# Creation of and use of "Craig-Bampton" Models Using MSC/NASTRAN

by

Ted L. Rose

Staff Scientist

MacNeal-Schwendler Corp.

MSC World Users Conference, March, 1990

## ABSTRACT

As structures become larger and more complex, many programs are requiring contractors to deliver "Craig-Bampton"* models of hardware instead of complete finite element models. This is identical to the default method used when performing component modal synthesis in MSC/NASTRAN using superelements. This method of component modal synthesis consists of reducing a finite element model into a set of generalized mass and stiffness matrices which can be connected to physical grid points. The generalized coordinates used to create these matrices consist of "constraint" modes and fixed-interface elastic modes of the structure.

This paper presents a simple method to create the generalized mass and stiffness matrices, and shows how to use them in further analysis.

* Coupling of Substructures for Dynamic Analysis, R. R. Craig and M. C. C. Bampton, AIAA Journal, vol 6, no 7, July, 1968.

I THE "CRAIG-BAMPTON" METHOD - "FIXED-BOUNDARY" COMPONENT MODAL SYNTHESIS (CMS)

A. Description of Methodology (matrices)

In order to perform "Craig-Bampton" component modal synthesis, it is necessary to divide the model into two sets of degrees of freedom (dof). The first set, which are called the "B" set represents the boundary points. The second set is the interior dof, which are called the "O" set.

A set of "constraint" modes is generated. Each "constraint" mode represents the motion of the model resulting from moving one boundary dof 1.0 unit, while holding all other boundary dof fixed. Therefore, there will be one "constraint" mode for each boundary dof.

In matrix form,

$$\begin{bmatrix} K_{oo} & K_{ob} \\ K_{bo} & K_{bb} \end{bmatrix} \begin{Bmatrix} O_{ob} \\ I_{bb} \end{Bmatrix} = \begin{Bmatrix} 0 \\ P_b \end{Bmatrix}$$

($P_b$ is not actually applied)

The first line gives:

$$\{O_{ob}\} = -[K_{oo}]^{-1}[K_{ob}]\{I_{bb}\}$$

giving the following "constraint" modes:

$$\{O_b\} = \begin{Bmatrix} O_{ob} \\ I_{bb} \end{Bmatrix}$$

Now the "B" dof are partitioned out and the "O" set equations are solved for the "fixed-boundary" modes, $\{O_{oo}\}$.

$$-w^2[M_{oo}]\{O_{oo}\} + [K_{oo}]\{O_{oo}\} = 0$$

as many of these "O" set modes as are desired are found. They are then concatenated with the "constraint" modes to form the generalized coordinates.

$$\{O_G\} = \begin{Bmatrix} O_{ob} & O_{oo} \\ I_{bb} & 0 \end{Bmatrix}$$

The mass and stiffness matrices are pre- and post-multiplied by these modes to obtain the "generalized" mass and stiffness.

$$[K_{Gen}] = \{O_G\}^T[K_{ff}]\{O_G\}$$

$$[M_{Gen}] = \{O_G\}^T[M_{ff}]\{O_G\}$$

(where the "F" set is the union of the "B" and "O" sets)

These "generalized" matrices contain physical dof representing the boundaries and "modal" coordinates representing the "fixed-boundary" component modes.

At this point, these matrices can be treated like any other structural matrices, and data recovery can be performed for the component in a manner similar to using modal coordinates. That is, the displacements of the generalized coordinates are multiplied by the associated vectors and added together to obtain the component displacements.

## II. WHY USE CMS

### A. Reduce Problem size (ECONOMY)

Structures are becoming larger and more complicated, requiring larger, more complex finite element models to represent their response. Solving a structure which consists of several complex models is time consuming, and restarts are almost as bad. If the number of dof needed to represent a component can be reduced, so will the effort and complexity.

### B. Allow multiple configurations (ease of use)

Some structures such as the space station have a number of possible configurations. Component modal synthesis allows dynamic solution of many configurations by simply combining the appropriate components.

"Multi-Level" CMS allows further reduction by assembling sets of components

### C. Allow different groups(companies) to create models

The deliverable model consists of boundary mass, stiffness and damping(if needed), allowing the following:

1. Duplicate IDs

Each component can be modeled individually, allowing repeated element and grid point numbers in different components.

2. Security

Since only "boundary" matrices are transmitted, no physical information describing the geometry or properties is sent. All "interior" information has been condensed to the boundaries.

3. Segmented model checkout

Since each segment is modeled separately, it can be checked individually, simplifying the check procedure.

## III. PROCEDURE USING MSC/NASTRAN

The first step is to divide the structure into components.

Next, the appropriate boundaries are defined and provided to each group (or company) which is modeling a component.

### A. Individual Contractor (Create Craig-Bampton model)

1. Create model

Create finite element model of each component using superelements. Multi-level models may be used for individual components.

Boundaries should be defined as belonging to the residual structure (not mandatory, but easiest for the user)

2. Check model

Each superelement can be checked individually, using both static and dynamic checks with PARAM,FIXEDB,-1

Checkout may include both static and modal test.

Test the model by applying loads which simulate the expected loading environment.

3. Reduce model by CMS

a. Once the model is ready, select the GRID points which will be loaded in the system analysis. These dof will be placed in the "U2" set by using SEUSET entries.

Example: Superelement 10 of the sample problem will have loads applied to GRID 8 dof 3, 4, or 5, so the following entries appear in the BULK DATA:

```
SEUSET,10,U2,8,345
DEFUSET,U2,U2
```

These entries, along with the following DMAP ALTER will create the reduced matrices for the superelement and allow loads to be applied to GRID point 8.

```
COMPILE PHASE1 SOUIN=MSCSOU NOREF NOLIST
$ ALTER FOR SOL 103 TO CREATE CRAIG-BAMPTON MODEL
ALTER 66 $ AFTER CALL SEGOA
TYPE DB ZUZR01,ZUZR02,ZUZR03 $
TYPE PARM NDDL I N ZUZR1 $
ZUZR1=SEID $
ADD       MLAA,MAA/ZUZR01 $
ADD       KLAA,KAA/ZUZR02 $
OUTPUT4   ZUZR01,ZUZR02,,,//0/12/-1 $
MATGEN    EQEXINS/INTEXT1/9/0/LUSETS $
UMERGE    USET,GOA,/GOG/'G'/'O'/'A' $
UPARTN    USET,GOG/GU2,,,/'G'/'U2'/'A'/1 $
UMERGE    USET,GU2,/GU2G/'G'/'U2'/'A' $
MPYAD     INTEXT1,GU2G,/PHIXT/1 $
TRNSP     PHIXT/PHIX1 $ G-SIZE MATRIX WITH TERMS FOR U2 SET
MATMOD    PHIX1,,,,,/U2NULL,/12//1 $ SEARCH FOR NULL COLUMNS
PARTN     PHIX1,U2NULL,/ZUZR03,,,/1 $
MATPRN    GU2G,U2NULL,ZUZR03// $
OUTPUT4   ZUZR03,,,,//0/12/-1 $
DBDIR     // $ OPTIONAL - PRINTS DATA BASE DICTIONARY
ENDALTER $
```

b. The next step is to determine the boundary points. These GRID points will represent the physical points which connect to the rest of the structure. Place these GRID points in the residual structure, and place all other GRID points in a superelement (superelement 10 for example).

c. Define the degrees of freedom which will be used to represent the component's modes (BE CONSERVATIVE). If a "fixed-boundary" normal modes analysis has already been done, use those results to determine the number. Depending on whether the model is to be used in a single-level, or multi-level system model, use either SCALAR points or GRID points to represent the component modes. As long as the total number of dof used for component modes is a multiple of 6, the resulting matrices may be used either way. These dof must be placed in the superelement's Q-SET by using SEQSET entries. They must also be exterior dof.

d. Perform CMS using SOL 103 and save the resulting data bases and files.

e. Check resulting matrices before sending them to the system integrator.

B. <u>System Integrator (Combine Craig-Bampton models)</u>

1. Check model

The first step upon receiving the matrices is to create a sample deck to read in the matrices and run a series of checks. Some recommended checks are, "rigid-body" checks (can the component be moved as a rigid-body without any strain energy?), geometry checks, and calculating the component modes of the reduced matrices.

2. Attach to "system" model

Once the matrices have been read into MSC/NASTRAN and checked, they can be attached to the "system" model as desired. Once again, a series of checks should be done to check displacement compatibility, etc.

3. Perform "system" analysis

Once the "system" model is ready, system modes can be calculated for the combined structure. After reviewing the system modes, a restart to perform a modal transient (or any other dynamic analysis) of the system can be performed, using the results of the modal run.

