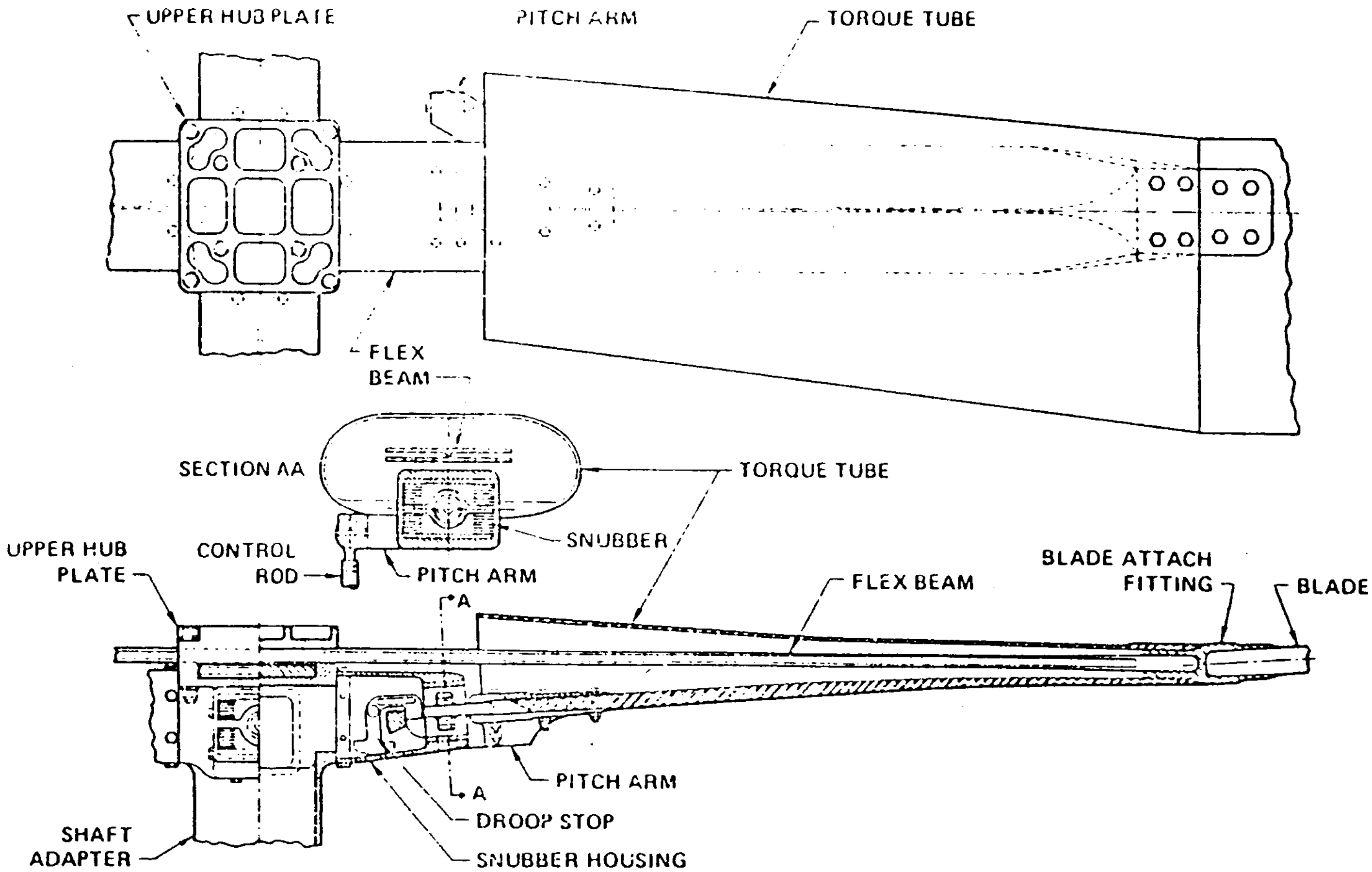


Fig. 3. Bearingless Rotor Concept with Torque Tube Enclosing the Flexibeam

Force and moment equilibrium require the following relations to be satisfied

$$V_x = \sum_{i=1}^{n} V_{x_i} \tag{6}$$

$$V_y = \sum_{i=1}^{n} V_{y_i} \tag{7}$$

$$V_z = \sum_{i=1}^{n} V_{z_i} \tag{8}$$

$$M_x = \sum_{i=1}^{n} (M_{x_i} + h_{y_i} V_{z_i} - h_{z_i} V_{y_i}) \tag{9}$$

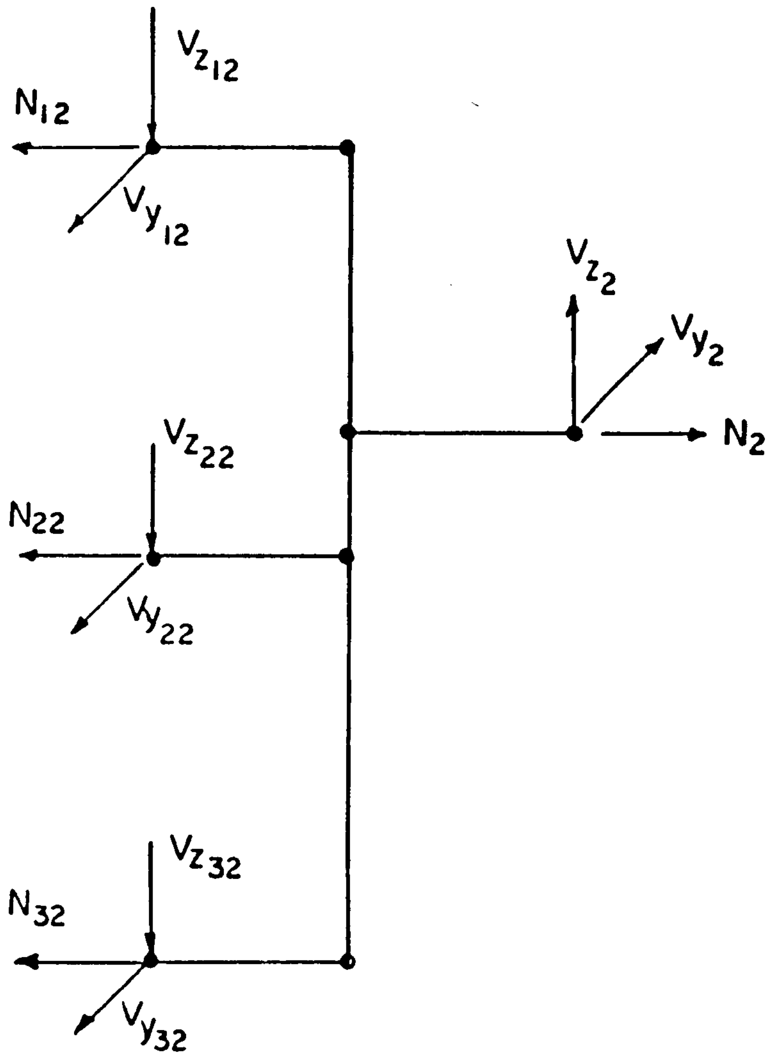
$$M_y = \sum_{i=1}^{n} (M_{y_i} + h_{z_i} V_{x_i}) \tag{10}$$

$$M_z = \sum_{i=1}^{n} M_{z_i} - h_{x_i} V_{x_i} \tag{11}$$

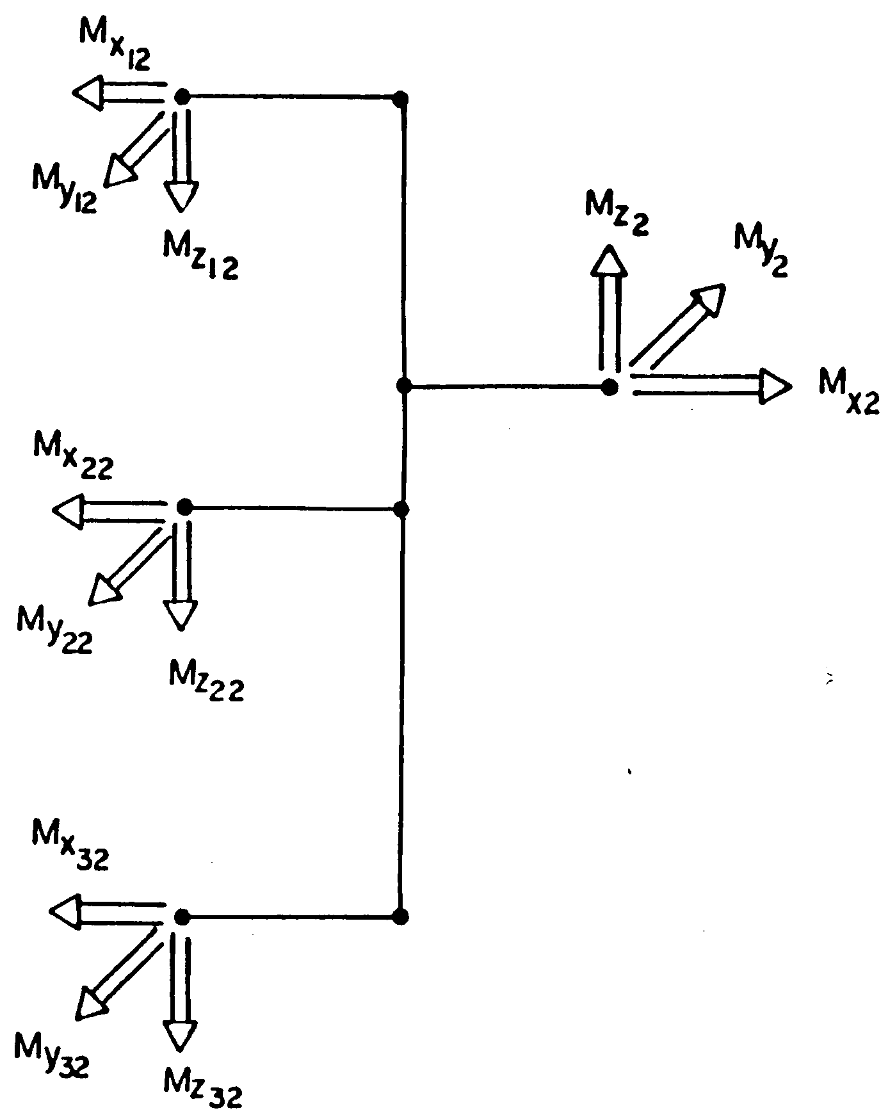
where n = number of load paths.

3.2 Compatibility Across the Clevis

Consider the plane of the clevis as shown in Fig. 5. The axial displacement of the ith load path can be written as:



A. Forces



B. Moments

Fig. 4. Free-Body Diagram of the Clevis

$$u_i = u - h_{z_i} w' - h_{y_i} v'$$

Rearrangement of the above equation yields:

$$u = u_i + h_{z_i} w' + h_{y_i} v' \tag{12}$$

The other compatibility conditions consistent with the rigid clevis are

$$w = w_i - h_{y_i} \phi_i \tag{13}$$

$$v = v_i + h_{z_i} \phi_i \tag{14}$$

$$w' = w_i' \tag{15}$$

$$v' = v_i' \tag{16}$$

$$\phi = \phi_i \tag{17}$$

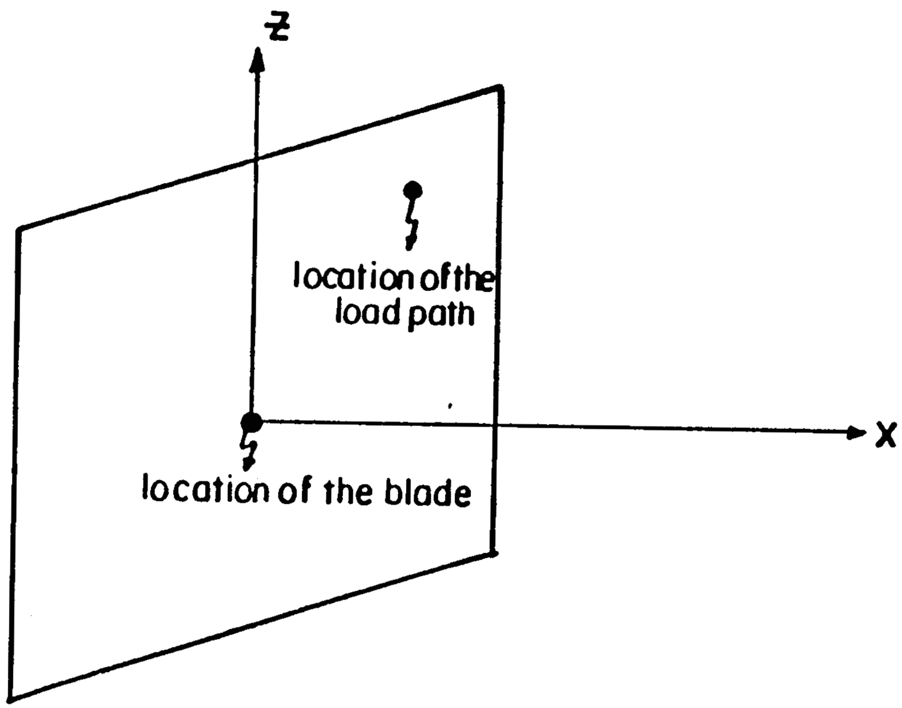


Fig. 5. Plane of the Clevis

4. MODIFIED GALERKIN'S METHOD

The basic equations of motion given by Eqs. (1) to (4) can be written as:

$$\frac{du}{dx} = \frac{V_x}{EA} \tag{18}$$

$$\frac{d\phi}{dx} = \frac{M_x}{GJ} \tag{19}$$

$$\frac{d^2v}{dx^2} = \frac{-C_{12}}{EI_1EI_2} M_y + \frac{C_{11}}{EI_1EI_2} M_z \tag{20}$$

$$\frac{d^2w}{dx^2} = \frac{C_{22}}{EI_1EI_2} M_y - \frac{C_{12}}{EI_1EI_2} M_z \tag{21}$$

$$- \frac{dV_x}{dx} \quad \Omega^2 mu = \omega^2 mu \tag{22}$$

$$- \frac{dM_x}{dx} + \Omega^2 m(k_{m_2}^2 - k_{m_1}^2)\cos 2\beta\phi$$

$$- \Omega^2 mex(-\sin\beta v' + \cos\beta w') + \Omega^2 me\sin\beta v$$

$$= \omega^2 mk_m^2\phi - \omega^2 me\sin\beta v + \omega^2 me\cos\beta w \tag{23}$$

$$\frac{d^2M_y}{dx^2} - (T \frac{dw}{dx})' - (\Omega^2 mcx\cos\beta\phi)'$$

$$= \omega^2 mw + \Omega^2 me\cos\beta\phi \tag{24}$$

$$\frac{d^2M_z}{dx^2} - (T\,\frac{dv}{dx})' + (\Omega^2 mex\sin\beta\phi)'$$

$$+ \Omega^2 me\sin\beta\phi - \Omega^2 mv = \omega^2 mv - \omega^2 me\sin\beta\phi \qquad (25)$$

where

$$C_{11} = EI_1\cos^2\beta + EI_2\sin^2\beta$$

$$C_{12} = (EI_2 - EI_1)\cos\beta\sin\beta$$

$$C_{22} = EI_1\sin^2\beta + EI_2\cos^2\beta$$

A convenient set of functions for Galerkin's method are selected with the following properties

1. Satisfy the root boundary conditions of the load paths
2. Satisfy the clevis compatibility conditions
3. Satisfy the blade tip conditions

The deflections and forces are expanded as finite series in terms of the selected functions as shown below. The choice of functions and their modifications are discussed in Chapter 5.

$$u = \sum_{n=1}^{N} a_{1n} u_n$$

$$u_i = \sum_{n=1}^{N} a_{1n} u_{n_i}$$

$$v = \sum_{n=1}^{N} a_{an} v_n$$

$$\left.\begin{aligned} v_i &= \sum_{n=1}^{N} a_{2n} v_{n_i} \\ w &= \sum_{n=1}^{N} a_{3n} w_n \\ w_i &= \sum_{n=1}^{N} a_{3n} w_{n_i} \\ \phi &= \sum_{n=1}^{N} a_{4n} \phi_n \\ \phi_i &= \sum_{n=1}^{N} a_{4n} \phi_{n_i} \end{aligned}\right\} \qquad (26)$$

$$\left.\begin{aligned} V_x &= \sum_{n=1}^{N} b_{1n} V_{xn} \\ V_{x_i} &= \sum_{n=1}^{N} b_{1n} v_{xn_i} \\ M_z &= \sum_{n=1}^{N} b_{2n} M_{zn} \end{aligned}\right\}$$

$$\left.\begin{aligned}
M_{z_i} &= \sum_{n=1}^{N} b_{2n} M_{zn_i} \\
M_y &= \sum_{n=1}^{N} b_{3n} M_{yn_i} \\
M_{y_i} &= \sum_{n=1}^{N} b_{3n} M_{yn_i} \\
M_x &= \sum_{n=1}^{N} b_{4n} M_{xn_i} \\
M_{x_i} &= \sum_{n=1}^{N} b_{4n} M_{xn_i}
\end{aligned}\right\} \tag{27}$$

