# COMPONENT MODE SYNTHESIS OF STRUCTURES WITH GEOMETRIC STIFFENING IN MSC/NASTRAN

By

Tarun Ghosh
Boeing North American, Inc.
Canoga Park, California

## ABSTRACT

Implementation of modal synthesis in MSC/NASTRAN is usually done using structural solution sequence based Direct Matrix Abstraction Programs, DMAPs. However, modal synthesis in MSC/NASTRAN without using structural solution based DMAPs is possible. But either method is tailored towards supporting components that have no non-linearity. Certain components, such as the solar arrays of the space station, exhibit non-linear behavior in the form of geometric stiffening. For structures with such components, the standard method of modal synthesis does not work. Special DMAPs need to be developed for these components. Realizing that the only difference between components with no geometric stiffening and those with geometric stiffening is in the method of obtaining the stiffness matrix, a simple solution is provided in this paper. The solution is to use the procedure for modal synthesis for components with no geometric stiffening by replacing the stiffness matrix with the one obtained from geometric stiffening. This approach, along with the recommended check runs, has been shown to work successfully in this paper. The method is also shown to be extremely efficient.

## INTRODUCTION

In today's world, the design and construction of large structures often requires the combined effort of several organizations. The International Space Station (ISS) is one such structure and is shown in Figure-1. It is an international joint venture and countries across the globe are participating. It is a giant station that will operate in space, and mankind will use it for research, commercial production and voyage to outer space. When completed, it will be about 300 feet long and weigh about a million pounds.

As different organizations build parts of the station, they also generate finite element models of these parts. MSC/NASTRAN for its inherent advantages is used to develop the finite element models. These models must be assembled in order to develop a total system model that could be used to perform loads analysis. The loads during on-orbit and launch operations are primarily dynamic in nature. All these factors make the component mode synthesis method an indispensable tool in the analysis of the space station.

The component mode synthesis method is a technique in finite element analysis. This technique involves representing a system as an assemblage of components, and is described in Craig[1], Craig[2] and Cook, et al[3]. In the system model, each component is represented in terms of its boundary and modal (generalized) degrees of freedom. Any loads applied at the interior of components need to be transferred to the boundary and generalized degrees of freedom. On the other hand, the response obtained is in terms of the boundary and vibration degrees of freedom of the component and needs to be transformed to obtain the response of the desired data recovery items. There are different techniques of doing this and each technique uses elaborate software.

The ISS is powered by solar energy. There are eight U. S. built solar arrays, each of which is about 1000 inches long, 360 inches wide and weighs about 2,200 lbs. A typical solar array is shown in Figure-2. It consists of a mast canister, a deployable mast, two blankets, a top piece and a bottom piece. The blanket is modeled as plate elements for which the stiffness is a function of the applied membrane load. So Solution 106 is used to iteratively obtain the stiffness matrix.

## THEORY

In finite element analysis, a physical model of a structure with applied loads is represented mathematically as

$$[m]\{\ddot{u}\}+[c]\{\dot{u}\}+[k]\{u\}=\{f(t)\} \tag{1}$$

$[m]$, $[c]$ and $[k]$ are the mass, stiffness and damping matrices, respectively. $\{u\}$, $\{\dot{u}\}$ and $\{\ddot{u}\}$ represent the generalized displacements, velocities and accelerations at the physical degrees of freedom and $\{f(t)\}$ represents the generalized forces at the physical degrees of freedom. The equations are generated by assembling elements defined mathematically in the physical domain.

For linear elastic structures, the equations in (1) can be transformed to the modal domain and expressed as follows

$$[\mathrm{M}]\{\ddot{h}\}+[\mathrm{C}]\{\dot{h}\}+[\mathrm{K}]\{h\}=\{\mathrm{F}(t)\} \tag{2}$$

where [M], [C] and [K] are the system modal mass, stiffness and damping matrices. $\{h\}$ is the generalized (modal) displacement vector and $\{F(t)\}$ is the modal load vector. The method of generating the equations in (2) from the equations in (1) requires an eigensolution of the undamped free system. That is, in equations (1) $[c]$ is assumed to be a null matrix and $\{f(t)\}$ is assumed to be a null vector. The eigenvectors so obtained are used to generate an eigenvector matrix $[\Phi]$. For linear systems,

$$\{u\}=[\Phi]\{\eta\}$$

$$\{\dot{u}\} = [\Phi]\{\dot{\eta}\} \tag{3}$$

$$\{\ddot{u}\} = [\Phi]\{\ddot{\eta}\}$$

Then, by replacing $\{u\}$, $\{\dot{u}\}$ and $\{\ddot{u}\}$ in (1) by right hand sides of (3) and pre-multiplying both sides by $[\Phi]^T$, one obtains (2). Actually, [C] in (2) comes from test data or other sources. For details of the method, one can refer to Cook[3] and Meirovitch[6].

The method above is the standard method of solving structural problems and is implemented in many finite element codes. The modal synthesis method is a slight variation of this method. It is also referred to as dynamic superelement or substructure analysis. Over the years, the method has proliferated and has several forms. Some of these are described in Craig[1] ,Craig[2] and Cook, et al[3]. The basic idea (not the detailed theory) behind modal synthesis will be explained below.

The equations in (1) can also be generated by assembling component mode models. In the standard method, a system model was treated as an assemblage of finite elements. In modal synthesis, a system model is treated as an assemblage of components or substructures. The motion (of the interior) of a component can be represented either in the form of equations (1) or equations (2). The motion at the degrees of freedom that are at the boundary of a component is always represented in the form of equations (1). Since a component has a boundary and an interior, a component mode model has both physical (boundary) and generalized (modal) degrees of freedom. It consists of

$[M_C]$ = component mass matrix

$[K_C]$ = component stiffness matrix

$[T_C]$ = component data recovery matrix

$[L_C]$ = component load transformation matrix

The details for obtaining the above matrices are based on Reference-7 and are as follows. Simply stated, it is assumed that the motion of the interior degrees of freedom of a component can be obtained from the summation of two types of motion as follows:

$$\{u_{CI}\} = [\Phi_{CT}]\{u_{CB}\} + [\Phi_{CQ}]\{u_{CQ}\} \tag{4}$$

where,

$[\Phi_{CT}]$ = Component attachment modes

$[\Phi_{CQ}]$ = Component vibration modes

$\{u_{CB}\}$ = Component boundary displacements

$\{u_{CI}\}$ = Component interior displacements

$\{u_{CQ}\}$ = Component modal displacements

In (4) the part corresponding to $[\Phi_{CT}]$ is from the static equilibrium of the component when there is motion of the boundary. The part corresponding to $[\Phi_{CQ}]$ is from the dynamic equilibrium of the component when the boundary is fixed.

