COMPLEX EIGENVALUE ANALYSIS OF ROTATING STRUCTURES

Jayant S. Patel
Principal Engineer
Teledyne Brown Engineering Company
Huntsville, Alabama

S. M. Seltzer
Sr. Research Engineer, Astrionics Lab.
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

INTRODUCTION

14

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[1,2]

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

$$\underline{F}^s = m^s \underline{A}^s$$

$$\underline{T}^s = \frac{{}^i d}{dt} \underline{H}^s \tag{1}$$

for the s^{th} rigid body of a flexible appendage; where m^s is the mass, $\underline{A}^s$ is the absolute translational acceleration vector. $\underline{F}^s$ and $\underline{T}^s$ are the sum of the external and connection force and torque vectors, respectively. $\underline{H}^s$ 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 ω (fig. 1), the expression for acceleration is written as

$$\underline{A}^s = \frac{{}^b d^2}{dt^2} (\underline{c} + \underline{u}^s) + 2\underline{\omega} \times \frac{{}^b d}{dt} (\underline{c} + \underline{u}^s) + \underline{\omega} \times (\underline{\omega} \times (\underline{c} + \underline{u}^s)) + \underline{\omega} \times (\underline{\omega} \times \underline{r}^s)$$

$$+ \frac{{}^i d}{dt} \underline{\omega} \times (\underline{c} + \underline{u}^s + \underline{r}^s) + \frac{{}^i d^2 \underline{R}}{dt^2} \tag{2}$$

$$\underline{T}^s = \frac{{}^i d}{dt} (I^s \cdot \omega^s) = \frac{{}^i d}{dt} (\underline{I}^s \cdot (\underline{\omega} + \underline{\beta}^s))$$

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

$$+ \underline{\omega} \times I^s \cdot \underline{\omega} + \underline{\omega} \times \underline{I}^s \cdot \frac{{}^b d}{dt} \underline{\beta}^s + \frac{{}^b d}{dt} \underline{\beta}^s \times \underline{I}^s \cdot \underline{\omega} \tag{3}$$

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

$$\underline{A}^s = \frac{{}^i d^2}{dt^2} (\underline{R} + \underline{c} + \underline{r}^s + \underline{u}^s) \tag{2-a}$$

$$\underline{T}^s = \underline{I}^s \cdot \frac{{}^i d^2}{dt^2} \underline{\beta}^s \quad . \tag{3-a}$$

In matrix notation the second term on the righ 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

$$\underline{c} = -\frac{1}{M} \sum_{s=1}^{n} m^s \underline{u}^s \tag{4}$$

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:

$$[M'] \{\ddot{u}\} + [G'] \{\dot{u}\} + [K''' + Ke + Kg] \{u\} = [F] . \tag{5}$$

The steady state equation in matrix form is:

$$[Ke + K'''] \{u\}^s = \{P\}^s . \tag{6}$$

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:

$$[M'] = [M]\ [T]$$
$$[G'] = [G]\ [T] \qquad (7)$$
$$[K'''] = [K'']\ [T]$$

Relations (7) afford a transformation

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

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

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

where

u = Vector of displacements from the steady state configuration of the nodal masses in spinning body frame.

T = 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 = NASTRAN generated mass matrix

G = Dummy module generated coriolis acceleration matrix

K'' = Dummy module generated centripetal acceleration matrix

Pg = Dummy module generated steady state centrifugal force vector

Kg = NASTRAN generated differential stiffness matrix using the above load vector Pg

Ke = NASTRAN generated elastic stiffness matrix.

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 WW with five elements using DMI* cards. The first of the two cards never changes for this program. WW(1), WW(2), WW(3) are the components of the spin vector in the body frame. Terms WW(4), WW(5) can take the values either 0.0 or 1.0. If WW(4) = 1.0 the structure is spinning about the mass center of vehicle, and if 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 WW(5) = 1.0. If 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 WW(4) and WW(5).

Case I. WW(4) = 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. WW(4) = 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. WW(4) = 1.0 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. WW(4) = 0.0 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 (PHI in DMAP) thus obtained are transformed to u_i (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.

```
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 Programer'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 Programer'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 Mgg 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 programer'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
    .
    .
    .
CALL READ
CALL CLOSE
```

