AN EFFICIENT PROCEDURE FOR DATA RECOVERY OF A CRAIG-BAMPTON COMPONENT

Edward T. Tong
Craig C.J. Chang
Members of Technical Staff
Space Division
Rockwell International
Downey, California

1994 MSC/NASTRAN World Users' Conference
Orlando, Florida
June, 1994

Introduction

Dynamic analyses of large and complex space structures are generally performed using the modal approach. Component modal syntheses are used frequently so that various companies, each designing their own hardware, can supply the component models to the integrator for system dynamic analysis. Of the various techniques of component modal reduction, the Craig-Bampton procedure is most commonly used. The deliverable data usually consists of the component Craig-Bampton generalized mass and stiffness matrices, and the Craig-Bampton transformation matrix. Linear Transformation Matrices(LTMs) associated with the model generated by the component companies are used to facilitate data recovery. Sizes of these LTMs range from a few key load indicators which are of interest to the integrator to an extensive data recovery set that drives the component's design. Component modal synthesis and data recovery therefore form a tandum in dynamic analysis of large space structures. The two widely used procedures to compute the LTMs are the mode displacement approach and the mode acceleration approach. Although the mode acceleration approach is generally perceived to be an improvement over the mode displacement approach, there are considerations when choosing a data recovery method.

This paper will formulate a simplified procedure to construct the Craig-Bampton LTMs using either the mode acceleration approach or the mode displacement approach. The procedure will be formulated for the general case of statically indeterminant structures. Further simplification can be made for statically determinant Craig-Bampton components. A general purpose DMAP routine of MSC/NASTRAN version 67 encompassing the tandum has been implemented. Sample problems to demonstrate the procedure with the DMAP are included. Although the DMAP is written for a Craig-Bampton component, it can be extended to a general modally reduced model.

A disadvantage of the mode acceleration recovery is the costly computation to construct the LTMs when an applied load at the interior set is present. One of the recent applications is the plume implingement on the solar arrays during Shuttle docking with the Space Station. This paper will discuss a cost saving technique and suggest an approach which reduces cost by utilizing a 'mixture' of the two approaches.

Background

Consider the basic equations of motion :

$$\begin{bmatrix} M_{aa} & M_{ao} \\ M_{oa} & M_{oo} \end{bmatrix} \begin{bmatrix} \ddot{x}_a \\ \ddot{x}_o \end{bmatrix} + \begin{bmatrix} C_{aa} & C_{ao} \\ C_{oa} & C_{oo} \end{bmatrix} \begin{bmatrix} \dot{x}_a \\ \dot{x}_o \end{bmatrix} + \begin{bmatrix} K_{aa} & K_{ao} \\ K_{oa} & K_{oo} \end{bmatrix} \begin{bmatrix} x_a \\ x_o \end{bmatrix} = \begin{bmatrix} F_a \\ F_o \end{bmatrix} \qquad (1)$$

where 'a' denotes the boundary set and 'o' represents the interior set.

The Craig-Bampton transformation (Reference 1) is :

$$\begin{bmatrix} x_a \\ x_o \end{bmatrix} = [\Phi_{cb}] \begin{bmatrix} x_a \\ q \end{bmatrix} \quad ; \quad [\Phi_{cb}] = \begin{bmatrix} I & 0 \\ \Phi_c & \Phi_n \end{bmatrix} \qquad (2)$$

where q are the modal coordinates,

Φ_c are the constraint modes, $\Phi_c = - K_{oo}^{-1} K_{oa}$

Φ_n are the normal modes with boundary constrained , and

$\{x_a , q\}$ are the generalized coordinates.

Upon transformation, eqn. (1) becomes

$$\begin{bmatrix} \overline{M}_{aa} & M_{aq} \\ M_{qa} & M_{qq} \end{bmatrix} \begin{bmatrix} \ddot{x}_a \\ \ddot{q} \end{bmatrix} + \begin{bmatrix} \overline{C}_{aa} & C_{aq} \\ C_{qa} & C_{qq} \end{bmatrix} \begin{bmatrix} \dot{x}_a \\ \dot{q} \end{bmatrix} + \begin{bmatrix} \overline{K}_{aa} & K_{aq} \\ K_{qa} & K_{qq} \end{bmatrix} \begin{bmatrix} x_a \\ q \end{bmatrix} = [\Phi_{cb}]^t \begin{bmatrix} F_a \\ F_o \end{bmatrix} \qquad (3)$$

where

$\overline{M}_{aa} = M_{aa} + M_{ao}\, \Phi_c + \Phi_c^t\, M_{oa} + \Phi_c^t\, M_{oo}\, \Phi_c,$