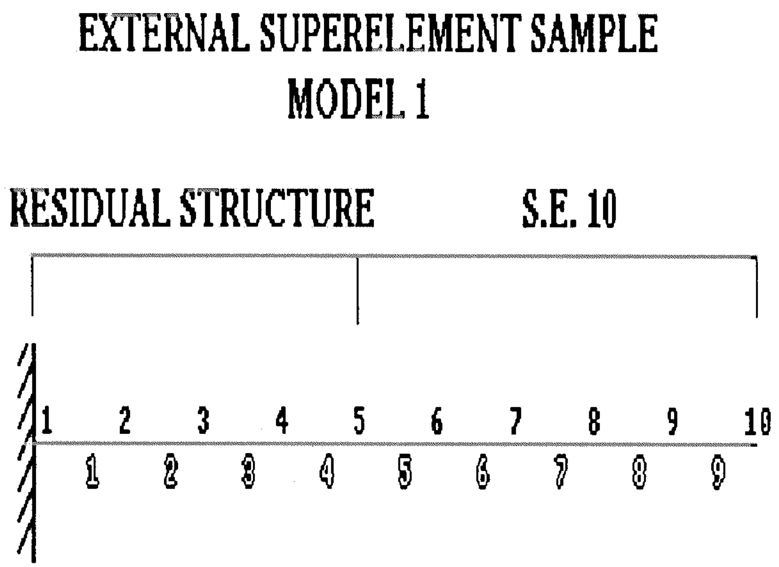
4. Review results

One of the most important parts of the analysis. The results should be checked to see that they are reasonable. Possible checks include checking total applied loads, maximum deformation, modal kinetic energy, modal "effective weight", deformed plots, and calculated values, such as stresses, forces, etc.

5. Send selected results to contractors

As a minimum, boundary displacements for the system solution should be sent (For both the transient and system modal analysis). An option is to have the component contractors provide data recovery matrices and multiply them by the boundary solutions.

SAMPLE PROBLEM




External Superelement Sample

The purpose of these samples is to demonstrate the creation of external superelement matrices by component modal synthesis, then the inclusion of the external superelement matrices into a "system" normal modes and transient analysis. The model used is a cantilever beam.

Two sets of samples were generated. The first (CBAR*.DAT) created only one external superelement and then combined it with the remaining area of the model for system analysis. This shows the "single-level" approach. The second sample (ML*.DAT) created two external superelements and then combined them together into a third before performing the system analysis. This sample demonstrated the use of a "multi-level" analysis, which would be best for the space station or other complex structures, since components could be placed into "assemblies", which would reduce the problem size and storage requirements for further processing.

In addition, a baseline run was performed on the beam without superelements for comparison.

The results of the analyses showed that the methods using external superelements worked correctly and gave good answers.

The structure was modeled using BAR elements and originally divided into one superelement (S.E. 10) and a residual structure. A series of runs were done to create boundary matrices for S.E. 10 and then to read them back in as an external superelement and perform the analysis of the model for normal modes and a modal transient. The problem decks are labeled CBAR*.dat and are described in the following.

CBAR.DAT - Baseline normal modes run. No superelements used, obtain the first 10 modes of the cantilever beam.

CBAR1.DAT - Perform "Craig-Bampton" modal synthesis for superelement 10 and write the boundary matrices to file in OUTPUT4 format (This allows transfer between different computers). A DMAP ALTER is added in the EXECUTIVE CONTROL to calculate the boundary mass and stiffness and a load scaling matrix, PHIXX. A special feature of this run is the creation of scaling matrix which will be used to apply loads to selected interior points of external superelement 10 for use in further runs. This matrix is created by telling MSC/NASTRAN the degrees of freedom in superelement 10 which might be loaded dynamically in future runs thru the use of an "SEUSET". The user specifies the desired DOF as belonging to the "U2" set, which is user defined. In the sample deck, DOF 3,4, and 5 of GRID point 8 are selected. The matrix PHIXX contains the component modes (including constraint modes) for the DOF in the "U2" set. These will be used in later runs to generate dynamic loads for the transient analysis (NOTE: Although this run uses the default "Craig-Bampton" method, any combination of B-set(fixed) and C-set(free) boundary conditions may be used for the component modal synthesis).

CBAR2.DAT - Read matrices from CBAR1 by INPUTT4 and store in a data base identified as belonging to superelement 10. Note that if all runs are done on the same type of computer, the data bases can be used with the "DBLOCATE" command. Since this paper is for general use, it used the INPUTT4 format, which is not dependent of the type of computer used.

CBAR3.DAT - This run contains the model for the rest of the beam. It attaches the matrices for superelement 10 to the residual structure as an external superelement and runs "system" modes. The DMAP alter in this run is the updated version of the one from my 1988 Users' Conference paper and performs rigid body checks on the combined matrices as well as calculating the effective modal weight and kinetic energy for the system modes.

CBAR4.DAT - Same as CBAR3, except that the grid points of the rest of the model are constrained. This gives the modes of superelement 10, along with rigid-body checks for it. (This run is for checking purposes only. It allows the user to check that the connections between the superelement and the rest of the structure are done properly. Any differences between the frequencies from this run and those from CBAR1 would indicate an error in the attachment. The rigid body checks also check that the attachments are done properly. Any reactions in these checks would indicate an error in the connections, or possibly an error made by the contractor in creating the matrices.)

CBAR5.DAT - Baseline modal transient run. Full model without superelements run for modal transient. The results of this run are only for comparison to the superelement runs.

CBAR6.DAT - Check run for superelement modal transient. Full model using superelements. This run placed the boundary loading vector for superelement 10 into a file for comparison to that calculated in later runs.

CBAR7.DAT - Calculate boundary loading vector for external superelement. This does not need to be a separate run, but was done as such for clarity. This run takes the load scaling matrix (PHIXX) from CBAR1 and multiplies it by a user-generated matrix (SCALE) to get the boundary loading vector (PA) and stores it for future use. The matrix SCALE is created by using DMI and is NDOF x 1 matrix, where NDOF is the number of DOF specified by the user in set "U2" in CBAR1, and the "1" is the number of loadings conditions (I haven't tried this for multiple LSEQ-DAREA combinations yet). The terms in the matrix SCALE represent the magnitude of the forces to be applied dynamically at the selected DOF. These loads will be applied dynamically in a later run. In this case, a force of magnitude -2.0 is applied at grid point 8 in DOF 3.

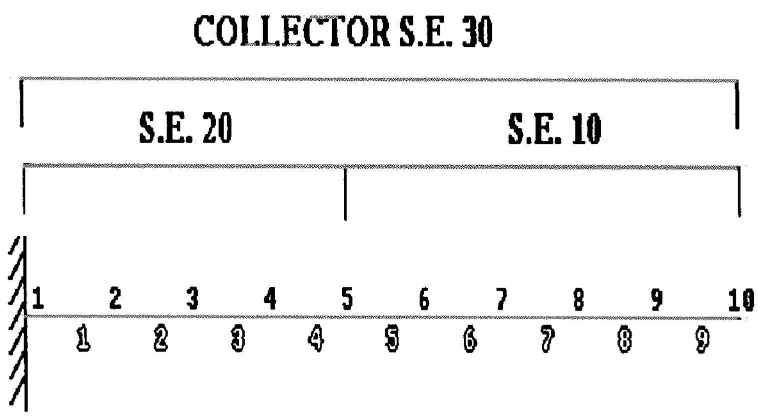
CBAR8.DAT - Modal transient analysis using external superelements. The boundary matrices for external superelement 8 are attached to the rest of the model and a complete modal transient (including normal modes) is performed. In addition to the data in run CBAR3, the dynamic loads are defined for the residual structure in this run. In this case, there are no loadings applied directly to the residual, so the LOADSET-LSEQ combination is used to tell MSC/NASTRAN to bring the PA vector from CBAR7 in and apply it dynamically. The results from this run are identical to run CBAR6.

CBAR9.DAT - Modal transient restart from run CBAR3. This run is identical to CBAR8, except it uses the modes calculated in CBAR3 for a modal transient restart. The commands SELG and SELR are used to control this. Once again, the answers are identical to run CBAR6.

Runs CBAR, CBAR3, CBAR5, CBAR6, and CBAR8 are for verification purposes only and would not be necessary in a production environment. Runs CBAR3 and CBAR6 would be recommended in a production analysis to verify that the procedure was followed properly.

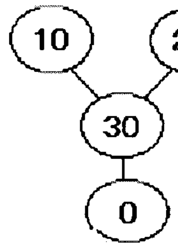
In a production environment, only runs CBAR1, CBAR2, CBAR4, CBAR7, and CBAR9 would be necessary. Runs similar to CBAR1 would be necessary for each contractor in order to create the boundary matrices, and the other runs would be done by the system integrator. Although the procedure to generate and use the load transform matrix (LTM), displacement transform matrix (DTM), and the acceleration transform matrix (ATM) is not shown here, it is a straight-forward procedure and could be easily done.

# EXTERNAL SUPERELEMENT SAMPLE
# MODEL 2




This is the same model as sample 1, except that now a multi-level superelement analysis is performed using external superelements.

Matrices are generated for superelements 10 and 20, then written to file. They are then brought in as external superelements for a multi-level analysis using the following tree:




The only grid point in the residual for this model is grid point 1, and superelement 30, which is a collector for superelements 10 and 20 could have its boundary matrices written out to a file and be used as an external superelement in future analyses.