Let,

$$[\Phi_C] = \begin{bmatrix} [I] & [0] \\ [\Phi_{CT}] & [\Phi_{CQ}] \end{bmatrix} \tag{5}$$

Then,

$$\begin{Bmatrix} \{u_{CB}\} \\ \{u_{CI}\} \end{Bmatrix} = [\Phi_C] \begin{Bmatrix} \{u_{CB}\} \\ \{u_{CQ}\} \end{Bmatrix} \tag{6}$$

The equations of motion for the component (neglecting damping at the component level) can be represented as

$$\begin{bmatrix} [m_{CBB}] & [m_{CBI}] \\ [m_{CIB}] & [m_{CII}] \end{bmatrix} \begin{Bmatrix} \{\ddot{u}_{CB}\} \\ \{\ddot{u}_{CI}\} \end{Bmatrix} + \begin{bmatrix} [k_{CBB}] & [k_{CBI}] \\ [k_{CIB}] & [k_{CII}] \end{bmatrix} \begin{Bmatrix} \{u_{CB}\} \\ \{u_{CI}\} \end{Bmatrix} = \begin{Bmatrix} \{f_{CB}\} \\ \{f_{CI}\} \end{Bmatrix} \tag{7}$$

The form of (7) is similar to that of (1), except that the matrices and vectors of (7) are partitioned. The subscripts B, I and C represent boundary, interior and component, respectively. Therefore, $[m_{CBI}]$ represents boundary (B) mass of component (C), excited by motion of interior (I) degrees of freedom. Differentiating both sides of equations (6) with respect to time gives

$$\begin{Bmatrix} \{\dot{u}_{CB}\} \\ \{\dot{u}_{CI}\} \end{Bmatrix} = [\Phi_C] \begin{Bmatrix} \{\dot{u}_{CB}\} \\ \{\dot{u}_{CQ}\} \end{Bmatrix}$$

$$\begin{Bmatrix} \{\ddot{u}_{CB}\} \\ \{\ddot{u}_{CI}\} \end{Bmatrix} = [\Phi_C] \begin{Bmatrix} \{\ddot{u}_{CB}\} \\ \{\ddot{u}_{CQ}\} \end{Bmatrix} \tag{8}$$

Substituting equations (6) and (8) in (7) and pre-multiplying both sides of (7) by $[\Phi_C]^T$ gives

$$[M_C] \begin{Bmatrix} \{\ddot{u}_{CB}\} \\ \{\ddot{u}_{CQ}\} \end{Bmatrix} + [K_C] \begin{Bmatrix} \{u_{CB}\} \\ \{u_{CQ}\} \end{Bmatrix} = [L_C] \begin{Bmatrix} \{f_{CB}\} \\ \{f_{CI}\} \end{Bmatrix} \tag{9}$$

where

$$\begin{aligned} [M_C] &= [\Phi_C]^T \begin{bmatrix} [m_{CBB}] & [m_{CBI}] \\ [m_{CIB}] & [m_{CII}] \end{bmatrix} [\Phi_C] \\ [K_C] &= [\Phi_C]^T \begin{bmatrix} [k_{CBB}] & [k_{CBI}] \\ [k_{CIB}] & [k_{CII}] \end{bmatrix} [\Phi_C] \\ [L_C] &= [\Phi_C]^T \end{aligned} \tag{10}$$

For downstream processing (assembling the system model) one needs the component mass, stiffness and load vectors. For upstream processing (component data recovery), one needs the data recovery matrix. Here, the word downstream means a later step and the word upstream means an earlier step. The system model can then be assembled from the component models and would be as follows

$$[M]\{\ddot{h}\} + [C]\{\dot{h}\} + [K]\{h\} = \{F(t)\} \tag{11}$$

where

$$\begin{aligned} [M] &= \sum [M_C] \\ [K] &= \sum [K_C] \\ \{F(t)\} &= \sum \{F_C(t)\} \end{aligned} \tag{12}$$

The system damping matrix is not needed when integrating the system model from the component models. It is assumed to be proportional to the mass and stiffness matrices as in Meirovitch[6]:

$$[C] = \alpha[M] + \beta[K] \tag{13}$$

Note that while the system model in the standard method as shown by the equations in (1) has only physical degrees of freedom, the system model in the modal synthesis method as shown by the equations in (11) has both modal and physical degrees of freedom. So (11) is the modal synthesis counterpart of (1) of the standard method and the rest of the analysis is as in the standard method. As in the standard method, the equations in (11) can be transformed to the modal domain and expressed as in (2) which is a set of second-order uncoupled differential equations. The mass, stiffness and damping matrices are, therefore, diagonal.

In modal synthesis, the interior degrees of freedom of a component are not carried upstream to the system level. Accordingly, no loads can be applied to component interior degrees of freedom at the system level. At the system level (equations (9)) only the component boundary and generalized degrees of freedom are available. These are referred to as the analysis (retained) set degrees of freedom. So one could use the component load transformation matrix, $[L_C]$, to express interior loads in terms of the analysis set degrees of freedom. Similarly, to compute responses at interior degrees of freedom from the analysis set degrees of freedom responses, one would use the component (displacement) mode data recovery matrix, $[T_C]$. The elements of $[T_C]$ and $[L_C]$ are obtained from component attachment and vibration modeshapes.

## METHODOLOGY

While the methodology for generating the reduced model of a linear component has only one step, the one for geometric stiffening has an additional step. The two step process works as follows: In the first step, the stiffness matrix from the last iteration of Solution 106 is printed out as a punched file – i.e. in DMIG form. It is important to output the file in this step in DMIG form, as the stiffness matrix needs to be in terms of physical degrees of freedom. A stiffness matrix printed out in OUTPUT4 or OUTPUT2 form may not work, as a matrix printed out in this form is in terms of a set which may not be the same in all solution sequences. The mass matrix is not printed out, as the iterations alter the stiffness matrix but not the mass matrix.

A simple DMAP that will output the global stiffness matrix from the last iteration in Solution 106 when using Version 70.5 of MSC/NASTRAN are as follows:

```
$
$ BEGIN NOTES DATED 8-18-98
$
$ ALTER IS FOR SOL 106 FOR VERSION 70.5
$ ALTER DEVELOPED BY T. K. GHOSH OF ROCKETDYNE.
$ THIS ALTER WILL GENERATE A PUNCH (I.E. DMIG) FILE CONTAINING
$ THE STIFFNESS MATRIX FROM LAST ITERATION.
$
$ BEGIN NOTES DATED 8-18-98
$
COMPILE SEKR, SOUIN=MSCSOU, LIST, REF $
ALTER 64
matgen eqexins/intext/9/0/lusets $
smpyad intext,knn,intext,,/knnext/3////1////6 $
matmod knnext,eqexins,,,,/mat1,/16/1 $
```

