INTRODUCTION

Even though the NASA Structural Analysis (NASTRAN) program is designed to solve numerous structural dynamic problems through the use of available rigid formats, an important class of problems, where the structures are spinning at a constant angular velocity, has been omitted. Rotating shafts, blades of spinning turbines, rotating linkages, and spin stabilized satellites are examples of problems falling within this class. These problems differ from the nonspinning structures in several significant ways. The accelerations of the masses in a nonrotating stationary frame are represented by the second derivatives with respect to time of the spatial variables. In the case of a structure spinning at a constant angular velocity, expressions for the accelerations of the discrete masses contain terms arising from the second derivatives of the spatial variables; in addition, they contain terms caused by coriolis accelerations, which are proportional to the velocities of the masses in the rotating frame. Finally, these expressions reflect the variations in steady-state centripetal accelerations caused by the small displacements of the masses in the rotating frame. The steady-state centrifugal forces set up the steady stresses that give rise to the geometric stiffness matrix.

Since NASTRAN does not construct coriolis and centripetal acceleration matrices, and a centrifugal load vector due to spin about a selected point or about the mass center of the structure, a Fortran subroutine to construct these matrices is added in NASTRAN. The rigid translational degrees of freedom can be removed by using a transformation matrix $T$ and its explicitly given inverse, $T^{-1}$. These matrices are generated in the above Fortran subroutine and their explicit expressions are given in Appendix A.
The complex eigenvalue subroutine of NASTRAN does not measure up to the excellence it has shown in assembling the matrices. If the user desires, an option is available to write out the matrices generated by NASTRAN on a magnetic tape which, in turn, can be used as the input to another eigenvalue program. The probable advantages in using another eigenvalue program are that the user may be able to solve a larger problem within the available core and he may use a more efficient eigenvalue routine if one is available to him. If it is required, the user can write out certain information generated by NASTRAN on a magnetic tape unit using the subroutines OUTPUT2 and WRTAPE used in this program.

THEORETICAL DESCRIPTION

The equations of motion of a spinning structure are briefly derived here to show how they differ from those of a nonspinning structure. The direct use of the Newton-Euler equations gives

\[ \begin{align*}
F_s &= m^s A^s \\
T_s &= \frac{\mathbf{H}^s}{dt}
\end{align*} \tag{1} \]

for the \(s\)th rigid body of a flexible appendage; where \(m^s\) is the mass, \(A^s\) is the absolute translational acceleration vector, \(F^s\) and \(T^s\) are the sum of the external and connection force and torque vectors, respectively. \(H^s\) is the angular momentum vector and \(i\) denoted differentiation in the inertial frame of reference.

For a rigid body of an appendage spinning nominally in the steady state with an angular velocity \(\omega\) (fig. 1), the expression for acceleration is written as

\[ \begin{align*}
A^s &= \frac{b_d^2}{dt^2} (c + u^s) + 2\frac{\omega}{dt} (c + u^s) + \omega \times (\omega \times (c + u^s)) + \omega \times (\omega \times r^s) \\
&+ \frac{i}{dt} \omega \times (c + u^s + r^s) + \frac{id^2 R}{dt^2}
\end{align*} \tag{2} \]
\[ T^s = \frac{i}{d} \left( I^s \cdot \omega^s \right) = \frac{i}{d} \left( I^s \cdot (\omega + \beta^s) \right) \]

\[ = I^s \cdot \left[ \frac{i}{d} \omega + \frac{b_d}{d^2} \beta^s + \omega \times \frac{b_d}{d} \beta^s \right] \]

\[ + \omega \times I^s \cdot \omega + \omega \times I^s \cdot \frac{b_d}{d} \beta^s + \frac{b_d}{d} \beta^s \times I^s \cdot \omega \]  

(3)

where \( c \) is a vector representing the location of the mass center at time \( t \) with respect to its steady state position. \( u^s \) and \( \beta^s \) are vectors representing the displacement and small rotation, respectively, of the \( s \)th rigid body from its steady state configuration. \( x^s \) is a vector representing the location of the \( s \)th rigid body of the appendage in its steady state configuration measured from the steady state mass center location. \( I^s \) is the inertia dyadic of the \( s \)th rigid body. Superscript \( b \) denotes differentiation in the reference frame \( b \) imbedded in the rigid body with the origin at the steady state mass center location.

For zero spin \( (\omega = 0) \), eq. (2) and (3) reduce to the familiar form

\[ \Delta^s = \frac{i}{d^2} (R + c + x^s + u^s) \]  

(2-a)

\[ T^s = I^s \cdot \frac{i}{d^2} \beta^s \]  

(3-a)

In matrix notation the second term on the right hand side of eq. (2), which is due to coriolis acceleration, gives rise to a skew-symmetric matrix; whereas, the third term, which is due to the centripetal acceleration, yields a symmetric matrix. The fourth term in eq. (2) and (3) represents a steady state centripetal acceleration which describes the steady state configuration. Stretching forces, moments and rotations obtained therefrom, give rise to the second order geometric stiffness matrix. In the absence of angular acceleration, the fifth term of eq. (2) vanishes. If rotational dynamics are the primary concern, the effect of translation of the orbit is disregarded and the last term of eq. (2) also vanishes. The last two terms in eq. (3) will cancel each other if the inertia matrix is diagonal with all the terms having the same magnitude. In the computer program no such restriction is imposed on the \( I^s \) matrix.
Conservation of linear momentum provides the relation

\[ c = -\frac{1}{M} \sum_{s=1}^{n} m^s u^s \]  \hspace{1cm} (4)

where \( M \) is the cumulative mass of all appendages and the central rigid body, and \( n \) is the total number of masses representing all of the appendages.

Conservation of angular momentum is not imposed. As a result, the central rigid body is restricted against variations in rotations. Conservation of linear momentum permits the translation of the central rigid body, thus allowing the coupling of the vibrations of all the appendages attached to the central rigid body.

The set of equations representing the motion of all the rigid bodies in the appendage about the steady state configuration is obtained by substituting eq. (4) into eq. (2) and writing the resulting eq. (2) and (3), for all rigid bodies in matrix form:

\[
[M'] \{\ddot{u}\} + [G'] \{\dot{u}\} + [K'''] + Ke + Kg \} \{u\} = \{F\} . \hspace{1cm} (5)
\]

The steady state equation in matrix form is:

\[
[Ke + K'''] \{u\}^S = \{P\}^S . \hspace{1cm} (6)
\]

The use of eq. (4) eliminates the remaining translational rigid body degrees-of-freedom. As a result, the mass matrix \( M' \) is a symmetric non-diagonal matrix. Matrix \( G' \) is in general, a fully populated skew-symmetric matrix of coriolis acceleration terms. Matrix \( K''' \) is a fully populated non-symmetric matrix of centrifugal acceleration terms. \( Ke \) and \( Kg \) are elastic and geometric (differential) stiffness matrices, respectively, (and are obtained from the NASTRAN program) and \( \{u\} \) is the vector of generalized displacements about the steady state configuration. In the absence of spin, matrices \( G' \), \( K''' \), and \( Kg \) will all be identically zero, and the eigenvalue problem reduces to the standard eigenvalue problem of a free-free structure or a cantilever. \( \{u\}^S \) is the vector of the steady state generalized displacements from the unstrained configuration \( \{r\} \) of the appendages. Since the steady state deformations \( \{u\}^S \) are very small compared to the unstrained configuration \( \{r\} \), it is assumed that the steady state configuration is given by \( \{r\} \) instead of \( \{r\} + \{u\}^S \). The steady state force vector \( \{P\}^S \) is used to obtain the geometric stiffness matrix, \( Kg \).
Matrices $M'$, $G'$, and $K'''$ have the following properties:

\[
\begin{align*}
[M'] &= [M] [T] \\
[G'] &= [G] [T] \\
[K'''] &= [K'''] [T]
\end{align*}
\] (7)

Relations (7) afford a transformation

\[
\{y\} = [T] \{u\}
\] (8)

Substitution of transformation (8) into eq. (5) gives

\[
[M] \{\ddot{y}\} + [G] \{\dot{y}\} + [K'''] + [K'e + Kg] T^{-1} \} \{y\} = 0
\] (9)

where

\[
\begin{align*}
{u} &= \text{Vector of displacements from the steady state configuration of the nodal masses in spinning body frame.} \\
{T} &= \text{Transformation matrix relates the displacements of nodal masses in the body frame with the origin at steady state mass center to the displacements in another body frame obtained by translating the above frame to the instantaneous mass center. In the absence of vibrations both above body frames coincide. If the axis of rotation and the origin of the body frame are both fixed in inertial space, $T$ and $T^{-1}$ become identity matrices, additionally.} \\
M &= \text{NASTRAN generated mass matrix} \\
G &= \text{Dummy module generated coriolis acceleration matrix} \\
K'' &= \text{Dummy module generated centripetal acceleration matrix} \\
Pg &= \text{Dummy module generated steady state centrifugal force vector} \\
Pg &= \text{NASTRAN generated differential stiffness matrix using the above load vector $Pg$} \\
Pg &= \text{NASTRAN generated elastic stiffness matrix.}
\end{align*}
\]