The following runstreams were used in this analysis:

ML1.DAT AND ML2.DAT - Perform "Craig-Bampton" modal syntheses for superelements 10 and 20 respectively and write matrices to a file in OUTPUT4 format. These runs are similar to CBAR1.

ML3.DAT - Read matrices from ML1 and ML2 by INPUTT4 and store in a data base. Similar to CBAR2, only each set of matrices is identified with the appropriate superelement id.

ML4.DAT - Generate boundary loads for superelement 10 and store in data base. Similar to CBAR7.

ML5.dat - Create assembly model and solve for normal modes. Combines external superelements 10 and 20 to create superelement 30. Since scalar points are used to model the component modes of superelements 10 and 20, they are retained as part of superelement 30 and not coupled into its modes (If it is desired to couple the modes in, then GRID points should be used to represent the modal coordinates in this run and the number of SEQSET dof in runs ML1 and ML2 should be multiples of 6). The residual structure modes are then calculated and match those of the prior problems.

ML6.DAT - Modal transient restart run from ML5. Similar to CBAR9, with the same results.

CBAR1.DAT

```
ASSIGN OUTPUT4=CBAR1.F12,NEW,UNIT=12,FORMATTED
$
$ SIMULATES GENERATION AND STORAGE OF MAA, KAA, AND PHIXX
$ BY A SUBCONTRACTOR. STORAGE MEDIA ARE AN OUTPUT4 FILE
$ AND A DATA BASE FILE.
$
SOL 103
DIAG 8
TIME 10
COMPILE PHASE1 SOUIN=MSCSOU NOREF NOLIST
ALTER 66 $ AFTER CALL SEGOA
TYPE DB ZUZR01,ZUZR02,ZUZR03 $
TYPE PARM NDDL I N ZUZR1 $
ZUZR1=SEID $
ADD      MLAA,MAA/ZUZR01 $
ADD      KLAA,KAA/ZUZR02 $
OUTPUT4  ZUZR01,ZUZR02,,,//0/12/-1 $
$
MATGEN   EQEXINS/INTEXT1/9/0/LUSETS $
UMERGE   USET,GOA,/GOG/'G'/'O'/'A' $
UPARTN   USET,GOG/GU2,,,/'G'/'U2'/'A'/1 $
UMERGE   USET,GU2,/GU2G/'G'/'U2'/'A' $
MPYAD    INTEXT1,GU2G,/PHIXT/1 $
TRNSP    PHIXT/PHIX1 $ G-SIZE MATRIX WITH TERMS FOR U2 SET
MATMOD   PHIX1,,,,,/U2NULL,/12//1 $ SEARCH FOR NULL COLUMNS
PARTN    PHIX1,U2NULL,/ZUZR03,,,/1 $
$
MATPRN   GU2G,U2NULL,ZUZR03// $
$
OUTPUT4  ZUZR03,,,,//0/12/-1 $
DBDIR    // $
CEND
TITLE = CREATE EXTERNAL SUPERELEMENT MODEL OF S.E. 10
SEALL = 10
SPC = 1
$
SUPER =10
  LABEL = CMS OF SUPERELEMENT 10
  PARAM,FIXEDB,-1  $ SUPERELEMENT COMPONENT MODES PRINTOUT
  DISP = ALL
  METHOD=1
  DISP = ALL
BEGIN BULK
PARAM,USETPRT,2
PARAM,GRDPNT,0
EIGR,1,MGIV,0.,100.,,10
$
$ ADD MODAL COORDINATES FOR S.E. 10
$
SPOINT,1001,THRU,1010
SEQSET1,10,0,1001,THRU,1010
$
```

```
$ DEFINE SEUSET U2 FOR INTERIOR POINTS WITH POTENTIAL
$  LOADINGS
$
DEFUSET,U2,U2
SEUSET,10,U2,8,345
GRID,1,,0.,0.,0.
=,*(1),=,*(10.),==
=(8)
GRID,11,,0.,1.,0.,,123456
GRID,12,,0.,1.,0.,,123456
CBAR,1,1,1,2,11
=,*(1),=,*(1),*(1),==
=(2)
CBAR,5,1,5,6,12
=,*(1),=,*(1),*(1),==
=(3)
$
$ DEFINE SUPERELEMENT
$
SESET,10,6,THRU,10
SESET,10,12
PBAR,1,1,1.,10.,10.,10.
PARAM,AUTOSPC,YES
MAT1,1,30.+6,,.3,.283
SPC1,1,123456,1
ENDDATA
```

CBAR2.DAT

```
ASSIGN INPUTT4=CBAR1.F12,UNIT=11,FORMATTED
$$ CREATE DATA BASE FOR USE AS EXTERNAL SUPERELEMENT
$$ IN SPACE STATION RUN
ASSIGN MASTER=CBAR2.MASTER
ASSIGN DBALL=CBAR2.DBALL
ASSIGN USROBJ=CBAR2.OBJ,TEMP $ THROWAWAY
ASSIGN USRSOU=CBAR2.SOU,TEMP $ THROWAWAY
$$ENDFMS
TIME 10
SOL 100
COMPILE USERDMAP SOUIN=MSCSOU
ALTER 2
PUTSYS(1,125) $ ALLOW FOR LATER RESTART WITHOUT WARNING
TYPE PARM,NDDL,I,N,SEID,PEID,ZUZR1 $ QUALIFIERS
TYPE DB MAA,KAA,ZUZR03 $
$ ZUZR03 IS THE PHIXX MATRIX
ZUZR1=10 $ ZUZR1 IS THE QUALIFIER FOR ZUZR0* DATA BLOCKS
SEID=10 $
PEID=10 $
INPUTT4   /MAA,KAA,ZUZR03,,/3/11/-1/-1 $ READ INPUT MATRICES
DBDIR     // $
CEND
TITLE = READ AND STORE MATRICES FOR CUPOLA IN DATA BASE
BEGIN BULK
ENDDATA
```

CBAR3.DAT

```
ASSIGN CBAR2=CBAR2.MASTER
DBLOC  DB=* LOGI=CBAR2 VERS=1
$$ CREATE RESIDUAL STRUCTURE DATA BASE
ASSIGN MASTER=CBAR3.MASTER
ASSIGN DBALL=CBAR3.DBALL
ASSIGN USROBJ=CBAR3.OBJ,TEMP $ THROWAWAY
ASSIGN USRSOU=CBAR3.SOU,TEMP $ THROWAWAY
$
$ ATTACH MATRICES FOR EXTERNAL SUPERELEMENT 10, RUN RIGID
$   BODY CHECKS, AND GET AND SAVE SYSTEM MODES
$
$ENDFMS
SOL 103
TIME 15
CEND
TITLE = CHECK EXTERNAL S.E. 10 MATRICES
$
$  ASSEMBLE MATRICES AND SOLVE RESIDUAL STRUCTURE
$
SPC = 1
METHOD = 1
DISP = ALL
BEGIN BULK
$
$  PERFORM EIGENVALUE ANALYSIS
$
EIGR,1,MGIV,0.,100.,,10
$
$ ADD CSUPER CARD TO BRING EXTERNAL S.E. 10 IN
$
SPOINT,1001,THRU,1010
CSUPER,10,,5,1001,THRU,1010
$
$ ADD GRIDS TO CONNECT EXTERNAL SUPERELEMENT TO
$ MODEL OF RESIDUAL STRUCTURE ONLY'
$
GRID,1,,0.,0.,0.
=,*(1),=,*(10.),==
=(3)
GRID,11,,0.,1.,0.,,123456
CBAR,1,1,1,2,11
=,*(1),=,*(1),*(1),==
=(2)
PBAR,1,1,1.,10.,10.,10.
PARAM,AUTOSPC,YES
MAT1,1,30.+6,,,.283
SPC1,1,123456,1
ENDDATA
```