The second step is similar to the single step process when generating the component mode model of a linear component in Solution 103. The DMAP used for a linear component is altered to read the stiffness matrix in DMIG form. The additional DMAP statements needed to do this when using Version 70.5 of MSC/NASTRAN are as follows:

```
$
$ ALTER IS FOR SOL 103 OF VERSION 70.5
$ ALTER DEVELOPED BY T. K. GHOSH OF ROCKETDYNE.
$ THIS ALTER WILL READ A PUNCH (I.E. DMIG) FILE CONTAINING
$ THE STIFFNESS MATRIX.
```

```
$
COMPILE SEMG, SOUIN=MSCSOU, LIST, REF $

$ ------------------------------------------------
ALTER 110
MTRXIN ,,MATPOOL,EQEXINS,SILS,/KNNEXT,,/LUSETS/  $  AD081998
PURGEX  /KJJZ,,,, $                                 AD081998
EQUIVX  KNNEXT/KJJZ/ALWAYS $                        AD081998
$ ------------------------------------------------
```

As Reference-4 shows, even after geometric stiffening, a detailed solar array model will have several hundred vibration modes in the frequency range of interest. It is necessary to identify and retain the dominant vibration modes. As in Reference-4, the strain energy of vibration will be used to identify the dominant modes. This is done by using the STRAIN option in CASE CONTROL to punch out the strain energy of a set of elements by vibration modes. Then the FORTRAN program of Figure-3 can be used to recreate the reduced matrices using only the retained modes. The FORTRAN program can do the additional task of eliminating unwanted data recovery items such as loads at interior sections of beam elements. The program can also add additional data recovery items such as longeron and batten loads of the solar array, which are functions of the mast base element loads.

Note that the method of this paper does in only three steps essentially what Reference-4 does in nine steps. Also, the method of this paper will work with that of Reference-8.

Generation of the system model from the component models, when one or more of the components has geometric stiffening, is the same as for when all the components have no geometric stiffening. The same alters (DMAPs) used for the synthesis and checking of the system model made from linear components can be used.

## MODEL CHECKING

It is always a good practice to check the reduced matrices. A system model consisting of only the reduced mass and stiffness matrices, and fixed at the boundary degrees of freedom, should give the same vibration modes as the physical (unreduced) model. This verifies the mass and stiffness matrices of the component mode model. A free-free frequency run using the reduced mass and stiffness matrices may be used to check if there is any grounding. To check the data recovery (also known as output transformation) and load transformation matrix columns corresponding to the vibration modes, simply run Solution 103 using the DMAP statements described above to read in a stiffness matrix saved in Solution 106. Do not use the DMAP statements used to generate the reduced matrices. To check the data recovery matrix and load transformation matrix columns corresponding to the attachments modes, simply run Solution 101 using the following DMAP statements to read in a stiffness matrix saved in Solution 106

```
$
$ ALTER IS FOR SOL 101 OF VERSION 70.5
$ ALTER DEVELOPED BY T. K. GHOSH OF ROCKETDYNE.
$ THIS ALTER WILL READ A PUNCH (I.E. DMIG) FILE CONTAINING
$ THE STIFFNESS MATRIX.
$
COMPILE SEMG, SOUIN=MSCSOU, LIST, REF $
ALTER 53
MTRXIN ,,MATPOOL,EQEXINS,SILS,/KNNEXT,,/LUSETS/  $
PURGEX  /KJJZ,,,, $
EQUIVX  KNNEXT/KJJZ/ALWAYS $
```

The data recovery items of the matrices should match the data recovery using the unreduced model in Solutions 101 and 103.

Checking the mass properties of the reduced model is much more difficult, and can be performed in one of two ways. One method is to use the DMIG form of the reduced mass and stiffness matrices that are also printed out along with the OUTPUT4 form. The MSC/NASTRAN provided alter to generate the system model can be used to obtain the component mass matrix. An alternative method is to use the method of Reference-9 to develop a DMAP that would give the mass properties of the component model.

## TEST CASE

A sample problem consisting of a cantilever beam, modeled using plate elements was used as a test case. The data deck for Solution 106 is shown in Figure-4. The first ten frequencies with geometric stiffening using Solution 106 are shown in Figure-5. The stiffness matrix from Solution 106 was saved and reused in altered Solution 103 to generate the component mode matrices. The frequencies from this step are shown in Figure-6 and they agree with those of Figure-5.

A sample data deck to post-process the reduced matrices in OUTPUT4 form using the computer code of Figure-3 is shown in Figure-7.

## DISCUSSION

The test case shows that the method of this paper correctly generates the component mode model of a component with geometric stiffening. This method is much simpler than the one in Reference-4. When the method of this paper was used to generate a reduced model of the solar array of Figure-2, it took less than half the time required when the method of Reference-4 was used. The method of this paper was developed, because it was too time consuming and difficult to update the DMAPs of Reference-4 which are based on Version 67.5 of MSC/NASTRAN. It is much simpler to modify the simple DMAPs to read and write a stiffness matrix than it is to modify the DMAPs of Reference-4 with the new version of MSC/NASTRAN. The absence of any elaborate user developed software demonstrates the simplicity of the new method.

This paper basically generalizes the method used in MSC/NASTRAN to generate a component mode model when there is no geometric stiffening so that it can be used even if there is geometric stiffening. This is implemented through a simple writing of the stiffness matrix in Solution 106 to a file and reading it back in Solution 103. In other words, the elaborate method of Reference-4 has been replaced by a simple read and a simple write.

## ACKNOWLEDGMENT

This work was performed at Rocketdyne Division, Boeing North American, Inc. The material presented here is based on work performed for the International Space Station under NASA contract HX3210. The author wishes to acknowledge the support of Rocketdyne and NASA management.

## REFERENCES

1. Craig, R. R., "A Review of Time-Domain and Frequency Domain Component Mode Synthesis Methods," Int. J. Analytical and Experimental Modal Analysis, Vol. 2, No. 2, 1987.
2. Craig, R. R., "Structural Dynamics", John Wiley & Sons, New York, 1981.
3. Cook, R. D., Malkus, D. S. and Plesha, M. E.," Concepts and Applications of Finite Element Analysis", John Wiley & Sons, 3rd Edition, 1989.
4. Bedrossian, H., and Rose, T., "DMAP Alters for Nonlinear Craig-Bampton Component Modal Synthesis", 1993 MSC World Users Conference.
5. MSC/NASTRAN User's Manual, Version 67, Volumes I & II, The MacNeal Schwendler Corp., Los Angeles, California, August 1991.
6. Meirovitch, L., "Elements of Vibration Analysis", Second Edition, McGraw-Hill Book Co, NewYork, 1986.
7. Ghosh, T. K., "Improved Method of Modal Synthesis in the analysis of International Space-station Structures", 38th AIAA/ASME/ASCE/AHS/ASC Structural Dynamics, and Materials Conference, Proceedings, Kissimmee, Florida, April 7-10, 1997.

8. Ghosh, T., "MSC/NASTRAN Based Component Mode Synthesis Analysis Without the use of DMAPs", 1996 MSC World Users Conference, Newport Beach, California.