The satisfaction of the following equations is the modified Galerkin's method for multiple-load-path blades which is equivalent to the quotient method described in Ref. 6. (N = Number of functions selected; n = number of load paths)

$$\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} \int_o^{\ell} [b_{3j} M''_{yj_i} - a_{3j}(T_i w'_{j_i})' + a_{4j}(\Omega^2 m_i e_i x \cos\beta_i \phi_{j_i})'] \, w_{k_i} dx \right\}$$

$$+ \sum_{j=1}^{N} \left\{ \int_{\ell}^{R} [b_{3j}M''_{y_j} - a_{3j}(Tw'_j)'\right.$$

$$\left. + a_{4j}(\Omega^2 mexcos\beta\phi_j)'] \, w_k dx \right\}$$

$$+ \left[\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} (-b_{3j}M'_{yj_i} + a_{3j}T_i w'_{j_i} + a_{4j}\Omega^2 m_i e_i xcos\beta_i \phi_{j_i}) \right.\right.$$

$$\left.\left. - (-b_{3j}M'_{y_j} + a_{3j}Tw_j + a_{4j}\Omega^2 mexcos\beta\phi_j) \right\} w_k \right]_{x=\ell}$$

$$+ \left[\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} (b_{3j}M_{yj_i} - h_{z_i} b_{1j} V_{xj_i}) - b_{3j}M_{yj} \right\} w'_k \right]_{x=\ell}$$

$$= \omega^2 \left[\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} \int_{o}^{\ell} m_i (a_{3j} w_{j_i} + a_{4j} e_i cos\beta_i \phi_{j_i}) w_{k_i} dx \right\} \right]$$

$$+ \omega^2 \left[\sum_{j=1}^{N} \int_{\ell}^{R} m(a_{3j}w_j + a_{4j}ecos\beta\phi_j) w_k dx \right] \tag{28}$$

where

$k = 1, 2, 3, \ldots N$

$$\sum_{j=1}^{N} < \sum_{i=1}^{n} \left\{ \int_{o}^{\ell} [b_{2j}M''_{zj_i} - a_{2j}(T_i v'_{j_i})' + a_{4j}(\Omega^2 m_i e_i x \sin\beta_i \phi_{j_i})' \right.$$

$$\left. + a_{4j}\Omega^2 m_i e_i \sin\beta_i \phi_{j_i} - a_{2j}\Omega^2 m_i v_{j_i}]\, v_{k_i} dx \right\} >$$

$$+ \sum_{j=1}^{N} \left\{ \int_{\ell}^{R} [b_{2j}M''_{zj} - a_{2j}(Tv'_j)' + a_{4j}(\Omega^2 m e x \sin\beta\phi_j)' \right.$$

$$\left. + a_{4j}\Omega^2 m e \sin\beta\phi_j - a_{2j}\Omega^2 m v_j]\, v_k dx \right\}$$

$$+ \left[\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} (-b_{2j}M'_{zj_i} + a_{2j}T_j v'_{j_i} - a_{4j}\Omega^2 m_i e_i x \sin\beta_i \phi_{j_i}) \right. \right.$$

$$\left. \left. - (-b_{2j}M'_{zj} + a_{2j}Tv'_j - a_{4j}\Omega^2 m e x \sin\beta\phi_j) \right\} v_k \right]_{x=\ell}$$

$$+ \left[\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} (b_{2j}M_{zj_i} - h_{x_i} b_{1j} V_{xj_i}) - b_{2j}M_{zj} \right\} v'_k \right]_{x=\ell}$$

$$= \omega^2 \left[\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} \int_{o}^{\ell} m_i (a_{2j} v_{j_i} - a_{4j} e_i \sin\beta_i \phi_{j_i}) v_{k_i} dx \right\} \right]$$

$$+ \omega^2 \left[\sum_{j=1}^{N} \int_{\ell}^{R} m(a_{2j} v_j - a_{4j} e \sin\beta\phi_j) v_k dx \right] \qquad (29)$$

where k = 1, 2, 3, ... N

$$\sum_{j=1}^{N}\left[\sum_{i=1}^{n}\int_{o}^{\ell}\left\{-b_{4j}M'_{xj_i} + \Omega^2 m_i e_i x(-a_{2j}\cos\beta_i v'_{j_i} + a_{3j}\sin\beta_i w'_{j_i})\right.\right.$$

$$\left.\left. + a_{2j}\Omega^2 m_i e_i \sin\beta_i v_{j_i} + a_{4j}\Omega^2 m_i (k^2_{m_{2_i}} - k^2_{m_{1_i}})\cos 2\beta_i \phi_{j_i}\right\}\phi_{k_i} dx\right]$$

$$+ \sum_{j=1}^{N}\left[\int_{\ell}^{R}\left\{-b_{4j}M'_{xj} + \Omega^2 mex(-a_{2j}\cos\beta v'_j + a_{3j}\sin\beta w'_j)\right.\right.$$

$$\left.\left. + a_{2j}\Omega^2 me\sin\beta v_j + a_{4j}\Omega^2 m(k^2_{m_2} - k^2_{m_1})\cos 2\beta\phi_j\right\}\phi_k dx\right]$$

$$+ < \sum_{j=1}^{N}\left\{\sum_{i=1}^{n}[b_{4j}M_{xj_i} + h_{y_i}(-b_{3j}M'_{yj_i} + a_{3j}T_i w'_{j_i}\right.$$

$$+ a_{4j}\Omega^2 m_i e_i x\cos\beta_i\phi_{j_i}) - h_{z_i}(-b_{2j}M'_{zj_i} + a_{2j}T_i v'_{j_i}$$

$$\left. - a_{4j}\Omega^2 m_i e_i x\sin\beta_i\phi_{j_i})] - b_{4j}M_{xj}\right\}\phi_k >_{x=\ell}$$

$$= \omega^2\left[\sum_{j=1}^{N}\left\{\sum_{i=1}^{n}\int_{o}^{\ell} m_i(a_{4j}k^2_{m_i}\phi_{j_i} + a_{3j}e_i\cos\beta_i w_{j_i}\right.\right.$$

$$\left.\left. - a_{2j}e_i\sin\beta_i v_{j_i})\,\phi_{k_i} dx\right\}\right]$$

$$+ \omega^2\left\{\sum_{j=1}^{N}\int_{\ell}^{R} m(a_{4j}k^2_m\phi_j + a_{3j}e\cos\beta w_j\right.$$

$$\left. - a_{2j}e\sin\beta v_j)\,\phi_k dx\right\} \qquad (30)$$

where k = 1, 2, 3, ... N

$$\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} \int_{o}^{\ell} (-b_{1j} V_{xj_i} - a_{1j} \Omega^2 m_i u_{j_i}) \, u_{k_i} dx \right\}$$

$$+ \sum_{j=1}^{N} \left\{ \int_{\ell}^{R} (-b_{1j} V'_{xj} - a_{1j} \Omega^2 m u_j) \, u_k dx \right\}$$

$$+ \left[\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} (b_{1j} V_{xj_i}) - b_{1j} V_{xj} \right\} u_k \right]_{x=\ell}$$

$$= \omega^2 \left[\sum_{j=1}^{N} \left\{ \int_{o}^{\ell} m_i a_{1j} u_{j_i} u_{k_i} dx \right\} \right]$$

$$+ \omega^2 \left[\int_{\ell}^{R} m a_{1j} u_j u_k dx \right] \tag{31}$$

where k = 1, 2, 3, ... N

$$\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} \int_{o}^{\ell} \left(a_{1j} u_{j_i} - \frac{b_{1j} V_{xj_i}}{AE_i} \right) V_{xk_i} dx \right\}$$

$$+ \sum_{j=1}^{N} \left\{ \int_{\ell}^{R} \left(a_{1j} u_j - \frac{b_{1j} V_{xj}}{EA} \right) V_{xk} dx \right\} = 0 \tag{32}$$

$$\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} \int_{o}^{\ell} \left(a_{4j} \phi_{j_i} - \frac{b_{4j} M_{xj_i}}{GJ_i} \right) M_{xk_i} dx \right\}$$

$$+ \sum_{j=1}^{N} \left\{ \int_{\ell}^{R} \left(a_{4j} \phi_j - \frac{b_{4j} M_{xj}}{GJ} \right) M_{xk} dx \right\} = 0 \tag{33}$$

$$\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} \int_{o}^{\ell} \left(a_{2j} v''_{j_i} + \frac{C_{12}}{EI_1 EI_2} b_{3j} M_{yj_i} \right. \right.$$

$$\left. \left. - \frac{C_{11}}{EI_1 EI_2} b_{2j} M_{zj_i} \right) M_{zk_i} dx \right\} +$$

$$\sum_{j=1}^{N} \left\{ \int_{\ell}^{R} \left(a_{2j} v''_j + \frac{C_{12}}{EI_1 EI_2} b_{3j} M_{yj} \right) \right.$$

$$\left. - \frac{C_{11}}{EI_1 EI_2} b_{2j} M_{zj} \right) M_{zk} \right\} = 0 \tag{34}$$

$$\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} \int_{o}^{\ell} \left(a_{3j} w''_{j_i} - \frac{c_{22}}{EI_1 EI_2} b_{3j} M_{yj_i} \right. \right.$$

$$\left. \left. + \frac{C_{12}}{EI_1 EI_2} b_{2j} M_{zj_i} \right) M_{yk_i} dx \right\} +$$

$$\sum_{j=1}^{N} \left\{ \int_{\ell}^{R} \left(a_{3j} w''_{j} - \frac{C_{22}}{EI_1 EI_2} b_{3j} M_{yj} \right. \right.$$

$$\left. \left. + \frac{C_{12}}{EI_1 EI_2} b_{2j} M_{zj} \right) M_{yk} dx \right\} = 0 \qquad (35)$$

5. SELECTION OF FUNCTIONS

5.1 Basic Functions

The selection of functions for multiple-load-path blades with fixed-conditions for load path members (inboard-ends) is presented in this section. The other inboard-end conditions can be incorporated easily in a straightforward manner. The CAMRAD program uses nonrotating uniform modes for the Galerkin's method; therefore, the same modes are employed for the multiple-load-path blades for uniformity. The non-rotating modes for uniform cantilever beams are given by:

1. Bending (Both Flapwise and Chordwise)

   a) Mode shapes

$$\alpha_r(x) = e_r(\sin \alpha_r \frac{x}{R} - \sinh \alpha_r \frac{x}{R}) + g_r(\cos \alpha_r \frac{x}{R} - \cosh \alpha_r \frac{x}{R}) \qquad (36)$$

where

$e_r = -(\sin \alpha_r - \sinh \alpha_r)/2\sin \alpha_r \sinh \alpha_r$

$g_r = (\cos \alpha_r + \cosh \alpha_r)/2\sin \alpha_r \sinh \alpha_r$

r = rth mode shape

$\alpha_1 = 1.875,\ \alpha_2 = 4.694,\ \alpha_3 = 7.855,\ \alpha_4 = 10.996$

$\alpha_5 = 14.137,\ \alpha_6 = 17.279,\ \alpha_7 = 20.420,\ \alpha_8 = 23.562$

$\alpha_9 = 26.704,\ \alpha_{10} = 29.845$

b) Frequencies

$$\omega_r = \alpha_r^2 \sqrt{EI/mR^4}$$

2) Stretching and Twisting

a) Mode shapes

$$\alpha_r(x) = \sin(\delta_r \frac{x}{R}) \tag{37}$$

where $\delta_r = (2r-1)\frac{\pi}{2}$

b) Frequencies

$$\omega_r = \delta_r \sqrt{EA/mR^2} \text{ , Stretching}$$

$$\omega_r = \delta_r \sqrt{GJ/mR^2} \text{ , Twisting}$$

The idealization for a two-load-path blade is shown in Fig. 6. The blade is divided into three regions

Region I: $0 \leq x \leq \ell_1$, all load path members

Region II: $\ell \leq x \leq R$, blade

Region III: $\ell_1 \leq x \leq \ell$, load path members

For regions I and II, the basic mode functions given by Eqs. (36) and (37) are used. It is therefore necessary to develop the clevis admissible comparison functions. At this stage the following two options are available.

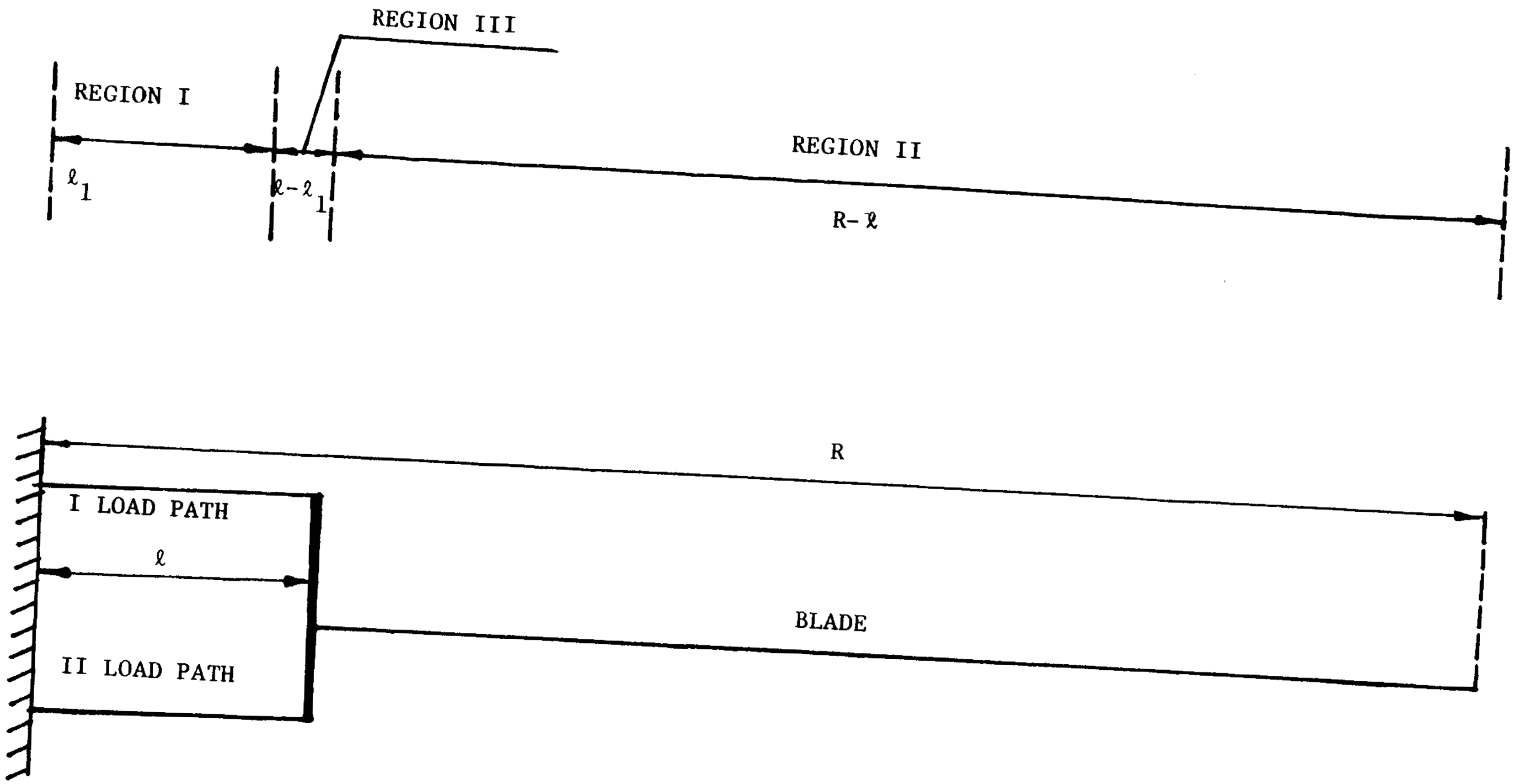


Fig. 6. Division of the Regions

1. Satisfy gometric compatibility only and account for the force conditions in the Galerkins' procedure.
2. Satisfy both geometric and force conditions.

Option 1 is selected here because the clevis compatibility functions do not depend on the eigenvalues while the clevis admissible comparison functions depend on the eigenvalues.

5.2 Clevis Compatibility Functions

1. Flapwise Bending

Assume the functions to be of the following form:

$$w_{r_i} = a^1_{0r_i} + a^1_{1r_i} x + a^1_{2r_u} x^2 + a^1_{3r_i} x^3 \tag{38}$$

The coefficients in the equation are determined by satisfying the displacement and slope continuity at $x = \ell_1$ (see Fig. 6) and geometric compatibility at the clevis [Eqs. (13) and (14)]. The resulting equations are given by:

$$a^1_{0r_i} + a^1_{1r_i}\ell_1 + a^1_{2r_i}\ell_1^2 + a^1_{3r_i}\ell_1^3 = w_{r_i}(\ell_1)$$

$$a^1_{1r_i} + 2a^1_{2r_i}\ell_1 + 3a^1_{3r_i}\ell_1^2 = w'_{r_i}(\ell_1)$$

$$a^1_{0r_i} + a^1_{1r_i}\ell + a^1_{2r_i}\ell^2 + a^1_{3r_i}\ell^3 = w_r(\ell) + h_{y_i}\phi_r(\ell)$$

$$a^1_{1r_i} + 2a^1_{2r_i}\ell + 3a^1_{3r_i}\ell^2 = w'_r(\ell)$$

Arranging the above equations into a matrix form yields:

Arranging the above equations into a matrix form yields:

$$\begin{Bmatrix} a^1_{0r_i} \\ a^1_{1r_i} \\ a^1_{2r_i} \\ a^1_{3r_i} \end{Bmatrix} = \begin{bmatrix} 1 & \ell_1 & \ell_1^2 & \ell_1^3 \\ 0 & 1 & 2\ell_1 & 3\ell_1^2 \\ 1 & \ell & \ell^2 & \ell^3 \\ 0 & 1 & 2\ell & 3\ell^2 \end{bmatrix}^{-1} \begin{Bmatrix} w_{r_i}(\ell_1) \\ w'_{r_i}(\ell_1) \\ w_r(\ell) + h_{y_i}\phi_r(\ell) \\ w'_r(\ell) \end{Bmatrix} \qquad (39)$$

where

subscript r = mode index

subscript i = load path number index

superscript 1 = flapwise bending

2. Chordwise Bending

Assume the functions to be of the following form:

$$v_{r_i} = a^2_{0r_i} + a^2_{1r_i}x + a^2_{2r_i}x^2 + a^2_{3r_i}x^3 \tag{40}$$

By employing a similar procedure used for the flapwise bending the following equation can be obtained for the chordwise bending clevis compatibility function.

$$\begin{Bmatrix} a^2_{0r_i} \\ a^2_{1r_i} \\ a^2_{2r_i} \\ a^2_{3r_i} \end{Bmatrix} = \begin{bmatrix} 1 & \ell_1 & \ell_1^2 & \ell_1^3 \\ 0 & 1 & 2\ell_1 & 3\ell_1^2 \\ 1 & \ell & \ell^2 & \ell^3 \\ 0 & 1 & 2\ell & 3\ell^2 \end{bmatrix}^{-1} \begin{Bmatrix} v_{r_i}(\ell_1) \\ v'_{r_i}(\ell_1) \\ v_r(\ell) - h_{z_i}\phi_r(\ell) \\ v'_r(\ell) \end{Bmatrix} \tag{41}$$

where superscript 2 indicates the chordwise bending.