$M_{qa} = M_{aq}^t = \Phi_n^t\, M_{oo}\, \Phi_c + \Phi_n^t\, M_{oa},$

$M_{qq} = I,$

$\overline{K}_{aa} = K_{aa} + K_{ao}\, \Phi_c,$

$K_{qa} = K_{aq} = 0,$

$K_{qq} = \omega^2,$

$C_{qq} = 2\zeta\omega,$

$\overline{C}_{aa} = C_{aq} = C_{qa} = 0$ (assuming only modal damping).

Two most commonly used data recovery LTM methods are the mode displacement and the mode acceleration methods. For the mode displacement approach of a modally reduced structure,

$$[\text{Load}] = [\Gamma_2]\begin{bmatrix} x_a \\ q \end{bmatrix} \tag{4}$$

where Γ_2 is the mode displacement LTM.

And for the mode acceleration approach, assuming negligible damping,

$$[\text{Load}] = [T_1]\begin{bmatrix} \ddot{x}_a \\ \ddot{q} \end{bmatrix} + [T_2]\begin{bmatrix} x_a \\ q \end{bmatrix} + [T_3]\begin{bmatrix} F_a \\ F_o \end{bmatrix} \tag{5}$$

where T_1 and T_2 are the LTMs for acceleration and displacement respectively and T_3 is the transformation matrix of the applied load. The 'Load' recovery is generically used here, it can be either displacement, interface loads, element forces or stresses. Every recovery will have a different set of Γ's and T's.

The methods of constucting the LTMs using the mode acceleration approach usually begins by recovering the $\ddot{x}_o$ from [$\ddot{x}_a$ $\ddot{q}$], then solving for x_o. These methods have been widely used (Reference 2 & 3) and together with the Craig-Bampton reduction will serve as a background for the rest of this paper.

Methodology

For a general structure, pre multiplying eqn. (1) by $[I \quad \Phi_c^t]$, assuming negligible damping,

$$\begin{bmatrix} I \\ \Phi_c \end{bmatrix}^t \begin{bmatrix} M_{aa} & M_{ao} \\ M_{oa} & M_{oo} \end{bmatrix} \begin{bmatrix} \ddot{x}_a \\ \ddot{x}_o \end{bmatrix} + \begin{bmatrix} I \\ \Phi_c \end{bmatrix}^t \begin{bmatrix} K_{aa} & K_{ao} \\ K_{oa} & K_{oo} \end{bmatrix} \begin{bmatrix} x_a \\ x_o \end{bmatrix} = \begin{bmatrix} I \\ \Phi_c \end{bmatrix}^t \left(\begin{bmatrix} F_a \\ F_o \end{bmatrix} + \begin{bmatrix} F_i \\ 0 \end{bmatrix} \right) \qquad (6)$$

where F_i is the interface loads.

Substituting the Craig-Bampton transformation (2) into (6), the mass term becomes

$$\begin{bmatrix} I \\ \Phi_c \end{bmatrix}^t \begin{bmatrix} M_{aa} & M_{ao} \\ M_{oa} & M_{oo} \end{bmatrix} \begin{bmatrix} I & 0 \\ \Phi_c & \Phi_n \end{bmatrix} = \begin{bmatrix} \overline{M}_{aa} & M_{aq} \end{bmatrix}$$

and the stiffness term is

$$\begin{bmatrix} I \\ \Phi_c \end{bmatrix}^t \begin{bmatrix} K_{aa} & K_{ao} \\ K_{oa} & K_{oo} \end{bmatrix} \begin{bmatrix} I & 0 \\ \Phi_c & \Phi_n \end{bmatrix} = \begin{bmatrix} \overline{K}_{aa} & 0 \end{bmatrix}$$

Upon substitution, eqn. (6) becomes

$$F_i = \begin{bmatrix} \overline{M}_{aa} & M_{aq} \end{bmatrix} \begin{bmatrix} \ddot{x}_a \\ \ddot{q} \end{bmatrix} + \begin{bmatrix} \overline{K}_{aa} & 0 \end{bmatrix} \begin{bmatrix} x_a \\ q \end{bmatrix} - \begin{bmatrix} I \\ \Phi_c \end{bmatrix}^t \begin{bmatrix} F_a \\ F_o \end{bmatrix} \qquad (7)$$

Comparing eqn. (7) with eqn. (5), it is clear that for the interface loads the LTMs are,

$$T_1 = \begin{bmatrix} \overline{M}_{aa} & M_{aq} \end{bmatrix} \; ; \; T_2 = \begin{bmatrix} \overline{K}_{aa} & 0 \end{bmatrix} \; ; \; T_3 = \begin{bmatrix} I \\ \Phi_c \end{bmatrix}^t$$

This shows that the interface loads LTMs T_1 and T_2 can be obtained by simply extracting terms from the generalized mass and stiffness matrices. T_3 is the transpose of the constraint modes. Therefore no computation is required.