CBAR4.DAT

```
ASSIGN CBAR2=CBAR2.MASTER $ DATA BASE WITH S.E. 10 MATRICES
DBLOCATE DB=*, LOGICAL=CBAR2,VERS=1 $
$
$ ATTACH MATRICES FOR EXTERNAL SUPERELEMENT 10, RUN RIGID
$   BODY CHECKS, AND CONSTRAIN R.S. TO CHECK S.E. MODES
$
ID CHECK, CBAR MATRICES
SOL 103
TIME 15
$
$ ALTER TO COMPUTE KINETIC ENERGY AND EFFECTIVE WEIGHT
$         FROM U.C. PAPER (1988) BY T. ROSE
$
$     MODIFIED FOR VERSION 66 - MAY, 1989
$
$     MODIFIED NOV 1, 1989 - NORMALIZE 'REAC' VECTORS
$
$     INCLUDES SECONDARY S.E.
$
$
$  ADD CALLS FROM SEKR FOR CHECKS AND "RIGID" BODY VECTORS
$
COMPILE SEKR SOUIN=MSCSOU $
ALTER 13 $                         AFTER LAST "TYPE" CARD
TYPE PARM,,I,N,MATTYP=0 $
TYPE PARM,,I,Y,(CHKSTIF=-1)
CALL CHECK USET//MATTYP/SEID/PEID $
ALTER 20 $                          AFTER GP4
MATTYP=1 $ CHECK KGG
CALL CHECK USETB//MATTYP/SEID/PEID $
ALTER 41 $                           AFTER ENDIF
$ END DO MPC
IF (NOMSET<>-1) THEN $
  MATTYP=2 $ CHECK KNN
  CALL CHECK USETB//MATTYP/SEID/PEID $
ELSE IF (CHKSTIF<>-1)THEN $
  MESSAGE //'NO M-SET FOR THIS SUPERELEMENT'/
           ' - KNN NOT CHECKED' $
ENDIF $
ALTER 120  $                         BEFORE RETURN
MATTYP=3 $ CHECK KAA
CALL CHECK USETB//MATTYP/SEID/PEID $
$
$ PASS PEID FROM SUPER3 TO SEDRCVR - FOR SECONDARY S.E.
$
COMPILE SUPER3 SOUIN=MSCSOU $
ALTER 115,115 $ REPLACE CALL SEDRCVR
IF ( NOOUT>-1 ) CALL SEDRCVR,
         UGVS,QGS,BGPDTS,EQEXINS,CSTMS,CASEDR,MPTS,DIT,
         ETT,OLB1,PJ1,ESTN,XYCDBDR,GEOM2S,GEOM3X,POSTCDB,
         ECTS,GPLS,EPTS,SILS,INDTA,KELM,KDICT,GPECT,VELEM,
         FORCE,XYCDB,PG,PCDBDR,USET,SLT,UHVF,OLBM,PGD,DLT,
```

```
          FRL,SPSEL,DYNAMICS,CRX,QGE/
          PUGV,OUGV1,OPG1,OQG1,OEF1X,OES1X,OSTR1,OGS1,
          OUGV2,OPG2,OQG2,OEF2, OES2, OSTR2,OGS2,
          OES1M,OES1G,OSTR1M,OSTR1G,EGPSTR,EGPSF,
          ONRGY1,OGPFB1/GRDPNT/APP/APP1/NOCOMPX/
          CURVPLOT/S,PFILE/NUMOUT1/NUMOUT2/BIGER1/
          BIGER2/NUMOUT/BIGER/SRTOPT/LSTRN/SRTELTYP/
          CURV/OUTOPT/OG/NINTPTS/S1M/S1G/S1AM/S1AG/
          DOPT/TINY/NOELOF/NOELOP/TABS/SEID/PEID/S1/S,CARDNO/
          PDRMSG/SCRSPECX/RSPECTRX/RSPRINT/TABID/INREL/GPFDR/
          NLHEAT/AERO/ICYCLIC/GEOMU/LOADU/POSTU/
          DBCDIAG/DBCCONVX/POST/S,CP/DBCOVWRT $
$
$ PASS PEID FROM PHASE1 TO SEMRM - FOR SECONDARY S.E.
$
COMPILE PHASE1  SOUIN=MSCSOU $
ALTER 64,64 $ REPLACE CALL SEMRM
CALL SEMRM   MJJ,MAPS,EQEXINS,SLIST,EMAP,GOAT,CASES,
             KFF,DYNAMICS,USET,SILS,GPLS,BGPDTS,CSTMS,
             LOO,MATPOOL,EST,GM/
             MGG,MAA,MFF,MOA1,PHIAZ,PHIOZ,CMLAMA,KLAA,
             GOAQ,MLAA1,MAR,MEA,MLAA/
             NOUP/ERROR/LUSETS/NOMSET/NOOSET/NORC/NOQSET/
             NOTSET/EPSMALC/EPSMALU/MAXRATIO/EPSBIG/INRLM/
             EPSRC/ASING/PRPHIVZ/NOASET/NORC/NOQSET/NOTSET/
             NORSET/NOOSET/SEID/PEID/WTMASS/NOA/NLHEAT/NOSSET
$
$ ADD PEID TO SEMRM AND PASS TO SEMR2 - FOR SECONDARY S.E.
$
COMPILE SEMRM  SOUIN=MSCSOU $
ALTER 1,1 $ REPLACE SUBDMAP STATEMENT
SUBDMAP SEMRM  MJJ,MAPS,EQEXINS,SLIST,EMAP,GOAT,CASES,
               KFF,DYNAMICS,USET,SILS,GPLS,BGPDTS,CSTMS,
               LOO,MATPOOL,EST,GM/
               MGG,MAA,MFF,MOA1,PHIAZ,PHIOZ,CMLAMA,KLAA,
               GOAQ,MLAA1,MAR,MEA,MLAA/NOUP/ERROR/
               LUSETS/NOMSET/NOOSET/NORC/NOQSET/NOTSET/
               EPSMALC/EPSMALU/MAXRATIO/EPSBIG/INRLM/EPSRC/
               ASING/PRPHIVZ/NOASET/NORC/NOQSET/NOTSET/
               NORSET/NOOSET/SEID/PEID/WTMASS/NOA/
               NLHEAT/NOSSET $
TYPE PARM,NDDL,I,N,PEID $
ALTER 26,26  $              REPLACE  EXISTING CALL SEMR2
   CALL SEMR2  MGG,GOAT,USET,GM/
      MAA,MFF,MOA1/ERROR/NOMSET/
      NOSSET/NOOSET/NORC/NOQSET/NOTSET/SEID/PEID $
$
$ MODIFY SEMR2 TO HAVE SEID AND PEID - TO PASS ON FOR MASS
$ CHECKS
$
COMPILE SEMR2 SOUIN=MSCSOU $
ALTER 1,1 $ REPLACE SUBDMAP STATEMENT
SUBDMAP SEMR2  MGG,GOAT,USET,GM/
        MAA,MFF,MOA1/ERROR/NOMSET/NOSSET/
```

```
         NOOSET/NORC/NOQSET/NOTSET/SEID/PEID $
TYPE PARM,NDDL,I,N,SEID,PEID $
ALTER 5  $           AFTER LAST "TYPE" STATEMENT
TYPE PARM,,I,N,MATTYP=0 $
MATTYP=4 $ CHECK MGG
CALL CHECK USET//MATTYP/SEID/PEID $
ALTER 7                 $      AFTER IF (  NOMSET......
MATTYP=5 $ CHECK MNN
CALL CHECK USET//MATTYP/SEID/PEID $
ALTER 17       $                AFTER ENDIF
MATTYP=6 $ CHECK MAA
CALL CHECK USET//MATTYP/SEID/PEID $
$
$ MODIFY SEDRCVR - ADD CALLS FOR KE AND EFW AND PASS PEID
$ FOR SECONDARY S.E.
$
COMPILE SEDRCVR SOUIN=MSCSOU $
ALTER 1,1 $ REPLACE SUBDMAP STATEMENT
SUBDMAP SEDRCVR  UGVS,QGS,BGPDTS,EQEXINS,CSTMS,CASEDR,MPTS,
                 DIT,ETT,OLB1,PJ1,EST,XYCDBDR,GEOM2S,GEOM3S,
                 POSTCDB,ECTS,GPLS,EPTS,SILS,INDTA,KELM,
                 KDICT,GPECT,VELEM,FORCE,XYCDB,PG,PCDBDR,
                 USET,SLT,UHVF,OLBM,PGD,DLT,
                 FRL,SPSEL,DYNAMICS,CRX,QGE/
                 PUGV,OUGV1,OPG1,OQG1,OEF1X,
                 OES1X,OSTR1,OGS1,OUGV2,OPG2,OQG2,
                 OEF2, OES2, OSTR2,OGS2,OES1M,OES1G,
                 OSTR1M,OSTR1G,EGPSTR,EGPSF,ONRGY1,OGPFB1/
                 GRDPNT/APP/APP1/NOCOMPS/CURVPLOT/PFILE/
                 NUMOUT1/NUMOUT2/BIGER1/BIGER2/NUMOUT/
                 BIGER/SRTOPT/LSTRN/SRTELTYP/CURV/OUTOPT/
                 OG/NINTPTS/S1M/S1G/S1AM/S1AG/DOPT/
                 TINY/NOELOF/NOELOP/TABS/SEID/PEID/S1/
                 CARDNO/PDRMSG/SCRSPEC/RSPECTRA/
                 RSPRINT/TABID/INREL/GPFDR/
                 NLHEAT/AERO/ICYCLIC/GEOMU/LOADU/POSTU/
                 DBCDIAG/DBCPROG/POST/CP/DBCOVWRT $
TYPE PARM,NDDL,I,N,PEID $
ALTER 109      $          AFTER OFP  OUGV1,OPG1,.......
CALL KEEFW UGVS/TEST99/SEID/PEID $
$
COMPILE CHECK $
SUBDMAP CHECK USETB//MATTYP/SEID/PEID $
$
$ SUBDMAP CHECK - CALC "RIGID-BODY" VECTORS & CHECK K AND M
$
$  MATTYP = FLAG FOR MATRIX TYPE AND SET
$          0 = CALC. "RIGID" BODY MATRICES
$          1 = KGG
$          2 = KNN
$          3 = KAA
$          4 = MGG
$          5 = MNN
$          6 = MAA
```

```
$
$   MODIFIED 5/1989 FOR VERSION 66 - SOL 103
$   MODIFIED 6/27/88 TO REMOVE PRINTOUT WITH UPSTREAM FOR
$                    TIPS
$
$ QUALIFIERS AND PARAMETERS
$
TYPE PARM,NDDL,I,N,PEID,MTEMP,MPC $
TYPE PARM,,I,N,SEID,PEID $
$  SET DEFAULTS
TYPE PARM,,I,Y,(CHKSTIF=-1) $ DEFAULT = NO STIFFNESS CHECKS
TYPE PARM,,I,Y,(CHKMASS=-1) $ - DEFAULT = NO MASS CHECKS
$
TYPE PARM,,I,N,(NOPAR=0)) $
TYPE PARM,NDDL,RS,Y,(WTMASS) $
TYPE PARM,,RS,N,(MW) $
TYPE PARM,NDDL,I,N,NOUP,ZUZR1,ZUZR2,ZUZR3 $
TYPE PARM,,CS,Y,(MASSWT) $
TYPE PARM,,I,N,MATTYP $
TYPE PARM,NDDL,CHAR8,N,APRCH,K2GG
$
TYPE DB,ZUZR02,ZUZR03 $ STORE RBG FOR FUTURE USE IN KEEFW
TYPE DB,EQEXINS,SLIST,EMAP,BGPDTS,CSTMS $
TYPE DB,KGG,KNN,KAA,MGG,MNN,MAA $
TYPE DB,GPLS,SILS $
TYPE DB,USET,KLAA,MAPS $
$
ZUZR1=SEID
IF(WTMASS=0.)WTMASS=1. $
MW=1./WTMASS $
MASSWT=CMPLX(MW,0.) $
$
  DBVIEW RBG=ZUZR02(WHERE ZUZR1=SEID)
  DBVIEW RBTG=ZUZR03(WHERE ZUZR1=SEID)
$
IF(MATTYP=0) THEN $
 $
 $ UPSTREAM Q-SET PARTITION VECTOR
 $
 IF(NOUP>0) THEN   $
  DBVIEW KLAAUP=KLAA(WHERE SEID=* AND WILDCARD=TRUE) $
  DBVIEW MAPSUP=MAPS(WHERE SEID=* AND WILDCARD=TRUE) $
  SEMA    EQEXINS,SLIST,EMAP,,KLAAUP,MAPSUP/KPP/SEID/
           LUSETS/QUAL='SEID'/0 $
  DIAGONAL  KPP/VGUQ/'COLUMN'/1. $
  $
  $ CREATE Q-SET PARTITION VECTOR & COMBINE W/UPSTREAM
  $  PARTITION
  $
  VEC USETB/ZUZR03/'G'/'COMP'/'Q'/ $
  ADD ZUZR03,VGUQ/ZUZR01/ $
  PARAML ZUZR01//'TRAILER'/5/V,N,NZWDS $
  PARAM //'SUB'/V,N,NOPAR/NZWDS/1 $
 ELSE $
```

```
 NOPAR=-1 $
 ENDIF $
 VECPLOT ,,BGPDTS,EQEXINS,CSTMS,,/RBTG1/GRDPNT//4 $
 COND NOPART,NOPAR $
 PARTN RBTG,ZUZR01,/RBX,,,/1 $
 MERGE RBX,,,,ZUZR01,/RBTG1/1 $
 LABEL NOPART $
 TRNSP RBTG1/RBG1/ $
 EQUIVX RBG1/ZUZR02/ALWAYS $
 EQUIVX RBTG1/ZUZR03/ALWAYS $
ENDIF $
$
IF (CHKSTIF>-1 AND MATTYP=1)THEN $
  $ CHECK KGG FOR CONSTRAINTS
  MPYAD KGG,RBG,/REACG/ $
  MPYAD RBTG,REACG,/CHKKGG/ $
  NORM  REACG/REACGNRM/ $
  MATGPR GPLS,USETB,SILS,REACGNRM//'H'/'G'//1.-2 $
  MATPRN CHKKGG // $
ENDIF $
$
$ CHECK KNN
$
IF (CHKSTIF>-1 AND MATTYP=2) THEN $
  $ CHECK KNN FOR CONSTRAINTS
  UPARTN USETB,RBG/RBN,,,/'G'/'N'/'M'/1 $
  TRNSP RBN/RBTN $
  MPYAD KNN,RBN,/REACN/ $
  MPYAD RBTN,REACN,/CHKKNN/ $
  NORM  REACN/REACNNRM/ $
  MATGPR GPLS,USET,SILS,REACNNRM//'H'/'N'//1.-2 $
  MATPRN CHKKNN // $
ENDIF $
$
IF (CHKSTIF>-1 AND MATTYP=3) THEN $
  UPARTN USET,RBG/RBA,,,/'G'/'A'/'O'/1 $
  $ CHECK KAA FOR CONSTRAINTS
  TRNSP RBA/RBTA $
  MPYAD KAA,RBA,/REACA/ $
  MPYAD RBTA,REACA,/CHKKAA/ $
  NORM  REACA/REACANRM/ $
  MATGPR GPLS,USET,SILS,REACANRM//'H'/'A'//1.-2 $
  MATPRN CHKKAA // $
  $ LABEL NOKAACHK $
ENDIF
$
$ AFTER MGG STORE
$
IF (CHKMASS>-1 AND MATTYP=4) THEN $
  MPYAD MGG RBG,/MGRB/ $
  MPYAD RBTG,MGRB,/MASS/ $
  ADD MASS,/WGHT/MASSWT $
  MATPRN WGHT// $
  PRTPARM //0/'MASSWT' $
```

```
  PRTPARM //0/'GRDPNT' $
ENDIF $
$
$ AFTER MAA STORE
$
IF(CHKMASS>-1 AND MATTYP=6) THEN $
  UPARTN USET,RBG/RBA,,,/'G'/'A'/'O'/1 $
  MPYAD MAA,RBA,/MARB $
  TRNSP RBA/RBTA/ $
  MPYAD RBTA,MARB,/MASSA/ $
  ADD MASSA,/WGHTA/MASSWT $
  MATPRN WGHTA// $
  PRTPARM //0/'MASSWT' $
  PRTPARM //0/'GRDPNT' $
ENDIF $
RETURN $
END $
$
COMPILE KEEFW $
SUBDMAP KEEFW UGVS1/TEST99/SEID/PEID $
$
$ SUBDMAP KEEFW - CALCULATE KE AND EFW IF REQUESTED
$
TYPE DB,EQEXINS,GPLS,SILS,USET,MGG,ZUZR02,MJJ,ZUZR01 $
TYPE PARM,NDDL,I,N,SEID $
TYPE PARM,,I,N,(LUSETD) $
TYPE PARM,,I,Y,(KEPRT=-1,EFWGT=-1,KEEFW=1) $
TYPE PARM,NDDL,I,N,NOUP,ZUZR1,ZUZR2,ZUZR3,PEID $
TYPE PARM,NDDL,RS,Y,(WTMASS) $
TYPE PARM,,RS,N,(MW) $
TYPE PARM,,CS,Y,(MASSWT) $
$
IF(WTMASS=0.)WTMASS=1. $
MW=1./WTMASS $
MASSWT=CMPLX(MW,0.) $
$
$ AFTER SDR1 - SYSTEM K.E.
$
IF (KEPRT<0 AND EFWGT<0)KEEFW=-1
 $
 IF (KEEFW>-1)THEN $ REQUEST MADE FOR KE OR EFW OR BOTH
  ZUZR1=SEID $
   IF (NOUP>0) THEN $
    MPYAD UGVS1,MGG,/PHITM/1//// $
     IF (EFWGT>-1)THEN $
       MPYAD PHITM,ZUZR02,/MER///// $
       TRNSP MER/MERT $
       TRNSP MER/MERTA $
       ADD MERT,MERTA/EFWWUP/MASSWT//1 $
       MATPRN EFWWUP// $
     ENDIF $
     IF (KEPRT>-1) THEN $
       TRNSP PHITM/PHITMT $
       ADD PHITMT,UGVS1/ENERGWUP///1 $
```

```
        MPYAD PHITM,UGVS1,/TOTEN1/ $
        DIAGONAL TOTEN1/TOTENWUP $
        MATPRN TOTENWUP// $
        MATGPR GPLS,USET,SILS,ENERGWUP//'H'/'G'//1.-2 $
$             1% FILTER ON ENERGY
      ENDIF
$
   ENDIF $
$
   MPYAD UGVS1,MJJ,/PHITMJ/1//// $
    IF(EFWGT>-1) THEN $
      MPYAD PHITMJ,ZUZR02,/MERJ///// $
      TRNSP MERJ/MERJT $
      TRNSP MERJ/MERJTA $
      ADD MERJT,MERJTA/EFWNOUP/MASSWT//1 $
      MATPRN EFWNOUP// $
    ENDIF $
$
    IF(KEPRT>-1) THEN $
      TRNSP PHITMJ/PHITMJT $
      ADD PHITMJT,UGVS1/ENERNOUP///1 $
      MPYAD PHITMJ,UGVS1,/TOTEN2/ $
      DIAGONAL TOTEN2/TOTENOUP $
      MATPRN TOTENOUP// $
      MATGPR GPLS,USET,SILS,ENERNOUP//'H'/'G'//1.-2 $
$         1% FILTER ON ENERGY
    ENDIF $
$
ENDIF $
RETURN $
END $
$
CEND
TITLE = CHECK EXTERNAL S.E. 10 MATRICES
SET 99 = 0
SEALL = 99
SPC = 1
METHOD = 1
DISP = ALL
BEGIN BULK
PARAM,CHKSTIF,1
PARAM,CHKMASS,1
PARAM,KEPRT,1
PARAM,EFWGT,1
$
$  PERFORM EIGENVALUE ANALYSIS
$
EIGR,1,MGIV,0.,100.,,10
$
$ ADD CSUPER CARD TO BRING EXTERNAL S.E. 10 IN
$
SPOINT,1001,THRU,1010
CSUPER,10,,5,1001,THRU,1010
$
```

```
$ ADD GRIDS TO CONNECT EXTERNAL SUPERELEMENT TO
$ MODEL OF RESIDUAL STRUCTURE ONLY'
$
GRID,1,,0.,0.,0.
=,*(1),=,*(10.),==
=(3)
GRID,11,,0.,1.,0.,,123456
CBAR,1,1,1,2,11
=,*(1),=,*(1),*(1),==
=(2)
PBAR,1,1,1.,10.,10.,10.
$
$ DENSITY REMOVED TO CHECK S.E. WEIGHT ALONE
$
PARAM,AUTOSPC,YES
MAT1,1,30.+6,,,
SPC1,1,123456,1,2,3,4,5
ENDDATA
```