3. Twisting

Assume the function to be of the following form:

$$\phi_{r_i} = b^1_{0r_i} + b^1_{1r_i}x \tag{42}$$

By satisfying the continuity at $x = \ell_1$ and compatibility at the clevis the following relation can be obtained for the coefficients in Eq. (42).

$$\begin{Bmatrix} b^1_{0r_i} \\ \\ b^1_{1r_i} \end{Bmatrix} = \begin{bmatrix} 1 & \ell_1 \\ \\ 1 & \ell \end{bmatrix}^{-1} \begin{Bmatrix} \phi_{r_i}(\phi_1) \\ \\ \phi_r(\ell) \end{Bmatrix} \tag{43}$$

The superscript 1 indicates twisting.

4. Stretching

Assume the functions to be of the following form:

$$u_{r_i} = b^2_{0r_i} + b^2_{1r_i} x \tag{44}$$

The coefficients in the above equations are computed by

$$\begin{Bmatrix} b^2_{0r_i} \\ \\ b^2_{r_i} \end{Bmatrix} = \begin{bmatrix} 1 & \ell_1 \\ \\ 1 & \ell \end{bmatrix}^{-1} \begin{Bmatrix} u_{r_i}(\ell_1) \\ \\ u(\ell) - h_{z_i} w'(\ell) - h_{y_i} v'(\ell) \end{Bmatrix} \tag{45}$$

The superscript 2 in the above equation indicates stretching.

The treatment for the selection of function is completed so far for the deflections, and it is necessary to establish the criteria for the forces. For simplicity the functions used for the forces are as follows.

$$\left.\begin{aligned} M_x &= \phi' \\ M_y &= w'' \\ M_z &= v'' \\ V_x &= u' \end{aligned}\right\} \tag{46}$$

For all regions (I, II and III), the basic functions defined in Eqs. (36) and (37) are employed while calculating the force functions given by Eq. (46). If the following equations are satisfied for the force function, then the modified Galerkin's procedure used in CAMRAD would reduce to the standard Galerkin form.

$$\left.\begin{aligned} M_x &= GJ\phi' \\ M_y &= (EI_1\cos^2\beta + EI_2\sin^2\beta)w'' + (EI_2\text{-}EI_1)\cos\beta\sin\beta v'' \\ M_z &= (EI_1\sin^2\beta + EI_2\cos^2\beta)v'' + (EI_2\text{-}EI_1)\cos\beta\sin\beta w'' \\ V_x &= EAu' \end{aligned}\right\} \tag{47}$$

In this situation, it is desirable to use fifth degree polynomials for bending degrees-of-freedom and quadratic functions for the twisting and stretching degrees-of-freedom for generating the clevis compatibility functions. However, the standard Galerkin form is not employed for parallel development with the CAMRAD approaches.

6. TENSION CALCULATIONS

The rotating blade mode shapes calculations are based on the linear formulation, and therefore the tension coefficient T appearing in Eqs. (2) and (3) should be known in advance; otherwise these equations will be nonlinear. Equation (5) can be employed for calculation of tension in the blade. The tensions in the load paths are calculated by assuming that the load paths are coincident with the blade at the clevis, i..e, $h_{y_i} = h_{z_i} = 0$. The tensions corresponding to this case are given by (Ref. 9):

$$T_i(x) = \int_x^{\ell} \Omega^2 m_i x dx + T_i(\ell) \tag{48}$$

where

$$T_i(\ell) \frac{\sum_{j=1}^{n} a_j}{\sum_{k=1}^{n} \left(\prod_{j=1}^{n,k} a_j \right)} \tag{49}$$

$$\prod_{j=1}^{n,i} a_j = a_1 a_2 \ldots a_n / a_i \tag{50}$$

$$a_i = \int_o^{\ell} \frac{dx}{EA_i} \tag{51}$$

It is to be noted that the tensions given by Eq. (48) correspond to the coincident load path case ($h_{y_i} = h_{z_i} = 0$), and for the noncoincident case (h_{y_i} and/or $h_{z_i} \neq 0$) the modified Galerkin's method is applied starting with the coincident load path case.

Modified Galerkin's Method for Tensions

The free-body diagram for tension calculations in the load path with blade detached is shown in Fig. 7. The equilibrium conditions forces at the clevis are given by:

$$\sum_{i=1}^{N} V_{x_i} = T(\ell) = \int_{\ell}^{R} \Omega^2 mxdx$$

$$\sum_{i=1}^{n} V_{y_i} = 0$$

$$\sum_{i=1}^{n} (M_{x_i} + h_{y_i}V_{z_i} - h_{z_i}V_{y_i}) = 0$$

$$\sum_{i=1}^{n} (M_{y_i} + h_{z_i}V_{x_i}) = 0$$

$$\sum_{i=1}^{n} (M_{z_i} - h_{y_i}V_{x_i}) = 0$$

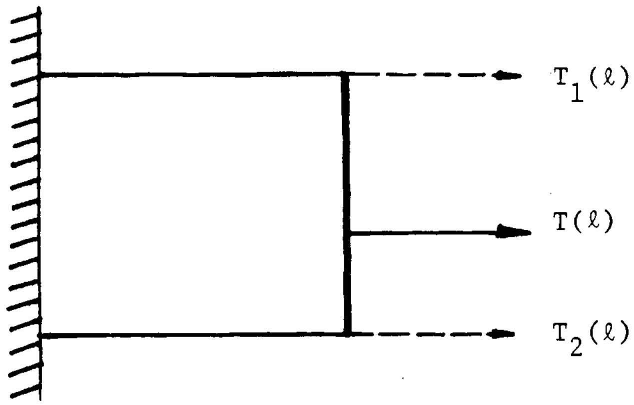


Fig. 7. Tension Distribution

A convenient set of functions are selected for the load path members with the following properties.

1. Satisfy the root boundary conditions of the load paths.
2. Satisfy the clevis compatibility conditions.

The deflections and forces in the load path members are expanded as finite series in terms of the selected functions as shown below

$$\left.\begin{aligned} u_i &= \sum_{n=1}^{N} a_{1n} u_{n_i} \\ v_i &= \sum_{n=1}^{N} a_{2n} v_{n_i} \\ w_i &= \sum_{n=1}^{N} a_{3n} w_{n_i} \\ \phi_i &= \sum_{n=1}^{N} a_{4n} \phi_{n_i} \\ V_{x_i} &= \sum_{n=1}^{N} b_{1n} V_{xn_i} \\ M_{z_i} &= \sum_{n=1}^{N} b_{2n} M_{zn_i} \end{aligned}\right\} \qquad (52)$$

$$\left.\begin{aligned} M_{y_i} &= \sum_{n=1}^{N} b_{3n} M_{yn_i} \\ M_{x_i} &= \sum_{n=1}^{N} b_{4n} M_{xn_i} \end{aligned}\right|$$

The adaptation of Galerkin's method yields the following equations:

$$\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} \int_{o}^{\ell} [b_{3j} M''_{yj_i} - a_{3j}(T_i w_{j_i})' + a_{4j}(\Omega^2 m_i e_i x \cos\beta_i \dot{\phi}_{j_i})'] w_{k_i} dx \right\}$$

$$+ \left[\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} (-b_{3j} M'_{yj_i} + a_{3j} T_i w'_{j_i} + a_{4j} \Omega^2 m_i e_i x \cos\beta_i \phi_{j_i}) w_k \right\} \right]_{x=\ell}$$

$$+ \left[\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} (b_{3j} M_{yj_i} - h_{z_i} b_{1j} V_{xj_i}) w'_k \right\} \right]_{x=\ell} = 0 \qquad (53)$$

$$
\begin{aligned}
&\sum_{n=1}^{N} \left\{ \sum_{i=1}^{n} \int_{o}^{\ell} [b_{2j} M''_{zj_i} - a_{2j}(T_i v'_{j_i})' + a_{4j}(\Omega^2 m_i e_i x \sin\beta_i \phi_j)' \right. \\
&\qquad \left. + a_{4j}\Omega^2 me_i e_i \sin\beta_i \phi_{j_i} - a_{2j}\Omega^2 m_i v_{j_i}] v_{k_i} dx \right\} \\
&+ \left[\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} (-b_{2j} M'_{zj} + a_{2j} T_i v'_{j_i} \right. \right. \\
&\qquad \left. \left. - a_{4j}\Omega^2 m_i e_i x \sin\beta_i \phi_{j_i}) v_k \right\} \right]_{x=\ell} \\
&+ \left[\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} (b_{2j}, M_{zj_i} - h_{y_i} b_{1j} V_{xj_i}) v'_k \right\} \right]_{x=\ell} = 0 \qquad (54)
\end{aligned}
$$

$$
\begin{aligned}
&\sum_{j=1}^{N} \left[\sum_{i=1}^{n} \int_{o}^{\ell} \left\{ -b_{4j} M'_{xj_i} + \Omega^2 m_i e_i x(-a_{2j}\cos\beta_i v'_{j_i} \right. \right. \\
&\qquad + a_{3j}\sin\beta_i w'_{j_i}) + a_{2j}\Omega^2 m_i e_i \sin\beta_i v_{j_i} \\
&\qquad \left. \left. + a_{4j}\Omega^2 m_i (k^2_{m_{2_i}} - k^2_{m_{1_i}}) \cos 2\beta_i \phi_{j_i} \right\} \phi_{k_i} dx \right]
\end{aligned}
$$

$$+ < \sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} [b_{4j} M_{xj_i} + h_{y_i} (-b_{3j} M'_{yj_i} + a_{3j} T_i w'_{j_i} \right.$$

$$+ a_{4j} \Omega^2 m_i e_i x \cos\beta_i \phi_{j_i}) - h_{z_i} (-b_{2j} M'_{zj_i}$$

$$\left. + a_{2j} T_i v'_{j_i} - a_{4j} \Omega^2 m_i e_i x \sin\beta_i \phi_{j_i})] \, \phi_k \right\} >_{x=\ell} = 0 \qquad (55)$$

$$\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} \int_o^{\ell} (-b_{1j} V'_{xj_i} - a_{1j} \Omega^2 m_i u_{j_i}) u_{k_i} dx \right\}$$

$$+ \left[\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} b_{1j} V_{xj_i} u_k \right\} \right]_{x=\ell} = T(\ell) u_k(\ell) \qquad (56)$$

$$\sum_{n=1}^{N} \left\{ \sum_{i=1}^{n} \int_o^{\ell} (a_{1j} u_{j_i} - \frac{b_{1j} V_{xj_i}}{EA_i}) V_{xk_i} dx \right\} = 0 \qquad (57)$$

$$\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} \int_o^{\ell} (a_{4j} \phi_{j_i} - \frac{b_{4j} M_{xj_i}}{GJ_i}) \, M_{xk_i} dx \right\} = 0 \qquad (58)$$

$$\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} \int_o^{\ell} (a_{2j} v''_{j_i} + \frac{c_{12}}{EI_1 EI_2} b_{3j} M_{yj_i} \right.$$

$$\left. - \frac{c_{11}}{EI_1 EI_2} b_{2j} M_{zj_i}) M_{zk_i} dx \right\} = 0 \qquad (59)$$

$$\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} \int_0^{\ell} \left(a_{3j} w''_{j_i} - \frac{c_{22}}{EI_1 EI_2} b_{3j} M_{yj_i} \right. \right.$$

$$\left. \left. + \frac{c_{12}}{EI_1 EI_2} b_{2j} M_{zj_i} \right) M_{yk_i} dx \right\} = 0 \tag{60}$$

where c_{11}, c_{12}, c_{22} are defined in Eq. (35). The solution of Eqs. (53) to (60) yields $T_i(\ell)$ [which is $V_{x_i}(\ell)$], and from Eq. (48) the tension coefficients in the load path members can be computed. At the most, two or three iterations may be needed starting with the coincident load path tensions to solve the nonlinear problem. The functions developed in Chapter 5 are used to solve the present problem also.

7. SOFTWARE DEVELOPMENT

7.1 Equations of Motion

The basic equations of motion given by Eqs. (18) to (25) are nondimensionalized as shown below.

$$\frac{d\bar{u}}{d\bar{x}} = \frac{\bar{V}_x}{\overline{EA}} \tag{61}$$

where $\bar{u} = u/R$; $\bar{x} = x/R$; $\overline{EA} = EAR^2/EI_R$

$\bar{V}_x = V_x R^2/EI_R$

$$\frac{d\bar{\phi}}{d\bar{x}} = \frac{\bar{M}_x}{\overline{GJ}} \tag{62}$$

where $\overline{GJ} = GJ/EI_R$; $\bar{M}_x = M_x R/EI_R$

$$\frac{d^2\bar{v}}{d\bar{x}^2} = \frac{-\overline{C}_{12}}{\overline{EI}_1\overline{EI}_2}\bar{M}_y + \frac{\overline{C}_{11}}{\overline{EI}_1\overline{EI}_2}\bar{M}_z \tag{63}$$

where $\overline{C}_{12} = (\overline{EI}_2 - \overline{EI}_1)\cos\bar{\beta}\sin\bar{\beta}$

$\overline{C}_{11} = \overline{EI}_1\cos^2\bar{\beta} + \overline{EI}_2\sin^2\bar{\beta}$

$\overline{EI}_1 = EI_1/EI_R$; $\overline{EI}_2 = EI_2/EI_R$; $\bar{v} = v/R$

$\bar{\beta} = \beta$; $\bar{M}_y = M_y R/EI_R$; $\bar{M}_z = M_z R/EI_R$

$$\frac{d^2\bar{w}}{d\bar{x}^2} = \frac{\bar{c}_{22}}{\overline{EI}_1\overline{EI}_2}\,\bar{M}_y - \frac{\bar{c}_{12}}{\overline{EI}_1\overline{EI}_2}\,\bar{M}_z \tag{64}$$

where $\bar{c}_{22} = \overline{EI}_1\sin^2\bar{\beta} + \overline{EI}_2\cos^2\bar{\beta}$

$\bar{w} = w/R$

$$\frac{-d\bar{V}_x}{d\bar{x}} - \bar{\Omega}^2\bar{m}\bar{u} = \omega^2\bar{m}\bar{u} \tag{65}$$

where $\bar{\Omega}^2 = \Omega^2 m_R R^4/EI_R$; $\bar{m} = m/m_R$

$\bar{\omega}^2 = \omega^2 m_R R^4/EI_R$

$$-\frac{d\bar{M}_x}{d\bar{x}} + \bar{\Omega}^2 m(\bar{k}_{m_2}^2 - \bar{k}_{m_1}^2)\cos 2\bar{\beta}\bar{\phi}$$

$$- \bar{\Omega}^2\bar{m}\bar{e}\bar{x}(-\sin\bar{\beta}\bar{v}' \cos\bar{\beta}\bar{w}') + \bar{\Omega}^2\bar{m}\bar{e}\sin\bar{\beta}\bar{v}$$

$$= \bar{\omega}^2\bar{m}\bar{k}_m^2\bar{\phi} - \bar{\omega}^2\bar{m}\bar{e}\sin\bar{\beta}\bar{v} + \bar{\omega}^2\bar{m}\bar{e}\cos\bar{\beta}\bar{w} \tag{66}$$

where $\bar{k}_{m_1}^2 = k_{m_1}^2/R^2$; $\bar{k}_{m_2}^2 = k_{m_2}^2/R^2$;

$\bar{k}_m^2 = k_m^2/R^2$; $\bar{e} = e/R$; $(\ )' = \frac{d}{dx}$

$$\frac{d^2\bar{M}_y}{d\bar{x}^2} - (\bar{T}\,\frac{d\bar{w}}{d\bar{x}})' - (\bar{\Omega}^2\bar{m}\bar{e}\bar{x}\cos\bar{\beta}\bar{\phi})'$$

$$= \bar{\omega}^2\bar{m}\bar{w} + \bar{\omega}^2\bar{m}\bar{e}\cos\bar{\beta}\bar{\phi} \tag{67}$$

where $\bar{T} = \bar{\Omega}^2 \int_{\bar{x}}^1 \bar{m}\bar{x}d\bar{x}$

$$\frac{d^2\bar{M}_z}{d\bar{x}^2} - (\bar{T}\frac{d\bar{v}}{d\bar{x}})' + (\bar{\Omega}^2\bar{m}\bar{e}\bar{x}\sin\beta\bar{\phi})'$$

$$+ \bar{\Omega}^2\bar{m}\bar{e}\sin\bar{\beta}\bar{\phi} - \bar{\Omega}^2\bar{m}\bar{v}$$

$$= \bar{\omega}^2\bar{m}\bar{v} - \bar{\omega}^2\bar{m}\bar{e}\sin\beta\bar{\phi} \qquad (68)$$

Nonuniform properties are allowed in the computer program and maximum number of stations allowed for inputting the properties are 11 for each load path member and 51 for the blade. The linear variation of properties is assumed for the intermediate locations.