The null term in T_2 points out that the interface loads are not dependent on the q set, and therefore does not relate to x_o. Hence the interface loads LTMs are exactly identical regardless of whether mode acceleration or mode displacement method is used.

The displacement LTMs and the element forces and/or stresses LTMs can be recovered by rewriting the second line of (1),

$$\begin{bmatrix} K_{oa} & K_{oo} \end{bmatrix} \begin{bmatrix} x_a \\ x_o \end{bmatrix} = - \begin{bmatrix} M_{oa} & M_{oo} \end{bmatrix} \begin{bmatrix} \ddot{x}_a \\ \ddot{x}_o \end{bmatrix} + F_o$$

rearranging, substituting (2)

$$x_o = - \begin{bmatrix} K_{oo} \end{bmatrix}^{-1} \begin{bmatrix} M_{oa} & M_{oo} \end{bmatrix} \begin{bmatrix} \Phi_{cb} \end{bmatrix} \begin{bmatrix} \ddot{x}_a \\ \ddot{q} \end{bmatrix} + \Phi_c \, x_a + F_o$$

and the displacement LTMs can be obtained by,

$$\begin{bmatrix} x_a \\ x_o \end{bmatrix} = \begin{bmatrix} D_1 \end{bmatrix} \begin{bmatrix} \ddot{x}_a \\ \ddot{q} \end{bmatrix} + D_2 \, x_a + D_3 \, F_o$$

where

$$D_1 = \begin{bmatrix} 0 \\ - K_{oo}^{-1} \begin{bmatrix} M_{oa} & M_{oo} \end{bmatrix} \begin{bmatrix} \Phi_{cb} \end{bmatrix} \end{bmatrix} \quad ;$$

$$D_2 = \begin{bmatrix} I \\ \Phi_c \end{bmatrix} \quad ; \quad D_3 = \begin{bmatrix} 0 \\ -K_{oo}^{-1} \end{bmatrix}$$

SDR2 subroutine is used to recover the element forces and stresses LTMs Λ_1, Λ_2 , Λ_3 respectively from D_1, D_2 , D_3. The element forces and/or stress can be written as,

$$\begin{bmatrix} \text{element} \\ \text{forces} \end{bmatrix} = \begin{bmatrix} \Lambda_1 \end{bmatrix} \begin{bmatrix} \ddot{x}_a \\ \ddot{q} \end{bmatrix} + \begin{bmatrix} \Lambda_2 \end{bmatrix} x_a + \begin{bmatrix} \Lambda_3 \end{bmatrix} F_o \qquad (8)$$

The recovery of Λ_3 for the complete interior set of a large space structure will be very costly. One way to reduce cost is to define only those dofs that will actually have an applied load. For example, the plume loads are all applied in the translational dofs. Let Θ be a subset of F_o of unit magnitude, then D_3 becomes,

$$D_3 = \begin{bmatrix} 0 \\ - K_{oo}^{-1} \, \Theta \end{bmatrix}$$

The cost of the SDR2 recovery of Λ_3 will therefore be reduced accordingly.

For a statically determinant structure, computations of the Craig-Bampton LTMs can be substantially simplified. The constraint modes of (2) are now the rigid body modes. This implies that the second term of (6) is

$$\begin{bmatrix} I \\ \Phi_c \end{bmatrix}^t \begin{bmatrix} K_{aa} & K_{ao} \\ K_{oa} & K_{oo} \end{bmatrix} = 0$$

hence,

$$F_i = \begin{bmatrix} \overline{M}_{aa} & M_{aq} \end{bmatrix} \begin{bmatrix} \ddot{x}_a \\ \ddot{q} \end{bmatrix} + \begin{bmatrix} I \\ \Phi_c \end{bmatrix}^t \begin{bmatrix} F_a \\ F_o \end{bmatrix}$$

and if there is no applied load at the component, then the interface loads are simply,

$$F_i = \begin{bmatrix} \overline{M}_{aa} & M_{aq} \end{bmatrix} \begin{bmatrix} \ddot{x}_a \\ \ddot{q} \end{bmatrix}$$