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
    .
    .
    .
CALL READ
CALL CLOSE
```

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

⋮

M(7) = M(6) 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 WRTRL 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

1. Likins, P. W.: Dynamics and Control of Flexible Space Vehicles. NASA TR-32-1329, Revision 1, January 1970.

2. Patel, J. S. and Seltzer, S. M.: Complex Eigenvalue Solution to a Spinning Skylab Problem. NASA TM-X-2378, Vol. II, September 1971.

3. "Eigenvalue Analysis of Rotating Structures." Job No. 310261, April 25, 1972. NASA Computation Lab. Bldg. 4663. Marshall Space Flight Center, Huntsville, Alabama 35805.

APPENDIX A

$$[K''] = \begin{bmatrix} [K']^1 & & & & & & \\ & [K']^2 & & & & 0 & \\ & & [K']^3 & & & & \\ & & & \cdot & & & \\ & 0 & & & \cdot & & \\ & & & & & \cdot & \\ & & & & & & [K']^n \end{bmatrix}_{6n \times 6n} \qquad \text{A-1}$$

where

$$[K']^i = \left[\begin{array}{c|c} [K'_{11}]^i & 0 \\ \hline 0 & [K'_{22}]^i \end{array}\right]$$

and

$$[K'_{11}]^i = \begin{bmatrix} -m^i(\omega_2^2 + \omega_3^2) & m^i\omega_1\omega_2 & m^i\omega_1\omega_3 \\ m^i\,\omega_1\omega_2 & -m^i(\omega_1^2 + \omega_3^2) & m^i\omega_2\omega_3 \\ m^i\omega_1\omega_3 & m^i\omega_2\omega_3 & -m^i(\omega_1^2 + \omega_2^2) \end{bmatrix}$$

$$[K'_{22}] = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix}$$

$$k_{11} = (I_{22}^i - I_{33}^i)(\omega_2^2 - \omega_3^2) + 4I_{23}^i\omega_2\omega_3 + I_{12}^i\omega_1\omega_2 + I_{13}^i\omega_1\omega_3$$

$$k_{12} = \begin{aligned}&(I^i_{33} - I^i_{11})\omega_1\omega_2 - 2I^i_{31}\omega_2\omega_3\\ &+ I^i_{21}(\omega_3^2 - \omega_2^2) - I^i_{32}\omega_1\omega_3\end{aligned}$$

$$k_{13} = \begin{aligned}&I^i_{31}(\omega_2^2 - \omega_3^2) - I^i_{32}\omega_1\omega_2\\ &-2I^i_{21}\omega_2\omega_3 - (I^i_{11} - I^i_{22})\omega_1\omega_3\end{aligned}$$

$$k_{21} = \begin{aligned}&I^i_{21}(\omega_3^2 - \omega_1^2) - I^i_{31}\omega_2\omega_3\\ &-(I^i_{22} - I^i_{33})\omega_1\omega_2 - 2I^i_{32}\omega_1\omega_3\end{aligned}$$

$$k_{22} = (I^i_{33} - I^i_{11})(\omega_3^2 - \omega_1^2) + I^i_{32}\omega_2\omega_3 + 4I^i_{31}\omega_1\omega_3 + I^i_{21}\omega_2\omega_1$$

$$k_{23} = \begin{aligned}&(I^i_{11} - I^i_{22})\omega_2\omega_3 - 2I^i_{21}\omega_1\omega_3\\ &-I^i_{32}(\omega_3^2 - \omega_1^2) - I^i_{31}\omega_1\omega_2\end{aligned}$$

$$k_{31} = \begin{aligned}&(I^i_{22} - I^i_{33})\omega_1\omega_3 - 2I^i_{32}\omega_1\omega_2\\ &-I^i_{31}(\omega_1^2 - \omega_2^2) - I^i_{21}\omega_2\omega_3\end{aligned}$$

$$k_{32} = I^i_{32}(\omega_1^2 - \omega_2^2) - I^i_{21}\omega_1\omega_3 - (I^i_{33} - I^i_{11})\omega_2\omega_3 - 2I^i_{31}\omega_1\omega_2$$

$$k_{33} = (I^i_{11} - I^i_{22})(\omega_1^2 - \omega_2^2) + 4I^i_{21}\omega_1\omega_2 + I^i_{31}\omega_1\omega_3 + I^i_{32}\omega_2\omega_3$$

$$[P^s] = \begin{bmatrix} [P']^1 \\ [P']^2 \\ [P']^3 \\ \vdots \\ [P']^n \end{bmatrix}_{6n} \qquad \text{where } [P']^i = \begin{bmatrix} [P'_1] \\ \text{----} \\ [P'_2] \end{bmatrix}, \tag{A-2}$$

and

$$[P'_1] = \begin{bmatrix} -m^i[-(\omega_2^2 + \omega_3^2)R_1^{'i} + \omega_1\omega_2 R_2^{'i} + \omega_1\omega_3 R_3^{'i} \\ -m^i[\omega_1\omega_2 R_1^{'i} - (\omega_1^2 + \omega_3^2)R_2^{'i} + \omega_2\omega_3 R_3^{'i} \\ -m^i[\omega_1\omega_3 R_1^{'i} + \omega_2\omega_3 R_2^{'i} - (\omega_1^2 + \omega_2^2)\ R_3^{1i} \end{bmatrix}$$

$$[P'_2] = \begin{bmatrix} (I_{22}^i - I_{33}^i)\omega_2\omega_3 - I_{32}^i(\omega_2^2 - \omega_3^2) - I_{31}^i\ \omega_1\omega_2 + I_{21}^i\ \omega_1\omega_3 \\ (I_{33}^i - I_{11}^i)\omega_1\omega_3 - I_{31}^i(\omega_3^2 - \omega_1^2) - I_{21}^i\omega_2\omega_3 + I_{32}^i\omega_1\omega_2 \\ (I_{11}^i - I_{22}^i)\omega_1\omega_2 - I_{21}^i(\omega_1^2 - \omega_2^2) - I_{32}^i\omega_1\omega_3 + I_{31}^i\omega_2\omega_3 \end{bmatrix}$$

$$R_1^{'i} = R_1^i - R_{1G}$$

$$R_2^{'i} = R_2^i - R_{2G}$$

$$R_3^{'i} = R_3^i - R_{3G}$$

$$\text{where } R_{jG} = \sum_{i=1}^{n} R_j^i m_i/M \qquad j = 1, 2, 3$$

$$[G] = \begin{bmatrix} [G]^1 & & & & \\ & [G]^2 & & & \\ & & [G]^3 & & 0 \\ & 0 & & \ddots & \\ & & & & [G]^n \end{bmatrix}_{6n \times 6n} \qquad \text{A-3}$$

$$\text{where } [G]^i = \left[\begin{array}{c|c} G_{11} & 0 \\ \hline 0 & G_{22} \end{array}\right]$$

and

$$G_{11} = \begin{bmatrix} 0 & -2m^i\omega_3 & 2m^i\omega_2 \\ 2m^i\omega_3 & 0 & -2m^i\omega_1 \\ -2m^i\omega_2 & 2m^i\omega_1 & 0 \end{bmatrix}$$

$$G_{22} = \begin{bmatrix} 0 & \begin{array}{c}(I^i_{33} - I^i_{11} - I^i_{22})\omega_3 \\ +2I^i_{32}\omega_2 + 2I^i_{31}\omega_1\end{array} & \begin{array}{c}(I^i_{33} + I^i_{11} - I^i_{22})\omega_2 \\ -2I^i_{32}\omega_3 - 2I^i_{21}\omega_1\end{array} \\ \begin{array}{c}(I^i_{11} + I^i_{22} - I^i_{33})\omega_3 \\ -2I^i_{32}\omega_2 - 2I^i_{31}\omega_1\end{array} & 0 & \begin{array}{c}(I^i_{11} - I^i_{22} - I^i_{33})\omega_1 \\ +2I^i_{21}\omega_2 + 2I^i_{31}\omega_3\end{array} \\ \begin{array}{c}(I^i_{22} - I^i_{33} - I^i_{11})\omega_2 \\ +2I^i_{32}\omega_3 + 2I^i_{21}\omega_1\end{array} & \begin{array}{c}(I^i_{22} + I^i_{33} - I^i_{11})\omega_1 \\ -2I^i_{21}\omega_2 - 2I^i_{31}\omega_3\end{array} & 0 \end{bmatrix}$$

$$
[T] = \left[\begin{array}{cccccccc}
\begin{pmatrix}1-\mu_1^1 & 0 & 0\\ 0 & 1-\mu_2^1 & 0\\ 0 & 0 & 1-\mu_3^1\end{pmatrix} &
\begin{pmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix} &
\begin{pmatrix}-\mu_1^2 & 0 & 0\\ 0 & -\mu_2^2 & 0\\ 0 & 0 & -\mu_3^2\end{pmatrix} &
\begin{pmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix} & \cdots &
\begin{pmatrix}-\mu_1^n & 0 & 0\\ 0 & -\mu_2^n & 0\\ 0 & 0 & -\mu_3^n\end{pmatrix} &
\begin{pmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix} \\
\begin{pmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix} &
\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix} &
\begin{pmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix} &
\begin{pmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix} & \cdots &
\begin{pmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix} &
\begin{pmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix} \\
\begin{pmatrix}-\mu_1^1 & 0 & 0\\ 0 & -\mu_2^1 & 0\\ 0 & 0 & -\mu_3^1\end{pmatrix} &
\begin{pmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix} & & & &
\begin{pmatrix}-\mu_1^n & 0 & 0\\ 0 & -\mu_2^n & 0\\ 0 & 0 & -\mu_3^n\end{pmatrix} &
\begin{pmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix} \\
\begin{pmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix} &
\begin{pmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix} & & \ddots & & \vdots & \vdots \\
\vdots & \vdots & & & & & \\
\begin{pmatrix}-\mu_1^1 & 0 & 0\\ 0 & -\mu_2^1 & 0\\ 0 & 0 & -\mu_3^1\end{pmatrix} &
\begin{pmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix} & & \cdots & &
\begin{pmatrix}1-\mu_1^n & 0 & 0\\ 0 & 1-\mu_2^n & 0\\ 0 & 0 & 1-\mu_3^n\end{pmatrix} &
\begin{pmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix} \\
\begin{pmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix} &
\begin{pmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix} & & \cdots & &
\begin{pmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix} &
\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix}
\end{array}\right]_{6n \times 6n}
$$

6n x 6n

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

$$
[T]^{-1} = \left[\begin{array}{cccccccccccccccccccc}
1+\frac{\mu_1^1}{S_{10}} & 0 & 0 & 0 & 0 & 0 & \frac{\mu_1^2}{S_{10}} & 0 & 0 & 0 & 0 & 0 & & \frac{\mu_1^n}{S_{10}} & 0 & 0 & 0 & 0 & 0 \\
0 & 1+\frac{\mu_2^1}{S_{20}} & 0 & 0 & 0 & 0 & 0 & \frac{\mu_2^2}{S_{20}} & 0 & 0 & 0 & 0 & \cdots & 0 & \frac{\mu_2^n}{S_{20}} & 0 & 0 & 0 & 0 \\
0 & 0 & 1+\frac{\mu_3^1}{S_{30}} & 0 & 0 & 0 & 0 & 0 & \frac{\mu_3^2}{S_{30}} & 0 & 0 & 0 & & 0 & 0 & \frac{\mu_3^n}{S_{30}} & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\mu_1^1}{S_{10}} & 0 & 0 & 0 & 0 & 0 & & & & & & & & \frac{\mu_1^n}{S_{10}} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{\mu_2^1}{S_{20}} & 0 & 0 & 0 & 0 & & & & & & & & 0 & \frac{\mu_2^n}{S_{20}} & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{\mu_3^1}{S_{30}} & 0 & 0 & 0 & & & & \ddots & & & & 0 & 0 & \frac{\mu_3^n}{S_{30}} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\
 & \vdots & & & \vdots & & & & & & & & & & \vdots & & & \vdots & \\
\frac{\mu_1^1}{S_{10}} & 0 & 0 & 0 & 0 & 0 & & & & & & & & 1+\frac{\mu_1^n}{S_{10}} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{\mu_2^1}{S_{20}} & 0 & 0 & 0 & 0 & & & & \cdots & & & & 0 & 1+\frac{\mu_2^n}{S_{20}} & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{\mu_3^1}{S_{30}} & 0 & 0 & 0 & & & & & & & & 0 & 0 & 1+\frac{\mu_3^n}{S_{30}} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & & & & \cdots & & & & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & 0 & 0 & 0 & 0 & 0 & 1
\end{array}\right]_{6n \times 6n}
$$

where

$$\mu_k^i = \begin{cases} \dfrac{m^i}{M_0 + \sum\limits_{j=1}^{n} m^j} & \text{if the motion of mass at node i in kth direction is not constrained to be zero} \\ 0 & \text{if the motion of mass at node i in kth direction is constrained to be zero} \end{cases}$$

$$S_{ko} = 1 - \sum_{j=1}^{n} \mu_k^i$$

where m^i is the mass at the ith node point of the total of 'n' nodes and I_{jk}^i is the moment of inertia of the rigid body at the ith node, M_o 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 x 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 NxN where N is the total degree-of-freedom of the problem.

APPENDIX B