CBAR5.DAT

```
ID CBAR, TEST PROBLEM FOR EXTERNAL SUPERELEMENTS
SOL 112
TIME 10
CEND
TITLE = MODAL TRANSIENT FOR NON-SUPERELEMENT MODEL
LABEL = SOLVE AS RESIDUAL STRUCTURE ONLY
SEALL = ALL
  SDAMP    = 100
  TSTEP    = 5
$
$  SET DEFAULT LOADSET FOR ALL S.E.
$
  LOADSET = 10
SPC = 1
SUBCASE 5
LABEL = RES STR - SOLVE PROBLEM FOR SYSTEM MODES
DLOAD = 10
METHOD = 1
DISP(PLOT) = ALL
OUTPUT(XYPLOT)
 XAXIS = YES
 YAXIS = YES
 XGRID = YES
 YGRID = YES
 XTITLE = TIME
 YTITLE =    Z  D I S P  O F  G R I D  5
XYPLOT DISP /5(T3)
BEGIN BULK
EIGR,1,MGIV,0.,100.,,10
GRID,1,,0.,0.,0.
=,*(1),=,*(10.),==
=(8)
GRID,11,,0.,1.,0.,,123456
CBAR,1,1,1,2,11
=,*(1),=,*(1),*(1),==
=(7)
PBAR,1,1,1.,10.,10.,10.
PARAM,AUTOSPC,YES
MAT1,1,30.+6,,,.283
SPC1,1,123456,1
TLOAD2,10,10,,,0.,5.,10.,,+T12
+T12         -1.
TSTEP          5      600     .001        5
DAREA,10,1,3,.0001
FORCE,12,8,,2.,0.,0.,-1.
LSEQ,10,10,12
TABDMP1,100,CRIT,,,,,,,+TDAMP
+TDAMP        0.     .01     200.      .01     ENDT
ENDDATA
```