As for the displacements and element forces and/or stresses recoveries, statically determinant implies that D_2 is the rigid body modes and therefore does not contribute to the computation of element forces and/or element stresses. The displacement LTMs remain the same but eliminating the second term of (8), element forces LTMs become,

$$\begin{bmatrix} \text{element} \\ \text{forces} \end{bmatrix} = [\Lambda_1] \begin{bmatrix} \ddot{x}_a \\ \ddot{q} \end{bmatrix} + [\Lambda_3] F_o$$

again, if there are no applied loads at the interior set, then

$$\begin{bmatrix} \text{element} \\ \text{forces} \end{bmatrix} = [\Lambda_1] \begin{bmatrix} \ddot{x}_a \\ \ddot{q} \end{bmatrix}$$

All the above mentioned procedure can be extended to a general modally reduced component. The transformation of any modal synthesis can be written in the general form,

$$\begin{bmatrix} x_a \\ x_o \end{bmatrix} = [\Phi] \begin{bmatrix} x_a \\ q \end{bmatrix} \quad ; \quad [\Phi] = \begin{bmatrix} I & 0 \\ \Phi_c & \Psi \end{bmatrix} \tag{9}$$

where Ψ is a hybrid set of modes. The Craig-Bampton reduction is just a special case of (9) when the 'c' set is null and Ψ are the cantilevered normal modes. For a general hybrid modally reduced component described above, all terms in (3) will remain the same with Ψ replacing Φ_n. Therefore all the LTMs are valid for the general modally reduced component with Ψ replacing Φ_n.

Implementation

A DMAP routine of MSC/NASTRAN version 67 using the normal modes solution 103 has been implemented. This DMAP routine will generate for the users, using mode acceleration or mode displacement approach, statically determinant or statically indeterminant, the following :

(i) Craig-Bampton generalized mass and stiffness matrices and the Craig-Bampton transformation matrix,
(ii) interface loads LTMs T_1, T_2 and T_3 as needed,
(iii) displacement LTMs D_1, D_2 and D_3 as needed,
(iv) element forces LTMs Λ_1, Λ_2 and Λ_3 as needed.

The DMAP procedure is listed in Appendix 1. Users are to put the interface dofs in the r set, i.e. SUPORT cards. Users are to define the following parameters in the BULK DATA:

(a)	PARAM, LTM, NO	---	no LTMs required
	MODEDISP	---	mode displacement method
	MODEACCE	---	mode acceleration method
(b)	PARAM, STATDET,0	---	default value, statically indeterminant
	1	---	statically determinant shortcut
(c)	PARAM, APPLOAD,0	---	default value, no apply load
	1	---	apply load LTM requested.

If the applied load LTM is requested, the user has an option to define a subset of the interior where there are applied loads. Users are to define the applied load dofs on the USET cards with the user set name 'U1'.

The output will be written on unit 21 in OUTPUT4 format.

A simple beam example has been used to illustrate the procedure. The example is listed in Appendix 2.

Discussions

One of the disadvantage of the mode acceleration approach is when the structural model is very large, and there is an applied load at the interior set. The computation required for D_3 and subsequent SDR2 for Λ_3 are very costly. This is because the mode acceleration approach is equivalent to computing a static solution at every time step.

In most dynamic analysis, especially in the preliminary design stage, it is only necessary to determine the maximum and/or the minimum loads. It is therefore quite wasteful to compute loads using the mode acceleration for every time step.

The 'mixed' approach will utilize the standard mode displacement approach to screen for the maximum and/or minimum load cases according to the users needs and mode acceleration correction will only be performed on those selected time steps that the maximum or minimum occured. This cuts down the costly mode acceleration correction for every time step and still be able to provide the maximum and minimum load accurately. The procedure is simply to restart and code the desired time steps for mode acceleration correction by OTIME.