Explicit forms of the above matrices are given in Appendix A.
DMAP DESCRIPTION

The following information and options are made available through the input of vector \( \text{WW} \) with five elements using DMI* cards. The first of the two cards never changes for this program. \( \text{WW}(1), \text{WW}(2), \text{WW}(3) \) are the components of the spin vector in the body frame. Terms \( \text{WW}(4), \text{WW}(5) \) can take the values either 0.0 or 1.0. If \( \text{WW}(4) = 1.0 \) the structure is spinning about the mass center of vehicle, and if \( \text{WW}(4) = 0.0 \) the structure is spinning about a fixed point in the space. The calculation of matrices \( T \) and \( T^{-1} \), which removes the rigid body translational degrees of freedom, is performed if \( \text{WW}(5) = 1.0 \). If \( \text{WW}(5) = 0.0 \), matrices \( T \) and \( T^{-1} \) are identity matrices which means that the structure is supported and does not have the rigid body translational degrees of freedom.

The following options can be exercised through the use of \( \text{WW}(4) \) and \( \text{WW}(5) \).

**Case I.** \( \text{WW}(4) = \text{WW}(5) = 1.0 \), GRID 1 constrained in all six directions. The structure is spinning about the vehicle mass center, and the rigid body translational degrees of freedom are removed. GRID No. 1 is connected by a rigid link to the mass center of the vehicle in the steady state configuration and one or more appendages are cantilevered from GRID No. 1. GRID No. 1 should be constrained in all six directions by use of SPC cards or permanent SPC on GRID cards.

**Case II.** \( \text{WW}(4) = \text{WW}(5) = 0.0 \), GRID 1 constrained in all six directions. The structure is assumed to be spinning about a point (GRID 1) fixed in inertial space, e.g., a spinning shaft with GRID 1 at bearing.

**Case III.** \( \text{WW}(4) = 1.0 \) \( \text{WW}(5) = 0.0 \), GRID 1 constrained in all six directions. Node No. 1 is rigidly connected to the steady state mass center which is fixed in inertial space. The structure is spinning about mass center.

**Case IV.** \( \text{WW}(4) = 0.0 \) \( \text{WW}(5) = 1.0 \), GRID 1 constrained in all six directions. The structure is spinning about GRID 1 with rigid body translational degrees of freedom removed.

* Refer to NASTRAN User's Manual for definitions of card names used herein.
The DMAP sequence given in Appendix B solves eq. (9) and eigenvectors \( y_i \) (\( \text{PHI} \) in DMAP) thus obtained are transformed to \( u_i \) (\( \text{PHID} \) in DMAP) which are the eigenvectors of eq. (8).

It is essential that two subcases are used in the case control deck as shown below for successful completion of the NASTRAN run.

```plaintext
CASE CONTROL DECK
TITLE

SUBCASE 1
DISPLACEMENT = ALL

SUBCASE 2
DSCOEFFICIENT = DEFAULT
BEGIN BULK
```

No provision for checkpoint is made since the time taken to assemble the matrices is just a fraction of the time taken to find a few eigenvalues.

**FUNCTIONAL MODULE PROGRAMING NOTES**

In writing a functional module for NASTRAN, the concept of open core should be employed even if the corresponding logic for an open core array is not used. This gives the generality and possibility of later expansions without having to alter the program extensively. This does not mean that the fixed dimensioned arrays cannot be used in NASTRAN functional modules. The details of the open core concept are given in Section 1.5 of the Programmer's Manual. Once the dimensions are set either by open core or by fixed locations, the next steps are either to retrieve the data (input blocks) to be used for further computations or to store the computed data (output blocks) in a prescribed format within NASTRAN. The data as described in Section 2.2 of the Programmer's Manual may be in the form of a matrix, a table or bulk data cards.

A matrix data are stored in two separate parts. One part constitutes the name of the matrix in alpha-numeric form (Header Information). The columns of the matrix are stored on random access peripheral equipments.
The second part is called the Trailer Information and is stored in FIAT which is an executive system table of NASTRAN. The first part is stored as a set of logical records: the first record is the Header information, and the second and subsequent logical records until the end of file is reached are the columns of a matrix. The Trailer informations, which is the collection of the properties (size, real, complex, symmetric, etc.) of all the matrices used in NASTRAN are given in the Programer's Manual but if new matrices are created, their Trailer information should be stored according to the instructions on page 2.2-2 of the Programer's Manual. Either of the above two parts describing a matrix can be called, as shown later, without disturbing the other.

Each of the matrices, whether constructed by NASTRAN or computed in a functional module and designated as an input or output block in a particular DMAP statement should be referred to by a file number. The numbering system of a file is standardized by NASTRAN as consisting of three digits. The first digit takes value 1 if it is an input block and 2 if it is an output block; the second and third digits refer to matrix location in the string of input or output data blocks. For example, file number 102 in DMAP statement DUMMOD1 given in Appendix B refers to the second input block which is an unreduced mass matrix $M_{gg}$ whereas 203 refers to the third output block which is the coriolis acceleration matrix $G$.

In order to read the desired matrix, the following set of calls to subroutines listed below will unpack and read the matrix data. In each of the subroutines the file number for the appropriate matrix data block must be included in appropriate argument.

```
CALL RDTRL
CALL OPEN
CALL FWDREC
CALL UNPACK
:
CALL UNPACK
CALL CLOSE
```

The subroutine RDTRL calls on the file number appearing in its argument for the Trailer information. A call on RDTRL can also be made after calling OPEN, if desired. Subroutine OPEN opens the file to be read. FWDREC positions the requested file forward one logical record thereby skipping the first record in this particular example. If for
some reason two logical records need to be skipped, FWDREC is called twice. Each call to UNPACK allows the reading of one column (a logical record) of a matrix at a time. The call to UNPACK can be put within a DO-loop once the information on the number of columns of the matrix is obtained from Trailer information. After the reading of columns is completed the subroutine CLOSE is called to close the file as soon as practicable.

If the data are in tabular form, instead of calling UNPACK, call READ to read the data. Each call to READ reads one logical record of the data. The programmer’s manual should be consulted for structure of the record read. The call to READ can be put either within a DO-loop once the information on the number of records is obtained from Trailer information or within an unending DO-loop in which case, when the end of file is reached, the transfer will be made to a statement number appearing in the argument of READ. The set of calls is shown as

```
CALL RDTRL
CALL OPEN
CALL FWDREC
CALL READ

```

If the data on Bulk Data Cards are desired to be retrieved the following set of calls to subroutines should be employed.

```
CALL PRELOC
CALL LOCATE
CALL READ

```

Subroutine PRELOC locates the file on which the bulk data card images are stored and LOCATE locates the desired number of cards in the file.

In DMAP statement DUMMOD1 file number 101 which is GEOM1, contains the geometric information from Bulk Data Cards. This file number is called in PRELOC subroutine. For each bulk data card to be read, subroutine READ should be called.
To pack the matrices calculated in a subroutine and appearing as output data block the following set of calls to subroutines is required.

CALL OPEN
CALL FNAME (finds Header information)
Prepare Trailer information (e.g. M(1), M(2), --M(7) )
   according to instructions appearing in Section 2.2-2
   of the Programer's Manual.
CALL WRITE (writes Header information)

Perform computations.

CALL PACK (----, ---- , WRITE, M)
   :
   CALL PACK
CALL WRTTRL (M(1))
CALL CLOSE

Subroutine OPEN opens the file to be written on and FNAME finds and stores the Header information as appearing in DMAP subroutine (e.g., if file number 204 is referred in OPEN, FNAME will go to the fourth output block of DUMMOD1 which is matrix AA and store AA as the Header information). Next prepare the Trailer information in the vector M according to the instructions in Programer's Manual with the following exceptions. On page 2.2-2 of the Programer's Manual M has the dimension 6 which is an error, it should be dimensioned M(7) and M(1) = File Number
M(2) = 0
M(3) = M(2) of the Programer's Manual
   :

Note that M(2) is set equal to zero and not one as implied in the Programer's Manual. If M(2) = 1, the information on the number of columns stored in Trailer information will show one more than the value desired. Hence when this Trailer information is used to read the columns, the READ will try to take it past the end of file resulting in fatal error.