7.2 Assumed Functions

The selected basic functions are given by Eqs. (36) and (37). These functions are nondimensionalized and differentiated with respect to $\bar{x}$ as shown below.

1. Bending (Both Flapwise and Chordwise)

   a) Deflections and Derivatives:

$$\alpha_r(\bar{x}) = \bar{e}_r(\sin\bar{\alpha}_r\bar{x} - \sinh\bar{\alpha}_r\bar{x}) + \bar{g}_r(\cos\bar{\alpha}_r\bar{x} - \cosh\bar{\alpha}_r\bar{x}) \qquad (69)$$

$$\alpha_r'(\bar{x}) = \bar{\alpha}_r\bar{e}_r(\cos\bar{\alpha}_r\bar{x} - \cosh\bar{\alpha}_r\bar{x}) + \bar{\alpha}_r\bar{g}_r(-\sin\bar{\alpha}_r\bar{x} - \sinh\bar{\alpha}_r\bar{x}) \qquad (70)$$

b) Moments and Derivatives:

$$\bar{M}_r(\bar{x}) = \bar{\alpha}_r^2 \bar{e}_r(-\sin \bar{\alpha}_r \bar{x} - \sinh \bar{\alpha}_r \bar{x}) +$$

$$\bar{\alpha}_r^2 \bar{g}_r(-\cos \bar{\alpha}_r \bar{x} - \cosh \bar{\alpha}_r \bar{x}) \quad (71)$$

$$\bar{M}_r(\bar{x}) = \bar{\alpha}_r^3 \bar{e}_r(-\cos \bar{\alpha}_r \bar{x} - \cosh \bar{\alpha}_r \bar{x}) +$$

$$\bar{\alpha}_r^3 \bar{g}_r(\sin \bar{\alpha}_r \bar{x} - \sinh \bar{\alpha}_r \bar{x}) \quad (72)$$

where $\bar{\alpha}_r = \alpha_r$, $\bar{g}_r = g_r$, $\bar{e}_r = e_r$

and are defined in Eq. (36)..

2. Twisting/Stretching

a) Deflections

$$\bar{\alpha}_r(x) = \sin(\bar{\delta}_r \bar{x}) \quad (73)$$

b) Moment/Force

$$\bar{M}_r(\bar{x}) = \bar{\delta}_r \cos(\bar{\delta}_r \bar{x}) \quad (74)$$

where $\delta_r = (2r-1)\,\frac{\pi}{2}$

The clevis compatibility functions are completed assuming $\ell_1 = 0.9$ and the coefficients for the bending deflection functions are stored in the triple subscripted variable A(I,J,K) with subroutine titled GFCTS. The notation for the arguments is as follows:

I = Number of selected functions

J = 1 $\rightarrow$ a_0

J = 2 $\rightarrow$ a_1

J = 3 $\rightarrow$ a_2

J = 4 $\rightarrow$ a_3

where a_0, a_1, a_2, a_3 are coefficients of the cubic polynomial

K = 1 $\rightarrow$ Flapwise bending, Load path 1

K = 2 $\rightarrow$ Chordwise bending, Load path 1

K = 3 $\rightarrow$ Flapwise bending, Load path 2

K = 4 $\rightarrow$ Chordwise bending, Load path 2

The coefficients for the twisting/stretching deflection functions are stored in the triple subscripted variable B(I,J,K) in the subroutine GFCTS. The notation for the arguments is as follows:

I = Number of selected function

J = 1 $\rightarrow$ b_0

J = 2 $\rightarrow$ b_1

where b_0, b_1 are coefficients of the linear function

K = 1 $\rightarrow$ Twisting, Load path 1

K = 2 $\rightarrow$ Stretching, Load path 1

K = 3 $\rightarrow$ Twisting, Load path 2

K = 4 $\rightarrow$ Stretching, Load path 2

7.3 Galerkin's Equations

The Galerkin's equations to solve the eigenvalues problem are given by Eqs. (28) to (31). To avoid the numerical differentiation of several products of the nonuniform blade properties, the above equations are rewritten as shown below.

$$- \sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} \int_{o}^{\ell} (b_{3j}M'_{yj_i} - a_{3j}D113_i w'_{j_i} + a_{4j}D12_i \phi'_{j_i}) w'_{k_i} dk \right\}$$

$$- \sum_{j=1}^{N} \int_{\ell}^{R} (b_{3j}M'_{yj} - a_{3j}D213w'_j + a_{4j}D22\phi'_j) w'_k dx$$

$$+ < \sum_{j=1}^{N} \left[\left\{ \sum_{i=1}^{n} (b_{3j}M_{yj_i} - h_{z_i} b_{ij} V_{xj_i}) \right\} - b_{3j}M_{yj} \right] w'_k >_{x=\ell}$$

$$= \omega^2 \sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} \int_{o}^{\ell} (D11_i a_{3j} w_{j_i} + a_{4j}D13_i \phi_{j_i}) w_{k_i} dx \right\}$$

$$+ \omega^2 \left\{ \sum_{j=1}^{N} \int_{\ell}^{R} (D21a_{3j}w_j + a_{4j}D23\phi_j) w_k dx \right\} \qquad (75)$$

$$- \sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} \int_{o}^{\ell} (b_{2j}M'_{zj_i} - a_{2j}D113_i v'_{j_i} + a_{4j}D114_i \phi_{j_i}) v'_{k_i} dx \right\}$$

$$+ \sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} \int_{o}^{\ell} (a_{4j}\bar{\Omega}^2 D15_i \phi_{j_i} - a_{2j}\bar{\Omega}^2 D11_i v_{j_i}) v_{k_i} dx \right\}$$

$$- \sum_{j=1}^{N} \left\{ \int_{\ell}^{R} (b_{2j} M'_{zj} - a_{2j} D213 v'_j + a_{4j} D214 \phi_j) v'_k dx \right\}$$

$$+ \sum_{j=1}^{N} \left\{ \int_{\ell}^{R} (a_{4j} \bar{\Omega}^2 D25 \phi_j - a_{2j} \bar{\Omega}^2 D21_i v_j) v_k dx \right\}$$

$$+ < \sum_{j=1}^{N} \left[\left\{ \sum_{i=1}^{n} (b_{2j} M_{zj_i} - h_{y_i} b_{1j} V_{xj_i}) \right\} - b_{2j} M_{zj} \right] v'_k >_{x=\ell}$$

$$= \omega^2 \sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} \int_{o}^{\ell} (D11_i a_{2j} v_{j_i} - a_{4j} D15_i \phi_{j_i}) v_{k_i} dx \right\}$$

$$+ \omega^2 \sum_{j=1}^{N} \left\{ \int_{\ell}^{R} (D21 a_{2j} v_j - a_{4j} D25 \phi_j) v_k dx \right\} \tag{76}$$

$$\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} \int_{o}^{\ell} (-b_{4j} M'_{xj_i} - D12_i a_{2j} v'_{j_i} + D14_i a_{3j} w'_{j_i} \right.$$

$$\left. + a_{2j} \bar{\Omega}^2 D15_i v_{j_i} + a_{4j} D16_i \phi_{j_i}) \phi_{k_i} dx \right\}$$

$$+ \sum_{j=1}^{N} \left\{ \int_{\ell}^{R} (-b_{4j} M'_{xj} - D212 a_{2j} v'_j + D24 a_{3j} w'_j \right.$$

$$\left. + a_{2j} \bar{\Omega}^2 D25_i v_j + a_{4j} D26 \phi_j) \phi_k dx \right\}$$

$$+ < \sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} [b_{4j}M_{xj_i} + h_{y_i}(-b_{3j}M'_{yj_i} + a_{3j}D113_i w'_{j_i} \right.$$

$$+ a_{4j}D12_i \phi_{j_i}) - h_{z_i}(-b_{2j}M'_{zj_i} + a_{2j}D113_i v'_{j_i}$$

$$\left. - a_{4j}D14_i \phi_{j_i})] - b_{4j}M_{xj} \right\} \phi_k >_{x=\ell}$$

$$= \omega^2 \sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} \int_o^{\ell} (D17_i a_{4j}\phi_{j_i} + D13_i a_{3j} w_{j_i} \right.$$

$$\left. - D15_i a_{2j} v_{j_i})\phi_{k_i} dx \right\}$$

$$+ \omega^2 \sum_{j=1}^{N} \left\{ \int_{\ell}^{R} (D27a4_j \phi_j + D23a_{3j} w_j \right.$$

$$\left. - D25a_{2j} v_j)\phi_k dx \right\} \tag{77}$$

$$\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} \int_o^{\ell} (-b_{1j}V'_{xj_i} - a_{1j}\bar{\Omega}^2 D11_i u_{j_i})u_{k_i} dx \right\}$$

$$+ \sum_{j=1}^{N} \left\{ \int_{\ell}^{R} (-b_{1j}V'_{xj} - a_{1j}\bar{\Omega}^2 D21u_j)u_k dx \right\}$$

$$+ \left[\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} (b_{1j} v_{xj_i}) - b_{1j} V_{xj} \right\} u_k \right]_{x=\ell}$$

$$= \omega^2 \sum_{j=1}^{N} \left\{ \int_o^{\ell} D11_i a_{1j} u_{j_i} u_{k_i} dx \right\}$$

$$+ \omega^2 \sum_{j=1}^{N} \left\{ \int_{\ell}^{R} D21 a_{1j} u_j u_k dx \right\} \tag{78}$$

$$\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} \int_o^{\ell} (a_{1j} u_{j_i} - D18_i b_{1j} V_{xj_i}) V_{xk_i} dx \right\}$$

$$+ \sum_{j=1}^{N} \left\{ \int_{\ell}^{R} (a_{1j} u_j - D28 b_{1j} V_{xj}) V_{xk} dx \right\} = 0 \tag{79}$$

$$\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} \int_o^{\ell} (a_{4j} \phi_{j_i} - D19_i b_{4j} M_{xj_i}) M_{xk_i} dx \right\}$$

$$+ \sum_{j=1}^{N} \left\{ \int_{\ell}^{R} (a_{4j} \phi_j - D29 b_{4j} M_{xj}) M_{xk} dx \right\} = 0 \tag{80}$$

$$\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} \int_{o}^{\ell} (a_{2j} v''_{j_i} + D111_i b_{3j} M_{yj_i} - D110_i b_{2j} M_{zj_i}) M_{zk_i} dx \right\}$$

$$+ \sum_{j=1}^{N} \left\{ \int_{\ell}^{R} (a_{2j} v''_{j} + D211 b_{3j} M_{yj} - D210 b_{2j} M_{zj}) M_{zk} \right\} = 0 \qquad (81)$$

$$\sum_{j=1}^{N} \left\{ \sum_{i=1}^{n} \int_{\ell}^{R} (a_{3j} w''_{j_i} - D112_i b_{3j} M_{yj_i} + D111_i b_{2j} M_{zj_i}) M_{yk_i} dx \right\}$$

$$+ \sum_{j=1}^{N} \left\{ \int_{\ell}^{R} (a_{3j} w_{j} - D212 b_{3j} M_{yj} + D211 b_{2j} M_{zj}) M_{yk} dx \right\} = 0 \qquad (82)$$

where

$Di1 = \bar{m}$

$Di2 = \bar{\Omega}^2 \bar{m}\bar{e}\bar{x}\cos\bar{\beta}$

$Di3 = \bar{m}\bar{e}\cos\bar{\beta}$

$Di4 = {}^{-2}\bar{m}\bar{e}\bar{x}\sin\bar{\beta}$

$Di5 = \bar{m}\bar{e}\sin\bar{\beta}$

$$Di6 = \bar{\Omega}^2\bar{m}(\bar{k}^2_{m_2} - \bar{k}^2_{m_1})\cos 2\bar{\beta}$$

$$Di7 = \bar{m}\bar{k}^2_m$$

$$Di8 = 1/\overline{EA}$$

$$Di9 = 1/\overline{GJ}$$

$$Di10 = \bar{C}_{11}/\overline{EI}_1\overline{EI}_2$$

$$Di11 = \bar{C}_{12}/\overline{EI}_1\overline{EI}_2$$

$$Di12 = \bar{C}_2/\overline{EI}_1\overline{EI}_2$$

$$Di13 = \bar{T}$$

$i = 1 \rightarrow$ Load Path Properties

$i = 2 \rightarrow$ Blade Properties

7.4 Eigenvalue Problem

Eqs. (75) to (82) can be written into a set of matrix equations of the following form for k = 1 to N.

$$\underset{NX4N}{[A_1]} \underset{4NX1}{\begin{Bmatrix} a_3 \\ a_4 \\ b_1 \\ b_3 \end{Bmatrix}} = \omega^2 \underset{NX2N}{[B_1]} \underset{2NX1}{\begin{Bmatrix} a_3 \\ a_4 \end{Bmatrix}} \tag{83}$$

$$\underset{NX4N}{[A_2]} \underset{4NX1}{\begin{Bmatrix} a_2 \\ a_4 \\ b_1 \\ b_2 \end{Bmatrix}} = \omega^2 \underset{NX2N}{[B_2]} \underset{2NX1}{\begin{Bmatrix} a_2 \\ a_4 \end{Bmatrix}} \tag{84}$$

$$\underset{NX6N}{[A_3]} \underset{6NX1}{\begin{Bmatrix} a_2 \\ a_3 \\ a_4 \\ b_2 \\ b_3 \\ b_4 \end{Bmatrix}} = \omega^2 \underset{NX3N}{[B_3]} \underset{3NX1}{\begin{Bmatrix} a_2 \\ a_3 \\ a_4 \end{Bmatrix}} \tag{85}$$

$$\underset{NX2N}{[A_4]} \begin{Bmatrix} a_1 \\ b_1 \end{Bmatrix} = \omega^2 \underset{NXN}{[B_4]} \begin{Bmatrix} a_1 \end{Bmatrix} \qquad (86)$$

$$\underset{NX2N}{[A_5]} \begin{Bmatrix} a_1 \\ b_1 \end{Bmatrix} = \begin{Bmatrix} 0 \end{Bmatrix} \qquad (87)$$

$$[A_6] \begin{Bmatrix} a_4 \\ b_4 \end{Bmatrix} = \begin{Bmatrix} 0 \end{Bmatrix} \qquad (88)$$

$$\underset{NX3N}{[A_7]} \begin{Bmatrix} a_2 \\ b_2 \\ b_3 \end{Bmatrix} = \begin{Bmatrix} 0 \end{Bmatrix} \qquad (89)$$

$$\underset{NX3N}{[A_8]} \begin{Bmatrix} a_3 \\ b_2 \\ b_3 \end{Bmatrix} = \begin{Bmatrix} 0 \end{Bmatrix} \qquad (90)$$

The above set of equations correspond to Eqs. (75) to (82) respectively. Eq. (83) can be partitioned into the following form:

$$\left|[A_{11}] \;\vdots\; A_{12}]\right| \begin{Bmatrix} a_3 \\ \hdashline a_4 \end{Bmatrix} + \left| [A_{13}] \;\vdots\; [A_{14}] \right| \begin{Bmatrix} b_1 \\ \hdashline b_3 \end{Bmatrix} \tag{91}$$

$$= \omega^2 \left|[B_{11}] \;\vdots\; [B_{12}]\right| \begin{Bmatrix} a_3 \\ a_4 \end{Bmatrix}$$

By employing a similar partition scheme for Eqs. (84) to (86), the four equations (83 to 86) can be combined into the following form

$$[E_1] \lfloor a_1 \quad a_2 \quad a_3 \quad a_4 \rfloor^T + [E_2] \lfloor b_1 \quad b_2 \quad b_3 \quad b_4 \rfloor^T$$

$$= \omega^2 \, [E_3] \, \lfloor a_1 \quad a_2 \quad a_3 \quad a_4 \rfloor^T \tag{92}$$

where

$$\underset{4N\text{X}4N}{[E_1]} = \begin{bmatrix} 0 & 0 & A_{11} & A_{12} \\ 0 & A_{21} & 0 & A_{22} \\ 0 & A_{31} & A_{32} & A_{33} \\ A_{41} & 0 & 0 & 0 \end{bmatrix}$$

$$\underset{4N\text{X}4N}{[E_2]} \quad \begin{bmatrix} A_{13} & 0 & A_{14} & 0 \\ A_{23} & A_{24} & 0 & 0 \\ 0 & A_{34} & A_{35} & A_{36} \\ A_{42} & 0 & 0 & 0 \end{bmatrix}$$

$$[E_1] = \begin{bmatrix} 0 & 0 & B_{11} & B_{12} \\ 0 & B_{21} & 0 & B_{22} \\ 0 & B_{31} & B_{32} & B_{33} \\ B_{41} & 0 & 0 & 0 \end{bmatrix}$$

Eqs. (87) to (90) can be combined into a matrix equation of the following form using a similar partition scheme employed in Eq. (91).

$$\lfloor b_1 \quad b_2 \quad b_3 \quad b_4 \rfloor^T = [E_4] \lfloor a_1 \quad a_2 \quad a_3 \quad a_4 \rfloor^T \tag{93}$$

where

$$E_4 = \begin{bmatrix} d_1 & 0 & 0 & 0 \\ 0 & c_{11} & c_{12} & 0 \\ 0 & c_{12} & c_{22} & 0 \\ 0 & 0 & 0 & d_2 \end{bmatrix}$$

$$[d_1] = -[A_{52}]^{-1} [A_{51}]$$

$$[d_2] = -[A_{62}]^{-1} [A_{61}]$$

$$\begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} = \begin{bmatrix} A_{72} & A_{73} \\ A_{82} & A_{83} \end{bmatrix}^{-1} \begin{bmatrix} A_{71} & 0 \\ 0 & A_{81} \end{bmatrix}$$

Substituting Eq. (3) into Eq. (92) yields

$$([D] - \omega^2[I]) \lfloor a_1 \quad a_2 \quad a_3 \quad a_4 \rfloor^T = \{0\} \tag{94}$$

where

$$[D] = [E_3]^{-1} ([E_1] + [E_2] [E_4])$$

REFERENCES

1. Fenaughty, R. R. and Noehren, W. L., "Composite Bearingless Tail Rotor for UTTAS," Journal of the American Helicopter Society, vol. 22, (3), July 1977.