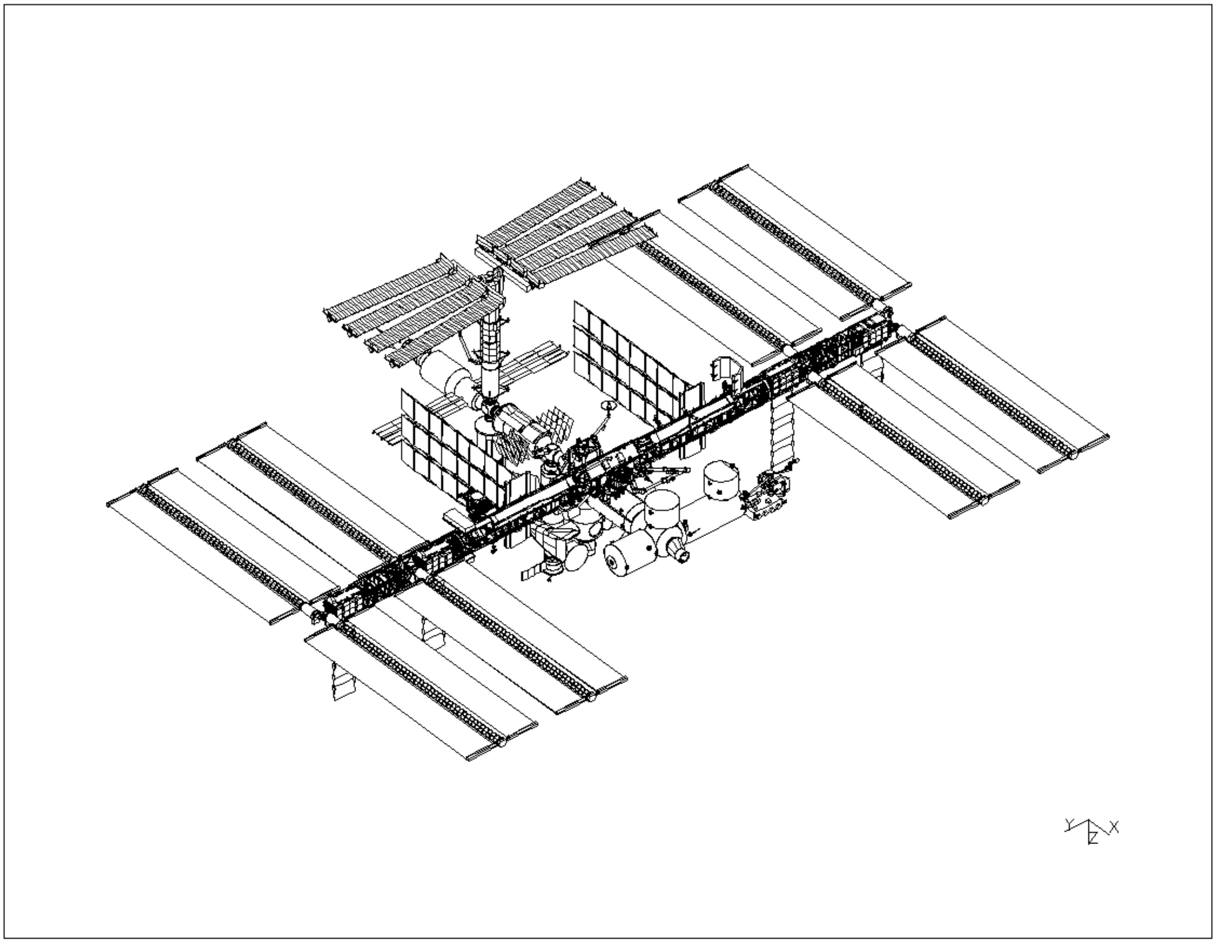


**FIGURE 1: INTERNATIONAL SPACE STATION**

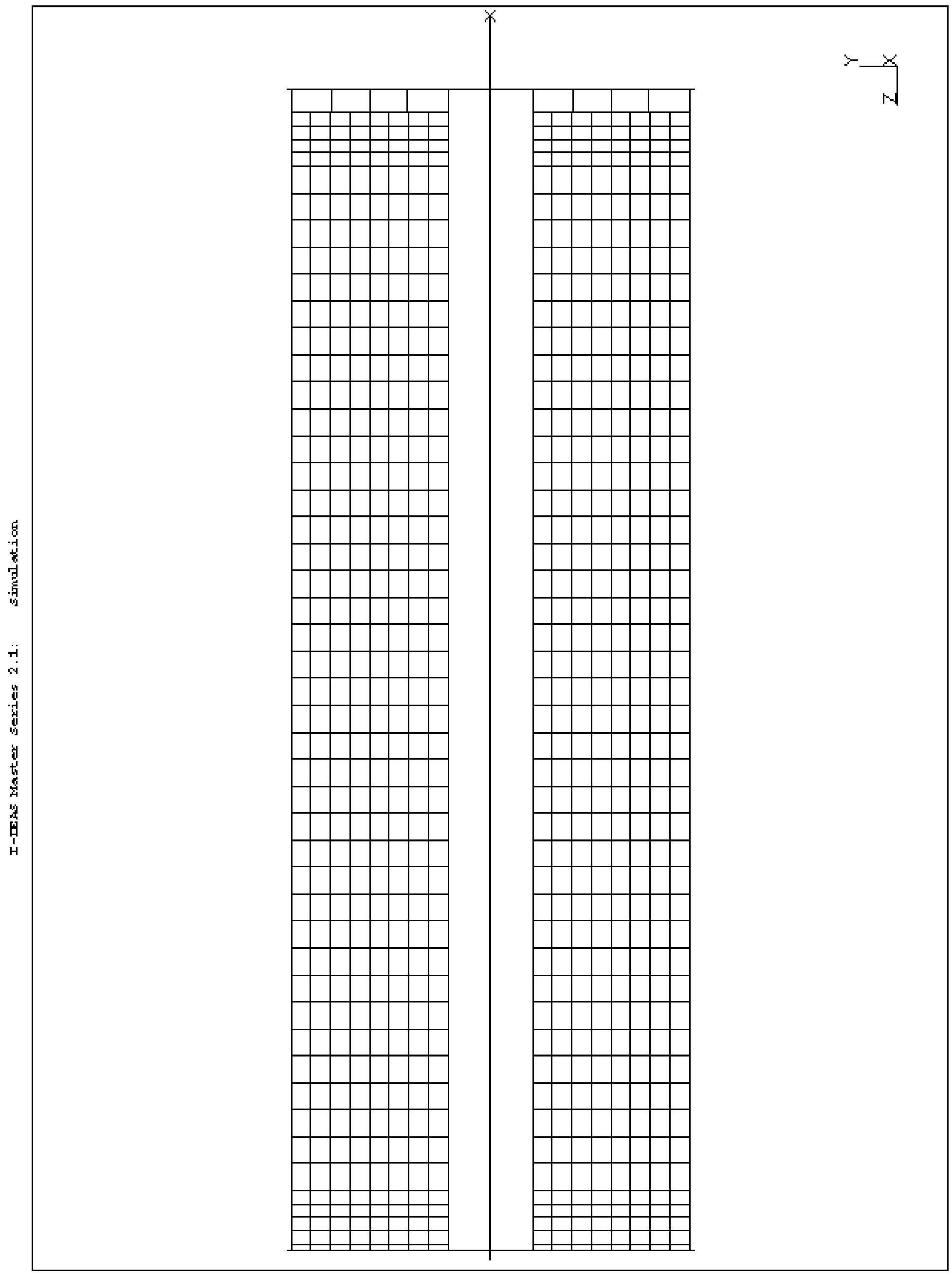


**FIGURE-2: DEPLOYED SOLAR ARRAY**

```
PROGRAM MODGEN
C
C  DESCRIPTION: POST-PROCESS REDUCED MATRICES GENERATED BY ALTER1G
C  DATE/VERSION: SEPTEMBER 1998
C  AUTHOR: TARUN GHOSH
C  PROGRAM: TO BE RUN WITH FILES OF MSC/NASTRAN VERSION 70.5 AND ABOVE
C  COMPUTER: ON HP 750 SERIES AND ABOVE (OR COMPATIBLE)
C  DOCUMENTATION: AS FOLLOWS
C  NOTES:
C       - THIS PROGRAM IS TO BE USED TO POST-PROCESS REDUCED MATRICES GEN-
C         ERATED BY ALTER1G IN OUTPUT4 FORM AS FOLLOWS:
C              (1) UPDATE REDUCED MATRICES BY ELIMINATING MODES BASED ON
C                  STRAIN ENERGY (OR USER SELECTED)
C              (2) UPDATE DATA RECOVEY MATRICES BY ELIMINATING OR ADDING
C                  DATA RECOVERY ITEMS
C
C       - ROWS ARE ADDED TO OTM BY MULTIPLYING OTM WITH [COEFF] WHERE [COEFF]
C         HAS SIZE NROWADD BY NUMBER OF RETAINED DOF IN COMPONENT MODEL.
C
C       - ROWS ARE REMOVED FROM OTM BASED ON THE CONTENTS OF COLUMN MATRIX [ROW].
C         ELEMENT VALUE OF 1 MEANS KEEP AND 0 MEANS DELETE.
C
C       - THE INPUT DATA MUST BE PREPARED BASED ON THE CONTENTS OF THE MATRIX
C         CONTAINING REDUCED MATRICES.
C
C       - FOLLOWING ARE THE INPUT/OUTPUT FILES:
C           UNIT 11: INPUT DATA
C           UNIT 12: OUTPUT
C           UNIT 13: STRAIN ENERGY BY VIBRATION MODES
C           UBIT 15: INPUT FILE CONTAINING REDUCED MATRICES IN OUTPUT4 FORM
C           UBIT 16: OUTPUT FILE CONTAINING REDUCED MATRICES IN OUTPUT4 FORM
C
C       - FOLLOWING ARE THE USER INPUT DATA:
C           CARD 1: NUMATTM,NUMVIBM,NUMGU2Q,NUMGU2T,NUMDIS,NUMFORC,
C                   NUMSTRS,NUMSPCF,NUMMPCF
C                   (8I10)
C                   NUMATTM: NUMBER OF ATTACHMENT MODES
C                   NUMVIBM: NUMBER OF VIBRATION MODES
C                   NUMGU2Q: NUMBER OF DOF IN [GU2Q]
C                   NUMGU2T: NUMBER OF DOF IN [GU2T]
C                   NUMDIS:  NUMBER OF ROWS IN [DIS]
C                   NUMFORC: NUMBER OF ROWS IN [FORC]
C                   NUMSTRS: NUMBER OF ROWS IN [STRS]
C                   NUMSPCF: NUMBER OF ROWS IN [SPCF]
C                   NUMMPCF: NUMBER OF ROWS IN [MPCF]
C
C           CARD 2: PERCENT
C                   (F10.5)
C                   PERCENT: PERCENT OF STRAIN ENERGY
C
C           CARD 3: NCOLRD
C                   (I10)
C                   NCOLRD: NUMBER OF USER DEFINED MODE SELECTION CARDS
C
C           CARD 4: JTHCOL,COL(JTHCOL)
C                   (2I10)
C                   JTHCOL: COLUMN NUMBER
C                   COL(JTHCOL): 1 TO KEEP, 0 TO ELIMINATE
C                   (NOTE THAT NUMBER OF CARD 4 IS EQUAL TO NCOLRD ON CARD 3)
C
C           CARD 5: MATNAME,MATTYPE,NROWADD,NROWREM,
C                   NOMEGA1,NOMEGA2,MATRIX1,MATRIX2,MATPRT
C                   (A8,5I8,2A8,I8)
C                   MATNAME: MATRIX NAME
C                   MATTYPE: MATRIX TYPE
C                           -1: STIFFNESS
C                            O: DAMPING OR MASS
C                            1: OTM
C                            2: LTM ASSOCIATED WITH ATT DOF, [GU2T]
C                            3: LTM ASSOCIATED WITH VIB DOF, [GU2Q]
C                   NROWADD: NUMBER OF ROWS TO BE ADDED
```