Subroutine WRITE writes the Header information and then one column at a time is packed by subroutine PACK. Call WRTTRL to write Trailer information and then call subroutine CLOSE.
If it is desired when developing functional modules, the number of input, output blocks and parameters can be altered by altering the information in MPL (subroutine XMPLBD). In the development of this program, four functional modules are written. In addition several other functional modules for other NASA projects at MSFC have also been written. All these functional modules are given dummy names as given in User's Manual. Because the procedure given in NASTRAN's program manual are either incomplete or in error, attempts to add new functional modules with unique name and unique MPL (Module Properties List) have not been met any degree of success.

It is acknowledged that most of the information presented in this section may be found throughout the Programer's Manual. However, it will take a considerable time to assemble and use. Here we have presented the information we collected through trial and error and months of diligent work by two expert programers. We present it with the hopes that someone wishing to write their new functional modules will not have to encounter the same difficulties.

ACKNOWLEDGEMENT

The help of Mr. Archie Jordan and Mr. Tommy Franklin of Computer Science Corporation of Huntsville, Alabama in successfully completing the programing of this altered version of NASTRAN to solve the Complex Eigenvalue Analysis of Rotating Structures is appreciated. This effort was made possible by Bettye L. Harrison of Marshall Space Flight Center's Computation Laboratory.

REFERENCES


\[ [k''] = \begin{bmatrix} [k^1]^1 \\ \vdots \\ 0 \\ \vdots \\ [k^1]^n \end{bmatrix} \quad 6n \times 6n \]

where

\[ [k'']^i = \begin{bmatrix} [k_{11}^i] \\ \vdots \\ 0 \\ \vdots \\ [k_{22}^i] \end{bmatrix} \]

and

\[ [k_{11}^i] = \begin{bmatrix} -im^i(\omega_2^2 + \omega_3^2) & im^i\omega_2 & m^i\omega_3 \\ im^i\omega_2 & -m^i(\omega_1^2 + \omega_3^2) & m^i\omega_3 \\ m^i\omega_3 & m^i\omega_2 & -m^i(\omega_1^2 + \omega_2^2) \end{bmatrix} \]

\[ [k_{22}^i] = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \]

\[ k_{11} = (i\omega_3 - i\omega_3)(\omega_2^2 - \omega_3^2) + 4i\omega_3^2\omega_2 \omega_3 \\
+ i\omega_1^2\omega_2 + i\omega_1\omega_2 \omega_3 \]
\[
\begin{align*}
\mathbf{k}_{12} &= (I_{33}^1 - I_{11}^1)\omega_1 \omega_2 - 2I_{31}^1 \omega_2 \omega_3 \\
&\quad + I_{21}^1 (\omega_3^2 - \omega_2^2) - I_{32}^1 \omega_1 \omega_3 \\
\mathbf{k}_{13} &= I_{31}^1 (\omega_2^2 - \omega_3^2) - I_{32}^1 \omega_1^3 \\
&\quad - 2I_{21}^1 \omega_2 \omega_3 - (I_{11}^1 - I_{22}^1) \omega_1 \omega_3 \\
\mathbf{k}_{21} &= I_{21}^1 (\omega_2^2 - \omega_1^2) - I_{31}^1 \omega_2 \omega_3 \\
&\quad - (I_{33}^1 - I_{33}^1) \omega_1^2 - 2I_{32}^1 \omega_1 \omega_3 \\
\mathbf{k}_{22} &= (I_{33}^1 - I_{11}^1) (\omega_3^2 - \omega_1^2) + I_{32}^1 \omega_2 \omega_3 \\
&\quad + 4I_{31}^1 \omega_1 \omega_3 + I_{21}^1 \omega_2 \omega_1 \\
\mathbf{k}_{23} &= (I_{11}^1 - I_{22}^1) \omega_2 \omega_3 - 2I_{21}^1 \omega_1 \omega_3 \\
&\quad - I_{32}^1 (\omega_3^2 - \omega_1^2) - I_{31}^1 \omega_1 \omega_2 \\
\mathbf{k}_{31} &= (I_{22}^1 - I_{33}^1) \omega_1 \omega_3 - 2I_{32}^1 \omega_1 \omega_2 \\
&\quad - I_{31}^1 (\omega_2^2 - \omega_2^2) - I_{21}^1 \omega_2 \omega_3 \\
\mathbf{k}_{32} &= I_{32}^1 (\omega_1^2 - \omega_2^2) - I_{21}^1 \omega_1 \omega_3 \\
&\quad - (I_{33}^1 - I_{11}^1) \omega_2 \omega_3 - 2I_{31}^1 \omega_1 \omega_2 \\
\mathbf{k}_{33} &= (I_{11}^1 - I_{22}^1) (\omega_1^2 - \omega_2^2) \\
&\quad + 4I_{21}^1 \omega_1 \omega_2 + I_{31}^1 \omega_1 \omega_3 + I_{32}^1 \omega_2 \omega_3
\end{align*}
\]
\[
[P^a] = \begin{bmatrix}
[P_1^a] \\
[P_2^a] \\
\vdots \\
[P_n^a]
\end{bmatrix} \quad \text{where} \quad [P_1^a] = \begin{bmatrix}
[P_1^1] \\
[P_2^1]
\end{bmatrix}
\]

and

\[
[P_1^1] = \begin{bmatrix}
-m^2 \omega_2^2 R_{11} + \omega_1 \omega_2 R_{11} + \omega_1 \omega_2 R_{11} \\
-m^2 [\omega_1 \omega_2 R_{11} - (\omega_1^2 + \omega_2^2) R_{11} + \omega_2 \omega_3 R_{11} \\
-m^2 [\omega_1 \omega_3 R_{11} + \omega_2 \omega_3 R_{11} - (\omega_1^2 + \omega_2^2) R_{11}]
\end{bmatrix}
\]

\[
[P_2^1] = \begin{bmatrix}
(I_{22} - I_{33}) \omega_2 \omega_3 - I_{32} (\omega_2^2 - \omega_3^2) - I_{31} (\omega_1 \omega_2 - \omega_2 \omega_1) + I_{31} \omega_1 \omega_3 \\
(I_{33} - I_{11}) \omega_1 \omega_3 - I_{13} (\omega_1^2 - \omega_3^2) - I_{12} \omega_1 \omega_3 + I_{12} \omega_1 \omega_2 \\
(I_{11} - I_{22}) \omega_1 \omega_2 - I_{21} (\omega_1^2 - \omega_2^2) - I_{21} \omega_1 \omega_3 + I_{21} \omega_2 \omega_3
\end{bmatrix}
\]

\[
R_1^i = R_1^i - R_{1G} \\
R_2^i = R_2^i - R_{2G} \\
R_3^i = R_3^i - R_{3G}
\]

where
\[
R_{jG} = \sum_{i=1}^{n} \frac{R_{jG}^i m_i}{M} \quad j = 1, 2, 3
\]
\[
\begin{bmatrix}
\mathcal{G}^1 & 0 \\
\mathcal{G}^2 & 0 \\
\vdots & \vdots \\
\mathcal{G}^n & 0 \\
\mathcal{G}^n & 0
\end{bmatrix}
\]

\[6n \times 6n\]

where \(\mathcal{G}^1 = \begin{bmatrix} G_{11} & 0 \\ \vdots & \vdots \\ 0 & G_{22} \end{bmatrix}\)

and

\[
\begin{bmatrix}
0 & -2m^i_2 & 2m^i_2 \\
2m^i_3 & 0 & -2m^i_1 \\
-2m^i_2 & 2m^i_1 & 0
\end{bmatrix}
\]

\[
\begin{bmatrix}
0 & (I_{33}^i - I_{11}^i - I_{22}^i)w_3 & (I_{33}^i + I_{11}^i - I_{22}^i)w_2 \\
(I_{33}^i + I_{11}^i - I_{22}^i)w_3 + 2I_{33}^i w_2 + 2I_{33}^i w_1 & 0 & -2I_{33}^i w_3 - 2I_{21}^i w_1 \\
-2I_{33}^i w_2 - 2I_{33}^i w_1 & (I_{11}^i - I_{22}^i - I_{33}^i)w_1 & 0 \\
(I_{22}^i - I_{33}^i - I_{11}^i)w_2 & (I_{22}^i + I_{33}^i - I_{11}^i)w_1 + 2I_{21}^i w_2 + 2I_{31}^i w_3 & 0 \\
+2I_{32}^i w_3 + 2I_{21}^i w_1 & -2I_{21}^i w_2 - 2I_{31}^i w_3 & 0
\end{bmatrix}
\]
$$[\Pi] = \begin{bmatrix}
(1-\mu_1 \ 0 \ 0)
(0 \ 1-\mu_2 \ 0)
(0 \ 0 \ 1-\mu_3)
(1-\mu_1 \ 0 \ 0)
(0 \ 1-\mu_2 \ 0)
(0 \ 0 \ 1-\mu_3)
\vdots
\vdots
\vdots
\vdots
\vdots
\vdots
\end{bmatrix}^{\mu_1 \mu_2 \mu_3}$$

$$6n \times 6n$$
The inverse of this matrix is expressed in the following form, rather than inverting $[T]$ by some matrix inversion technique.