```
 1 BEGIN     EIGENVALUE ANALYSIS OF ROTATING STRUCTURES. $
 2 GP1       GEØM1,GEØM2,/GPL,EQEXIN,GPDT,CSTM,BGPDT,SIL/V,N,LUSET/C,N,123/
             V,N,NØGPDT  $
 3 SAVE      LUSET  $
 4 GP2       GEØM2,EQEXIN/ECT  $
 5 PLTSET    PCDB,EQEXIN,ECT/PLTSETX,PLTPAR,GPSETS,ELSETS/V,N,NSIL/V,N,
             JUMPPLØT $
 6 SAVE      NSIL,JUMPPLØT $
 7 PRTMSG    PLTSETX// $
 8 SETVAL    //V,N,PLTFLG/C,N,1/V,N,PFILE/C,N,0 $
 9 SAVE      PLTFLG,PFILE $
10 CØND      P1,JUMPPLØT $
11 PLØT      PLTPAR,GPSETS,ELSETS,CASECC,BGPDT,EQEXIN,SIL,,/PLØTX1/V,N,
             NSIL/V,N,LUSET/V,N,JUMPPLØT/V,N,PLTFLG/V,N,PFILE $
12 SAVE      JUMPPLØT,PLTFLG,PFILE $
13 PRTMSG    PLØTX1// $
14 LABEL     P1 $
15 GP3       GEØM3,EQEXIN,GEØM2/SLT,GPTT/C,N,123/V,N,NØGRAV/C,N,123 $
16 TA1       ,ECT,EPT,BGPDT,SIL,GPTT,CSTM/EST,,GEI,ECPT,GPCT/V,N,LUSET/C,N,
             123/V,N,NØSIMP/C,N,0/V,N,NØGENL/V,N,GENEL $
17 SAVE      NØGENL,NØSIMP,GENEL  $
18 CØND      ERRØR1,NØSIMP $
19 PURGE     ØGPST/GENEL $
20 SMA1      CSTM,MPT,ECPT,GPCT,DIT/KGGX,,GPST/V,N,NØGENL/V,N,NØK4GG $
21 SMA2      CSTM,MPT,ECPT,GPCT,DIT/MGG,/V,Y,WTMASS=1.0/V,N,NØMGG/V,N,NØBGG/
             V,Y,CØUPMASS=-1 $
22 SAVE      NØMGG  $
23 CØND      LBL1,GRDPNT $
24 CØND      ERRØR4,NØMGG $
25 GPWG      BGPDT,CSTM,EQEXIN,MGG/ØGPWG/V,Y,GRDPNT=-1/V,Y,WTMASS $
26 ØFP       ØGPWG,,,,,//V,N,CARDNØ $
27 SAVE      CARDNØ  $
28 LABEL     LBL1 $
29 EQUIV     KGGX,KGG/NØGENL $
30 CØND      LBL11,NØGENL $
31 SMA3      GEI,KGGX/KGG/V,N,LUSET/V,N,NØGENL/V,N,NØSIMP $
32 LABEL     LBL11 $
33 PARAM     //C,N,MPY/V,N,NSKIP/C,N,0/C,N,0 $
34 GP4       CASECC,GEØM4,EQEXIN,SIL,GPDT/RG,YS,USET/V,N,LUSET/V,N,MPCF1/V,
             N,MPCF2/V,N,SINGLE/V,N,ØMIT/V,N,REACT/V,N,NSKIP/V,N,REPEAT/V,
             N,NØSET/V,N,NØL/V,N,NØA $
35 SAVE      MPCF1,MPCF2,SINGLE,ØMIT,REACT,NSKIP,REPEAT,NØSET,NØL,NØA $
36 CØND      ERRØR5,NØL $
37 PURGE     GM/MPCF1/GØ,KØØB,LØØ,UØØ,PØ,UØØV,RUØV/ØMIT/PS,KFS,KSS,QG/
             SINGLE/UBØØV/ØMIT/YBS,PBS,KBFS,KBSS,KDFS,KDSS/SINGLE $
38 EQUIV     KGG,KNN/MPCF1/MGG,MNN/MPCF1 $
39 CØND      LBL4D,REACT $
40 JUMP      ERRØR2 $
41 LABEL     LBL4D $
42 CØND      LBL4,GENEL $
43 GPSP      GPL,GPST,USET,SIL/ØGPST $
44 ØFP       ØGPST,,,,,//V,N,CARDNØ $
45 SAVE      CARDNØ $
46 LABEL     LBL4 $
47 CØND      LBL2,MPCF2 $
48 MCE1      USET,RG/GM $
49 MCE2      USET,GM,KGG,MGG,,/KNN,MNN,,  $
50 LABEL     LBL2 $
51 EQUIV     KNN,KFF/SINGLE/MNN,MFF/SINGLE $
52 CØND      LBL3,SINGLE $
```