CBAR6.DAT

```
ID CBAR, TEST PROBLEM FOR EXTERNAL SUPERELEMENTS
SOL 112
TIME 10
COMPILE SELR,SOUIN=MSCSOU, NOLIST, NOREF
ALTER 86
MATPRN PA// $
$OUTPUT4 PA,,,,//-1/12/-1 $ WRITE PA TO FILE FOR COMPARISON
CEND
TITLE = MODAL TRANSIENT FOR NON-SUPERELEMENT MODEL
LABEL = SOLVE AS RESIDUAL STRUCTURE ONLY
SEALL = ALL
  SDAMP   = 100
  TSTEP   = 5
$
$  SET DEFAULT LOADSET FOR ALL S.E.
$
  LOADSET = 10
$
SPC = 1
$
SUBCASE 5
SUPER = ALL
LABEL = RES STR - SOLVE PROBLEM FOR SYSTEM MODES
DLOAD = 10
METHOD = 1
DISP(PLOT) = ALL
$
$  PLOT RESULTS
$
OUTPUT(XYPLOT)
 XAXIS = YES
 YAXIS = YES
 XGRID = YES
 YGRID = YES
 XTITLE = TIME
 YTITLE =    Z  D I S P  O F  G R I D  5
XYPLOT DISP /5(T3)
BEGIN BULK
SPC1,1,123456,1
$ SID   DAREA   DELAY   TYPE      T1       T2       F    PHASE
TLOAD2,10,10,,,0.,5.,10.,,+T12
$            C         B
+T12        -1.
$            SID    N(1)   DT(1)   NO(1)
TSTEP          5     600    .001       5
$
FORCE,12,8,,2.,0.,0.,-1.
$
LSEQ,10,10,12
$
TABDMP1,100,CRIT,,,,,,,+TDAMP
```

```
+TDAMP          0.     .01    200.      .01    ENDT
EIGR,1,MGIV,0.,100.,,10
$
$ ADD MODAL COORDINATES FOR S.E. 10
$
SPOINT,1001,THRU,1010
SEQSET1,10,0,1001,THRU,1010
$
GRID,1,,0.,0.,0.
=,*(1),=,*(10.),==
=(8)
GRID,11,,0.,1.,0.,,123456
GRID,12,,0.,1.,0.,,123456
CBAR,1,1,1,2,11
=,*(1),=,*(1),*(1),==
=(2)
CBAR,5,1,5,6,12
=,*(1),=,*(1),*(1),==
=(3)
$
$ DEFINE SUPERELEMENT
$
SESET,10,6,THRU,10
SESET,10,12
PBAR,1,1,1.,10.,10.,10.
PARAM,AUTOSPC,YES
MAT1,1,30.+6,,.3,.283
SPC1,1,123456,1
ENDDATA
```

CBAR7.DAT

```
ASSIGN MASTER=CBAR2.MASTER
RESTART
$ CALCULATE LOADING MATRIX
TIME 10
SOL 100
COMPILE USERDMAP SOUIN=MSCSOU
ALTER 2
PUTSYS(1,125) $ ALLOW FOR LATER RESTART WITHOUT WARNING
TYPE PARM,NDDL,I,N,SEID,PEID,ZUZR1 $ QUALIFIERS
TYPE DB ZUZR03,PA $
$ ZUZR03 IS THE PHIXX MATRIX
ZUZR1=10 $ ZUZR1 IS THE QUALIFIER FOR ZUZR0* DATA BLOCKS
SEID=10 $ SET QUALIFIER FOR PA
PEID=10 $ SET QUALIFIER FOR PA
DMIIN     DMI,DMINDX/SCALE,,,,,,,,, $
MATPRN    ZUZR03,SCALE// $
MPYAD     ZUZR03,SCALE,/PA $
MATPRN    PA// $
DBDIR     // $
CEND
TITLE = CREATE LOADING PA FOR SUPERELEMENT 10
BEGIN BULK
$
$ USE DMI TO ENTER SCALING FOR PHIXX -
$
$ FOR THIS CASE IT IS A 3X1 MATRIX - LOAD = -2 POUNDS ON
$     GRID 8 DOF 3
$
DMI,SCALE,0,2,1,,,3,1
DMI,SCALE,1,1,-2.
ENDDATA
```