$$
[T]^{-1} =
\begin{pmatrix}
\frac{\mu_1}{S_{10}} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{\mu_2}{S_{20}} & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{\mu_3}{S_{30}} & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & \ddots & \ddots \\
0 & 0 & 0 & 0 & 0 & \frac{\mu_n}{S_{10}} \\
0 & 0 & 0 & 0 & 0 & \frac{\mu_n}{S_{20}} \\
0 & 0 & 0 & 0 & 0 & \frac{\mu_n}{S_{30}} \\
0 & 0 & 0 & 0 & 0 & \frac{\mu_n}{S_{40}} \\
0 & 0 & 0 & 0 & 0 & \frac{\mu_n}{S_{50}} \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots \\
0 & 0 & 0 & 0 & \frac{\mu_n}{S_{10}} & 0 \\
0 & 0 & 0 & \frac{\mu_n}{S_{20}} & 0 & 0 \\
0 & 0 & \frac{\mu_n}{S_{30}} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\end{pmatrix}
$$
where
\[
\mu_{ik} = \begin{cases} 
\frac{m_i}{M_0 + \sum_{j=1}^{n} m_j} & \text{if the motion of mass at node } i \text{ in } k\text{th direction is not constrained to be zero} \\
0 & \text{if the motion of mass at node } i \text{ in } k\text{th direction is constrained to be zero}
\end{cases}
\]

\[
S_{k0} = 1 - \sum_{j=1}^{n} \mu_{ik}
\]

where \(m_i\) is the mass at the ith node point of the total of 'n' nodes and \(I_{ik}\) is the moment of inertia of the rigid body at the ith node, \(M_0\) is the mass of the central rigid body and \(\{\omega\}\) is the spin vector.

Matrices \([G]\), \([K'']\), \([T]\), and \([T^{-1}]\) are the square matrices of the dimension \(6n \times 6n\). Rows and columns corresponding to the degree-of-freedom which are either constrained to be zero or have no mass should be removed. This will reduce the above matrices to \(N\times N\) where \(N\) is the total degree-of-freedom of the problem.
APPENDIX B

1 BEGIN 
EIGENVALUE ANALYSIS OF ROTATING STRUCTURES. $
2 GP1 GEOM1,GEOM2;/GPL,EQEXIN,GPDT,CSTM,BGPDT,SIL/V,N,LUSET/C,N,123/
3 V,N,N0GPDT $ 
4 SAVE LUSET $ 
5 GP2 GEOM2,EQEXIN/ECT $ 
6 PLTSET PCDB,EQEXIN/ECT/PLTSETX,PLTPAR,SPSETS,ELSETS/V,N,NSIL/V,N,
7 JUMPLOT $ 
8 SAVE NSIL,JUMPLOT $ 
9 PRTRSG PLTSETX/ // $ 
10 COND PL,JUMPPLOT $ 
11 PL0T PLTPAR,GPSETS,ELSETS,CASECC,BGPDT,EQEXIN,PLT,;/PL0T1X/V,N,
12 SAVE NSIL/V,N,LUSET/V,N,JUMPPLOT/V,N,PLTFLG/V,N,PFILE $ 
13 SAVE JUMPPLOT,PLTFLG,PFILE $ 
14 LABEL PI $ 
15 GP3 GEOM3,EQEXIN,GEOM2/SLT1GPTT/C,N,123/V,N,N0GRAV/C,N,123 $ 
16 TAI; ,ECT,EPT,BGPDT;SIL;GPTT,CSTM/EST,;GEI,ECTP,GPCT/V,N,LUSET/C,N,
17 123/V,N,N0SIMP/C,N,0/V,N,N0GENL/V,N,GENEL $ 
18 SAVE N0GENL,N0SIMP,GENEL $ 
19 PURGE BGPST/GENEL $ 
20 SMA1 CSTM,MPT,ECTP,GPTT;DIT/KGXX,GPS/T/V,N,N0GENL/V,N,N0K4GG $ 
21 SMA2 CSTM,MPT,ECTP,GPTT;DIT/MGXX;/V,Y,WTMMASS=1.0/V,N,N0MGG/V,N,N0BGG/
22 V,Y,CG0UPM0SS=-1 $ 
23 SAVE N0MGG $ 
24 COND ERR0R3,J0N0SIMP $ 
25 PURGE BGPST/GENEL $ 
26 GPW0 BGPDT,CSTM;E0GXX,00G/X,00G/W,V,Y,GRDPNT=1/V,Y,WTMMASS $ 
27 0 FP BGPW,;:///V,N,CARDN0 $ 
28 SAVE CARDN0 $ 
29 LABEL LBL1 $ 
30 EQU1V KGXX,KG0X/N0GENL $ 
31 COND LBL1;GR0PN0T $ 
32 COND ERR0R4,J0N0MGG $ 
33 GPW1 BGPDT,CSTM;E0GXX,00G/X,00G/W,V,Y,GRDPNT=1/V,Y,WTMMASS $ 
34 0 FP BGPW,;:///V,N,CARDN0 $ 
35 SAVE CARDN0 $ 
36 LABEL LBL1 $ 
37 EQU1V KGXX,KG0X/N0GENL $ 
38 COND LBL1;GR0PN0T $ 
39 COND ERR0R4,J0N0MGG $ 
40 GPW1 BGPDT,CSTM;E0GXX,00G/X,00G/W,V,Y,GRDPNT=1/V,Y,WTMMASS $ 
41 0 FP BGPW,;:///V,N,CARDN0 $ 
42 SAVE CARDN0 $ 
43 LABEL LBL1 $ 
44 COND LBL4D;REACT $ 
45 JUMP ERR0R2 $ 
46 LABEL LBL4D $ 
47 COND LBL4,G0NEL $ 
48 COND ERR0R3,J0N0SIMP $ 
49 GPSP GPL;GPST,LUSET,SIL/0GPST $ 
50 0 FP 0GPST,;:///V,N,CARDN0 $ 
51 SAVE CARDN0 $ 
52 LABEL LBL4 $ 
53 COND LBL2,MPCF2 $ 
54 MCE1 USET,KG0M $ 
55 MCE2 USET,GM,KG0X,KG0X,;//,KNN,MNN,$ $ 
56 LABEL LBL2 $ 
57 EQU1V KNN,KFF/SG0NE/L0M,0FF/SINGLE $ 
58 COND LBL3,SINGLE $ 
59
COND  LBL15,N0D $ 109
0FP  ØPHID,,,,,//V,N,CARDN0 $ 110
SAVE  CARDN0 $ 111
LABEL  LBL15 $ 112
COND  LBL16,N0P $ 113
EQUIV  PHID,CPHIP/N0A $ 114
COND  LBL17,N0A $ 115
SDR1  USETD,,PHID,,,,G00,GM,,KFS,OP/CPHIP,,QPC/C,N,1/C,N,DYNAMICS $ 116
LABEL  LBL17 $ 117
SDR2  CASECC,CSTM,MPD,DIT,EOQNY,STLO,,,,CLAMA,QPC,CPHIP,EST,OPQPC1, 118
ØCPHIP,ØESC1,ØEFC1/C,N,CEIG $ 119
0FP  ØCPHIP,ØOPO1,ØEFC1ØESC1,,//V,N,CARDN0 $ 120
SAVE  CARDN0 $ 121
LABEL  LBL16 $ 122
LABEL  FINIS $ 123
PRTPARM //C,N,-1/CIN,M0DES$ 124
LABEL  ERR0R2 $ 125
PRTPARM //C,N,-2/CIN,DIFSTIFS$ 126
LABEL  ERR0R4 $ 127
PRTPARM //C,N,-4/CIN,DIFSTIFS$ 128
LABEL  ERR0RS $ 129
PRTPARM //C,N,-5/CIN,DIFSTIFS$ 130
LABEL  FINIS $ 131
END  $ 132
Description of DMAP Operations for Eigenvalue Analysis of Rotating Structures

2. GP1 generates coordinate system transformation matrices, table of grid point locations, and tables for relating internal and external grid point numbers.