Along with the efficiency of the mode acceleration method there are misuses. To use the mode acceleration approach effectively, the component model must have a set of resonably accurate modes. General guidelines are to include modes to 50-100% above the maximum frequency of interest in the forcing functions (Reference 4). One example of the misuse is the following : - a component model with modes up to 5 Hertz; combined with other component models with better modal fidelity of 30 Hertz; generate system modes up to 15 Hertz: go through dynamic analysis and expects the mode acceleration method will recover loads accurately. In this case, the mode acceleration approach will only correct the static contribution and assume no dynamic contribution above 5 Hertz. Since the static mode is generally a good approximation for the fundamental mode, the results will be erroneous if some of the recovered elements respond significantly to higher modes with frequencies above 5 Hertz. Furthermore, if there is any coupling between components then the interface forcing functions will not be accurate either.

Damping has been assumed negligible for the mode acceleration approach. When damping is significant, the mode displacement approach could be prefered over the mode acceleration approach. Reference 5 suggested mode displacement approach with residual flexibility, this will take care of the damping and the modal truncation problems. However, it will not be practical to compute residual flexibility if the applied load covers most of the component.

References

1. Craig, R.R. and Bampton, M.C.C., "Coupling of Substructures for Dynamic Analysis" AIAA journal, Vol. 6, No. 7, July 1968.

2. Rose, Ted., "Component Modal Synthesis using External Superelements" MSC/NASTRAN publication, May 1992.

3. Flanigan C.C., "Efficient and Accurate Procedures for Calculating Data Recovery Matrices" MSC/NASTRAN World Users' Conference, March 1989.

4. Flanigan C.C., "Accurate and Efficient Mode Acceleration Data Recovery for Superelement models" MSC/NASTRAN World Users' Conference, March 1988.

5. Rose, Ted, "Using Residual Vectors in MSC/NASTRAN Dynamic Analysis to Improve Accuracy" MSC/NASTRAN World Users' Conference, 1991.