```
 53 SCE1      USET,KNN,MNN,,/KFF,KFS,KSS,MFF,,  $
 54 LABEL     LBL3 $
 55 EQUIV     KFF,KAA/ØMIT/MFF,MAA/ØMIT $
 56 CØND      LBL5,ØMIT $
 57 SMP1      USET,KFF,MFF,,/GØ,KAA,KØØB,LØØ,UØØ,MAA,MØØB,MØAB,, $
 58 LABEL     LBL5 $
 59 DUMMØD1   GEØM1,MGG,BGPDT,WW,USET,,,/KP,PG,G,AA,T,TI,RPP,NDØF/ $
 60 EQUIV     PG,PL/NØSET $
 61 RBMG2     KAA/LLL,ULL $
 62 CØND      LBL10,NØSET $
 63 SSG2      USET,GM,YS,KFS,GØ,,PG/,PØ,PS,PL $
 64 LABEL     LBL10 $
 65 SSG3      LLL,ULL,KAA,PL,LØØ,UØØ,KØØB,PØ/ULV,UØØV,RULV,RUØV/V,N,ØMIT/
              V,Y,IRES=-1 $
 66 CØND      LBL9,IRES $
 67 MATGPR    GPL,USET,SIL,RULV//C,N,L $
 68 MATGPR    GPL,USET,SIL,RUØV//C,N,Ø $
 69 LABEL     LBL9 $
 70 SDR1      USET,PG,ULV,UØØV,YS,GØ,GM,PS,KFS,KSS,/UGV,PGG,QG/C,N,1/C,N,DSØ$
 71 DUMMØD2   UGV,,,,,,/UGVX,,,,,,/V,Y,IUGV=0  $
 72 SAVE      IUGV  $
 73 MATPRN    UGVX,,,,//  $
 74 CØND      LBB,IUGV  $
 75 DSMG1     CASECC,GPTT,SIL,EDT,UGV,CSTM,MPT,ECPT,GPCT,DIT/KPGG/
              V,N,DSCØSET $
 76 SAVE      DSCØSET $
 77 ADD       KPGG,KGGX/KDGG/C,Y,ALPHA=(1.0,0.0)/C,Y,BETA=(1.0,0.0) $
 78 EQUIV     KDGG,KDNN/MPCF2 $
 79 CØND      LBL2D,MPCF2 $
 80 MCE2      USET,GM,KDGG,,,/KDNN,,, $
 81 LABEL     LBL2D $
 82 EQUIV     KDNN,KDFF/SINGLE $
 83 CØND      LBL3D,SINGLE $
 84 SCE1      USET,KDNN,,,/KDFF,KDFS,KDSS,,, $
 85 LABEL     LBL3D $
 86 EQUIV     KDFF,KDAA/ØMIT $
 87 CØND      LBL5D,ØMIT $
 88 SMP1      USET,KDFF,,,/GØØ,KDAA,KØØB1,LØØ1,UØØ1,,,,, $
 89 LABEL     LBL5D $
 90 LABEL     LBB  $
 91 EQUIV     KAA,KDAA/IUGV/GØ,GØØ/IUGV  $
 92 MATPRN    KAA,GØ,,,// $
 93 MPYAD     KDAA,TI,/KSUM/C,N,0/C,N,1/C,N,0/C,N,1 $
 94 MPYAD     KAA,TI,/KSUM2/C,N,0/C,N,1/C,N,0/C,N,1 $
 95 ADD       KSUM,KP/KSUM1/C,Y,ALPHA=(1.0,0.0)/C,Y,BETA=(1.0,0.0) $
 96 DPD       DYNAMICS,GPL,SIL,USET/GPLD,SILD,USETD,TFPØØL,,,,,,EED,EQDYN/
              V,N,LUSET/V,N,LUSETD/V,N,NØTFL/V,N,NØDLT/V,N,NØPSDL/
              V,N,NØFRL/V,N,NØNLFT/V,N,NØTRL/V,N,NØEED/C,N,123/V,N,NØUE $
 97 ØUTPUT2   RPP,NDØF,TI,,//C,Y,P1=-1/C,Y,P2=-1 $
 98 ØUTPUT2   MAA,KSUM2,KSUM,KP,G//C,Y,P1=-1/C,Y,P2=-1 $
 99 CEAD      KSUM1,G,MAA,EED,CASECC/PHI,CLAMA,ØCEIGS/V,N,EIGVS $
100 SAVE      EIGVS $
101 ØFP       ØCEIGS,CLAMA,,,,//V,N,CARDNØ $
102 SAVE      CARDNØ $
103 CØND      LBL16,EIGVS $
104 MPYAD     TI,PHI,/PHID/C,N,0/C,N,1/C,N,0/C,N,1  $
105 MATPRN    PHID,PHI,,,//  $
106 MØDC      AA,PHID//C,N,-1  $
107 VDR       CASECC,EQDYN,USETD,PHID,CLAMA,,/ØPHID,/C,N,CEIGN/C,N,DIRECT/
              C,N,0/V,N,NØD/V,N,NØP/C,N,0 $
108 SAVE      NØD,NØP $
```

```
109 CØND        LBL15,NØD $
110 ØFP         ØPHID,,,,,//V,N,CARDNØ $
111 SAVE        CARDNØ $
112 LABEL       LBL15 $
113 CØND        LBL16,NØP $
114 EQUIV       PHID,CPHIP/NØA $
115 CØND        LBL17,NØA $
116 SDR1        USETD,,PHID,,,GØØ,GM,,KFS,,/CPHIP,,QPC/C,N,1/C,N,DYNAMICS $
117 LABEL       LBL17 $
118 SDR2        CASECC,CSTM,MPT,DIT,EQDYN,SILD,,,,CLAMA,QPC,CPHIP,EST,/,ØQPC1,
                ØCPHIP,ØESC1,ØEFC1,/C,N,CEIG $
119 ØFP         ØCPHIP,ØQPC1,ØEFC1,ØESC1,,//V,N,CARDNØ $
120 SAVE        CARDNØ $
121 LABEL       LBL16 $
122 JUMP        FINIS  $
123 LABEL       ERRØR1  $
124 PRTPARM     //C,N,-1/C,N,MØDES$
125 LABEL       ERRØR2  $
126 PRTPARM     //C,N,-2/C,N,DIFFSTIF$
127 LABEL       ERRØR4  $
128 PRTPARM     //C,N,-4/C,N,DIFFSTIF$
129 LABEL       ERRØRS  $
130 PRTPARM     //C,N,-5/C,N,DIFFSTIF$
131 LABEL       FINIS  $
132 END         $
```

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 $[K^x_{gg}]$ 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 $[K^x_{gg}]$ 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 $[R_g]\{u_g\} = 0$ 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 $[R_g] = [R_m \mid R_n]$ and solves for multipoint constraint transformation matrix $[G_m] = -[R_m]^{-1}[R_n]$.

49. MCE2 partitions stiffness and mass matrices

$$[K_{gg}] = \left[\begin{array}{c|c} K_{nn} & K_{nm} \\ \hline K_{mn} & K_{mm} \end{array}\right] \quad \text{and} \quad [M_{gg}] = \left[\begin{array}{c|c} M_{nn} & M_{nm} \\ \hline M_{mn} & M_{mm} \end{array}\right]$$

and performs matrix reductions

$$[K_{nn}] = [\bar{K}_{nn}] + [G_m^T][K_{mn}] + [K_{mn}^T][G_m]$$

$$+ [G_m^T][K_{mm}][G_m] \quad \text{and}$$

$$[M_{nn}] = [\bar{M}_{nn}] + [G_m^T][M_{mn}] + [M_{mn}^T][G_m]$$

$$+ [G_m^T][M_{mm}][G_m] .$$

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.

$$[K_{nn}] = \left[\begin{array}{c|c} K_{ff} & K_{fs} \\ \hline K_{sf} & K_{ss} \end{array}\right] \qquad \text{and} \qquad [M_{nn}] = \left[\begin{array}{c|c} M_{ff} & M_{fs} \\ \hline M_{sf} & M_{ss} \end{array}\right] .$$

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

$$[K_{ff}] = \left[\begin{array}{c|c} \bar{K}_{aa} & K_{ao} \\ \hline K_{oa} & K_{oo} \end{array}\right] \qquad \text{and} \qquad [M_{ff}] = \left[\begin{array}{c|c} \bar{M}_{aa} & M_{ao} \\ \hline M_{oa} & M_{oo} \end{array}\right]$$

solves for transformation matrix $[G_o] = -[K_{oo}]^{-1}[K_{oa}]$, and performs matrix reductions $[K_{aa}] = [\bar{K}_{aa}] + [K_{oa}^T][G_o]$ and $[M_{aa}] = [\bar{M}_{aa}] + [M_{oa}^T][G_o] + [G_o^T][M_{oa}]$

$$+ [G_o^T][M_{oo}][G_o] .$$

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 $\{P_l\}$ if no constraints applied.

61. RMBG2 decomposes constrained stiffness matrix $[K_{ll}] = [L_{ll}][U_{ll}]$.

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

63. SSG2 applies constraints to static load vectors

$$\{P_g\} = \begin{Bmatrix} \bar{P}_n \\ -- \\ P_m \end{Bmatrix} , \qquad \{P_n\} = \{\bar{P}_n\} + [G_m^T]\{P_m\},$$

$$\{P_n\} = \begin{Bmatrix} \bar{P}_f \\ -- \\ P_s \end{Bmatrix} , \qquad \{P_f\} = \{\bar{P}_f\} - [K_{fs}]\{Y_s\} ,$$

$$\{P_f\} = \begin{Bmatrix} P_a \\ -- \\ P_o \end{Bmatrix} \quad \text{and} \quad \{P_l\} = \{P_a\} + [G_o^T]\{P_o\} .$$

65. SSG3 solves for displacements of independent coordinates

$$\{u_l\} = [K_{ll}]^{-1}\{P_l\} ,$$

solves for displacements of omitted coordinates

$$\{u_o^o\} = [K_{oo}]^{-1}\{P_o\},$$

calculates residual vector (RULV) and residual vector error ratio for independent coordinates

$$\{\delta P_l\} = \{P_l\} - [K_{ll}]\{u_l\}$$

$$\epsilon_l = \frac{\{u_l^T\}\{\delta P_l\}}{\{P_l^T\}\{u_l\}}$$

and calculates residual vector (RUOV) and residual vector error ratio for omitted coordinates

$$\{\delta P_o\} = \{P_o\} - [K_{oo}] \{u_o^o\} ,$$

$$\epsilon_o = \frac{\{u_o^T\} \{\delta P_o\}}{\{P_o^T\} \{u_o^o\}} .$$

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

$$\{u_o\} = [G_o] \{u_l\} + \{u_o^o\} ,$$

$$\begin{Bmatrix} u_a \\ -- \\ u_o \end{Bmatrix} = \{u_f\} , \qquad \begin{Bmatrix} u_f \\ -- \\ Y_s \end{Bmatrix} = \{u_n\} ,$$

$$\{u_m\} = [G_m] \{u_n\} , \qquad \begin{Bmatrix} u_n \\ -- \\ u_m \end{Bmatrix} = \{u_g\} ,$$

and recovers single-point forces of constraint

$$\{q_s\} = - \{P_s\} + [K_{fs}^T] \{u_f\} + [K_{ss}] \{Y_s\} .$$

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

$$[K_{gg}^x] + [K_{gg}^p] = [K_{gg}^d] .$$

78. Equivalence $[K^d_{gg}]$ to $[K^d_{nn}]$ if no multipoint constraints.

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

80. MCE2 partitions differential stiffness matrix

$$[K^d_{gg}] = \left[\begin{array}{c|c} \bar{K}^d_{nn} & K^d_{nm} \\ \hline K^d_{mn} & K^d_{mm} \end{array}\right]$$

and performs matrix reduction $[K^d_{nn}] = [\bar{K}^d_{nn}] + [G^T_m] [K^d_{mn}]$

$$+ [K^d_{mn}] [G_m] + [G^T_m] [K^d_{mm}] [G_m].$$

82. Equivalence $[K^d_{nn}]$ to $[K^d_{ff}]$ if no single-point constraints.

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

84. SCE1 partitions out single-point constraints.

$$[K^d_{nn}] = \left[\begin{array}{c|c} K^d_{ff} & K^d_{fs} \\ \hline K^d_{sf} & K^d_{ss} \end{array}\right]$$

86. Equivalence $[K^d_{ff}]$ to $[K^d_{aa}]$ if no omitted coordinates.

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

88. SMP1 partitions constrained stiffness matrix

$$[K^d_{ff}] = \left[\begin{array}{c|c} \bar{K}^d_{aa} & K^d_{ao} \\ \hline K^d_{oa} & K^d_{oo} \end{array}\right] ,$$

solves for transformation matrix $[G_{oo}] = -[K^d_{oo}]^{-1} [K^d_{oa}]$, and performs matrix reductions $[K^d_{aa}] = [\bar{K}^d_{aa}] + [K^{Td}_{oa}] [G_{oo}]$.

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

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

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

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 x 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 x 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 $[M_{aa}]$, $[K_{aa}]$, $[K^d_{aa}]$, $[K''']$, $[G]$ are output on magnetic tape.

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

$[K^d_{aa}]$ 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

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

is given by complex eigenvalue analysis step #91.

$\{\phi_d\}$, the eigenvector of

$$[MTp^2 + GTp + K''' T + K_e + K_g]\{\phi_d\} = 0$$

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

105. Eigenvectors $\{\phi_d\}$ and $\{\phi\}$ are printed.

106. $[\delta^T \delta]$, 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

$$\{\phi_o\} = [G^d_{oo}]\{\phi_d\}, \quad \begin{Bmatrix} \phi_d \\ \text{---} \\ \phi_o \end{Bmatrix} = \{\phi_f + \phi_e\},$$

$$\begin{Bmatrix} \phi_f + \phi_e \\ \text{-------} \\ \phi_s \end{Bmatrix} = \{\phi_n + \phi_e\}, \{\phi_m\} = [G^d_m]\ \{\phi_n + \phi_e\},$$

$$\begin{Bmatrix} \phi_n + \phi_e \\ \text{-------} \\ \phi_m \end{Bmatrix} = \{\phi_p\}$$

and recovers single-point forces of constraint

$$\{q_s\} = [K^T_{fs}]\ \{\phi_f\}.$$

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.

124. Normal mode analysis error message No. 1 - Mass matrix required for real eigenvalue analysis.

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.