4. GP2 generates Element Connection Table with internal indices.

5. PLTSET transforms user input into a form used to drive structure plotter.

7. PRTMSG prints error messages associated with structure plotter.

10. Go to DMAP No. 14 if no undeformed structure plot request.

11. PLOT generates all requested undeformed structure plots.

13. PRTMSG prints plotter data and engineering data for each undeformed plot generated.

15. GP3 generates Grid Point Temperature Table.

16. TA1 generates element tables for use in matrix assembly and stress recovery.

18. Go to DMAP No. 123 and print error message if there are no structural elements.

20. SMA1 generates stiffness matrix \([ K_{gg}^x ]\) and Grid Point Singularity Table.

21. SMA2 generates mass matrix \([ M_{gg} ]\).

23. Go to DMAP No. 28 if no weight and balance request.

24. Go to DMAP No. 127 and print error message if no mass matrix exists.

25. GPWG generates weight and balance information.

26. OFP formats weight and balance information and places it on the system output file for printing.
29. Equivalence $[K^g_{gg}]$ to $[K_{gg}]$ if no general elements.

30. Go to DMAP No. 32 if no general elements.

31. SMA3 adds general elements to $[K^x_{gg}]$ to obtain stiffness matrix $[K_{gg}]$.

34. GP4 generates flags defining members of various displacement sets (USET), forms multipoint constraint equations $[R^g_g \{u\}_g] = 0$ and forms enforced displacement vector $[Y^s]$.

36. Go to DMAP No. 129 and print error message if no independent degrees of freedom are defined.

38. Equivalence $[K_{gg}]$ to $[K_{nn}]$ and $[M_{gg}]$ to $[M_{nn}]$ if no multipoint constraints.

39. Go to DMAP No. 41 if no free-body supports supplied.

40. Go to DMAP No. 125 and print error message if free-body supports are present.

42. Go to DMAP No. 46 if general elements present.

43. GPSP determines if possible grid point singularities remain.

44. OFP formats table of possible grid point singularities and places it on the system output file for printing.

47. Go to DMAP No. 50 if MCE1 and MCE2 have already been executed for current set of multipoint constraints.

48. MCE1 partitions multipoint constraint equations $[R^g_g] = [R^m_m \mid R^n_n]$ and solves for multipoint constraint transformation matrix $[G^m_m] = -[R^m_m]^{-1}[R^n_n]$.

49. MCE2 partitions stiffness and mass matrices

$$[K_{gg}] = \begin{bmatrix} K_{nn} & K_{nm} \\ K_{mn} & K_{mm} \end{bmatrix}$$

and

$$[M_{gg}] = \begin{bmatrix} M_{nn} & M_{nm} \\ M_{mn} & M_{mm} \end{bmatrix}$$
and performs matrix reductions

\[
\begin{bmatrix}
K_{nn}'
\end{bmatrix} = \begin{bmatrix}
\bar{K}_{nn}' + G_{m}^{T} [ K_{mn} ] + K_{mn}^{T} [ G_{m} ] \\
y [ G_{m}^{T} [ K_{mn} ] + [ K_{mn} ] [ G_{m} ] \end{bmatrix}
\]

\[
[ M_{nn}' ] = [ \bar{M}_{nn}' + [ G_{m}^{T} [ M_{mn} ] + [ M_{mn} ] [ G_{m} ] \\
y + [ G_{m}^{T} [ M_{mn} ] [ G_{m} ] ] .
\]

51. Equivalence \([ K_{nn}' ] \) to \([ K_{ff} ] \) and \([ M_{nn}' ] \) to \([ M_{ff} ] \) if no single-point constraints.

52. Go to DMAP No. 54 if no single-point constraints.

53. SCE1 partitions out single-point constraints.

\[
[ K_{nn} ] = \begin{bmatrix}
K_{ff} & K_{fs} \\
- K_{sf} & K_{ss}
\end{bmatrix}
\quad \text{and} \quad
[ M_{nn} ] = \begin{bmatrix}
M_{ff} & M_{fs} \\
- M_{sf} & M_{ss}
\end{bmatrix}.
\]

55. Equivalence \([ K_{ff} ] \) to \([ K_{aa} ] \) and \([ M_{ff} ] \) to \([ M_{aa} ] \) if no omitted coordinates.

56. Go to DMAP No. 58 if no omitted coordinates.

57. SMP1 partitions constrained stiffness and mass matrices

\[
[ K_{ff} ] = \begin{bmatrix}
\bar{K}_{aa} & K_{ao} \\
K_{oa} & K_{oo}
\end{bmatrix}
\quad \text{and} \quad
[ M_{ff} ] = \begin{bmatrix}
\bar{M}_{aa} & M_{ao} \\
M_{oa & M_{oo}}
\end{bmatrix}
\]

solves for transformation matrix \([ G_{o} ] = - [ K_{oo} ]^{-1} [ K_{oa} ] \),

and performs matrix reductions \([ K_{aa} ] = [ \bar{K}_{aa} ] + [ K_{oa} ] [ G_{o} ] \)

and \([ M_{aa} ] = [ \bar{M}_{aa} ] + [ M_{oa} ] [ G_{o} ] + [ G_{o} T ] [ M_{oa} ] \\
y + [ G_{o} T ] [ M_{oo} ] [ G_{o} ] .
\]

220
59. Dummy module DUMMOD1 constructs coriolis acceleration matrix \([ G] \), centripetal acceleration matrix \([ K''']\), transformation matrix \([ T]\) and its inverse \([ T]^{-1}\), and centrifugal load vector \(\{ P \}_g\). Rows and columns corresponding to the degrees of freedom constrained to be zero or have no mass have been removed from \([ G]\), \([ K''']\), \([ T]\), and \([ T]^{-1}\). Centrifugal load vector \(\{ P \}_g\) is in \(g\)-set and is reduced in the following D-MAP statements.

60. Equivalence \(\{ P \}_g\) to \(\{ P \}_1\) if no constraints applied.

61. RMBG2 decomposes constrained stiffness matrix \([ K''] = [ L'] [ U'']\).

62. Go to DMAP No. 64 if no constraints applied.

63. SSG2 applies constraints to static load vectors

\[
\{ P \}_g = \begin{pmatrix} \hat{P}_n \\ \hat{P}_m \end{pmatrix}, \quad \{ P \}_n = \{ \hat{P}_n \} + [ G]^T \{ P \}_m, \quad \{ P \}_n = \begin{pmatrix} \hat{P}_f \\ \hat{P}_s \end{pmatrix}, \quad \{ P \}_f = \{ \hat{P}_f \} - [ K_{fs}' ] \{ Y_s \}, \quad \{ P \}_f = \begin{pmatrix} P_a \\ P_o \end{pmatrix}, \quad \{ P \}_1 = \{ P_a \} + [ G_o^T ] \{ P \}_o.
\]

65. SSG3 solves for displacements of independent coordinates

\[
\{ u_1 \} = [ K_{11}]^{-1} \{ P \}_1,
\]
solves for displacements of omitted coordinates

\[
\{ u_o \} = [ K_{oo}]^{-1} \{ P \}_o,
\]
calculates residual vector (RULV) and residual vector error ratio for independent coordinates

\[
\delta P_1 = \{ P \}_1 - [ K_{11}] \{ u_1 \}
\]

\[
\epsilon_1 = \frac{\{ u_1 \}^T \{ \delta P_1 \}}{\{ P \}_1 \{ u_1 \}}
\]
and calculates residual vector (RUOV) and residual vector error ratio for omitted coordinates

\[ \{ \delta P_o \} = \{ P_o \} - \left[ K_{oo} \right] \{ u_o^0 \} , \]

\[ \epsilon_o = \frac{\{ u_o^T \} \{ \delta P_o \}}{\{ P_o^T \} \{ u_o^0 \}} . \]

66. Go to DMAP No. 69 if residual vector is not to be printed.

67. Print residual vector for independent coordinates (RULV).

68. Print residual vector for omitted coordinates (RUOV).

70. SDR1 recovers dependent displacements

\[ \{ u_o \} = \left[ G \right] \{ u_1 \} + \{ u_o^0 \} , \]

\[ \begin{bmatrix} u_a \\ u_o \end{bmatrix} = \begin{bmatrix} u_f \\ y_s \end{bmatrix} , \quad \begin{bmatrix} u_f \\ u_m \end{bmatrix} = \begin{bmatrix} u_n \\ u_m \end{bmatrix} = \begin{bmatrix} u_g \end{bmatrix} , \]

and recovers single-point forces of constraint

\[ \{ q_s \} = - \left[ P_s \right] \{ u_1 \} + \left[ K_{fs}^T \right] \{ u_f \} + \left[ K_{ss} \right] \{ y_s \} . \]

71. DUMMOD2 checks if vector \{ u_g \} is a null vector. IUGV = -1 if \{ u_g \} is null (geometric stiffness matrix KDGG is also a null matrix) otherwise IUGV = 0.