2. Staley, J. A., Gabel, R. and MacDonald, H. I., "Full Scale Ground and Air Resonance Testing of the Army-Boeing Vertol Bearingless Main Rotor," Paper No. 79-23, American Helicopter Society 35th Annual National Forum, Washington, D.C., May 1979.

3. Dixon, P. G. C. and Bishop, H. E., "The Bearingless Main Rotor," Journal of the American Helicopter Society, vol. 25 (3), July 1980.

4. Bousman, W. G., Ormiston, R. A. and Mirick, P. H., "Design Considerations for Bearingless Main Rotor Hubs," American Helicopter Society 39th Annual National Forum, St. Louis, MO, May 1983.

5. Johnson, Wayne, "A Comprehensive Analytical Model of Rotorcraft Aerodynamics and Dynamics," NASA TM 81182, June 1980.

6. Lang, K. W. and Nemat Nasser, S., "An Approach for Estimating Vibration Characteristics of Nonuniform Rotor Blades," AIAA Journal, vol. 17, No. 9, Sep. 1979.

7. Houbolt, J. C. and Brooks, G. W., "Differential Equations of Motion for Combined Flapwise Bending, Chordwise Bending, and Torsion of Twisted Nonuniform Rotor Blades," NACA Report 1346, 1958.

8. Dixon, P. G. C., "Design, Development and Flight Demonstration of the Loads and Stability Characteristics of Bearingless Main Rotor," USA AVRADCOM-TR-80-D3, 1980.