```
      SUBRØUTINE DUMØD1
      INTEGER TYPIN, TYPØUT, TDGF, SYSBUF, CØRSZ
      REAL M, MU(300)
      EXTERNAL WRITE
      EXTERNAL READ
      DIMENSIØN HEAD(5), HEAD2(5), HEAD3(5), HEAD4(5), IGRID(2)
     D          ,ICØNM(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(5)
      DIMENSIØN XI11(300)
     D          ,XI21(300), XI22(300), XI31(300), XI32(300), XI33(300)
     D          ,XMGG(300), DØF(6,300)
      CØMMØN /DUMØPN/ XKP
      CØMMØN /PACKX/ TYPIN, TYPØUT, I1, N1, INCR
      CØMMØN /UNPAKX/ ITYPE, JJ, N, JINCR
      CØMMØN /SYSTEM/ SYSBUF, ØUTAPE
      CØMMØN XXX
      EQUIVALENCE (A(1), ID)
     1            ,(A(7), IA)
     2            ,(B(2), IG)
C
C     ØPEN CØRE ARRAY
C
      EQUIVALENCE (XKP(2000), NDEGF)
     E            ,(XKP(3800), M    )
     E            ,(XKP(4100), NFRE )
     E            ,(XKP(4400), TDGF )
     E            ,(XKP(4700), A    )
     E            ,(XKP(4708), B    )
     E            ,(XKP(4721), R    )
     E            ,(XKP(4725), HEAD )
     E            ,(XKP(4730), HEAD1)
     E            ,(XKP(4735), HEAD2)
     E            ,(XKP(4740), HEAD3)
     E            ,(XKP(4745), HEAD4)
     E            ,(XKP(4750), AA   )
      EQUIVALENCE (XKP(5650), RP    )
     E            ,(XKP(6550), MU   )
     E            ,(XKP(6850), XI11 )
     E            ,(XKP(7150), XI21 )
     E            ,(XKP(7450), XI22 )
     E            ,(XKP(7750), XI31 )
     E            ,(XKP(8050), XI32 )
     E            ,(XKP(8350), XI33 )
     E            ,(NDEGF, DØF)
      DATA NAM1, NAM2, NAM3, NAM4 / 201, 202, 203, 204 /
C***********************************************************************
C
C     IFL(101) REFERS TØ GEØM1-THE FIRST INPUT DATA BLØCK ØF DMAP
C              STATEMENT DUMMØD1.
C     ICØNM    REFERS TØ MGG-THE SECØND INPUT DATA BLØCK ØF DUMMØD1.
C     IBG      REFERS TØ BGPDT-THE THIRD INPUT DATA BLØCK ØF DUMMØD1.
C     IWW(104) REFERS TØ WW-THE BLØCK CØNTAINING W-S AND R-T FLAGS.
C     NAM5     REFERS TØ USET-THE FIFTH INPUT DATA BLØCK ØF DUMMØD1.
C
C     THE GENERAL PRØCEDURE FØR READING A DATA BLØCK IS-FIRST,
C     ØPEN THE FILE CØNTAINING THE BLØCK SUCH AS ICØNM,IBG ETC.
C     THEN SKIP THE HEADER RECØRD BY CALLING FWDREC. THE NEXT STEP
C     IS TØ DETERMINE IF THE DATA IS A MATRIX EG. MGG. FØR MATRICES,
```