74. Go to DMAP No. 90 if IUGV = -1.

75. DSMG1 generates differential stiffness matrix \[ K_{gg}^P \].

77. ADD elastic and geometric stiffness matrices in g-set

\[ \left[ K_{gg}^x \right] + \left[ K_{gg}^P \right] = \left[ K_{gg}^d \right] . \]
78. Equivalence \([K_{gg}^d]\) to \([K_{nn}^d]\) if no multipoint constraints.

79. Go to DMAP No. 81 if no multipoint constraints.

80. MCE2 partitions differential stiffness matrix

\[
[K_{gg}^d] = \begin{bmatrix}
K_{nn}^d & K_{nm}^d \\
K_{mn}^d & K_{mm}^d
\end{bmatrix}
\]

and performs matrix reduction
\[
[K_{nn}^d] = [K_{nn}^d] + [G_m^T] [K_{mn}^d]
\]

\[
+ [K_{mn}^d] [G_m] + [G_m^T] [K_{mm}^d] [G_m].
\]

82. Equivalence \([K_{nn}^d]\) to \([K_{ff}^d]\) if no single-point constraints.

83. Go to DMAP No. 85 if no single-point constraints.

84. SCE1 partitions out single-point constraints.

\[
[K_{nn}^d] = \begin{bmatrix}
K_{ff}^d & K_{fs}^d \\
K_{sf}^d & K_{ss}^d
\end{bmatrix}
\]

86. Equivalence \([K_{ff}^d]\) to \([K_{aa}^d]\) if no omitted coordinates.

87. Go to DMAP No. 89 if no omitted coordinates.

88. SMP1 partitions constrained stiffness matrix

\[
[K_{ff}^d] = \begin{bmatrix}
-K_{aa}^d & K_{ao}^d \\
K_{oa}^d & -K_{oo}^d
\end{bmatrix},
\]

solves for transformation matrix \([G_{oo}] = -[K_{oo}]^{-1} [K_{oa}]\),
and performs matrix reductions $[K_{aa}^d] = [K_{aa}^d] + [K_{oa}^T][G_{oo}]$.

91. Equivalence $[K_{aa}^d]$ to $[K_{aa}]$ and $[G_o]$ to $[G_{oo}]$ if geometric stiffness matrix $[K_{gg}^d]$ is a null matrix.

93. Multiplies the matrices $[K_{aa}^d][T]^{-1} = [KSUM]$.

94. Multiplies the matrices $[K_{aa}^d][T]^{-1} = [KSUM2]$.

95. Adds matrix $KSUM$ and the centripetal acceleration matrix $[K''']$.

96. DPD generates flags defining members of various displacement sets used in dynamic analysis (USETD), tables relating internal and external grid point numbers, including extra points introduced for dynamic analysis, and prepared Transfer Function Pool and Eigenvalue Extraction Data.

97. Matrices $[R_p']$, $[NDOF]$ and $[T]^{-1}$ are output on magnetic tape.

$[R_p']$ is $(n \times 4)$ matrix where $n$ = no. of grid points. First three columns represent the coordinates of grid points in basic coordinates and fourth column stores the mass data at grid points.

$[NDOF]$ is $(3 \times n)$ matrix. Value of 1.5 is written if the translational D.O.F. at a grid point is not constrained by SPC, MPC, OMIT or permanent SPC on GRID cards. Otherwise it is written 0.0.

98. Matrices $[M_{aa}]$, $[K_{aa}]$, $[K_{aa}^d]$, $[K''']$, $[G]$ are output on magnetic tape.

$[K_{aa}]$ is the reduced elastic stiffness matrix

$[K_{aa}^d]$ is the reduced (elastic + geometric) stiffness matrix

$[K''']$ is the reduced centripetal acceleration matrix

$[G]$ is the reduced coriolis acceleration matrix.

99. CEAD extracts complex eigenvalues from the equation

$$[M_{dd}p^2 + B_{dd}p + K_{dd}]\{u_d\} = 0$$
and normalizes eigenvectors according to one of the following user requests:

1. Unit magnitude of selected coordinate
2. Unit magnitude of largest component.

101. OFP formats the summary of complex eigenvalues and summary of eigenvalue extraction information and places them on the system output file for printing.

103. Go to DMAP No. 121 if no eigenvalues found.

104. \( \{ \phi \} \), the eigenvector of

\[
[Mp^2 + Gp + \left( K'' + [K_e + K_g] T^{-1} \right)] \{ \phi \} = 0
\]

is given by complex eigenvalue analysis step #91.

\( \{ \phi_d \} \), the eigenvector of

\[
[MT_p^2 + GT_p + K''^T + K_e + K_g] \{ \phi_d \} = 0
\]

is obtained in this step \( \{ \phi_d \} = [T]^{-1} \{ \phi \} \).

105. Eigenvectors \( \{ \phi_d \} \) and \( \{ \phi \} \) are printed.

106. \( \begin{bmatrix} \delta^T \delta \end{bmatrix} \), a (3 x 3) matrix for each of the eigenvector \( \{ \phi_d \} \) is constructed and printed.

107. VDR prepares eigenvectors for output, using only the independent degrees of freedom.

109. Go to DMAP No. 112 if no output request for the independent degrees of freedom.

110. OFP formats the eigenvectors for independent degrees of freedom and places them on the system output file for printing.

113. Go to DMAP No. 121 is no output request involving dependent degrees of freedom or forces and stresses.

114. Equivalence \( \{ \phi_d \} \) to \( \{ \phi_p \} \) if no constraints applied.

115. Go to DMAP No. 117 if no constraints applied.
116. SDR1 recovers dependent components of eigenvectors

\[ \{ \phi_o \} = [G_{oo}] \{ \phi_d \}, \quad \{ \phi_f + \phi_e \} \]

\[ \{ \phi_f + \phi_e \} = \{ \phi_n + \phi_e \}, \quad \{ \phi_m \} = [G_{m}] \{ \phi_n + \phi_e \}, \]

\[ \{ \phi_n + \phi_e \} = \{ \phi_p \} \]

and recovers single-point forces of constraint

\[ \{ q_s \} = [K_{fs}^T] \{ \phi_f \}. \]

118. SDR2 calculates element forces and stresses (OESC1, OEFC1) and prepares eigenvectors and single-point forces of constraint for output (OCPHIP, OQPC1).

119. OFP formats tables prepared by SDR2 and places them on the system output file for printing.

122. Go to DMAP No. 131 and make normal exit.


126. Static analysis with differential stiffness error message No. 2 - Free body support not allowed.

128. Static analysis with differential stiffness error message No. 4 - Mass matrix required for weight and balance calculations.

130. Static analysis with differential stiffness error message No. 5 - No independent degrees of freedom have been defined.
SUBROUTINE DUMBDL
INTEGER TYPIN, TYPOUT, TDGF, SYSBUF, CORSZ
REAL M, MU(300)
EXTERNAL WRITE
EXTERNAL READ
DIMENSION HEAD(5), HEAD2(5), HEAD3(5), HEAD4(5), ISIDI(2)
D,(IC0NM(2), A(8), B(13), R(4), RG(4), W(5), M(300))
D,TDGF(300), NDEGF(6,300), NFRE(300), XKP(1)
D,AA(3,300), RP(300,3), MM(7)
D,RHEAD(15)
DIMENSION XI(1300)
D,XI21(300), XI22(300), XI31(300), XI32(300), XI33(300)
D,XMGG(300), D0F(6,300)
COMMON /DUMBPL/ XKP
COMMON /PACKXY TYPIN, TYPOUT, II, NI, INC
COMMON /UNPACKXY ITYPE, JJ, N, JINC
COMMON /SYSTEM/ SYSBUF, OUTAPE
COMMON XXX
EQUIVALENCE (AI(1), ID)
1, (AL(T), IA)
2, (BL2), IG)
C
OPEN CORE ARRAY
C
EQUIVALENCE (XXP(12000), NDEGF)
E, (XXP(3800), M )
E, (XXP(4100), NFRE)
E, (XXP(4400), TDGF)
E, (XXP(4700), A )
E, (XXP(4708), B )
E, (XXP(4721), R )
E, (XXP(4725), HEAD)
E, (XXP(4730), HEAD1)
E, (XXP(4735), HEAD2)
E, (XXP(4740), HEAD3)
E, (XXP(4745), HEAD4)
E, (XXP(4750), AA )
EQUIVALENCE (XXP(5650), RP )
E, (XXP(6550), MU )
E, (XXP(6850), XI11)
E, (XXP(7150), XI21)
E, (XXP(7450), XI22)
E, (XXP(7750), XI31)
E, (XXP(8050), XI32)
E, (XXP(8350), XI33)
E, (NDEGF, D0F)
DATA NAM1, NAM2, NAM3, NAM4 / 201, 202, 203, 204 /
C
IF(I101) REFERS TO GEOM1-THE FIRST INPUT DATA BLOCK OF DMAP
STATEMENT DUMBODL.
C
IC0NM REFERS TO MGG-THE SECOND INPUT DATA BLOCK OF DUMBODL.
C
IB6 REFERS TO BGPD-THE THIRD INPUT DATA BLOCK OF DUMBODL.
C
INW(104) REFERS TO Ww-THE BLOCK CONTAINING W-S AND R-T FLAGS.
C
NAM5 REFERS TO USE-THE FIFTH INPUT DATA BLOCK OF DUMBODL.
C
THE GENERAL PROCEDURE FOR READING A DATA BLOCK IS-FIRST,
OPEN THE FILE CONTAINING THE BLOCK SUCH AS IC0NM, IB6 ETC.
THEN SKIP THE HEADER RECORD BY CALLING FWDREC. THE NEXT STEP
IS TO DETERMINE IF THE DATA IS A MATRIX EG. MGG. FOR MATRICES,
THE NEXT STEP IS TO CALL UNPACK. EACH CALL TO UNPACK BRINGS INTO MEMORY ONE COLUMN OF THE MATRIX. FOR NON-MATRIX INPUT, THE RECORD MUST BE LOOKED UP IN THE NA crank PROGRAMMER'S MANUAL. EACH RECORD CAN THEN BE BROUGHT INTO MEMORY BY CALLING READ. ONE CALL TO READ BRINGS IN ONE RECORD.