9. Murthy, V. R., and Joshi, A. M., "Free Vibration Characteristics of Multiple-Load-Path Blades by the Transfer Matrix Method," Journal of the American Helicopter Society, vol. 31, (4), pp. 43-50, Oct. 1986.

```
      PROGRAM MOGALK
C     ******************************************************************
C     DYNAMIC ANALYSIS OF MULTIPLE-LOAD-PATH BEARINGLESS ROTOR BLADES
C     MODIFIED GALERKIN'S METHOD COUPLED FLAPWISE BENDING, CHORDWISE
C     BENDING, TWISTING, AND STRECHING MOTION OF PRETWISTED NONUNIFORM
C     ROTATING BLADES.
C     ******************************************************************
      REAL MASS,MASS1(2,11),MASS2(51),KM1S1(2,11),KM1S2(51),KM2S2(51),
     1KM2S1(2,11)
      DIMENSION STA1(11),STA2(51),sta(51),EIY1(2,11),EIY2(51),EIZ1(2,
     111),EIZ2(51),EA1(2,11),EA2(51),BETA1(2,11),BETA2(51),GJ1(2,11),
     2GJ2(51),E1(2,11),E2(51),FD(2,2),D21(51),D22(51),D23(51),D24(51),
     3D25(51),D26(51),D27(51),D28(51),D29(51),D210(51),D211(51),
     4D212(51),D213(51),D11(2,11),D12(2,11),D13(2,11),D14(2,11),
     5D15(2,11),D16(2,11),D17(2,11),D18(2,11),D19(2,11),D110(2,11),
     6D111(2,11),D112(2,11),D113(2,11),BB(3),AT(2),DS(20,20),
     7A(5,4,4),B(5,2,4),D214(51),FRE(51)
      COMPLEX DSC(20,20),LAMDA(20),VECT(20,20),HL(20,20),H1(20,20),
     1CNT(20),COLM(20)
      INTEGER INTH(20,2)
      COMMON/XM1/NPATH
      COMMON/XM2/SPAN1,SPAN2,SPAN3,FD,OMEGAN
      COMMON/XM3/D11,D12,D13,D14,D15,D16,D17,D18,D19,D110,D111,D112,
     1D113,D21,D22,D23,D24,D25,D26,D27,D28,D29,D210,D211,D212,D213
      COMMON/XM4/STA,NS
      COMMON/XM5/HS1,HS2
      COMMON/XGFCT1/A,B
      COMMON/XGPR1/DS
```

```
C     ---------------------------------------------------------------
C     THIS SECTION READS THE DATA OF THE SYSTEM
C     ---------------------------------------------------------------
C
      READ(20,100) NPATH,NS1,NS2,MODES,ITT,NV
      READ(20,105) SPAN1,SPAN2,OMEGA
      READ(20,105) (STA1(I),I=1,NS1)
      READ(20,105) (STA2(I),I=1,NS2)
      DO 400 I=1,NPATH
        READ(20,105) (MASS1(I,J),J=1,NS1)
        READ(20,105) (EIY1(I,J),J=1,NS1)
        READ(20,105) (EIZ1(I,J),J=1,NS1)
        READ(20,105) (GJ1(I,J),J=1,NS1)
        READ(20,105) (E1(I,J),J=1,NS1)
        READ(20,105) (BETA1(I,J),J=1,NS1)
        READ(20,105) (KM1S1(I,J),J=1,NS1)
        READ(20,105) (KM2S1(I,J),J=1,NS1)
        READ(20,105) (EA1(I,J),J=1,NS1)
  400 CONTINUE
      READ(20,105) (MASS2(I),I=1,NS2)
      READ(20,105) (EIY2(I),I=1,NS2)
      READ(20,105) (EIZ2(I),I=1,NS2)
      READ(20,105) (GJ2(I),I=1,NS2)
      READ(20,105) (E2(I),I=1,NS2)
      READ(20,105) (BETA2(I),I=1,NS2)
      READ(20,105) (KM1S2(I),I=1,NS2)
      READ(20,105) (KM2S2(I),I=1,NS2)
      READ(20,105) (EA2(I),I=1,NS2)
```

```
      DO 401 I=1,NPATH
      DO 401 J=1,2
  401 FD(I,J)=0.0
      IF (NPATH.GT.1) READ(20,105) ((FD(I,J),J=1,2),I=1,NPATH)
C     ------------------------------------------------------------------
C     THIS SECTION PRINTS THE DATA OF THE SYSTEM
C     ------------------------------------------------------------------
C
      SPAN=SPAN1*SPAN2
      WRITE(22,200)
      WRITE(22,205)
      WRITE(22,347) MODES
      WRITE(22,371) NV
      WRITE(22,225) SPAN
      WRITE(22,230) SPAN1
      WRITE(22,235) SPAN2
      WRITE(22,240) OMEGA
      WRITE(22,245) NS1
      WRITE(22,250) NS2
      WRITE(22,315) NPATH
      WRITE(22,316) ITT
      IF (OMEGA.LE.0) ITT=1
      DO 405 I=1,NPATH
         IF (I.EQ.1) WRITE(22,255)
         IF (I.EQ.2) WRITE(22,260)
         WRITE(22,270)
         WRITE(22,275) (STA1(J),J=1,NS1)
         WRITE(22,280)
```

```
        WRITE(22,275)  (MASS1(I,J),J=1,NS1)
        WRITE(22,285)
        WRITE(22,275)  (EIY1(I,J),J=1,NS1)
        WRITE(22,350)
        WRITE(22,275)  (EIZ1(I,J),J=1,NS1)
        WRITE(22,352)
        WRITE(22,275)  (GJ1(I,J),J=1,NS1)
        WRITE(22,354)
        WRITE(22,275)  (E1(I,J),J=1,NS1)
        WRITE(22,356)
        WRITE(22,275)  (BETA1(I,J),J=1,NS1)
        WRITE(22,358)
        WRITE(22,275)  (KM1S1(I,J),J=1,NS1)
        WRITE(22,360)
        WRITE(22,275)  (KM2S1(I,J),J=1,NS1)
        WRITE(22,286)
        WRITE(22,275)  (EA1(I,J),J=1,NS1)
        IF (NPATH.GT.1) WRITE(22,321) I,FD(I,1),FD(I,2)
405 CONTINUE
    WRITE(22,290)
    WRITE(22,270)
    WRITE(22,275)  (STA2(I),I=1,NS2)
    WRITE(22,280)
    WRITE(22,275)  (MASS2(I),I=1,NS2)
    WRITE(22,285)
    WRITE(22,275)  (EIY2(I),I=1,NS2)
    WRITE(22,350)
    WRITE(22,275)  (EIZ2(I),I=1,NS2)
```

```
      WRITE(22,352)
      WRITE(22,275) (GJ2(I),I=1,NS2)
      WRITE(22,354)
      WRITE(22,275) (E2(I),I=1,NS2)
      WRITE(22,356)
      WRITE(22,275) (BETA2(I),I=1,NS2)
      WRITE(22,358)
      WRITE(22,275) (KM1S2(I),I=1,NS2)
      WRITE(22,360)
      WRITE(22,275) (KM2S2(I),I=1,NS2)
      WRITE(22,286)
      WRITE(22,275) (EA2(I),I=1,NS2)
      NS=NS1
      DO 406 J=1,NS
      STA(J)=STA1(J)
  406 CONTINUE
      HH=SPAN1/10.0
      DO 418 I=1,NPATH
         DO 410 J=1,NS1
            D21(J)=MASS1(I,J)
            D22(J)=EIY1(I,J)
            D23(J)=EIZ1(I,J)
            D24(J)=GJ1(I,J)
            D25(J)=E1(I,J)
            D26(J)=BETA1(I,J)
            D27(J)=KM1S1(I,J)
            D28(J)=KM2S1(I,J)
            D29(J)=EA1(I,J)
```

```
410 CONTINUE
    CALL INTPOL(11,D21,HH)
    CALL INTPOL(11,D22,HH)
    CALL INTPOL(11,D23,HH)
    CALL INTPOL(11,D24,HH)
    CALL INTPOL(11,D25,HH)
    CALL INTPOL(11,D26,HH)
    CALL INTPOL(11,D27,HH)
    CALL INTPOL(11,D28,HH)
    CALL INTPOL(11,D29,HH)
    DO 415 J=1,11
      MASS1(I,J)=D21(J)
      EIY1(I,J)=D22(J)
      EIZ1(I,J)=D23(J)
      GJ1(I,J)=D24(J)
      E1(I,J)=D25(J)
      BETA1(I,J)=D26(J)
      KM1S1(I,J)=D27(J)
      KM2S1(I,J)=D28(J)
      EA1(I,J)=D29(J)
415 CONTINUE
    IF (I.EQ.1) WRITE(22,255)
    IF (I.EQ.2) WRITE(22,260)
    WRITE(22,280)
    WRITE(22,295)
    WRITE(22,275) (D21(J),J=1,11)
    WRITE(22,285)
    WRITE(22,275) (D22(J),J=1,11)
```

```
      WRITE(22,350)
      WRITE(22,275) (D23(J),J=1,11)
      WRITE(22,352)
      WRITE(22,275) (D24(J),J=1,11)
      WRITE(22,354)
      WRITE(22,275) (D25(J),J=1,11)
      WRITE(22,356)
      WRITE(22,275) (D26(J),J=1,11)
      WRITE(22,358)
      WRITE(22,275) (D27(J),J=1,11)
      WRITE(22,360)
      WRITE(22,275) (D28(J),J=1,11)
      WRITE(22,286)
      WRITE(22,275) (D29(J),J=1,11)
  418 CONTINUE
      NS=NS2
      DO 419 J=1,NS
      STA(J)=STA2(J)
  419 continue
      HH=SPAN2/100.0
      CALL INTPOL(51,MASS2,HH)
      CALL INTPOL(51,EIY2,HH)
      CALL INTPOL(51,EIZ2,HH)
      CALL INTPOL(51,GJ2,HH)
      CALL INTPOL(51,E2,HH)
      CALL INTPOL(51,BETA2,HH)
      CALL INTPOL(51,KM1S2,HH)
      CALL INTPOL(51,KM2S2,HH)
```

```
      CALL INTPOL(51,EA2,HH)
      CALL INTPOL(51,CH,HH)
      CALL INTPOL(51,QE,HH)
      WRITE(22,300)
      WRITE(22,280)
      WRITE(22,275) (MASS2(J),J=1,51)
      WRITE(22,285)
      WRITE(22,275) (EIY2(J),J=1,51)
      WRITE(22,350)
      WRITE(22,275) (EIZ2(J),J=1,51)
      WRITE(22,352)
      WRITE(22,275) (GJ2(J),J=1,51)
      WRITE(22,354)
      WRITE(22,275) (E2(J),J=1,51)
      WRITE(22,356)
      WRITE(22,275) (BETA2(J),J=1,51)
      WRITE(22,358)
      WRITE(22,275) (KM1S2(J),J=1,51)
      WRITE(22,360)
      WRITE(22,275) (KM2S2(J),J=1,51)
      WRITE(22,286)
      WRITE(22,275) (EA2(J),J=1,51)
      WRITE(22,205)
C     ----------------------------------------------------------------
C     THIS SECTION NONDIMENSIONALIZED THE DATA
C     ----------------------------------------------------------------
C
      PI=4.0*ATAN(1.0)
```

```
      OMEGA=OMEGA*PI/30.0
      OMEGAS=OMEGA*OMEGA
      SPANS=SPAN*SPAN
      EIY=EIY2(1)
      MASS=MASS2(1)
      FACT=SQRT(SPANS*SPANS*MASS/EIY)
      OMEGAN=OMEGAS*FACT*FACT
      FA=SPANS/EIY
      SPAN1=SPAN1/SPAN
      SPAN2=SPAN2/SPAN
      HS1=SPAN1/10.0
      HS2=SPAN2/50.0
      SPAN3=0.7*SPAN1
      DO 420 I=1,NPATH
      DO 420 J=1,11
      BETA1(I,J)=BETA1(I,J)*PI/180.0
      KM1S1(I,J)=KM1S1(I,J)/(MASS1(I,J)*SPANS)
      KM2S1(I,J)=(KM2S1(I,J)/MASS1(I,J)+E1(I,J)*E1(I,J))/SPANS
      E1(I,J)=E1(I,J)/SPAN
      MASS1(I,J)=MASS1(I,J)/SPAN
      EIY1(I,J)=EIY1(I,J)/EIY
      EIZ1(I,J)=EIZ1(I,J)/EIY
      GJ1(I,J)=GJ1(I,J)/EIY
      EA1(I,J)=EA1(I,J)*FA
  420 CONTINUE
      DO 4201 J=1,51
      BETA2(J)=(BETA2(J))*PI/180.0
      KM1S2(J)=KM1S2(J)/(MASS2(J)*SPANS)
```

```
      KM2S2(J)=(KM2S2(J)/MASS2(J)+E2(J)*E2(J))/SPANS
      E2(J)=E2(J)/SPAN
      EIY2(J)=EIY2(J)/EIY
      EIZ2(J)=EIZ2(J)/EIY
      GJ2(J)=GJ2(J)/EIY
      EA2(J)=FA*EA2(J)
 4201 CONTINUE
C     ------------------------------------------------------------------
C     THIS SECTION CALCULATES THE COEFFICIENTS IN THE GALERKIN'S EQUATIONS
C     ------------------------------------------------------------------
C
      X=0.0
      DO 430 I=1,NPATH
      DO 430 J=1,11
      TC1=OMEGA*MASS1(I,J)
      C=COS(BETA1(I,J))
      S=SIN(BETA1(I,J))
      D11(I,J)=MASS1(I,J)
      D12(I,J)=TC1*E1(I,J)*X
      D14(I,J)=D12(I,J)*S
      D12(I,J)=D12(I,J)*C
      D13(I,J)=MASS1(I,J)*E1(I,J)
      D15(I,J)=D13(I,J)*S
      D13(I,J)=D13(I,J)*C
      D16(I,J)=TC1*(KM2S1(I,J)-KM1S1(I,J))*(C*C-S*S)
      D17(I,J)=MASS1(I,J)*(KM2S1(I,J)+KM1S1(I,J))
      D18(I,J)=1.0/EA1(I,J)
      D19(I,J)=1.0/GJ1(I,J)
```

```
      TC2=EIY1(I,J)*EIZ1(I,J)
      D110(I,J)=(EIY1(I,J)*C*C+EIZ1(I,J)*S*S)/TC2
      D111(I,J)=(EIZ1(I,J)-EIY1(I,J))*C*S/TC2
      D112(I,J)=(EIY1(I,J)*S*S+EIZ1(I,J)*C*C)/TC2
      X=X+HS1
  430 CONTINUE
      X=SPAN1
      DO 440 J=1,51
      TC1=OMEGAN*MASS2(J)
      C=COS(BETA2(J))
      S=SIN(BETA2(J))
      D21(J)=MASS2(J)
      D22(J)=TC1*E2(J)*X
      D24(J)=D22(J)*S
      D22(J)=D22(J)*C
      D23(J)=MASS2(J)*E2(J)
      D25(J)=D23(J)*S
      D23(J)=D23(J)*C
      D26(J)=TC1*(KM2S2(J)-KM1S2(J))*(C*C-S*S)
      D27(J)=MASS2(J)*(KM2S2(J)+KM1S2(J))
      D28(J)=1.0/EA2(J)
      D29(J)=1.0/GJ2(J)
      TC2=EIY2(J)*EIZ2(J)
      D210(J)=(EIY2(J)*C*C+EIZ2(J)*S*S)/TC2
      D211(J)=(EIZ2(J)-EIY2(J))*C*S/TC2
      D212(J)=(EIY2(J)*S*S+EIZ2(J)*C*C)/TC2
      X=X+HS2
  440 CONTINUE
```

```
      X=SPAN1
      H=HS2/24.0
      DO 450 I=1,51
      D214(I)=MASS2(I)*H*X
      X=X+HS2
  450 CONTINUE
      D213(51)=0.0
      D213(50)=(9.0*D214(51)+19.0*D214(50)-5.0*D214(49)+D214(48))
     1*OMEGAN
      DO 455 I=2,49
      J=51-I
      D213(J)=D213(J+1)+(-D214(J+2)+13.0*(D214(J+1)+D214(J))-
     1D214(J-1))*OMEGAN
  455 CONTINUE
      D213(1)=D213(2)+(D214(4)-5.0*D214(3)+19.0*D214(2)+9.0*D214(1))
     1*OMEGAN
      GO TO (460,465),NPATH
  460 BB(1)=D213(1)
      GO TO 480
  465 H=SPAN1/24.0
      DO 475 J=1,NPATH
      AT(J)=(9.0*D18(J,1)+19.0*D18(J,2)-5.0*D18(J,3)+D18(J,4))*H
      DO 470 K=1,9
      AT(J)=AT(J)+H*(-D18(J,K)+13.0*D18(J,K+1)+13.0*D18(J,K+2)
     1-D18(J,K+3))
  470 CONTINUE
      AT(J)=AT(J)+H*(D18(J,8)-5.0*D18(J,9)+19.0*D18(J,10)+9.0*D18(J,11))
  475 CONTINUE
```

```
      BB(1)=AT(1)*D213(1)/(AT(1)+AT(2))
      BB(2)=AT(2)*D213(1)/(AT(1)+AT(2))
  480 DO 495 JJ=1,NPATH
      X=0.0
      DO 485 I=1,11
      D214(I)=MASS1(JJ,I)*H*X
      X=X+HS1
  485 CONTINUE
      D113(JJ,11)=0.0
      D113(JJ,10)=(9.0*D214(11)+19.0*D214(10)-5.0*D214(9)+D214(8))
     1*OMEGAN
      DO 490 I=2,9
      J=11-I
      D113(JJ,J)=D113(JJ,J+1)+C-D214(J+2)+13.0*(D214(J+1)+D214(J)
     1-D214(J-1))*OMEGAN
  490 CONTINUE
      D113(JJ,1)=D113(JJ,2)+(D214(4)-5.0*D214(3)+19.0*D214(2)
     1+9.0*D214(1))*OMEGAN
  495 CONTINUE
      DO 500 I=1,NPATH
      DO 500 J=1,11
  500 D113(I,J)=D113(I,J)+BB(I)
      IF (NPATH.GT.1.AND.ITT.GT.1) CALL TITER(ITT)
C     ------------------------------------------------------------------
C     THIS SECTION CALCULATES GALERKIN'S FUNCTIONS AND WRITES THE
C     COEFFICIENTS OF THE CLEVIS COMPATIBILITY POLYNOMIALS
C     ------------------------------------------------------------------
C
```

```
      CALL GFCTS(MODES)
      N1=2*NPATH
      WRITE(22,361)
      DO 505 K=1,MODES
      WRITE(22,366)
      DO 505 I=1,N1,2
      IF (I.EQ.1) WRITE(22,362)
      IF (I.EQ.2) WRITE(22,363)
      WRITE(22,364)
      WRITE(22,275) (A(K,J,I),J=1,4)
      WRITE(22,365)
      WRITE(22,275) (A(K,J,I+1),J=1,4)
      WRITE(22,367)
      WRITE(22,275) (B(K,J,I),J=1,2)
      WRITE(22,368)
      WRITE(22,275) (B(K,J,I+1),J=1,2)
  505 CONTINUE
C     ------------------------------------------------------------------
C     THIS SECTION CALCULATES THE EIGENVALUES AND EIGENVECTORS
C     ------------------------------------------------------------------
C
      N4=4*NPATH
      DO 510 I=1,N4
      DO 510 J=1,N4
  510 DSC(I,J)=CMPLX(DS(I,J),0.0)
      INTH(1,1)=N4
      INTH(2,1)=NV
      CALL EECM(DSC,LAMDA,VECT,HL,H1,CNT,COLM,INTH)
```

```
      WRITE(22,200)
      WRITE(22,205)
      WRITE(22,369)
      WRITE(22,205)
      WRITE(22,275) (LAMDA(J),J=1,N4)
      DO 515 J=1,N4
  515 FRE(J)=SQRT(REAL(LAMDA(J)))
      WRITE(22,370)
      WRITE(22,275) (FRE(J),J=1,N4)
      IF (NV.EQ.0) GO TO 600
      WRITE(22,371)
      DO 520 I=1,N4
  520 WRITE(22,275) (VECT(I,J),J=1,NV)
      WRITE(22,372)
      DO 530 J=1,NV
      AMAX=REAL(VECT(I,J))
      DO 525 I=2,N4
  525 IF(ABS(AMAX).LT.ABS(REAL(VECT(I,J)))) AMAX=REAL(VECT(I,J))
      DO 530 I=1,N4
  530 VECT(I,J)=VECT(I,J)/AMAX
      DO 535 I=N4
  535 WRITE(22,275) (VECT(I,J),J=1,NV)
      WRITE(22,205)
C     -------------------------------------------------------------------
C     FORMATS
C     -------------------------------------------------------------------
C
```

```
100 FORMAT(5I5)
105 FORMAT(5E14.7)
200 FORMAT(1H1)