```
C     THE NEXT STEP IS TØ CALL UNPACK. EACH CALL TØ UNPACK BRINGS INTØ
C     MEMØRY ØNE CØLUMN ØF THE MATRIX. FØR NØN-MATRIX INPUT, THE RECØRD
C     STRUCTURE MUST BE LØØKED UP IN THE NASTRAN PRØGRAMMER-S MANUAL.
C     EACH RECØRD CAN THEN BE BRØUGHT INTØ MEMØRY BY CALLING READ.
C     ØNE CALL TØ READ BRINGS IN ØNE RECØRD.
C
C     THE EXCEPTIØN IN THIS CASE IS DATA READ EXACTLY AS IT APPEARS ØN
C     BULK DATA CARDS. FØR THIS, THE PRØCEDURE IS CALL PRELØC, THEN
C     LØCATE, THEN READ FØR EACH BULK DATA CARD. THIS IS USED FØR IFL.
C
C**********************************************************************
      DATA NAM5, NAM6 /205, 206/
      DATA NAM7,NAM8 /207,208/
      DATA IGRID /4501,45/  IFL /101/
      DATA TWØ, EØR /2,1/
      DATA ICØN /102/ ICØNM /1501,15/IBG /103/
C
C   THIS IS THE MAIN RØUTINE FØR CØMPUTING THE K-PRIME, P-PRIME, G AND
C     MASS MATRICES USED IN SØLVING THE RØTATING FLEXIBLE STRUCTURE
C     PRØBLEM
C
C
C   READ IN ØMEGA VALUES AND RG + T-INVERSE FLAGS
C
      IWW = 104
      ITYPE = 1
      JJ = 1
      N = 5
      JINCR = 1
      CALL ØPEN($1000, IWW, XKP(LCØL+1), 0)
      CALL FWDREC($1000, IWW)
      CALL FNAME(IWW, RHEAD(1))
      CALL UNPACK($1000, IWW, W, READ)
      CALL CLØSE(IWW,1)
C
C     WRITE CØMMENTS ØN ØUTPUT LISTING
C
      WRITE(6,601)
  601 FØRMAT(1HI, - RESTRICTIØNS ***-//
     F- 1. ID NØ. 1 ØN GRID CARDS SHØULD BE USED FØR THE CENTRAL RIGID-/
     F-    BØDY ØTHERWISE THE TRANSFØRMATIØN MATRICES T AND T-INVERSE -/
     F-    WILL BE INCØRRECT-/
     F- 2. THE USE ØF THE T AND T-INVERSE PERMIT THE BASE MØTIØN WITH -/
     F-    THE TRANSLATIØNAL RIGID BØDY MØTIØN SWEPT ØUT FRØM THE EQS.-/
     F-    ØF MØTIØN -)
      WRITE(6,604)
  604 FØRMAT(1H ,
     F -3. IT IS ASSUMED THAT THE AXIS ØF RØTATIØN PASSES THRØUGH THE-/
     F-    CENTER ØF MASS ØF THE ENTIRE VEHICLE. IF THE AXIS ØF RØTATIØN
     F-/
     F-    IS TØ PASS THRØUGH A PØINT ØTHER THAN THE MASS CENTER, THE-/
     F-    CØRRESPØNDING GRID PØINT SHØULD BE DEFINED AS GRID NØ. 1-)
      WRITE(6,605)
  605 FØRMAT(
     F- 4. LØCATIØN ØF THE CENTER ØF RØTATIØN (C.R.) GØVERNS THE-/
     F-    CALCULATIØNS ØF DELTA TRANSPØSE DELTA MATRICES AND THE-/
     F-    CENTRIFUGAL FØRCES WHICH IN TURN AFFECTS THE GEØMETRIC-/
     F-    STIFFNESS MATRIX; THE ØPTIØN TØ SELECT THE C.R. IS AVAILABLE-
     F /
     F-    THRU DMI CARDS. W(4)=0.0-CG IS NØT CALCULATED AND C.R. IS-/
     F-    ASSUMED TØ BE GRID NØ. 1; W(4)=1.0-CG IS CALCULATED AND C.R.-
```

```
      F /                                                               DM112200
      F-     IS C.G.-                                                   DM112300
      F-     W(5)=1.0 PERMITS THE CALCULATIØNS ØF T AND T-INVERSE.-/    DM112400
      F-     W(5)=0.0 MAKES T AND T-INVERSE IDENTITY MATRICES.-/        DM112500
      F-     IN GENERAL THE FØLLØWING CØMBINATIØNS SHØULD BE USED.-/    DM112600
      F-     W(4)=W(5)=0.0 ØR W(4)=W(5)=1.0. W(4)=W(5)=0.0 ASSUMES THE-) DM112700
       WRITE(6,899)                                                     DM112800
  899 FØRMAT(                                                           DM112900
      F-     STRUCTURE RØTATES ABØUT GRID NØ. 1 AND ØNLY THE CANTILEVER-/ DM113000
      F-     MØDES ARE AVAILABLE. W(4)=W(5)=1.0 ASSUMES THE STRUCTURE-/ DM113100
      F-     RØTATES ABØUT THE C.G. WITH TRANSLATIØNAL RIGID BØDY DØF-/ DM113200
      F-     SWEPT ØUT-/                                                DM113300
      F- 5.  IN THE CALCULATIØN ØF THE CENTER ØF MASS ØF THE VEHICLE ALL-/DM113400
      F-     THE GRID PØINTS WITH MASSES ARE USED. IF THERE IS A DUMMY- / DM113500
      F-     GRID PØINT FØR THE DEFINITIØN ØF THE PLANE FØR CBAR CARDS,-/ DM113600
      F-     DØ NØT PUT ANY MASS AT THAT DUMMY GRID PØINT ØTHERWISE WRØNG-DM113700
      F/                                                                DM113800
      F-     C. M. WILL BE CØMPUTED-)                                   DM113900
       WRITE(6,606)                                                     DM114000
  606 FØRMAT(                                                           DM114100
      F- 6.  THE NUMBER ØF GRID PØINTS IN THE PRØBLEM SHØULD BE LESS-/  DM114200
      F-     THAN 300. THIS ALLØWS UP TØ 1800 DØF.-)                    DM114300
       WRITE(6,602)                                                     DM114400
  602 FØRMAT(1H0,- THE FØLLØWING DATA WAS TAKEN FRØM GRID CARDS BY THE DDM114500
      FUMMY MØDULE***-/)                                                DM114600
       WRITE(6,612) W(1),W(2),W(3)                                      DM114700
  612 FØRMAT(/  1H (     -SPIN RATE VECTØR*** ØMEGA1 = -F10.4, 5X,      DM114800
      F                  -ØMEGA2 = -F10.4, 5X, -ØMEGA3 = -F10.4)        DM114900
       IF(W(4) .LT. 0.01) WRITE(6,695)                                  DM115000
  695 FØRMAT(1H ,-THE FØLLØWING ANALYSIS ASSUMES RØTATIØN ABØUT GRID NØ.DM115100
      1 1.-)                                                            DM115200
       IF(W(4) .GT. 0.0) WRITE(6,696)                                   DM115300
  696 FØRMAT(1H ,-IN THIS ANALYSIS THE STRUCTURE SPINS ABØUT THE C.G. ØFDM115400
      1 THE SYSTEM DESCRIBED IN THE BULK DATA-)                         DM115500
       IF(W(5) .LT. 0.01) WRITE(6,697)                                  DM115600
  697 FØRMAT(1H ,-THE FØLLØWING ANALYSIS GIVES CANTILEVER MØDES ØF A SPIDM115700
      FNNING STRUCTURE BY MAKING T AND-/1H ,-T-INVERSE IDENTITY MATRICES-DM115800
      F)                                                                DM115900
       IF(W(5) .GT. 0.0) WRITE(6,698)                                   DM116000
  698 FØRMAT(1H ,-THE FØLLØWING ANALYSIS CALCULATES THE MATRICES T AND TDM116100
      F-INVERSE THUS SWEEPING ØUT THE-/1H ,-TRANSLATIØNAL RIGID BØDY DØF-DM116200
      F)                                                                DM116300
       WRITE(6,611)                                                     DM116400
C                                                                       DM116500
C      SET UP PACK CØMMØN AND LØCATE END ØF CØRE                        DM116600
C                                                                       DM116700
  611 FØRMAT(1H0,- NØDE    DØF       MASS        I11        I21       I22 DM116800
      F     I31       I32        I33         R1         R2        R3-/)  DM116900
       TYPIN = 1                                                        DM117000
       TYPØUT = 1                                                       DM117100
       II = 1                                                           DM117200
       INCR = 1                                                         DM117300
       LZ = CØRSZ(XXX, XKP)                                             DM117400
       LCØL = LZ - SYSBUF                                               DM117500
       IL = LCØL                                                        DM117600
       IXX = LZ - 2*SYSBUF -2                                           DM117700
       IBGR = IXX - SYSBUF - 1                                          DM117800
       DØ 750 II=1,3                                                    DM117900
  750 RB(II) = 0.0                                                      DM118000
       XN = 0.0                                                         DM118100
```