# FIGURE-3: PROGRAM LISTING

```
C                   NROWREM: NUMBER OF ROWS TO BE REMOVED
C                   NOMEGA1: GREATER THAN 0 TO CREATE MATRIX1
C                   NOMEGA2: GREATER THAN 0 TO CREATE MATRIX2
C                   MATRIX1: NAME OF MATRIX WITH COLUMNS TIMES OMEGA
C                   MATRIX2: NAME OF MATRIX WITH COLUMNS TIMES OMEGA**2
C                   MATPRT:  DO NOT PRINT IF .NE. 0
C                   (NOTE THAT ENTRIES NROWADD THRU MATRIX2 ARE USED ONLY WITH OTM)
C
C           CARD 6: IROW,JCOL,DUMMY
C                   (2I10,F10.5)
C                   IROW: ROW NUMBER
C                   JCOL: COLUMN NUMBER
C                   DUMMY: VALUE OF COEFF (IROW, JCOL)
C                   (NOTE THAT IROW LESS THAN ZERO SIGNIFIES END OF CARD TYPE 6)
C
C           CARD 7: IROW,JROW,INCR
C                   (3I10)
C                   IROW: FROM ROW TO BE REMOVED
C                   JROW: TO ROW TO BE REMOVED
C                   INCR: INCREMENT
C                   (NOTE THAT IROW LESS THAN ZERO SIGNIFIES END OF CARD TYPE 7)
C
C                   REPEATE CARD TYPE 5, 6 AND 7 TO READ IN ALL MATRICES
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/AA/SPCFD(1000,1000),COEFF(100,1000)
      CHARACTER*8 MATNAME,MATRIX1,MATRIX2
      CHARACTER*56 TITLE1
      INTEGER COL(1000),ROW(1000)
      DIMENSION MODES(1000),OMEGA2(1000)
C
      READ(11,1001) NUMATTM,NUMVIBM,NUMGU2Q,NUMGU2T,NUMDIS,NUMFORC,
     &              NUMSTRS,NUMSPCF,NUMMPCF
      WRITE(12,1001) NUMATTM,NUMVIBM,NUMGU2Q,NUMGU2T,NUMDIS,NUMFORC,
     &              NUMSTRS,NUMSPCF,NUMMPCF
      READ(11,1002) PERCENT
      WRITE(12,2002) PERCENT
      NCOL=NUMATTM+NUMVIBM
      DO 102 II=1,NCOL
        COL(II)=1
  102 CONTINUE
      IF(PERCENT.GT.0.001) THEN
        CALL ENERGY(PERCENT,NMODES,MODES)
        IF(NUMVIBM.NE.NMODES) GO TO 9001
        DO 103 II=1,NMODES
        COL(NUMATTM+II)=MODES(II)
  103   CONTINUE
      ELSE
      ENDIF
      READ(11,1001) NCOLRD
      WRITE(12,1001) NCOLRD
      IF(NCOLRD.GT.0) THEN
        DO 101 II=1,NCOLRD
        READ(11,1001) JTHCOL,COL(JTHCOL)
        WRITE(12,1001) JTHCOL,COL(JTHCOL)
  101   CONTINUE
      ELSE
      ENDIF
      WRITE(12,1003) (II,COL(II),II=1,NCOL)
C
    1 READ(11,1004,END=9999) MATNAME,MATTYPE,NROWADD,NROWREM,
     &                       NOMEGA1,NOMEGA2,MATRIX1,MATRIX2,MATPRT
      WRITE(12,1004) MATNAME,MATTYPE,NROWADD,NROWREM,
     &                       NOMEGA1,NOMEGA2,MATRIX1,MATRIX2,MATPRT
      IF(MATTYPE.LE.1) CALL MATOPT4(MATNAME,NCOL,COL,MATTYPE,0,
     &                              NEWNCOL,NEWNROW,TITLE1,MATPRT)
      IF(MATTYPE.GT.1) CALL MATOPT4(MATNAME,NCOL,COL,MATTYPE,NUMATTM,
     &                              NEWNCOL,NEWNROW,TITLE1,MATPRT)
C
      IF(NROWADD.GT.0) THEN
```

```
        NROW=NEWNROW
        DO 108 JJ=1,NROW
```

# FIGURE-3: PROGRAM LISTING (CONTD.)

```
        DO 108 II=1,NROWADD
        COEFF(II,JJ)=0.0
  108   CONTINUE
    5   READ(11,1005) IROW,JCOL,DUMMY
        WRITE(12,1005) IROW,JCOL,DUMMY
        IF(IROW.LT.0) GO TO 6
        COEFF(IROW,JCOL)=DUMMY
        GO TO 5
    6   CALL MATADDR(MATNAME,NROW,ROW,MATTYPE,NUMATTM,
     &   NCOL,NEWNROW,TITLE1,NROWADD)
      ELSE
      ENDIF
C
      IF(NROWREM.GT.0) THEN
        NROW=NEWNROW
        DO 106 II=1,NROW
        ROW(II)=1
  106   CONTINUE
    4   READ(11,1001) IROW,JROW,INCR
        WRITE(12,1001) IROW,JROW,INCR
        IF(IROW.LT.0) GO TO 3
        DO 107 II=IROW,JROW,INCR
        ROW(II)=0
  107   CONTINUE
        GO TO 4
    3   CONTINUE
        CALL MATREMR(MATNAME,NROW,ROW,MATTYPE,NUMATTM,
     &              NCOL,NEWNROW,TITLE1)
      ELSE
      ENDIF
C
    2 IF(MATPRT.NE.0) GO TO 10
      WRITE(16,1022) NEWNCOL,NEWNROW,TITLE1
      IROW=1
      NW=NEWNROW
      DO 105 ICOL=1,NEWNCOL
      WRITE(16,1033) ICOL,IROW,NW,
     &                      (SPCFD(I,ICOL),I=IROW,IROW+NW-1,1)
  105 CONTINUE
      WRITE(16,1033) NEWNCOL+1,NEWNROW+1,NEWNCOL/NEWNCOL,DUMMY
C
   10 CONTINUE
C
      IF(NOMEGA1.GT.0) THEN
      WRITE(16,2044) NEWNCOL,NEWNROW,MATRIX1
      IROW=1
      NW=NEWNROW
      DO 111 ICOL=1,NEWNCOL
      WRITE(16,1033) ICOL,IROW,NW,
     &  (SQRT(OMEGA2(ICOL))*SPCFD(I,ICOL),I=IROW,IROW+NW-1,1)
  111 CONTINUE
      WRITE(16,1033) NEWNCOL+1,NEWNROW+1,NEWNCOL/NEWNCOL,DUMMY
      ELSE
      ENDIF
C
      IF(NOMEGA2.GT.0) THEN
      WRITE(16,2044) NEWNCOL,NEWNROW,MATRIX2
      IROW=1
      NW=NEWNROW
      DO 112 ICOL=1,NEWNCOL
      WRITE(16,1033) ICOL,IROW,NW,
     &       (OMEGA2(ICOL)*SPCFD(I,ICOL),I=IROW,IROW+NW-1,1)
  112 CONTINUE
      WRITE(16,1033) NEWNCOL+1,NEWNROW+1,NEWNCOL/NEWNCOL,DUMMY
      ELSE
      ENDIF
C
```

```
      IF(MATTYPE.LT.0) THEN
      DO 109 ICOL=1,NEWNCOL
      IF(ICOL.LE.NUMATTM) OMEGA2(ICOL)=0.0
      IF(ICOL.GT.NUMATTM) OMEGA2(ICOL)=SPCFD(ICOL,ICOL)
  109 CONTINUE
```