205 FORMAT(//2X,'****************************************',
   $'**************************')
225 FORMAT(//5X,'RADIUS OF THE ROTOR (INCHES)   =',E14.7)
230 FORMAT(//5X,'LENGTH OFF INBOARD SEGMENTS (IN)=',E14.7)
235 FORMAT(//5X,'LENGTH OF THE BLADE (INCHES)    =',E14.7)
240 FORMAT(//5X,'ROTATIONAL SPEED (RPM)          =',E14.7)
245 FORMAT(//5X,'NUMBER OF DATA POINTS FOR INBOARD SEGMENTS=',I5)
250 FORMAT(//5X,'NUMBER OF DATA POINTS FOR BLADE           =',I5)
255 FORMAT(//5X,'PROPERTIES OF THE FIRST LOAD PATH')
260 FORMAT(//5X,'PROPERTIES OF THE SECOND LOAD PATH')
270 FORMAT(//5X,'DATA POINT LOCATIONS IN INCHES')
275 FORMAT(4(6X,E14.7))
280 FORMAT(//5X,'MASS PER UNIT LENGTH (LB-SEC.**2/IN**2)')
285 FORMAT(//5X,'FLAPWISE BENDING STIFFNESS (LB-IN**2)')
286 FORMAT(//5X,'AXIAL STIFFNESS(LB) ')
290 FORMAT(//5X,'PROPERTIES OF THE BLADE')
295 FORMAT(//5X,'INTERPOLATED VALUES FOR INBOARD SEGMENTS, 11',
   $'EQUIDISTANT VALUES')
300 FORMAT(//5X,'INTERPOLATED VALUES FOR THE BLADE, 51',
   $'EQUIDISTANT VALUES')
315 FORMAT(//5X,'MUNBER OF LOAD PATHS')
316 FORMAT(//5X,'NUMBER OF FLNSION ITERATIONS =',I5)
321 FORMAT(//5X,'Y,Z DISTANCES BETWEEN LOADPATH NO:',I2,
   $/5X,' AND THE BLADE(IN) ARE=',F6.2,5X,F6.2)
347 FORMAT(//5X,'NUMBER OF MODES TO BE USED =',I5)
```

```
350 FORMAT(//5X,'CHORDWISE BENDING STIFFNESS(LB-IN**2)')
352 FORMAT(//5X,'TORSIONAL STIFFNESS(LB-IN**2)')
354 FORMAT(//5X,'DISTANCE BETWEEN MASS AND ELASTIC AXIS(IN)=')
356 FORMAT(//5X,'TWIST OF THE BLADE',/5X,'(DEGREES) =')
358 FORMAT(//5X,'MASS MOMENT OF INERTIA ABOUT THE CHORD(LB-SEC**2)')
360 FORMAT(//5X,'MASS MOMENT OF INERTIA ABOUT AN AXIS',
   $/5X,'PERPINDICULAR TO THE CHORD THROUGH THE CENTER OF',
   $/5X,'GRAVITY(LB-SEC**2)')
361 FORMAT(//5X,'THE COEFFICIENTS OF THE CLEVIS COMPATIBILITY
   $FUNCTIONS ARE')
362 FORMAT(//5X,'COEFFICENTS OF THE FIRST LOAD PATH')
363 FORMAT(//5X,'COEFFICENTS OF THE SECOND LOAD PATH')
364 FORMAT(//5X,'FLAPWISE BENDING')
365 FORMAT(//5X,'CHORDWISE BENDING')
366 FORMAT(//5X,'MODE NUMBER')
367 FORMAT(//5X,'TWISTING')
368 FORMAT(//5X,'STRETCHING')
369 FORMAT(//5X,'EIGENVALUES ARE AS FOLLOWS')
370 FORMAT(//5X,'NATURAL FREQUENCIES ARE AS FOLLOWS')

371 FORMAT(//5X,'EIGENVECTORS ARE AS FOLLOWS')
372 FORMAT(//5X,'NORMALIZED EIGENVECTORS')
C
600 STOP
    END
C   **********************************************************************
    SUBROUTINE EECM(A,LAMDA,VECT,HL,H,CNT,COLM,INTH)
C   **********************************************************************
```

```
C     ------------------------------------------------------------------
C     COMPUTES EIGENVALUES AND EIGENVECTORS OF A COMPLEX MATRIX
C     ------------------------------------------------------------------
C
      COMPLEX A(20,20),LAMDA(20),SHIFT(3),TEMP,TEMP1,TEMP2,SINN,COSS,
     $HL(60,60),CNT(20),VECT(20,20),COLM(20),H(60,60)
      LOGICAL TWICE,CON
      INTEGER INTH(20,2),R,RP1,RP2
      N=INTH(1,1)
      M=N
      NV=INTH(2,1)
      DO 1 I=1,M
      DO 1 J=1,M
    1 HL(J,I)=A(J,I)
      NCAL=N
      SHIFT(1)=(0.0,0.0)
      ICOUNT=0
      IF (N-2) 2,37,3
    2 LAMDA(1)=A(1,1)
      VECT(1,1)=(1.0,1.0)
      GO TO 63
C
C  REDUCE MATRIX TO HESSENBURG GORM
C
    3 NM2=N-2
      DO 14 R=1,NM2
      RP1=R+1
      RP2=R+2
```

```
      ABIG=0.0
      INTH(R,1)=RP1
      DO 4 I=RP1,N
      ABSSQ=REAL(HL(I,R))**2+AIMAG(HL(I,R))**2
      IF (ABSSQ.LE.ABIG) GO TO 4
      INTH(R,1)=I
      ABIG=ABSSQ
    4 CONTINUE
      INTER=INTH(R,1)
      IF (INTER.EQ.0.0) GO TO 14
      IF (INTER.EQ.RP1) GO TO 7
      DO 5 I=R,N
      TEMP=HL(RP1,I)
      HL(RP1,I)=HL(INTER,I)
    5 HL(INTER,I)=TEMP
      DO 6 I=1,N
      TEMP=HL(I,RP1)
      HL(I,RP1)=HL(I,INTER)
    6 HL(I,INTER)=TEMP
    7 DO 8 I=RP2,N
      COLM(I)=HL(I,R)/HL(RP1,R)
    8 HL(I,R)=COLM(I)
      DO 10 I=1,RP1
      TEMP=(0.0,0.0)
      DO 9 J=RP2,N
    9 TEMP=TEMP+HL(I,J)*COLM(J)
   10 HL(I,RP1)=HL(I,RP1)+TEMP
      DO 12 I=RP2,N
```

```
      TEMP=(0.0,0.0)
      DO 11 J=RP2,N
   11 TEMP=TEMP+HL(I,J)*COLM(J)
   12 HL(I,RP1)=HL(I,RP1)+TEMP-COLM(I)*HL(RP1,RP1)
      DO 13 I=RP2,N
      DO 13 J=RP2,N
   13 HL(I,J)=HL(I,J)-COLM(I)*HL(RP1,J)
   14 CONTINUE
C
C  CALCULATE EPSILON
C
      EPS=0.0
      DO 15 I=1,N
   15 EPS=EPS+CABS(HL(1,I))
      DO 17 I=2,N
      SUM=0.0
      IM1=I-1
      DO 16 J=IM1,N
   16 SUM=SUM+CABS(HL(I,J))
      IF (SUM.GT.EPS) EPS=SUM
   17 CONTINUE
      FN=N
      EPS=SQRT(FN)*EPS*1.0E-12
      IF(EPS.LT.1.0E-12) EPS=1.0E-12
      DO 18 I=1,N
      DO 18 J=1,N
   18 H(J,I)=HL(J,I)
   19 IF (N-2)20,37,21
```

```
   20 LAMDA(M)=HL(1,1)+SHIFT(1)
      GO TO 39
   21 MN1=M-N+1
      CON=.FALSE.
      IF (REAL(HL(N,N)).NE.0.0.OR.AIMAG(HL(N,N)).NE.0.0) CON=.TRUE.
      IF (.NOT.CON) GO TO 22
      HL(N,N)=HL(N,N)
      IF (ABS(REAL(HL(N,N-1)/HL(N,N)))+ABS(AIMAG(HL(N,N-1)/HL
     $(N,N)))-1.0E-12) 23,23,22
   22 IF(ABS(REAL(HL(N,N-1)))+ABS(AIMAG(HL(N,N-1))).GE.EPS) GO TO 24
   23 LAMDA(MN1)=HL(N,N)+SHIFT(1)
      ICOUNT=0
      N=N-1
      GO TO 19
C
C  DETERMINE SHIFT
C
   24 TEMP=HL(N,N)*HL(N-1,N-1)-HL(N,N-1)*HL(N-1,N)
      TEMP1=HL(N-1,N-1)+HL(N,N)
      SHIFT(2)=(TEMP1+CSQRT((TEMP1)**2-4.0*(TEMP)))/2.0
      IF (SHIFT(2).NE.(0.0,0.0)) GO TO 25
      SHIFT(3)=TEMP1
      GO TO 26
   25 SHIFT(3)=TEMP/SHIFT(2)
   26 IF (CABS(SHIFT(2)-HL(N,N)).LT.CABS(SHIFT(3)-HL(N,N))) GO TO 27
      INDEX=3
      GO TO 28
   27 INDEX=2
```

```
   28 IF (CABS(HL(N-1,N-2)).GE.EPS) GO TO 29
      LAMDA(MN1)=SHIFT(2)+SHIFT(1)
      LAMDA(MN1+1)=SHIFT(3)+SHIFT(1)
      ICOUNT=0
      N=N-2
      GO TO 19
   29 SHIFT(1)=SHIFT(1)+SHIFT(INDEX)
C
C  PERFORM GIVEN ROTATIONS, QR ITERATIONS
C
      DO 30 I=1,N
   30 HL(I,I)=HL(I,I)-SHIFT(INDEX)
      IF (ICOUNT.LE.40) GO TO 31
      NCAL=M-N
      GO TO 39
   31 NM1=N-1
      TEMP1=HL(1,1)
      TEMP2=HL(2,1)
      DO 36 R=1,NM1
      RP1=R+1
      SUM=SQRT(REAL(TEMP1)**2+AIMAG(TEMP1)**2+REAL(TEMP2)**2+AIMAG
     $(TEMP2)**2)
      IF (SUM.NE.0.0) GO TO 32
      TEMP1=HL(RP1,RP1)
      TEMP2=HL(R+2,RP1)
      GO TO 36
   32 COSS=TEMP1/SUM
      SINN=TEMP2/SUM
```

```
      INDEX=MAX0(R-1,1)
      DO 33 I=INDEX,N
      TEMP=CONJG(COSS)*HL(R,I)+CONJG(SINN)*HL(RP1,I)
      HL(RP1,I)=-SINN*HL(R,I)+COSS*HL(RP1,I)
   33 HL(R,I)=TEMP
      TEMP1=HL(RP1,RP1)
      TEMP2=HL(R+2,RP1)
      DO 34 I=1,R
      TEMP=COSS*HL(I,R)+SINN*HL(I,RP1)
      HL(I,RP1)=-CONJG(SINN)*HL(I,R)+CONJG(COSS)*HL(I,RP1)
   34 HL(I,R)=TEMP
      INDEX=MIN0(R+2,N)
      DO 35 I=RP1,INDEX
      HL(I,R)=SINN*HL(I,RP1)
   35 HL(I,RP1)=CONJG(COSS)*HL(I,RP1)
   36 CONTINUE
      ICOUNT=ICOUNT+1
      GO TO 21
   37 TEMP1=HL(N-1,N-1)+HL(N,N)
      TEMP2=HL(N,N)*HL(N-1,N-1)-HL(N,N-1)*HL(N-1,N)
      TEMP=(TEMP1+CSQRT((TEMP1)**2-4.0*(TEMP2)))/2.0
      IF(REAL(TEMP).NE.0.0) GO TO 38
      LAMDA(M)=SHIFT(1)
      LAMDA(M-1)=TEMP1+SHIFT(1)
      GO TO 39
   38 LAMDA(M)=TEMP+SHIFT(1)
      LAMDA(M-1)=TEMP2/TEMP+SHIFT(1)
C
```

```
C   VALUSE IN DECENDING ORDER OF ABSOLUTE MAGITUDE
C
   39 CONTINUE
      IF(M.NE.2) GO TO 40
      EPS=AMAX1(CABS(LAMDA(1)),CABS(LAMDA(2)))*1.0E-12
      IF (EPS.LT.1.0E-12) EPS=1.0E-12
      H(1,1)=HL(1,1)
      H(1,2)=HL(1,2)
      H(2,1)=HL(2,1)
      H(2,2)=HL(2,2)
   40 CONTINUE
      IF(NCAL.LE.1) GO TO 143
      K=NCAL+1
      DO 41 I=1,NCAL
      K=K-1
      CNT(I)=LAMDA(K)
   41 CONTINUE
      L=NCAL-1
      DO 43 I=1,L
      K=I+1
      DO 42 J=K,NCAL
      IF (CABS(CNT(I)).GE.CABS(CNT(J))) GO TO 42
      TEMP=CNT(J)
      CNT(J)=CNT(I)
      CNT(I)=TEMP
   42 CONTINUE
   43 LAMDA(I)=CNT(I)
      LAMDA(NCAL)=CNT(NCAL)
```

```
C
C   CALCULATES VECTORS
C
  143 CONTINUE
      NC=MIN0(NV,NCAL)
      IF(NC.EQ.0) GO TO 63
      NM1=M-1
      DO 62 L=1,NC
      DO 45 I=1,M
      DO 44 J=1,M
   44 HL(J,I)=H(J,I)
   45 HL(I,I)=HL(I,I)-LAMDA(L)
      DO 49 I=1,NM1
      COLM(I)=(0.0,0.0)
      INTH(I,2)=0
      IP1=I+1
      IF (CABS(HL(I+1,I)).LE.CABS(HL(I,I))) GO TO 47
      INTH(I,2)=I
      DO 46 J=1,M
      TEMP=HL(IP1,J)
      HL(IP1,J)=HL(I,J)
   46 HL(I,J)=TEMP
   47 IF(HL(I,I).EQ.(0.0,0.0)) GO TO 49
      COLM(I)=-HL(IP1,I)/HL(I,I)
      DO 48 J=IP1,M
   48 HL(IP1,J)=HL(IP1,J)+COLM(I)*HL(I,J)
   49 CONTINUE
      DO 50 I=1,M
```

```
50 CNT(I)=(1.0,0.0)
   TWICE=.FALSE.
51 IF (HL(M,M).EQ.(0.0,0.0)) HL(M,M)=EPS
   CNT(M)=CNT(M)/HL(M,M)
   DO 53 I=1,NM1
   K=M-1
   DO 52 J=K,NM1
52 CNT(K)=CNT(K)-HL(K,J+1)*CNT(J+1)
   IF(HL(K,K).EQ.(0.0,0.0)) HL(K,K)=EPS
53 CNT(K)=CNT(K)/HL(K,K)
   ABIG=0.0
   DO 54 I=1,N
   SUM=ABS(REAL(CNT(I)))+ABS(AIMAG(CNT(I)))
   IF(SUM.GT.ABIG) ABIG=SUM
54 CONTINUE
   DO 55 I=1,M
55 CNT(I)=CNT(I)/ABIG
   IF(TWICE) GO TO 57
   DO 56 I=1,NM1
   IF(INTH(I,2).EQ.0) GO TO 56
   TEMP=CNT(I)
   CNT(I)=CNT(I+1)
   CNT(I+1)=TEMP
56 CNT(I+1)=CNT(I+1)+COLM(I)*CNT(I)
   TWICE=.TRUE.
   GO TO 51
57 IF(M.EQ.2) GO TO 60
   NM2=M-2
```

```
      DO 59 I=1,NM2
      N1I=M-1-I
      NI1=M-I+1
      DO 58 J=NI1,M
   58 CNT(J)=H(J,N1I)*CNT(N1I+1)+CNT(J)
      INDEX=INTH(N1I,1)
      TEMP=CNT(N1I+1)
      CNT(N1I+1)=CNT(INDEX)
   59 CNT(INDEX)=TEMP
C
C  NORMALIZE VECTORS TO UNIT LENGTH
C
   60 CONTINUE
      TEMP=(0.0,0.0)
      DO 61 I=1,M
   61 TEMP=TEMP+CNT(I)*CNT(I)
      TEMP=CSQRT(TEMP)
      IF(TEMP.EQ.(0.0,0.0)) TEMP=(1.0,0.0)
      DO 62 I=1,M
      VECT(I,L)=CNT(I)/TEMP
   62 CONTINUE
   63 CONTINUE
      INTH(1,1)=NCAL
      RETURN
      END
C     *******************************************************************
      SUBROUTINE GFCTS(M)
C     *******************************************************************
```

```
C     ------------------------------------------------------------------
C     THIS SUBROUTINE CALCULATES THE GALERKIN'S FUNCTIONS
C     ------------------------------------------------------------------
C
      REAL MYL(5,2,11),MYLP(5,2,11),MZL(5,2,11),MZLP(5,2,11),MXL(5,2,11)
     1,MXLP(5,2,11),MY(5,51),MYP(5,51),MZ(5,51),MZP(5,51),MX(5,51),
     2MXP(5,51)
      DIMENSION AL(5),WL(5,2,11),WLP(5,2,11),VL(5,2,11),VLP(5,2,11),
     1TL(5,2,11),VXL(5,2,11),VXLP(5,2,11),W(5,51),WP(5,51),V(5,51),
     2VP(5,51),T(5,51),U(5,51),VX(5,51),VXP(5,51),C(4,4),CT(4,4),
     3B(5,2,4),CT1(4,4),A(5,4,4),FD(2,2),ul(5,2,11)
      DATA AL/1.875,4.694,7.855,10.996,14.137/
      COMMON/XM1/NPATH
      COMMON/XM2/SPAN1,SPAN2,SPAN3,FD,OMEGAN
      COMMON/XGFCT1/A,B
      COMMON/XGFCT2/MYL,MYLP,MZL,MZLP,MXL,MXLP,MY,MYP,MZ,MZP,MX,MXP,WL,
     1WLP,VL,VLP,TL,UL,VXL,VXLP,W,WP,V,VP,T,U,VX,VXP
      DO 220 I=1,M
      X=0.0
      SN=SIN(AL(I))
      SH=SINH(AL(I))
      CN=COS(AL(I))
      CH=COSH(AL(I))
      E=(-SN+SH)/2.0*SN*SH
      G=(CN+CH)/2.0*SN*SH
      D=(2.0*FLOAT(I)-1.0)*PI/2.0
      DO 210 J=1,11
      X1=X*D
```

```
      X=X*AL(I)
      SN=SIN(X)
      SH=SINH(X)
      CN=COS(X)
      CH=COSH(X)
      CN1=COS(X1)
      SN1=SIN(X1)
      DO 200 K=1,NPATH
      WL(I,K,J)=E*(SN-SH)+G*(CN-CH)
      WLP(I,K,J)=AL(I)*E*(CN-CH)+AL(I)*G*(-SN-SH)
      MYL(I,K,J)=AL(I)**2*E*(-SN-SH)+AL(I)**2*G*(-CN-CH)
      MYLP(I,K,J)=AL(I)**3*E*(-CN-CH)+AL(I)**3*G*(SN-SH)
      VL(I,K,J)=WL(I,K,J)
      VLP(I,K,J)=WLP(I,K,J)
      MZL(I,K,J)=MYL(I,K,J)
      MZLP(I,K,J)=MYLP(I,K,J)
      TL(I,K,J)=SN1
      MXL(I,K,J)=D*CN1
      MXLP(I,K,J)=-D*D*SN1
      UL(I,K,J)=TL(I,K,J)
      VXL(I,K,J)=MXL(I,K,J)
      VXLP(I,K,J)=MXLP(I,K,J)
  200 CONTINUE
      X=X+HS1
  210 CONTINUE
      X=SPAN1
      DO 220 J=1,51
      X1=X*D
```

```
      X=AL(I)
      SN=SIN(X)
      SH=SINH(X)
      CN=COS(X)
      CH=COSH(X)
      CN1=COS(X1)
      SN1=SIN(X1)
      W(I,J)=E*(SN-SH)+G*(CN-CH)
      WP(I,J)=AL(I)*E*(CN-CH)+AL(I)*G*(-SN-SH)
      MY(I,J)=AL(I)**2*E*(-SN-SH)+AL(I)**2*G*(-CN-CH)
      MYP(I,J)=AL(I)**3*E*(-CN-CH)+AL(I)**3*G*(SN-SH)
      V(I,J)=W(I,J)
      VP(I,J)=WP(I,J)
      MZ(I,J)=MYP(I,J)
      T(I,J)=SN1
      MX(I,J)=D*CN1
      MXP(I,J)=-D*D*SN1
      U(I,J)=T(I,J)
      VX(I,J)=MX(I,J)
      VXP(I,J)=MXP(I,J)
      X=X+HS2
  220 CONTINUE
      C(1,1)=1.0
      C(1,2)=SPAN3
      C(1,3)=SPAN3*SPAN3
      C(1,4)=C(1,3)*SPAN3
      C(2,1)=0.0
      C(2,2)=1.0
```

```
C(2,3)=2.0*SPAN3
C(2,4)=3.0*SPAN3*SPAN3
C(3,1)=1.0
C(3,2)=SPAN1
C(3,3)=SPAN1*SPAN1
C(3,4)=C(3,3)*SPAN1
C(4,1)=0.0
C(4,2)=1.0
C(4,3)=2.0*SPAN1
C(4,4)=3.0*SPAN1*SPAN1
CALL SOLUTN(C,4,-1,4)
N=NPATH+1
n2=2*npath
DO 240 I=1,M
DO 230 K=1,N2,2
K1=K
IF(K.GT.1) K1=K-1
CT(1,K)=WL(I,K1,10)
CT(2,K)=WLP(I,K1,10)
CT(3,K)=W(I,1)+FD(K1,1)*T(I,1)
CT(4,K)=WP(I,1)
CT(1,K+1)=VL(I,K1,10)
CT(2,K+1)=VLP(I,K1,10)
CT(3,K+1)=V(I,1)-FD(K1,2)*T(I,1)
CT(4,K+1)=VP(I,1)
B(I,1,K)=(SPAN1*TL(I,K1,10)-SPAN3*T(I,1))/(SPAN1-SPAN3)
B(I,2,K)=(T(I,1)-TL(I,K1,10))/(SPAN1-SPAN3)
B(I,1,K+1)=(SPAN1*UL(I,K1,10)-SPAN3*U(I,1))/(SPAN1-SPAN3)
```

```
      B(I,2,K+1)=(U(I,1)-UL(I,K1,10))/(SPAN1-SPAN3)
  230 CONTINUE
      CALL MATMUL(C,CT,CT1,4,4,N2)
      DO 240 II=1,4
      DO 240 IJ=1,N2
      A(I,II,IJ)=CT1(II,IJ)
  240 CONTINUE
      DO 260 I=1,MODES
      DO 260 J=1,NPATH
      IJ=2*J-1
      TL(I,J,11)=0.0
      UL(I,J,11)=0.0
      WL(I,J,11)=0.0
      VL(I,J,11)=0.0
      WLP(I,J,11)=0.0
      VLP(I,J,11)=0.0
      WL(I,J,11)=A(I,L,IJ)*SPAN1**(L-1)+WL(I,J,11)
      VL(I,J,11)=A(I,L,IJ+1)*SPAN1**(L-1)+VL(I,J,11)
      WLP(I,J,11)=FLOAT(L-1)*A(I,L,IJ)*SPAN1**(L-2)+WLP(I,J,11)
      VLP(I,J,11)=FLOAT(L-1)*A(I,L,IJ+1)*SPAN1**(L-2)+VLP(I,J,11)
  250 CONTINUE
      TL(I,J,11)=B(I,1,IJ)+B(I,2,IJ)*SPAN1
      UL(I,J,11)=B(I,1,IJ+1)+B(I,2,IJ+1)*SPAN1
  260 CONTINUE
      RETURN
      END
C     *******************************************************************
      SUBROUTINE GPROC(MODES)
```

```
C     ****************************************************************