```
C
C   LØCATE FILE THAT GRID PØINTS ARE STØRED ØN
C
      CALL PRELØC($1000, XKP(IL), IFL)
      CALL LØCATE($1000,XKP(IL), IGRID, IFLG)
C
C   LØCATE FILE THAT BASIC GRID PØINT CØ-ØRDINATES ARE STØRED ØN
C
      CALL ØPEN($1000, IBG, XKP(IBGR), 0)
      CALL FWDREC($700, IBG)
      NØDE = 0
C     BUILD MASS AND INERTIA TABLE
C
C     UNPACK MGG MATRIX TØ GET MASS AND INERTIAS
C
      N = 300
      CALL ØPEN($1000, ICØN, XKP(IXX), 0)
      CALL FWDREC($1000, ICØN)
      CALL FNAME(ICØN, RHEAD(1))
C
C     WHEN AN END ØF FILE IS ENCØUNTERED BY FWDREC, READING ØF MGG STØPS
C
  950 CALL FWDREC($699, ICØN)
      CALL FWDREC($699, ICØN)
      NØDE = NØDE + 1
      ND1 = (NØDE-1)*6
      CALL UNPACK($951 , ICØN, XMGG, READ)
      M(NØDE) = XMGG(ND1+3)
      XM = XM + M(NØDE)
  951 CALL UNPACK($952 , ICØN, XMGG, READ)
      XI11(NØDE) = XMGG(ND1+4)
  952 CALL UNPACK($953 , ICØN, XMGG, READ)
      XI21(NØDE) = XMGG(ND1+4)
      XI22(NØDE) = XMGG(ND1+5)
  953 CALL UNPACK($950 , ICØN, XMGG, READ)
      XI31(NØDE) = XMGG(ND1+4)
      XI32(NØDE) = XMGG(ND1+5)
      XI33(NØDE) = XMGG(ND1+6)
      GØ TØ 950
  699 NØDE = 0
  700 CØNTINUE
      CALL READ($710, $710, IFL, A, 8, 0, IFLG)
      NØDE = NØDE + 1
C
C   BASIC GRID PØINT INFØ.
      CALL READ($710, $710, IBG, R, 4, 0, IFLG)
C   IF MØTIØN AT A GRID PØIT IS RESTRAINED IN ALL DIRECTIØNS THE LØGIC
C   IN THE CØDE CAUSES THE GRID PØINT TØ BE DISREGARDED
C
      WRITE(6,610) ID, IA, M(NØDE), XI11(NØDE), XI21(NØDE), XI22(NØDE)
     W             ,XI31(NØDE), XI32(NØDE), XI33(NØDE), R(2), R(3),
     W              R(4)
  610 FØRMAT(1I5,1I8, 10F10.4)
      IF(W(4) .GT. 0.0) GØ TØ 720
      IF(ID .GT. 1) GØ TØ 725
      DØ 733 II=1,3
  733 RG(II) = R(II+1)
      GØ TØ 725
  720 CØNTINUE
      DØ 701 II=1,3
  701 RG(II) = RG(II) + M(NØDE)*R(II+1)
```

```
  725 CØNTINUE                                                          DM124300
      GØ TØ 700                                                         DM124400
  710 CØNTINUE                                                          DM124500
 1000 CØNTINUE                                                          DM124600
      IF(W(4) .EQ. 0.0) GØ TØ 719                                       DM124700
      DØ 734 II=1,3                                                     DM124800
  734 RG(II) = RG(II)/XM                                                DM124900
  719 CØNTINUE                                                          DM125000
      WRITE(6,607)                                                      DM125100
  607 FØRMAT(1H ,-RG GIVES THE CØØRDINATES ØF THE CENTER ØF RØTATIØN-)  DM125200
      WRITE(6,615) XM, RG(1), RG(2), RG(3)                              DM125300
  615 FØRMAT(1H0,- TØTAL MASS = -,1E15.8, -  RG(1) =-,1E15.8, -  RG(2) =DM125400
     F-, 1E15.8, -  RG(3) =-, 1E15.8)                                   DM125500
      WRITE(6,620)                                                      DM125600
  620 FØRMAT(1H0, - MØTIØN CØNSTRAINTS IN 1 THRU 6 DIRECTIØN AND TØTAL DDM125700
     FØF AT EACH NØDE-//- NØDE    CØNSTRAINTS    TØTAL DØF -)           DM125800
      CALL CLØSE(IFL, 1)                                                DM125900
      CALL CLØSE(ICØN, 1)                                               DM126000
C***********************************************************************DM126100
C     NAM1(201) REFERS TØ PG-THE FIRST ØUTPUT DATA BLØCK ØF DUMMØD1.    DM126200
C     ...                                                               DM126300
C     NAM8(208) REFERS TØ DØF-THE EIGHTH ØUTPUT DATA BLØCK ØF DUMMØD1.  DM126400
C                                                                       DM126500
C     THE PRØCEDURE FØR PACKING A MATRIX IS-ØPEN THE FILE(EG. 201),     DM126600
C     CALL FNAME, CALL WRITE TØ WRITE THE HEADER RECØRD, THEN PACK.     DM126700
C     ALSØ BUILD THE TRAILER ARRAY, THEN CALL WRTTRL. SEE 2.2-1 ØF THE  DM126800
C     NASTRAN PRØGRAMMER-S MANUAL FØR TRAILER INFØRMATIØN.              DM126900
C                                                                       DM127000
C     ALWAYS CLØSE THE FILES ØUT AS SØØN AS PØSSIBLE                    DM127100
C                                                                       DM127200
      CALL CLØSE(IBG, 1)                                                DM127300
      CALL DFRE(LCØL , NØDE)                                            DM127400
      DØ 672 II=1,NØDE                                                  DM127500
      NFRE(II) = NDEGF(1,II) + NDEGF(2,II) + NDEGF(3,II)                DM127600
      NFREE = NDEGF(4,II) + NDEGF(5,II) + NDEGF(6,II)                   DM127700
  672 TDGF(II) = NFRE(II) + NFREE                                       DM127800
      WRITE(6,625) (II, (NDEGF(JJ,II),JJ=1,6), TDGF(II), II=1,NØDE)     DM127900
  625 FØRMAT(1I5, 9X, 6I1, 1I11)                                        DM128000
      CALL ØPEN($300, NAM1, XKP(LCØL+1), 1)                             DM128100
      CALL FNAME(NAM1, HEAD(1) )                                        DM128200
      CALL WRITE(NAM1, HEAD(1), TWØ, EØR)                               DM128300
C                                                                       DM128400
C  ZERØ ØUT RØWS AND CØLS. ØF EACH 6X6 SUBMATRIX WHERE THE DEGREE ØF    DM128500
C  FREEDØM IS NØT USED ( EG FØR NDEG(3) = 0 , THIRD RØW AND CØL ARE ZERØDM128600
C  THE NØN-ZERØ ELEMENTS ARE THEN MØVED TØ THE TØP LEFT CØRNER ØF SUBMATDM128700
C                                                                       DM128800
C     ZERØ ØPEN CØRE                                                    DM128900
C                                                                       DM129000
      NCØR = 6*NØDE                                                     DM129100
      DØ 817 II=1,NCØR                                                  DM129200
      XKP(II) = 0.0                                                     DM129300
  817 CØNTINUE                                                          DM129400
      DØ 818 II=1,NØDE                                                  DM129500
      DØ 818 JJ=1,6                                                     DM129600
  818 ITT = NDEGF(JJ,II) + ITT                                          DM129700
      CALL KPRIM(NØDE, W)                                               DM129800
  300 CØNTINUE                                                          DM129900
      CALL CLØSE(NAM1, 1)                                               DM130000
      CALL ØPEN($501, NAM2, XKP(LCØL+1), 1)                             DM130100
      CALL FNAME(NAM2, HEAD2(1) )                                       DM130200
```