# FIGURE-3: PROGRAM LISTING (CONTD.)

```
      WRITE(12,2011) (OMEGA2(I),I=1,NEWNCOL)
      ELSE
      ENDIF
C
      GO TO 1
C
 1001 FORMAT(8I10)
 1002 FORMAT(4F10.5)
 1003 FORMAT(2I10)
 1004 FORMAT(A8,5I8,2A8,I8)
 1005 FORMAT(2I10,F10.5)
 1022 FORMAT(I8,I8,A56)
 1033 FORMAT(I8,I8,I8/(5E16.9))
 2044 FORMAT(2I8,'       2       1',A8,'1P,5E16.9')
 2002 FORMAT(T5,'PERCENT=',F10.5)
 2011 FORMAT(5E16.5)
 9001 WRITE(12,2901)
 2901 FORMAT(' FATAL ERROR 9001*********')
 9999 STOP
      END
C
      SUBROUTINE MATOPT4(MATNAME,NCOL,COL,MATTYPE,NUMATTM,
     &NEWNCOL,NEWNROW,TITLE1,MATPRT)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/AA/SPCFD(1000,1000),COEFF(100,1000)
      INTEGER COL(NCOL)
      CHARACTER*56 TITLE1
      CHARACTER*8 MATNAME
      INTEGER COLREM(1000),ROWREM(1000)
C
      READ(15,1022,END=9002) NCOL,NROW,TITLE1
 1022 FORMAT(I8,I8,A56)
      WRITE(12,1022) NCOL,NROW,TITLE1
      IF(MATPRT.EQ.-1) WRITE(16,1022) NCOL,NROW,TITLE1
      DO 100 II=1,NROW
      DO 100 JJ=1,NCOL
      SPCFD(II,JJ)=0.0
  100 CONTINUE
      NEWNCOL=NCOL
      NEWNROW=NROW
    1 CONTINUE
      READ(15,1033,END=9006) ICOL,IROW,NW,
     &                      (SPCFD(I,ICOL),I=IROW,IROW+NW-1,1)
      IF(MATPRT.EQ.-1) WRITE(16,1033) ICOL,IROW,NW,
     &                      (SPCFD(I,ICOL),I=IROW,IROW+NW-1,1)
 1033 FORMAT(I8,I8,I8/(5E16.9))
      IF(ICOL.LT.NCOL) GO TO 1
      READ(15,1033,END=9007) ICOL,IROW,NW,DUMMY
      IF(MATPRT.EQ.-1) WRITE(16,1033) ICOL,IROW,NW,DUMMY
      IF(MATTYPE.EQ.2) GO TO 2
      IJ=0
      DO 101 JJ=1,NCOL
      IF(COL(JJ+NUMATTM).EQ.0) GO TO 101
      IJ=IJ+1
      DO 102 II=1,NROW
      SPCFD(II,IJ)=SPCFD(II,JJ)
  102 CONTINUE
  101 CONTINUE
      NEWNCOL=IJ
      IF(MATTYPE.GT.0) GO TO 2
      IJ=0
      DO 103 II=1,NROW
      IF(COL(II).EQ.0) GO TO 103
      IJ=IJ+1
```

```
      DO 104 JJ=1,NEWNCOL
      SPCFD(IJ,JJ)=SPCFD(II,JJ)
  104 CONTINUE
  103 CONTINUE
      NEWNROW=IJ
    2 CONTINUE
      IROW=1
      NW=NEWNROW
```

# FIGURE-3: PROGRAM LISTING (CONTD.)

```
      DO 105 ICOL=1,NEWNCOL
  105 CONTINUE
      RETURN
 9002 WRITE(12,2902)
 2902 FORMAT(' FATAL ERROR 9002*********')
      RETURN
 9006 WRITE(12,2906)
 2906 FORMAT(' FATAL ERROR 9006*********')
      RETURN
 9007 WRITE(12,2907)
 2907 FORMAT(' FATAL ERROR 9007*********')
      RETURN
      END
C
      SUBROUTINE MATADDR(MATNAME,NROW,ROW,MATTYPE,NUMATTM,
     &NCOL,NEWNROW,TITLE1,NROWADD)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/AA/SPCFD(1000,1000),COEFF(100,1000)
      INTEGER ROW(NROW)
      CHARACTER*56 TITLE1
      CHARACTER*8 MATNAME
C
      DO 101 II=1,NROWADD
      DO 101 JJ=1,NCOL
      SPCFD(NROW+II,JJ)=0.0
      DO 101 KK=1,NROW
      SPCFD(NROW+II,JJ)=SPCFD(NROW+II,JJ)+COEFF(II,KK)*SPCFD(KK,JJ)
  101 CONTINUE
      NEWNROW=NROW+NROWADD
      RETURN
      END
C
      SUBROUTINE MATREMR(MATNAME,NROW,ROW,MATTYPE,NUMATTM,
     &NCOL,NEWNROW,TITLE1)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER ROW(NROW)
      COMMON/AA/SPCFD(1000,1000),COEFF(100,1000)
      CHARACTER*56 TITLE1
      CHARACTER*8 MATNAME
C
      IJ=0
      DO 103 II=1,NROW
      IF(ROW(II).EQ.0) GO TO 103
      IJ=IJ+1
      DO 104 JJ=1,NCOL
      SPCFD(IJ,JJ)=SPCFD(II,JJ)
  104 CONTINUE
  103 CONTINUE
      NEWNROW=IJ
    2 CONTINUE
      IROW=1
      NW=NEWNROW
      DO 105 ICOL=1,NCOL
  105 CONTINUE
      RETURN
      END
C
      SUBROUTINE ENERGY(PERCENT,NSET,MODES)
      IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*1 DOLLAR
      REAL PERCNT
```

```
      DIMENSION X(2000),Y(2000),Z(2000),V(2000),W(2000),MODES(2000)
C
      PERCNT=PERCENT
C
C     READ THE INPUT FILE
C
      NTCOM=0
      DO 100 J=1,9999999
      READ(13,1,END=991)DOLLAR
    1 FORMAT(A1)
      IF(DOLLAR.EQ.'$')NTCOM=NTCOM+1
```

# FIGURE-3: PROGRAM LISTING (CONTD.)

```
  100 CONTINUE
  991 NDATA=J-1
      REWIND 13
C
      NCOM=0
      DO 200 J=1,9999999
      READ(13,1,END=291)DOLLAR
      IF(DOLLAR.EQ.'$')NCOM=NCOM+1
      IF(DOLLAR.NE.'$')GO TO 291
  200 CONTINUE
  291 CONTINUE
      REWIND 13
C
      NSET=NTCOM/NCOM
      NDAT=NDATA/NSET-NCOM
      WRITE(12,1661)NCOM,NTCOM,NSET,NDATA,NDAT
 1661 FORMAT(//,'  NCOM =',I5,/
     *         ,'  NTCOM=',I5,/
     *         ,'  NSET =',I5,/
     *         ,'  NDATA=',I5,/
     *         ,'  NDAT =',I5,//)
      WRITE(12,4999)
 4999 FORMAT(3X,'MODE      MODES','        TOTAL STRAIN ',
     *      6X,'  STRAIN ENERGY',
     *       6X,'  PERCENT',/,4X,'NO     SELECTED',10X,'ENERGY',
     *      13X,'  IN SET',13X,'ENERGY',/)
C
C        READ NONLINEAR STRAIN ENERGIES
C
      IK=0
      DO 2100 K=1,NSET
      DO 201 JJ=1,NCOM
  201 READ(13,1)DOLLAR
      ENSUM=0.0
      ENPCT=0.0
      DO 202 KK=1,NDAT
      READ(13,203)X(KK),Y(KK)
  203 FORMAT(18X,2F18.0)
      ENSUM=ENSUM+X(KK)
      ENPCT=ENPCT+Y(KK)
      ENTOT=ENSUM*100.0/ENPCT
  202 CONTINUE
      V(K)=ENSUM
C     Z(K)=ENPCT
      W(K)=ENTOT
      Z(K)=V(K)*100.0/W(K)
      ENPCT=Z(K)
      IF(ENPCT.GT.PERCNT)IK=IK+1
      IF(ENPCT.GT.PERCNT)
     *WRITE(12,3999)K,IK,W(K),V(K),Z(K)
 3999 FORMAT(I5,I11,9X,1PE12.4,9X,1PE12.4,7X,1PE12.4)
      IF(ENPCT.LT.PERCNT)
     *WRITE(12,3998)K,W(K),V(K),Z(K)
 3998 FORMAT(I5,20X,1PE12.4,9X,1PE12.4,7X,1PE12.4)
      IF(ENPCT.GT.PERCNT)
     *MODES(K)=IK
      IF(ENPCT.LT.PERCNT)
     *MODES(K)=0
```

```
2100 CONTINUE
     RETURN
     END
```