C     ----------------------------------------------------------------
C     THIS SUBROUTINE FORMULATES THE EIGENVALUE PROBLEM
C     ----------------------------------------------------------------
      REAL INTG
      DIMENSION A11(5,5),A12(5,5),A21(5,5),A22(5,5),A31(5,5),A32(5,5),
     1A33(5,5),A41(5,5),A13(5,5),A14(5,5),A23(5,5),A24(5,5),A34(5,5),
     2A35(5,5),A36(5,5),A42(5,5),B11(5,5),B12(5,5),B21(5,5),b33(5,5),
     3B22(5,5),B31(5,5),B32(5,5),B41(5,5),A51(5,5),A52(5,5),A61(5,5),
     4A62(5,5),A71(5,5),A72(5,5),A73(5,5),A83(5,5),D11(2,11),D12(2,11),
     5D13(2,11),D14(2,11),D15(2,11),D16(2,11),D17(2,11),D18(2,11),
     6D19(2,11),D110(2,11),D21(51),D22(51),D23(51),d111(2,11),
     7D24(51),D25(51),D26(51),D27(51),D28(51),D29(51),D210(51),D211(51),
     8D212(51),F11(2,11),F12(2,11),F13(2,11),F14(2,11),F15(2,11),F21
     9(51),F22(51),F23(51),F24(51),F25(51),FD(2,2),F16(2,11),F17(2,11),
     1F26(51),F27(51),D1(5,5),D2(5,5),C(10,10),C1(10,10),C2(10,10),
     2WL(5,2,11),WLP(5,2,11),VL(5,2,11),VLP(5,2,11),TL(5,2,11),UL(5,2,
     311),VXL(5,2,11),VXLP(5,2,11),W(5,51),WP(5,51),V(5,51),VP(5,51),
     4t(5,51),U(5,51),VX(5,51),VXP(5,51),A(5,4,4),B(5,2,4),E1(20,20),
     5e2(20,20),E3(20,20),E4(20,20),DS(20,20),d112(2,11),d113(2,11),
     6d213(51),a81(5,5),a82(5,5)
      REAL MYL(5,2,11),MYLP(5,2,11),MZL(5,2,11),MZLP(5,2,11),
     1mxl(5,2,11),MXLP(5,2,11),MY(5,51),MYP(5,51),MZ(5,51),MZP(5,51),
     2mx(5,51),MXP(5,51)
      COMMON/XM1/NPATH
      COMMON/XM2/SPAN1,SPAN2,SPAN3,FD,OMEGAN
      COMMON/XM3/D11,D12,D13,D14,D15,D16,D17,D18,D19,D110,D111,D112,D113
     1,D21,D22,D23,D24,D25,D26,D27,D28,D29,D210,D211,D212,D213
```

```
      COMMON/XGFCT2/MYL,MYLP,MZL,MZLP,MXL,MXLP,MY,MYP,MZ,MZP,MX,MXP,WL,
     1WLP,VL,VLP,TL,UL,VXL,VXLP,W,WP,V,VP,T,U,VX,VXP
      COMMON/XGPR1/DS
      DO 120 K=1,MODES
      DO 120 J=1,MODES
      DO 100 I=1,NPATH
      DO 100 L=1,11
      F11(I,L)=D113(I,L)*WLP(J,I,L)*WLP(K,I,L)
      F12(I,L)=D12(I,L)*TL(J,I,L)*WLP(K,I,L)
      F13(I,L)=-MYLP(J,I,L)*WLP(K,I,L)
      F14(I,L)=D11(I,L)*WL(J,I,L)*WL(K,I,L)
      F15(I,L)=D13(I,L)*TL(J,I,L)*WL(K,I,L)
  100 CONTINUE
      DO 110 L=1,51
      F21(L)=D213(L)*WP(J,L)*WP(K,L)
      F22(L)=D22(L)*T(J,L)*WP(K,L)
      F23(L)=-MYP(J,L)*WP(K,L)
      F24(L)=D21(L)*W(J,L)*W(K,L)
      F25(L)=D23(L)*T(J,L)*W(K,L)
  110 CONTINUE
      A11(K,J)=INTG(F11,F12)
      A12(K,J)=INTG(F12,F22)
      A14(K,J)=INTG(F13,F23)
      B11(K,J)=INTG(F14,F24)
      B12(K,J)=INTG(F15,F25)
      A14(K,J)=A14(K,J)+W(K,1)*(MYL(J,1,11)+MYL(J,2,11)-MY(J,1))
      A13(K,J)=-FD(1,2)*VXL(J,1,11)-FD(2,2)*VXL(J,2,11)
  120 CONTINUE
```

```
      DO 150 K=1,MODES
      DO 150 J=1,MODES
      DO 130 I=1,NPATH
      DO 130 L=1,11
      F11(I,L)=D113(I,L)*VLP(J,I,L)*VLP(K,I,L)
      F12(I,L)=D14(I,L)*TL(J,I,L)*VLP(K,I,L)
      F13(I,L)=-MZLP(J,I,L)*VLP(K,I,L)
      F14(I,L)=D11(I,L)*VL(J,I,L)*VL(K,I,L)
      F15(I,L)=-D15(I,L)*TL(J,I,L)*VL(K,I,L)
  130 CONTINUE
      DO 140 L=1,51
      F21(L)=D213(L)*VP(J,L)*VP(K,L)
      F22(L)=D24(L)*T(J,L)*VP(K,L)
      F23(L)=-MZP(J,L)*VP(K,L)
      F24(L)=D21(L)*V(J,L)*V(K,L)
      F25(L)=-D25(L)*T(J,L)*V(K,L)
  140 CONTINUE
      A21(K,J)=INTG(F11,F21)
      A22(K,J)=INTG(F12,F22)
      A24(K,J)=INTG(F13,F23)
      B21(K,J)=INTG(F14,F24)
      B22(K,J)=INTG(F15,F25)
      A24(K,J)=A24(K,J)+V(K,1)*(MZL(J,1,11)+MZL(J,2,11)-MZ(J,1))
      A23(K,J)=-FD(1,1)*VXL(J,1,11)-FD(2,1)*VXL(J,2,11)
  150 CONTINUE
      DO 180 K=1,MODES
      DO 180 J=1,MODES
      DO 160 I=1,NPATH
```

```
      DO 160 L=1,11
      F11(I,L)=-D12(I,L)*VLP(J,I,L)*TL(K,I,L)+OMEGAN*D15(I,L)*
     1VL(J,I,L)*TL(K,I,L)
      F12(I,L)=D14(I,L)*WLP(J,I,L)*TL(K,I,L)
      F13(I,L)=D16(I,L)*TL(J,I,L)*TL(K,I,L)
      F14(I,L)=-MXLP(J,I,L)*TL(K,I,L)
      F15(I,L)=D17(I,L)*TL(J,I,L)*TL(K,I,L)
      F16(I,L)=D13(I,L)*WL(J,I,L)*TL(K,I,L)
      F17(I,L)=-D15(I,L)*VL(J,I,L)*TL(K,I,L)
  160 CONTINUE
      DO 170 L=1,51
      F21(L)=-D22(L)*VP(J,L)*T(K,L)+OMEGAN*D25(L)*V(J,L)*T(K,L)
      F22(L)=D24(L)*WP(J,L)*T(K,L)
      F23(L)=D26(L)*T(J,L)*T(K,L)
      F24(L)=-MXP(J,L)*T(K,L)
      F25(L)=D27(L)*T(J,L)*T(K,L)
      F26(L)=D23(L)*W(J,L)*T(K,L)
      F27(L)=-D25(L)*V(J,L)*T(K,L)
  170 CONTINUE
      A31(K,J)=INTG(F11,F12)
      A32(K,J)=INTG(F12,F22)
      A33(K,J)=INTG(F13,F23)
      A36(K,J)=INTG(F14,F24)
      B31(K,J)=INTG(F17,F27)
      B32(K,J)=INTG(F16,F26)
      B33(K,J)=INTG(F15,F25)
      A36(K,J)=A36(K,J)+T(K,1)*(MXL(J,1,11)+MXL(J,2,11)-MX(J,1))
      A35(K,J)=A35(K,J)-FD(1,1)*MYLP(J,1,11)*T(K,1)-FD(2,1)
```

```
     1*MYLP(J,2,11)*T(K,1)
      A32(K,J)=A32(K,J)+T(K,1)*(D113(1,11)*WLP(J,1,11)*FD(1,1)
     1+D113(2,11)*WLP(J,2,11)*FD(2,1))
      A33(K,J)=A33(K,J)+T(K,1)*(FD(1,1)*D12(1,11)*T(J,1))
     1+FD(2,1)*D12(2,11)*T(J,2)+T(K,1)*(FD(2,1)*D14(2,11)*
     2TL(J,1,11)+FD(2,2)*D14(2,11)*TL(J,2,11))
      A34(K,J)=A34(K,J)+T(K,1)*(FD(1,2)*MZLP(J,1,11)+FD(2,2)
     1*MZLP(J,2,11))
      A31(K,J)=A31(K,J)*T(K,1)*(-FD(1,2)*D113(1,11)*VLP(J,1,11)
     1-FD(2,2)*D113(2,11)*VLP(J,2,11))
  180 CONTINUE
      DO 210 K=1,MODES
      DO 210 J=1,MODES
      DO 190 I=1,NPATH
      DO 190 L=1,11
      F11(I,L)=-OMEGAN*D11(I,J)*UL(J,I,L)*UL(K,I,L)
      F12(I,L)=-VXLP(J,I,L)*UL(K,I,L)
      F13(I,L)=D11(I,L)*UL(J,I,L)*UL(K,I,L)
  190 CONTINUE
      DO 200 L=1,51
      F22(L)=-OMEGAN*D21(L)*U(J,L)*U(K,L)
      F22(L)=-VXP(J,L)*U(K,L)
      F23(L)=D21(L)*U(J,L)*U(K,J)
  200 CONTINUE
      A41(K,J)=INTG(F11,F21)
      A42(K,J)=INTG(F12,F22)
      B41(K,J)=INTG(F13,F23)
      A42(K,J)=A42(K,J)+U(K,1)*(VXL(J,1,11)+VXL(J,2,11)-VX(J,1))
```

```
210 CONTINUE
    DO 240 K=1,MODES
    DO 240 J=1,MODES
    DO 220 I=1,NPATH
    DO 220 L=1,11
    F11(I,L)=UL(J,I,L)*VXL(K,I,L)
    F12(I,L)=-D18(I,J)*VXL(J,I,L)*VXL(K,I,L)
    F13(I,L)=TL(J,I,L)*MXL(K,I,L)
    F14(I,L)=-D19(I,L)*MXL(J,I,L)*MXL(K,I,L)
220 CONTINUE
    DO 230 L=1,51
    F21(L)=U(J,L)*VX(K,L)
    F22(L)=-D28(L)*VX(J,L)*VX(K,L)
    F23(L)=T(J,L)*MX(K,L)
    F24(L)=-D29(L)*MX(J,L)*MX(K,L)
230 CONTINUE
    A51(K,J)=INTG(F11,F21)
    A52(K,J)=INTG(F12,F22)
    A61(K,J)=INTG(F13,F23)
    A62(K,J)=INTG(F14,F24)
240 continue
    DO 270 K=1,MODES
    DO 270 J=1,MODES
    DO 250 I=1,NPATH
    DO 250 L=1,11
    F11(I,L)=MZL(J,I,L)*MZL(K,I,L)
    F12(I,L)=D111(I,L)*MYL(J,I,L)*MZL(K,I,L)
    F13(I,L)=-D110(I,L)*MYL(J,I,L)*MZL(K,I,L)
```

```
      F14(I,L)=MYL(J,I,L)*MYL(K,I,L)
      F15(I,L)=-D112(I,L)*MYL(J,I,L)*MYL(K,I,L)
      F16(I,L)=D112(I,L)*MZL(J,I,L)*MYL(K,I,L)
  250 CONTINUE
      DO 260 L=1,51
      F21(L)=MZ(J,L)*MZ(K,L)
      F22(L)=D211(L)*MY(J,L)*MZ(K,L)
      F23(L)=-D210(L)*MY(J,L)*MZ(K,L)
      F24(L)=MY(J,L)*MZ(K,L)
      F25(L)=-D112(2,L)*MY(J,L)*MY(K,L)
      F26(L)=D112(2,L)*MY(J,L)*MZ(K,L)
  260 CONTINUE
      A71(J,K)=INTG(F11,F21)
      A72(J,K)=INTG(F13,F23)
      A73(J,K)=INTG(F12,F22)
      A81(J,K)=INTG(F14,F24)
      A82(J,K)=INTG(F16,F26)
      A83(J,K)=INTG(F15,F25)
  270 CONTINUE
      CALL SOLUTN(A52,MODES,-1,MODES)
      CALL MATMUL(A52,A51,D1,5,5,5)
      CALL SOLUTN(A62,MODES,-1,MODES)
      CALL MATMUL(A62,A61,D2,5,5,5)
      DO 280 I=1,MODES
      DO 280 J=1,MODES
      C(I,J)=A72(I,J)
      C(I,J+MODES)=A73(I,J)
      C(I+MODES,J)=A82(I,J)
```

```
      C(I+MODES,J+MODES)=A83(I,J)
      C1(I,J)=A71(I,J)
      C1(I,J+MODES)=0.0
      C1(I+MODES,J)=0.0
      C1(I+MODES,J+MODES)=0.0
  280 CONTINUE
      N2=2*MODES
      CALL SOLUTN(C,N2,-1,N2)
      CALL MATMUL(C,C1,C2,N2,N2,N2)
      N3=3*MODES
      N4=4*MODES
      DO 285 I=1,N4
      DO 285 J=1,N4
      E1(I,J)=0.0
      E2(I,J)=0.0
      E3(I,J)=0.0
      E4(I,J)=0.0
  285 CONTINUE
      DO 290 I=1,MODES
      DO 290 J=1,MODES
      E1(I,J+N2)=A11(I,J)
      E1(I,J+N3)=A12(I,J)
      E1(I+MODES,J+MODES)=A21(I,J)
      E1(I+MODES,J+N3)=A22(I,J)
      E1(I+N2,J+MODES)=A31(I,J)
      E1(I+N2,J+N2)=A32(I,J)
      E1(I+N2,J+N3)=A33(I,J)
      E1(I+N3,J)=A41(I,J)
```

```
      E2(I,J)=A13(I,J)
      E2(I,J+N2)=A14(I,J)
      E2(I+MODES,J)=A23(I,J)
      E2(I+MODES,J+MODES)=A24(I,J)
      E2(I+N2,J+MODES)=A34(I,J)
      E2(I+N2,J+N2)=A35(I,J)
      E2(I+N2,J+N3)=A36(I,J)
      E2(I+N3,J)=A42(I,J)
      E3(I,J+N2)=B11(I,J)
      E3(I,J+N3)=B12(I,J)
      E3(I+MODES,J+MODES)=B21(I,J)
      E3(I+MODES,J+N3)=B22(I,J)
      E3(I+N2,J+N2)=B32(I,J)
      E3(I+N2,J+MODES)=B31(I,J)
      E3(I+N2,J+N3)=B33(I,J)
      E3(I+N3,J)=B41(I,J)
      E4(I,J)=-D1(I,J)
      E4(I+N3,J+N3)=-D2(I,J)
      E4(I+MODES,J+MODES)=C2(I,J)
      E4(I+MODES,J+N2)=C2(I,J+MODES)
      E4(I+N2,J+MODES)=C2(I+MODES,J)
      E4(I+N2,J+N2)=C2(I+MODES,J+MODES)
  290 CONTINUE
      CALL MATMUL(E2,E4,DS,N4,N4,N4)
      CALL SOLUTN(E3,N4,-1,N4)
      DO 300 I=1,N4
      DO 300 J=1,N4
      E2(I,J)=E1(I,J)+DS(I,J)
```

```
  300 CONTINUE
      CALL MATMUL(E3,E2,DS,N4,N4,N4)
      RETURN
      END
C     ******************************************************************
      FUNCTION INTG(F1,F2)
C     ******************************************************************
C     ------------------------------------------------------------------
C     THIS FUNCTION INTEGRATES THE FUNCTION IN THE DOMAIN
C     ------------------------------------------------------------------
C
      REAL INTG
      DIMENSION F1(2,11),F2(51)
      COMMON/XM1/ NPATH
      COMMON/XM5/ HS1,HS2
      H1=HS1/24.0
      H2=HS2/24.0
      SUM=0.0
      DO 110 J=1,NPATH
      SUM=SUM+(9.0*F1(J,1)+19.0*F1(J,2)-5.0*F1(J,3)+F1(J,4))*H1
      DO 100 K=1,8
      SUM=SUM+H1*(-F1(J,K)+13.0*F1(J,K+1)+13.0*F1(J,K+2)-F1(J,K_3))
  100 CONTINUE
      SUM=SUM+H1*(F1(J,8)-5.0*F1(J,9)+19.0*F1(J,10)+9.0*F1(J,11))
  110 CONTINUE
      SUM=SUM+(9.0*F2(1)+19.0*F2(2)-5.0*F2(3)+F2(4))*H2
      DO 120 K=1,48
      SUM=SUM+H2*(-F2(K)+13.0*F2(K+1)+13.0*F2(K+2)-F2(K+3))
```

```
  120 CONTINUE
      INTG=SUM+H2*(F2(48)-5.0*F2(49)+19.0*F2(50)+9.0*F2(51))
      RETURN
      END
C     ***********************************************************************
      SUBROUTINE INTPOL(N,A,H)
C     ***********************************************************************
C     -----------------------------------------------------------------------
C     THIS SUBROUTINE INTERPOLATES FOR THE REQUIRED VALUES
C     -----------------------------------------------------------------------
      DIMENSION A(51),STA(51),TABLE(51,1),B(51)
      COMMON/XM4/STA,NS
      NN=N-1
      A(N)=A(NS)
      NM1=NS-1
      DO 20 I=1,NM1
   20 TABLE(I,1)=(A(I+1)-A(I))/(STA(I+1)-STA(I))
      XARG=H
      DO 35 I=2,NN
      DO 25 J=1,NS
      IF (J.EQ.NS.OR.XARG.LE.STA(J)) GO TO 30
   25 CONTINUE
   30 MAX=J
      IF (MAX.LE.2) MAX=2
      ISUB=MAX-1
      YEST=TABLE(ISUB,1)
      B(I)=YEST*(XARG-STA(ISUB))+A(ISUB)
   35 XARG=XARG+H
```

```
      DO 40 J=2,NN
   40 A(J)=B(J)
      RETURN
      END
C     *****************************************************************
      SUBROUTINE MATMUL(A,B,C,L,M,N)
C     *****************************************************************
C     -----------------------------------------------------------------
C     MATRIX MULTIPLICATION
C     -----------------------------------------------------------------
C
      DIMENSION A(L,M),B(M,N),C(L,N)
      DO 100 I=1,L
      DO 100 J=1,N
      C(I,J)=0.0
      DO 100 K=1,M
      C(I,J)=C(I,J)+A(I,M)*B(M,J)
  100 CONTINUE
      RETURN
      END
C     ***************************************************************
      SUBROUTINE SOLUTN(A,N,INDIC,NRC)
C     ***************************************************************
C     ---------------------------------------------------------------
C     SOLUTION OF SIMULTANEOUS EQUATIONS & INVERSE
C     ---------------------------------------------------------------
C
      DIMENSION A(NRC,NRC),X1(100),IROW(100),JCOL(100),JORD(100),
```

```
   1Y(100)
    IND=0
    MAX=N
    IF (INDIC.GE.0) MAX=N+1
    DETER=1.0
    DO 80 K=1,N
    KM1=K-1
    PIVOT=0.0
    DO 60 I=1,N
    DO 60 J=1,N
    IF (K.EQ.1) GO TO 55
    DO 50 ISCAN=1,KM1
    DO 50 JSCAN=1,KM1
    IF (I.EQ.IROW(ISCAN)) GO TO 60
    IF (J.EQ.JCOL(JSCAN)) GO TO 60
 50 CONTINUE
 55 IF (ABS(A(I,J)).LE.ABS(PIVOT)) GO TO 60
    PIVOT=A(I,J)
    IROW(K)=I
    JCOL(K)=J
 60 CONTINUE
    IF (ABS(PIVOT).GT.0.1E-15) GO TO 65
    DETER=0.0
    IND=1
    WRITE(50,61)
 61 FORMAT('MATRIX IS ALGORITHMICALLY SINGULAR')
    RETURN
 65 IROWK=IROW(K)
```

```
      JCOLK=JCOL(K)
      DETER=DETER*PIVOT
      DO 70 J=1,MAX
   70 A(IROWK,J)=A(IROWK,J)/PIVOT
      A(IROWK,JCOLK)=1.0/PIVOT
      DO 80 I=1,N
      AIJCK=A(I,JCOLK)
      IF (I.EQ.IROWK) GO TO 80
      A(I,JCOLK)=-AIJCK/PIVOT
      DO 75 J=1,MAX
   75 IF (J.NE.JCOLK) A(I,J)=A(I,J)-AIJCK*A(IROWK,J)
   80 CONTINUE
      DO 85 I=1,N
      IROWI=IROW(I)
      JCOLI=JCOL(I)
      JORD(IROWI)=JCOLI
      IF (INDIC.GE.0) X1(JCOLI)=A(IROWI,MAX)
   85 CONTINUE
      INTCH=0
      NM1=N-1
      DO 90 I=1,NM1
      IP1=I+1
      DO 90 J=IP1,N
      IF (JORD(J).GE.JORD(I)) GO TO 90
      JTEMP=JORD(j)
      JORD(J)=JORD(I)
      JORD(I)=JTEMP
      INTCH=INTCH+1
```

```
   90 CONTINUE
      IF (INTCH/2*2.NE.INTCH) DETER=-DETER
      IF (INDIC.LE.0) GO TO 94
      RETURN
   94 DO 100 J=1,N
      DO 95 I=1,N
      IROWI=IROW(I)
      JCOLI=JCOL(I)
      Y(JCOLI)=A(IROWI,J)
   95 CONTINUE
      do 100 I=1,N
      A(I,J)=Y(I)
  100 CONTINUE
      DO 110 I=1,N
      DO 105 J=1,N
      IROWJ=IROW(J)
      JCOLJ=JCOL(J)
      Y(IROWJ)=A(I,JCOLJ)
  105 CONTINUE
      DO 110 J=1,N
      A(I,J)=Y(J)
  110 CONTINUE
      RETURN
      END
```