THE EXCEPTION IN THIS CASE IS DATA READ EXACTLY AS IT APPEARS ON BULK DATA CARDS, FOR THIS THE PROCEDURE IS CALL PRELBC, THEN LOCATE, THEN READ FOR EACH BULK DATA CARD. THIS IS USED FOR IFL.

DATA NAM5; NAM6 /205, 206/
DATA NAM7; NAM8 /207,208/
DATA IGRID /4501,45/ I.FL /101/
DATA TW2, E0R /2,1/
DATA IC0N /102/ IC0NM /1501,15/IE6 /103/

THIS IS THE MAIN ROUTINE FOR COMPUTING THE K-PRIME, P-PRIME, G AND Mass matrices used in solving the rotating flexible structure

READ IN OMEGA VALUES AND RG + T-INVERSE FLAGS

IWW = 104
ITYPE = 1
JJ = 1
N=5
JINC = 1
CALL OPEN($1000, IWW, XKPLC0L+1), 0)
CALL FWDREC($1000, IWW)
CALL FNAME(IWW, RHEAD(1))
CALL UNPACK($1000, IWW, R I READ)
CALL CLOSE(IWW, 1)

WRITE COMMENTS ON OUTPUT LISTING

WRITE(6,601) 601 FORMAT(1H1, - RESTRICTIONS ****-//
F- 1. ID NO. 1 ON GRID CARDS SHOULD BE USED FOR THE CENTRAL RIGID -DM110000
F- BODY OTHERWISE THE TRANSFORMATION MATRICES T AND T-INVERSE -DM101000
F- WILL BE INCORRECT -/ DM102000
F- 2. THE USE OF THE T AND T-INVERSE PERMIT THE BASE MOTION WITH -DM103000
F- THE TRANSLATIONAL RIGID BODY MOTION SWEEP OUT FROM THE Eqs.-/DM104000
F- 0F MOTION -)
F- WRITE(6,604)

WRITE(6,605) 605 FORMAT(1H1, 4. LOCATION OF THE CENTER OF ROTATION (C.R.) GOVERNS THE-/ DM111000
F- CENTER OF MASS OF THE ENTIRE VEHICLE. IF THE AXIS OF ROTATION DI110000
F- IS TO PASS THROUGH A POINT OTHER THAN THE MASS CENTER, THE-/ DM111100
F- CORRESPONDING GRID POINT SHOULD BE DEFINED AS GRID NO. 1-) DM111200
F- WRITE(6,605)

WRITE(6,605)
F/ OM112290
F- IS C.G.-
F- W(5)=1.0 PERMITS THE CALCULATIONS OF T AND T-1/NEVERSE.-/ DM112300
F- W(5)=0.0 MAKES T AND T-1/NEVERSE IDENTITY MATRICES.-/ DM112500
F- IN GENERAL THE FOLLOWING COMBINATIONS SHOULD BE USED.-/ DM112650
F- W(4)=W(5)=0.0 OR W(4)=W(5)=1.0. W(4)=W(5)=0.0 ASSUMES THE- DM112700
F/ WRITE(6,899)
DM112800
899 FORMAT(1H)
F- STRUCTURE ROTATES ABOUT GRID W0. 1 AND ONLY THE CANTILEVER-/ DM113050
F- MODES ARE AVAILABLE. W(4)=W(5)=1.0 ASSUMES THE STRUCTURE-/ DM113120
F- ROTATES ABOUT THE C.G. WITH TRANSLATIONAL RIGID BODY DOF-/ DM113210
F- SWEPT OUT/- DM113300
F- W141=W151=0.0 OR W141=W151=1.0. W141=W151=0.0 ASSUMES THE-) DM113500
F- C. M. WILL BE COMPUTED-)
WRITE(6,606)
DM114000
606 FORMAT(1H)
F- 6. THE NUMBER OF GRID POINTS IN THE PROBLEM SHOUL) BE LESS-/ DM114200
F- THAN 300, THIS ALL5WS UP TO 1800 DOF.-/ DM114350
WRITE(6,6021)
DM114600
602 FORMAT(1H)- THE FOLLOWING DATA WAS TAKEN FROM GRID CARDS BY THE DM114550
FUMMY MODULE***-/
WRITE(6,612) W(1),W(2),W(3)
DM114700
612 FORMAT(1H)- -SPIN RATE VECTOR*** $MEGA1 = -F1O.4, 5X,
DM114800
F- $MEGA2 = -F10.4, 5X, -$MEGA3 = -F10.41 DM114900
IF(W(4) .LT. 0.01) WRITE(6,695)
DM115000
695 FORMAT(1H)- THE FOLLOWING ANALYSIS ASSUMES ROTATION ABOUT GRID NO. DM115100
1 1.-) DM115200
IF(W(4) .GT. 0.0) WRITE(6,696)
DM115300
696 FORMAT(1H)- IN THIS ANALYSIS THE STRUCTURE SPINS ABOUT THE C.G. OFDM115400
1 THE SYSTEM DESCRIBED IN THE BULK DATA-)
DM115500
IF(W(5) .LT. 0.01) WRITE(6,697)
DM115600
697 FORMAT(1H)- THE FOLLOWING ANALYSIS GIVES CANTILEVER MODES OF A SPIDM115700
IMPLICIT STRUCTURE BY MAKING T. AND-/1H , T-1/NEVERSE IDENTITY MATRICES-DM1158500
F) DM115900
IF(W(5) .GT. 0.0) WRITE(6,698)
DM116000
698 FORMAT(1H)- THE FOLLOWING ANALYSIS CALCULATES THE MATRICES T AND TDM116100
1-1/NEVERSE THUS SWEEPING OUT THE-/1H ,-TRANSLATIONAL RIGID BODY DOF-DM1162500
F) DM116300
WRITE(6,611)
DM116400
C SET UP PACK COMMON AND LOCATE END OF CORE DM116600
C DM116700
611 FORMAT(1H)- NODE DOF MASS I11 I21 I22 DM116800
F I31 I32 I33 R1 R2 R3-) DM116900
TYPIN = 1 DM117000
TYPNUT = 1 DM117100
IL = 1 DM117200
INCR = 1 DM117300
LZ = COSRZ(XXX, XKP) DM117400
LCOL = LZ - SYSBUF DM117500
IL = LCOL DM117600
IXX = LZ - 2*SYSBUF -2 DM117700
IBGR = IXX - SYSBUF - 1 DM117800
DB 750 IL=I3 DM118000
750 RO{11} = 0.0 DM118100
XN = 0.0 DM118130
LOCATE FILE THAT GRID POINTS ARE STORED ON
CALL PRELOC($1000; XKP(11), IFL)
CALL LOCATE($1000; XKP(11), IGRID, IFLG)

LOCATE FILE THAT BASIC GRID POINT CO-ORDINATES ARE STORED ON
CALL OPEN($1000, IBG, XKP(IBGR), 0)
CALL FWDREC($700, IBG)
NODE = 0
BUILD MASS AND INERTIA TABLE
UNPACK MGG MATRIX TO GET MASS AND INERTIAS

N = 300
CALL OPEN($1000, ICON, XKP(1XX), 0)
CALL FWDREC($1000, ICON)
CALL FNAMEN(ICON, RHEAD(1))