**FIGURE-3: PROGRAM LISTING (CONTD.)**

```
ID GHOSH, SOL106

TIME 100

DIAG 8,14,56

SOL 106 $ NonLinear Statics

CEND

TITLE = CANTILEVERED BEAM MADE OF PLATES

SET 100 = 1, 2, 3

SET 101 = 101, 201

SET 102 = 1, 2

SET 103 = 101, 201, 111, 211
ESE = 100
SPCF= 101
ELFO= 102
DISP= 103
METHOD    = 888
SUBCASE   10601 $ NonLinear Statics
   LOAD = 10601
 NLPARM = 10602
 SPC = 1
BEGIN BULK
$.......2.......3.......4.......5.......6.......7.......8.......9.......0
$
$                  Cantilevered Beam Made of Plates Model
$
$            \201 202  203  204  205  206  207  208  209  210  211
$  Y         \*----*----*----*----*----*----*----*----*----*----*
$  ^         \|    |    |    |    |    |    |    |    |    |    |
$  |         \| 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 |
$  |         \|    |    |    |    |    |    |    |    |    |    |
$  +--->X    \*----*----*----*----*----*----*----*----*----*----*
$            \101 102  103  104  105  106  107  108  109  110  111
$
$.......2.......3.......4.......5.......6.......7.......8.......9.......0
GRID    101             0.      0.      0.
GRID    102             1.      0.      0.
=       *1      =       *1.     ==
=8
GRID    201             0.      1.      0.
GRID    202             1.      1.      0.
=       *1      =       *1.     ==
=8
$
CQUAD4  1       1       101     102     202     201
=       *1      =       *1      *1      *1      *1
=8
MAT1    1       10.E6           .3      2.588-4 1.E-6   0.
PSHELL  1       1       .1      1
FORCE   10601   111             1.      0.      0.      1.
NLPARM  10602   2               AUTO
PARAM   LGDISP  +1
PARAM,TINY,0.0
PARAM,COUPMASS,1
PARAM,DBDICT,2
PARAM,GRDPNT,0
PARAM,NMLOOP,2
PARAM,USETPRT,11
PARAM,NSPOINT,010
EIGRL,888,,,10
SPC1,1,123456,101,201
ENDDATA
```

# FIGURE-4: SAMPLE PROBLEM

```
1    CANTILEVERED BEAM MADE OF PLATES
SEPTEMBER   3, 1998  MSC/NASTRAN  4/28/98   PAGE   307

0

                                                        R E A L   E I G E N V A L U E S
   MODE    EXTRACTION      EIGENVALUE            RADIANS             CYCLES
    NO.       ORDER
```

| MODE NO. | EXTRACTION ORDER | EIGENVALUE | RADIANS | CYCLES |
|---|---|---|---|---|
| 1 | 1 | 4.027513E+04 | 2.006866E+02 | 3.194027E+01 |
| 2 | 2 | 1.617668E+06 | 1.271876E+03 | 2.024253E+02 |
| 3 | 3 | 4.154381E+06 | 2.038230E+03 | 3.243943E+02 |
| 4 | 4 | 1.342324E+07 | 3.663774E+03 | 5.831077E+02 |
| 5 | 5 | 1.571083E+07 | 3.963689E+03 | 6.308407E+02 |
| 6 | 6 | 5.639206E+07 | 7.509464E+03 | 1.195168E+03 |
| 7 | 7 | 1.408952E+08 | 1.186993E+04 | 1.889158E+03 |
| 8 | 8 | 1.651239E+08 | 1.285006E+04 | 2.045150E+03 |
| 9 | 9 | 1.729998E+08 | 1.315294E+04 | 2.093355E+03 |
| 10 | 10 | 4.175458E+08 | 2.043394E+04 | 3.252162E+03 |

**FIGURE-5: FREQUENCIES FROM SOLUTION 106**

```
1    CREATE EXTERNAL SUPERELEMENT MODEL OF S.E. 10          CBAR1G
SEPTEMBER   3, 1998  MSC/NASTRAN  4/28/98   PAGE   347

      TEST RESIDUAL VECTORS
SUPERELEMENT 10
0
SUBCASE 1
```

| | | | R E A L   E I G E N V A L U E S | |
|---|---|---|---|---|
| MODE NO. | EXTRACTION ORDER | EIGENVALUE | RADIANS | CYCLES |
| 1 | 1 | 4.027414E+04 | 2.006842E+02 | 3.193988E+01 |
| 2 | 2 | 1.617667E+06 | 1.271875E+03 | 2.024253E+02 |
| 3 | 3 | 4.154380E+06 | 2.038230E+03 | 3.243943E+02 |
| 4 | 4 | 1.342324E+07 | 3.663774E+03 | 5.831077E+02 |
| 5 | 5 | 1.571084E+07 | 3.963690E+03 | 6.308408E+02 |
| 6 | 6 | 5.639206E+07 | 7.509464E+03 | 1.195168E+03 |
| 7 | 7 | 1.408952E+08 | 1.186993E+04 | 1.889158E+03 |
| 8 | 8 | 1.651239E+08 | 1.285006E+04 | 2.045150E+03 |
| 9 | 9 | 1.729998E+08 | 1.315294E+04 | 2.093355E+03 |
| 10 | 10 | 4.175458E+08 | 2.043394E+04 | 3.252162E+03 |

**FIGURE-6: FREQUENCIES FROM SOLUTION 103**

```
         12         10          6          6         24         16          0
0
          0
       34.0
          3
         13          1
         14          0
         17          1
       KAA      -1
       MAA
       BAA
      GU2Q       3
      GU2T       2
       DIS       1      00      00      01      01     VEL     ACC
      FORC       1      02     10
          1          1       0.5
          1          2       0.5
          2          3       0.5
          2          4       0.5
        -99
         01         05        01
         06         10        01
        -99
```

**FIGURE-7: INPUT DATA FOR SAMPLE PROBLEM)**