```
      CALL WRITE(NAM2, HEAD2(1), TWØ, EØR)
      CALL ØPEN($501, IBG, XKP(IBGR), 0)
      CALL FWDREC($501, IBG)
      CALL PPRIM(RG, NØDE, W)
  501 CØNTINUE
      CALL CLØSE(NAM2, 1)
      CALL CLØSE(IBG, 1)
C
C   ZERØ ØUT ØPEN CØRE
C
      DØ 2001 II=1,N1
 2001 XKP(II) = 0.0
      CALL ØPEN($3000, NAM3, XKP(LCØL+1), 1)
      CALL FNAME(NAM3, HEAD3(1) )
      CALL WRITE(NAM3, HEAD3(1), TWØ, EØR)
      CALL GMAT(XM, NØDE, W)
 3000 CØNTINUE
      CALL CLØSE(NAM3, 1)
C
C  ZERØ ØUT ØPEN CØRE
C
      DØ 3001 II=1,N1
      XKP(II) = 0.0
 3001 CØNTINUE
      CALL ØPEN($4000, NAM4, XKP(LCØL+1), 1)
      CALL FNAME(NAM4, HEAD4(1) )
      CALL WRITE(NAM4, HEAD4(1), TWØ, EØR)
C
      CALL AMAT(NØDE)
      CALL CLØSE(NAM4, 1)
C
      CALL ØPEN($4000, NAM5, XKP(LCØL+1), 1)
      CALL FNAME(NAM5, HEAD4(1))
      CALL WRITE(NAM5, HEAD4(1), TWØ, EØR)
      CALL TMAT(XM,NØDE,W,ITT)
      CALL CLØSE(NAM5, 1)
C
      CALL ØPEN($4000, NAM6, XKP(LCØL+1), 1)
      CALL FNAME(NAM6, HEAD4(1))
      CALL WRITE(NAM6, HEAD4(1), TWØ, EØR)
      CALL TIMAT(NØDE,W,ITT,XM)
      CALL CLØSE(NAM6, 1)
C
C     PACK RP MATRIX
C
      CALL ØPEN($4000, NAM7, XKP(LCØL+1), 1)
      CALL FNAME(NAM7, HEAD4(1))
      CALL WRITE(NAM7, HEAD4(1), TWØ, EØR)
      N1 = NØDE
      MM(1) = 207
      MM(2) = 0
      MM(3) = NØDE
      MM(4) = 2
      MM(5) = 1
      MM(6) = NØDE
      DØ 3500 II=1,3
      CALL PACK(RP(1,II), NAM7, WRITE, MM)
 3500 CØNTINUE
C
C     PACK MASSES
C
```

```
      CALL PACK(M, NAM7, WRITE, MM)
      CALL WRTTRL(MM(1))
      CALL CLØSE(NAM7, 1)
C
C     PACK DØF
C
      CALL ØPEN($4000, NAM8, XKP(LCØL+1), 1)
      CALL FNAME(NAM8, HEAD4(1))
      CALL WRITE(NAM8, HEAD4(1), TWØ, EØR)
      DØ 3550 II=1,6
      DØ 3550 J1=1,NØDE
 3550 IF(NDEGF(II,J1) .GT. 0) DØF(II,J1) = 1.5
      N1 = 3
      MM(1) = 208
      MM(2) = 0
      MM(3) = 6
      MM(4) = 2
      MM(5) = 1
      MM(6) = 6
      DØ 3600 II=1,NØDE
      CALL PACK(  DØF(1,II),NAM8,WRITE, MM)
 3600 CØNTINUE
      CALL WRTTRL(MM(1))
      CALL CLØSE(NAM8,1)
 4000 CØNTINUE
      RETURN
      END
```

For complete listing of this program on Univac 1108 Computer, write to Reference 3.

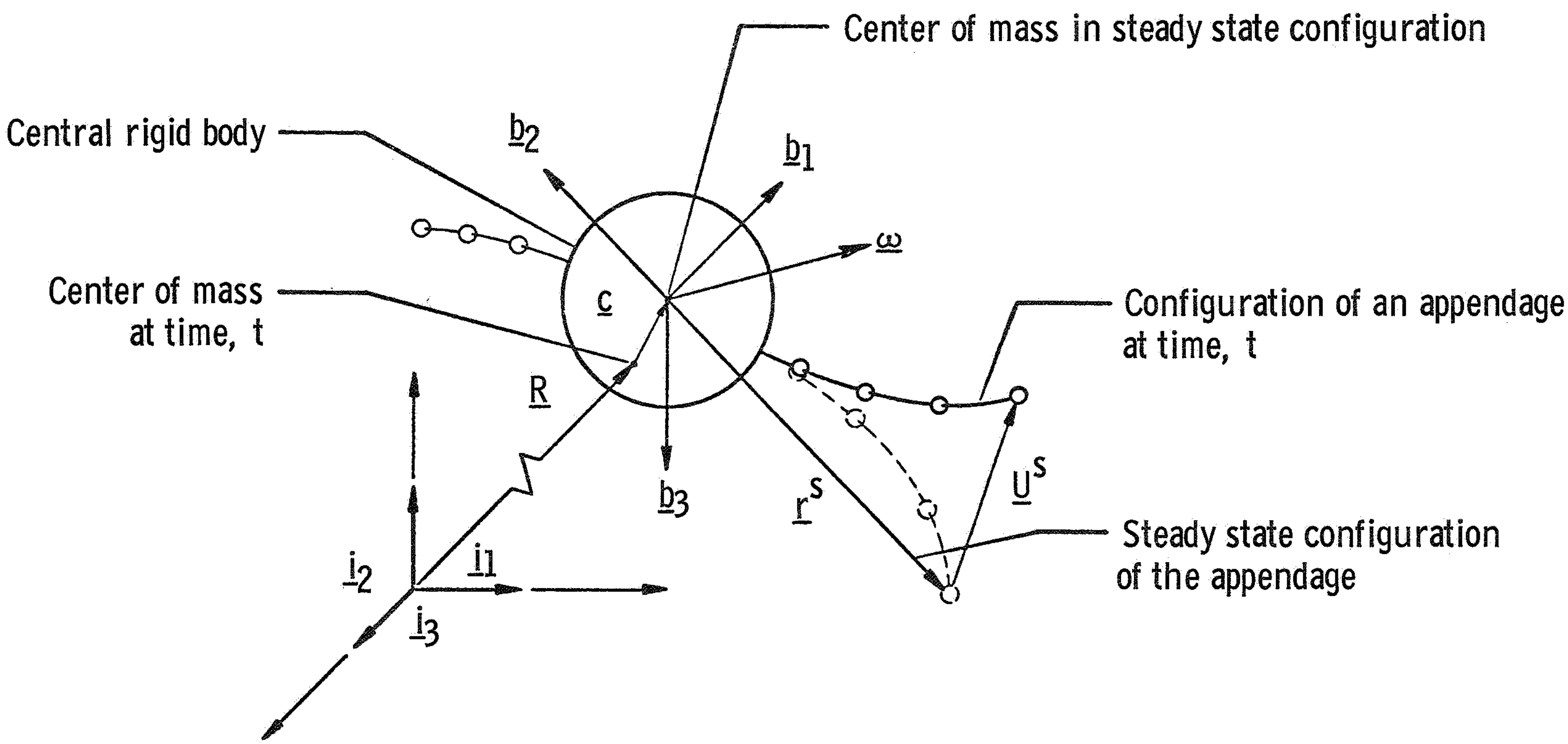

FIGURE 1. GEOMETRY OF SPINNING FLEXIBLE APPENDAGE AND CENTRAL BODY