CBAR8.DAT

```
ASSIGN CBAR2=CBAR2.MASTER
DBLOC  DB=* LOGI=CBAR2 VERS=2
$ MODAL, TRANSIENT USING EXTERNAL SUPERELEMENT
SOL 112
TIME 10
$
$ ATTACH MATRICES FOR EXTERNAL SUPERELEMENT 10
$  GET SYSTEM MODES AND RUN MODAL TRANSIENT
$
CEND
TITLE = MODAL TRANSIENT FOR EXTERNAL SUPERELEMENT MODEL
LABEL = SOLVE FOR SYSTEM MODES AND THEN TRANSIENT
SET 99 = 0
SEALL = 99
LOADSET = 10
SDAMP   = 100
TSTEP   = 5
$
SPC = 1
$
SUBCASE 5
SET 99 = 0
SUPER = 99
LABEL = RES STR - SOLVE PROBLEM FOR SYSTEM MODES
DLOAD = 10
METHOD = 1
DISP(PLOT) = ALL
$
$  PLOT RESULTS
$
OUTPUT(XYPLOT)
 XAXIS = YES
 YAXIS = YES
 XGRID = YES
 YGRID = YES
 XTITLE = TIME
 YTITLE =    Z  D I S P  O F  G R I D  5
XYPLOT DISP /5(T3)
BEGIN BULK
$
$  PERFORM EIGENVALUE ANALYSIS
$
EIGR,1,MGIV,0.,100.,,10
$
$ ADD CSUPER CARD TO BRING EXTERNAL S.E. 10 IN
$
SPOINT,1001,THRU,1010
CSUPER,10,,5,1001,THRU,1010
$
$ ADD GRIDS TO CONNECT EXTERNAL SUPERELEMENT TO
$ MODEL OF RESIDUAL STRUCTURE ONLY'
$
```

```
GRID,1,,0.,0.,0.
=,*(1),=,*(10.),==
=(3)
GRID,11,,0.,1.,0.,,123456
CBAR,1,1,1,2,11
=,*(1),=,*(1),*(1),==
=(2)
PBAR,1,1,1.,10.,10.,10.
PARAM,AUTOSPC,YES
MAT1,1,30.+6,,,.283
SPC1,1,123456,1
$ SID    DAREA    DELAY    TYPE       T1        T2        F     PHASE
TLOAD2,10,10,,,0.,5.,10.,,+T12
$               C         B
+T12           -1.
$              SID     N(1)    DT(1)    NO(1)
TSTEP           5      600     .001         5
$
DAREA,10,2,3,.0001
$
TABDMP1,100,CRIT,,,,,,,+TDAMP
+TDAMP          0.     .01     200.      .01     ENDT
$
$ FAKE FORCE CARD AND LSEQ TO TELL NASTRAN TO USE
$    SUPERELEMENT LOADS FROM EXTERNAL SUPERELEMENT
$
FORCE,12,1,,0.,0.,0.,-1.
$
LSEQ,10,10,12
$
ENDDATA
```

CBAR9.DAT

```
$ RESTART FROM CBAR3 TO RUN MODAL TRANSIENT
$ GET EXTERNAL SUPERELEMENT DATA BASE
ASSIGN CBAR2=CBAR2.MASTER
DBLOC  DB=* LOGI=CBAR2 VERS=2
$ GET RESIDUAL STRUCTURE DATA BASE
RESTART
ASSIGN MASTER=CBAR3.MASTER
$
SOL 112
TIME 10
$
$ ATTACH MATRICES FOR EXTERNAL SUPERELEMENT 10
$ USE SYSTEM MODES TO RUN MODAL TRANSIENT (RESTART)
$
CEND
TITLE = MODAL TRANSIENT FOR SUPERELEMENT MODEL
SUBTITLE = RESTART TO RUN MODAL TRANSIENT
$
SUBCASE 5
   LABEL = RES STR - SOLVE PROBLEM FOR TRANSIENT SOLUTION
   LOADSET = 10
   SDAMP   = 100
   TSTEP   = 5
   SPC = 1
   DLOAD = 10
   METHOD = 1
   DISP(PLOT) = ALL
$
$  PLOT RESULTS
$
OUTPUT(XYPLOT)
 XAXIS = YES
 YAXIS = YES
 XGRID = YES
 YGRID = YES
 XTITLE = TIME
 YTITLE =    Z  D I S P  O F  G R I D  5
XYPLOT DISP /5(T3)
BEGIN BULK
TLOAD2,10,10,,,0.,5.,10.,,+T12
+T12            -1.
TSTEP             5      600     .001        5
DAREA,10,2,3,.0001
TABDMP1,100,CRIT,,,,,,,+TDAMP
+TDAMP           0.     .01    200.      .01    ENDT
$
$ FAKE FORCE CARD AND LSEQ TO TELL NASTRAN TO USE
$  SUPERELEMENT LOADS FROM EXTERNAL SUPERELEMENT
$
FORCE,12,1,,0.,0.,0.,-1.
LSEQ,10,10,12
ENDDATA
```

ML1.DAT

```
ASSIGN OUTPUT4=ML1.F12,UNIT=12,FORMATTED
$
$ SIMULATES GENERATION AND STORAGE OF MAA, KAA, AND PHIXX
$ BY A SUBCONTRACTOR. STORAGE MEDIA ARE AN OUTPUT4 FILE
$ AND A DATA BASE FILE.
$
SOL 103
DIAG 8
TIME 10
COMPILE PHASE1 SOUIN=MSCSOU NOREF NOLIST
ALTER 66 $ AFTER CALL SEGOA
TYPE DB ZUZR01,ZUZR02,ZUZR03 $
TYPE PARM NDDL I N ZUZR1 $
ZUZR1=SEID $
ADD       MLAA,MAA/ZUZR01 $
ADD       KLAA,KAA/ZUZR02 $
OUTPUT4   ZUZR01,ZUZR02,,,//0/12/-1 $
$
MATGEN    EQEXINS/INTEXT1/9/0/LUSETS $
UMERGE    USET,GOA,/GOG/'G'/'O'/'A' $
UPARTN    USET,GOG/GU2,,,/'G'/'U2'/'A'/1 $
UMERGE    USET,GU2,/GU2G/'G'/'U2'/'A' $
MPYAD     INTEXT1,GU2G,/PHIXT/1 $
TRNSP      PHIXT/PHIX1 $ G-SIZE MATRIX WITH TERMS FOR U2 SET
MATMOD    PHIX1,,,,,/U2NULL,/12//1 $ SEARCH FOR NULL COLUMNS
PARTN     PHIX1,U2NULL,/ZUZR03,,,/1 $
MATPRN    GU2G,U2NULL,ZUZR03// $
OUTPUT4   ZUZR03,,,,//0/12/-1 $
DBDIR     // $
CEND
TITLE = CREATE EXTERNAL SUPERELEMENT MODEL OF S.E. 10
SEALL = 10
LABEL = CMS OF SUPERELEMENT 10
   SUPER =10
   METHOD=1
   SPC = 1
   DISP = ALL
BEGIN BULK
PARAM,USETPRT,2
PARAM,GRDPNT,0
EIGR,1,MGIV,0.,100.,,10
$
$ ADD MODAL COORDINATES FOR S.E. 10
$
SPOINT,1001,THRU,1010
SEQSET1,10,0,1001,THRU,1010
$
$
$ ADD MODAL COORDINATES FOR S.E. 20
$
SPOINT,2001,THRU,2010
SEQSET1,20,0,2001,THRU,2010
```

```
$
$ DEFINE SEUSET U2 FOR INTERIOR POINTS WITH POTENTIAL
$ LOADINGS
$
DEFUSET,U2,U2
SEUSET,10,U2,8,345
GRID,1,,0.,0.,0.
=,*(1),=,*(10.),==
=(8)
GRID,11,,0.,1.,0.,,123456
GRID,12,,0.,1.,0.,,123456
CBAR,1,1,1,2,11
=,*(1),=,*(1),*(1),==
=(2)
CBAR,5,1,5,6,12
=,*(1),=,*(1),*(1),==
=(3)
$
$ DEFINE SUPERELEMENTS
$
SESET,10,6,THRU,10
SESET,10,12
$
SESET,20,2,3,4,5,11
$
PBAR,1,1,1.,10.,10.,10.
PARAM,AUTOSPC,YES
MAT1,1,30.+6,,.3,.283
SPC1,1,123456,1
ENDDATA
```

ML2.DAT

```
ASSIGN OUTPUT4=ML2.F12,UNIT=12,FORMATTED
$
$ SIMULATES GENERATION AND STORAGE OF MAA, KAA, AND PHIXX
$ BY A SUBCONTRACTOR. STORAGE MEDIA ARE AN OUTPUT4 FILE
$ AND A DATA BASE FILE.
$
SOL 103
DIAG 8
TIME 10
COMPILE PHASE1 SOUIN=MSCSOU NOREF NOLIST
ALTER 66 $ AFTER CALL SEGOA
TYPE DB ZUZR01,ZUZR02,ZUZR03 $
TYPE PARM NDDL I N ZUZR1 $
ZUZR1=SEID $
ADD       MLAA,MAA/ZUZR01 $
ADD       KLAA,KAA/ZUZR02 $
OUTPUT4   ZUZR01,ZUZR02,,,//0/12/-1 $
$
MATGEN    EQEXINS/INTEXT1/9/0/LUSETS $
UMERGE    USET,GOA,/GOG/'G'/'O'/'A' $
UPARTN    USET,GOG/GU2,,,/'G'/'U2'/'A'/1 $
UMERGE    USET,GU2,/GU2G/'G'/'U2'/'A' $
MPYAD     INTEXT1,GU2G,/PHIXT/1 $
TRNSP     PHIXT/PHIX1 $ G-SIZE MATRIX WITH TERMS FOR U2 SET
MATMOD    PHIX1,,,,,/U2NULL,/12//1 $ SEARCH FOR NULL COLUMNS
PARTN     PHIX1,U2NULL,/ZUZR03,,,/1 $
$
MATPRN    GU2G,U2NULL,ZUZR03// $
$
OUTPUT4   ZUZR03,,,,//0/12/-1 $
DBDIR     // $
CEND
TITLE = CREATE EXTERNAL SUPERELEMENT MODEL OF S.E. 20
SEALL = 20
LABEL = CMS OF SUPERELEMENTS 10 AND 20
   SPC = 1
   SUPER =20
   METHOD=1
   DISP = ALL
BEGIN BULK
PARAM,GRDPNT,0
EIGR,1,MGIV,0.,100.,,10
$
$ ADD MODAL COORDINATES FOR S.E. 20
$
SPOINT,2001,THRU,2010
SEQSET1,20,0,2001,THRU,2010
$
$ DEFINE SEUSET U2 FOR INTERIOR POINTS WITH POTENTIAL
LOADINGS
$
DEFUSET,U2,U2
```

```
SEUSET,10,U2,8,345
GRID,1,,0.,0.,0.
=,*(1),=,*(10.),==
=(8)
GRID,11,,0.,1.,0.,,123456
CBAR,1,1,1,2,11
=,*(1),=,*(1),*(1),==
=(2)
$
$ DEFINE SUPERELEMENT
$
SESET,20,2,3,4,11
$
PBAR,1,1,1.,10.,10.,10.
PARAM,AUTOSPC,YES
MAT1,1,30.+6,,.3,.283
SPC1,1,123456,1
ENDDATA
```