WHEN AN END OF FILE IS ENCOUNTERED BY FWDREC, READING OF MGG STOPS

950 CALL FWDREC($699, ICON)
   CALL FWDREC($699, ICONF)
   NODE = NODE + 1
   ND1 = (NODE-1)*6
   CALL UNPACK($951, ICON, XMGG; READ)
   M(NODE) = XMGG(ND1+3)
   XM = XM + M(NODE)
951 CALL UNPACK($952, ICON, XMGG; READ)
   XI11(NODE) = XMGG(ND1+4)
952 CALL UNPACK($953, ICON, XMGG; READ)
   XI21(NODE) = XMGG(ND1+4)
   XI21(NODE) = XMGG(ND1+5)
953 CALL UNPACK($950, ICON, XMGG; READ)
   XI31(NODE) = XMGG(ND1+4)
   XI32(NODE) = XMGG(ND1+5)
   XI33(NODE) = XMGG(ND1+6)
G0 TO 950
699 NODE = 0
700 CONTINUE
   CALL READ($710, $710, IFL; A; 8, 0, IFLG)
   NODE = NODE + 1
BASIC GRID POINT INFO.
   CALL READ($710, $710, IBG; R, 4, 0, IFLG)
   IF MOTION AT A GRID POINT IS RESTRAINED IN ALL DIRECTIONS THE LOGIC
   IN THE CODE CAUSES THE GRID POINT TO BE DISREGARDED
   WRITE(6,610) ID, IA, M(NODE), XI11(NODE), XI21(NODE), XI22(NODE)
   W, XI31(NODE), XI32(NODE), XI33(NODE), R(2), R(3),
   W R(4)
   FORMAT(115,118, 10F10:4)
   IF(IA.GT.0.0) G0 TO 720
   IF(IA.GT.1.0) G0 TO 725
   DQ 733 II=1,3
733 RG(II) = R(II+1)
   G0 TO 725
720 CONTINUE
   DQ 701 II=1,3
701 RG(II) = RG(II) + M(NODE)*R(II+1)

230
725 CONTINUE
710 CONTINUE
1000 CONTINUE
IF(WIN(4) .EQ. 0.0) GO TO 719
GO TO 700
734 RG(II) = RG(II)/XM
719 CONTINUE
WRITE(6,607)
607 FORMAT(1H* - RG GIVES THE CENTER OF ROTATION - )
WRITE(6,615) XM, RG(1), RG(2), RG(3)
615 FORMAT(1H* - TOTAL MASS = -1E15.8, - RG(1) = -1E15.8, - RG(2) = -1E15.8, - RG(3) = -1E15.8)
F= -1E15.8* - RG(3) = -1E15.8

620 FORMAT(1H* - MOTION CONSTRAINTS IN 1 THRU 6 DIRECTION AND TOTAL DOF - )
CALL CLOSE(FL, 1)
CALL CLOSE(ICON, 1)
C******************************************************************************
C NAM1(201) REFERS TO PG-THE FIRST OUTPUT DATA BLOCK OF DUMM01.
C NAM2(200) REFERS TO D0F-THE EIGHTH OUTPUT DATA BLOCK OF DUMM01.
C THE PROCEDURE FOR PACKING A MATRIX IS-OPEN THE FILE(FG. 201),
C CALL FNAME, CALL WRITE TO WRITE THE HEADER RECORD, THEN PACK.
C ALSO BUILD THE TRAILER ARRAY, THEN CALL WRITRA. SEE 2.2-1 OF THE
C NASTRAN PROGRAMMER'S MANUAL FOR TRAILER INFORMATION.
C ALWAYS CLOSE THE FILES OUT AS SOON AS POSSIBLE
C CALL CLOSE(IG, 1)
CALLymmetric MTRel ROWs AND C0LS. OF EACH 6X6 SUBMATRIX WHERE THE DEGREE
OF FREEDOM IS NOT USED,i E FOR NDEG(3) = 0 , THIRD ROW AND COL ARE ZER0.
C THE NON-ZERO ELEMENTS ARE THEN MOVED TO THE TOP LEFT CORNER OF SUBMAT.
C CALL CLOSE(IG, 1)
CALL OPEN(500, NAM1, XKPLC0L+1), 1)
CALL FNAME(NAM1, HEAD1(1))
CALL WRITE(NAM1, HEAD1(1), TWO, END)
C ZER0 OUT ROWS AND C0LS. OF EACH 6X6 SUBMTRIX WHERE THE DEGREE OF
C FREEDOM IS NOT USED,i E FOR NDEG(3) = 0 , THIRD ROW AND COL ARE ZER0.
C THE NON-ZERO ELEMENTS ARE THEN MOVED TO THE TOP LEFT CORNER OF SUBMAT.
C ZER0 OUT ROWS AND C0LS. OF EACH 6X6 SUBMTRIX WHERE THE DEGREE OF
C FREEDOM IS NOT USED,i E FOR NDEG(3) = 0 , THIRD ROW AND COL ARE ZER0.
C THE NON-ZERO ELEMENTS ARE THEN MOVED TO THE TOP LEFT CORNER OF SUBMAT.
C CALL OPEN(500, NAM1, XKPLC0L+1), 1)
CALL FNAME(NAM1, HEAD1(1))
C******************************************************************************
CALL WRITE(NAM2, HEAD2(1), TW0, E0R)
CALL OPEN($501, IBG, XKP(1BGR), 0)
CALL FNDREC($501, IBG)
CALL PRIMM(RG, NODE, W)
501 CONTINUE
CALL CLOSE(NAM2, 1)
CALL CLOSE(IBGR, 1)
C
C #INIT OPEN CORC
C
DO 2001 II=1,N1
2001 XKPI(II) = 0.0
CALL OPEN($3000, NAM3$, XKP(LC0L+1), 1)
CALL FNAME(NAM3$, HEAD3(1)
CALL WRITE(NAM3$, HEAD3(1), TW0, E0R)
CALL GMATNXM, NODE, W)
CALL CLOSE(NAM3, 1)
3000 CONTINUE
CALL CLOSE(NAM3, 1)
C
C #INIT OPEN CORC
C
DO 3001 II=1,N1
3001 XKPI(II) = 0.0
CALL OPEN($4000, NAM4$, XKP(LC0L+1), 1)
CALL FNAME(NAM4$, HEAD4(1)
CALL WRITE(NAM4$, HEAD4(1), TW0, E0R)
C
CALL AMAT(NODE)
CALL CLOSE(NAM4, 1)
C
CALL OPEN($4000, NAM5$, XKP(LC0L+1), 1)
CALL FNAME(NAM5$, HEAD4(1)
CALL WRITE(NAM5$, HEAD4(1), TW0, E0R)
CALL TMAT(NODE, W, WIT, XM
CALL CLOSE(NAM5, 1)
C
CALL OPEN($4000, NAM6$, XKP(LC0L+1), 1)
CALL FNAME(NAM6$, HEAD4(1)
CALL WRITE(NAM6$, HEAD4(1), TW0, E0R)
CALL TMAT(NODE, W, WIT, XM
CALL CLOSE(NAM6, 1)
C
C PACK RP MATRIX
C
CALL OPEN($4000, NAM7$, XKP(LC0L+1), 1)
CALL FNAME(NAM7$, HEAD4(1)
CALL WRITE(NAM7$, HEAD4(1), TW0, E0R)
N1 = NODE
MM(1) = 207
MM(2) = 0
MM(3) = NODE
MM(4) = 2
MM(5) = 1
MM(6) = NODE
DO 3500 II=1,3
CALL PACKRP(II, NAM7$, WRITE, MM)
3500 CONTINUE
C
C PACK MASSES
C
232
CALL PACK(M, NAM7$ WRITE, MM)
CALL WRTRRL(MM(1))
CALL CLOSE(NAM7, 1)

CALL OPEN($4000, NAM8, XKP(LC0L+1), 1)
CALL FNAME(NAM8, HEAD4(1))
CALL WRITE(NAM8, HEAD4(1)$ TW0, E0R)
DO 3550 II=1,6
DO 3550 J1=1,N0DE
3550 IF(ND4EGF(II,J1).GT. 0) DOF(II,J1) = 1,5
   NI = 3
   MM(1) = 208
   MM(2) = 0
   MM(3) = 6
   MM(4) = 2
   MM(5) = 1
   MM(6) = 6
   DO 3600 II=1,NI
   CALL PACK(I, DOF(II)*NAM8, WRITE, MM)
3600 CONTINUE
CALL WRTRRL(MM(1))
CALL CLOSE(NAM8, 1)
4000 CONTINUE
RETURN
END

For complete listing of this program on Univac 1108 Computer, write to Reference 3.
FIGURE 1. GEOMETRY OF SPINNING FLEXIBLE APPENDAGE AND CENTRAL BODY