ML3.DAT

```
ASSIGN INPUTT4=ML1.F12,UNIT=11,FORMATTED
ASSIGN INPUTT4=ML2.F12,UNIT=12,FORMATTED
$ CREATE DATA BASE FOR USE AS EXTERNAL SUPERELEMENT
$ IN SPACE STATION RUN
ASSIGN MASTER=ML3.MASTER
ASSIGN DBALL=ML3.DBALL
ASSIGN USROBJ=ML3.OBJ,TEMP $ THROWAWAY
ASSIGN USRSOU=ML3.SOU,TEMP $ THROWAWAY
$
TIME 10
SOL 100
COMPILE USERDMAP SOUIN=MSCSOU
ALTER 2
PUTSYS(1,125) $ ALLOW FOR LATER RESTART WITHOUT WARNING
TYPE PARM,NDDL,I,N,SEID,PEID,ZUZR1 $ QUALIFIERS
TYPE DB MAA,KAA,ZUZR03 $
$ ZUZR03 IS THE PHIXX MATRIX
ZUZR1=10 $ ZUZR1 IS THE QUALIFIER FOR ZUZR0* DATA BLOCKS
SEID=10 $
PEID=10 $
INPUTT4    /MAA,KAA,ZUZR03,,/3/11/-1/-1 $ READ INPUT MATRICES
ZUZR1=20 $ ZUZR1 IS THE QUALIFIER FOR ZUZR0* DATA BLOCKS
SEID=20 $
PEID=20 $
INPUTT4    /MAA,KAA,,,/2/12/-1/-1 $ READ INPUT MATRICES
DBDIR      // $
CEND
TITLE = READ AND STORE MATRICES FOR CUPOLA IN DATA BASE
BEGIN BULK
ENDDATA
```

ML4.DAT

```
ASSIGN MASTER=ML3.MASTER
RESTART
$$ CUPOLA, CALCULATE LOADING MATRIX
TIME 10
SOL 100
COMPILE USERDMAP SOUIN=MSCSOU
ALTER 2
PUTSYS(1,125) $ ALLOW FOR NOKEEP AND LATER RESTART
TYPE PARM,NDDL,I,N,SEID,PEID,ZUZR1 $ QUALIFIERS
TYPE DB ZUZR03,PA $
$ ZUZR03 IS THE PHIXX MATRIX
ZUZR1=10 $ ZUZR1 IS THE QUALIFIER FOR ZUZR0* DATA BLOCKS
SEID=10 $ SET QUALIFIER FOR PA
PEID=10 $ SET QUALIFIER FOR PA
DMIIN     DMI,DMINDX/SCALE,,,,,,,,, $
MATPRN    ZUZR03,SCALE// $
MPYAD     ZUZR03,SCALE,/PA $
MATPRN    PA// $
DBDIR     // $
CEND
TITLE = CREATE LOADING PA FOR SUPERELEMENT 10
BEGIN BULK
$
$ USE DMI TO ENTER SCALING FOR PHIXX -
$
$ FOR THIS CASE IT IS A 3X1 MATRIX - LOAD = -2 POUNDS ON
$    GRID 8 DOF 3
$
DMI,SCALE,0,2,1,,,3,1
DMI,SCALE,1,1,-2.
ENDDATA
```

# ML5.DAT

```
ASSIGN ML3=ML3.MASTER
DBLOC  DB=* LOGI=ML3 VERS=2
$ CREATE RESIDUAL STRUCTURE DATA BASE
ASSIGN MASTER=ML5.MASTER
ASSIGN DBALL=ML5.DBALL
ASSIGN USROBJ=ML5.OBJ,TEMP $ THROWAWAY
ASSIGN USRSOU=ML5.SOU,TEMP $ THROWAWAY
$
$ ATTACH MATRICES FOR EXTERNAL SUPERELEMENTS
$  AND GET AND SAVE SYSTEM MODES
$
SOL 103
TIME 15
CEND
TITLE = COMBINE EXTERNAL S.E. 10 & 20 TO GET 30 AND SOLVE
$
$  TELL MSC/NASTRAN TO ASSEMBLE MATRICES
$    AND SOLVE RESIDUAL STRUCTURE
$
SET 99=0,30
SUPER=99
SPC = 1
METHOD = 1
DISP = ALL
BEGIN BULK
$
$  PERFORM EIGENVALUE ANALYSIS
$
EIGR,1,MGIV,0.,100.,,10
$
$ ADD SUPERELEMENT TREE TO DEFINE MULTI-LEVEL MODEL
$
DTI,SETREE,1,10,30,20,30,30,0
$
$ ADD CSUPER CARD TO BRING EXTERNAL S.E. 10 IN
$
SPOINT,1001,THRU,1010
CSUPER,10,,5,1001,THRU,1010
$
$ ADD CSUPER CARD TO BRING EXTERNAL S.E. 20 IN
$
SPOINT,2001,THRU,2010
CSUPER,20,,1,5,2001,THRU,2010
$
$ DEFINE S.E. 30
$
SESET,30,5
SEQSET1,30,0,3001,THRU,3010
SPOINT,3001,THRU,3010
$
$ ADD PLOTEL TO TELL NASTRAN THAT S.E 30 CONNECTS
$  GRIDS 5 & 1
```

```
$
PLOTEL,99,1,5
$
$ ADD GRIDS TO CONNECT EXTERNAL SUPERELEMENT TO
$
$ MODEL OF RESIDUAL STRUCTURE ONLY
$
GRID,1,,0.,0.,0.
GRID,5,,40.,0.,0.
PARAM,AUTOSPC,YES
SPC1,1,123456,1
ENDDATA
```

ML6.DAT

```
$ GET EXTERNAL SUPERELEMENT DATA BASE
ASSIGN ML3=ML3.MASTER
DBLOC  DB=* LOGI=ML3 VERS=2
$ GET RESIDUAL STRUCTURE DATA BASE
RESTART
ASSIGN MASTER=ML5.MASTER
$ MODAL TRANSIENT USING EXTERNAL SUPERELEMENT
SOL 112
TIME 10
$
$ RESTART FROM NORMAL MODES RUN TO RUN MODAL TRANSIENT -
$    EXTERNAL S.E. 10 AND 20 COMBINED INTO S.E. 30
$
CEND
TITLE = MODAL TRANSIENT FOR EXTERNAL SUPERELEMENT MODEL
LABEL = SOLVE USING MULTI-LEVEL EXTERNAL SE
SET 99=0,30
SUPER=99
LOADSET = 10
SDAMP   = 100
TSTEP   = 5
SPC = 1
LABEL = RES STR - SOLVE PROBLEM FOR SYSTEM MODES
DLOAD = 10
METHOD = 1
DISP(PLOT) = ALL
$
$  PLOT RESULTS
$
OUTPUT(XYPLOT)
SEPLOT 30
 XAXIS = YES
 YAXIS = YES
 XGRID = YES
 YGRID = YES
 XTITLE = TIME
 YTITLE =    Z  D I S P  O F  G R I D  5
XYPLOT DISP /5(T3)
BEGIN BULK
TLOAD2,10,10,,,0.,5.,10.,,+T12
+T12           -1.
TSTEP            5      600     .001        5
DAREA,10,1,3,.0001
TABDMP1,100,CRIT,,,,,,,+TDAMP
+TDAMP          0.     .01     200.      .01    ENDT
$
$ FAKE FORCE CARD AND LSEQ TO TELL NASTRAN
$  TO USE SUPERELEMENT LOADS FROM EXTERNAL SUPERELEMENT
$
FORCE,12,1,,0.,0.,0.,-1.
LSEQ,10,10,12
ENDDATA
```