APPENDIX 1

```
$
$ CRAIG-BAMPTON REDUCTION -- USER TO PUT  FIXED BOUNDARY DOFS IN RSET
$                                   (SUPORT CARDS)
$ OUTPUT     - GENERALIZED MASS & STIFFNESS
$            - CRAIG-BAMPTON TRANSFORMATION MATRIX
$
$ TO REQUEST LTM -- USER TO DEFINE THE FOLLOWING PARAMETERS
$   (i)   PARAM, LTM, NO           ---   DEFAULT, NO LTM REQUEST
$                     MODEDISP     ---   MODE DISPLACEMENT METHOD
$                     MODEACCE     ---   MODE ACCELERATION METHOD
$   (ii)  PARAM, STATDET, 0        ---   DEFAULT VALUE, STATIC INDETERMINANT
$                         1        ---   STATIC DETERMINANT
$   (ii)  PARAM, APPLOAD, 0        ---   DEFAULT VALUE, NO APPLIED LOAD
$                         1        ---   APPLIED LOAD LTM REQUESTED
$
$  OUTPUT    - STATIC INDETERMINANT
$                  - INTERFACE LOADS LTM T1, T2, T3
$                  - DISPLACEMENT LTM D1, D2, D3
$                  - ELEMENT FORCES LTM L1, L2, L3
$            - STATIC DETERMINANT
$                  - INTERFACE LOADS LTM T1,  T3
$                  - DISPLACEMENT LTM D1, D2, D3
$                  - ELEMENT FORCES LTM L1, L3
$      IF THERE IS NO APPLIED LOAD IN THE LSET, THEN THERE IS NO T3, D3, L3
$
$      TO SAVE COST ON APPLIED LOAD LTM, USER TO INPUT USER DEFINE SET U1
$      IN THE BULK DATA TO REFLECT ALL DOFS WHICH ARE SUBJECT TO APPLIED
$      LOADS, U1 MUST BE A SUBSET OF LSET
$
COMPILE MODERS, SOUIN=MSCSOU, NOLIST, NOREF $
ALTER 6
TYPE  PARM,,I,N,NOASET,NOBSET,NOMSET,NOGSET,NOOSET,NORC,NOUSET $
TYPE  PARM,,I,N,NOQSET,NOSSET,NOTSET,NOVSET,NOA,NOSET,TOTCOL $
TYPE  PARM,,I,Y,STATDET,APPLOAD   $
TYPE  PARM,,CHAR8,Y,LTM='NO'      $
CALL PMLUSET  USET//S,NOASET/S,NOBSET/S,NOCSET/S,NOGSET/S,NOLSET/
        S,NOOSET/S,NOQSET/S,NORSET/S,NOSSET/S,NOTSET/
        S,NOVSET/S,NOA/S,NOSET/S,NORC/S,NOMSET/S,NOUSET $
UPARTN USET,MMAA/MRR,MLR,MRL,MLL/'A'/'R'/'L'  $
UPARTN USET,MKAA/KRR,KLR,KRL,KLL/'A'/'R'/'L'  $
ALTER 17,17
ALTER 80,80
READ  KLL,MLL,,,EED,,CASES,/LAMA,PHIX,MI,OEIGS/READAPP/
        S,N,NEIGV/NSKIP $
ALTER 82,82
REIGL  KLL,MLL,DYNAMICS,CASES,,,,/LAMA,PHIX,MI,EIGVMAT,/READAPP/
        S,N,NEIGV/NSKIP $
ALTER 84
```

```
TOTCOL = NORSET + NEIGV        $
MATGEN,/LAM1/6/TOTCOL/0/TOTCOL  $
MATGEN,/LAM2/6/NORSET/0/NORSET  $
MATGEN,/LAM3/6/NOLSET/0/NOLSET  $
CALL DBSTORE LAM1,LAM2,LAM3,,//0/1/'   '/0 $
MATGEN,/MERGCB/6/TOTCOL/NORSET/NEIGV $
MATGEN,/IDENT/1/NORSET             $
UMERGE USET,IDENT,DM/PHIC/'A'/'R'/'L' $
UMERGE USET,,PHIX/PHIY/'A'/'R'/'L'   $
MERGE  PHIC,,PHIY,,MERGCB,/PHICB/1    $
MPYAD  MLL,DM,MLR/M1    $
MPYAD  DM,M1,MRR/M2/1    $
MPYAD  MRL,DM,M2/MBAR    $
MPYAD  PHIX,M1,/MQA/1    $
TRNSP  MQA/MAQ          $
LAMX,  ,LAMA/LMAT/-1     $
MATMOD LMAT,,,,,/MT,/1/4 $
MATMOD MT,,,,,/MQQ,/28  $
MERGE  MBAR,MQA,MAQ,MQQ,MERGCB,/MGEN/-1/2/6  $
MATMOD LMAT,,,,,/KQQ,/28 $
MPYAD  KRL,DM,KRR/KBAR  $
MERGE  KBAR,,,KQQ,MERGCB,/KGEN/-1/2/6  $
OUTPUT4 MGEN,KGEN,PHICB,,//-1/21   $
MATPRN MGEN,KGEN,PHICB,,// $
IF (LTM = 'NO') THEN       $
MESSAGE //'*****     NO LTM REQUEST      *****'       $
EXIT  $
ENDIF  $
$
$  EXTRACT INTERFACE LOADS LTM
$  OUTPUT - LTM T1, T2, T3
$
MERGE  MBAR,,MAQ,,MERGCB,/T1/1 $
IF (STATDET = 0) THEN       $
MERGE  KBAR,,,,MERGCB,/T2/1 $
ENDIF              $
IF (APPLOAD = 1) THEN  $
TRNSP  PHIC/T3       $
ENDIF            $
OUTPUT4 T1,T2,T3,,//0/21    $
$
$  GENERATE DISPLACEMENTS LTM
$  OUTPUT - LTM D1, D2, D3
$
IF (LTM = 'MODEACCE') THEN      $
MESSAGE //'*****    MODE ACCELERATION LTM REQUEST      *****'      $
MPYAD  MLL,PHIX,/M3           $
MERGE  M1,,M3,,MERGCB,/TEMP1/1  $
SOLVE  KLL,TEMP1/DD1/1/-1       $
UMERGE USET,,DD1/D1/'A'/'R'/'L' $
EQUIVX PHIC/D2/ALWAYS  $
ELSE IF (LTM = 'MODEDISP') THEN $
MESSAGE //'*****    MODE DISPLACEMENT LTM REQUEST      *****'      $
EQUIVX PHICB/D2/ALWAYS $
```

```
ENDIF          $
IF (APPLOAD = 1) THEN  $
MESSAGE //'*****    APPLIED LOAD LTM REQUEST     *****'      $
PARAML USET//'USET'/////'U1'/S,N,NOU1 $
IF (NOU1 > 1) THEN    $
MESSAGE //'***** SHORTCUT APPLIED LOAD LTM REQUEST  *****'       $
MATGEN,/DUM2/1/NOU1   $
UMERGE USET,DUM2,/P1/'L'/'U1'/'C'/1  $
SOLVE  KLL,P1/DD3/1/-1 $
ELSE           $
SOLVE  KLL,/DD3/3     $
ENDIF          $
UMERGE USET,,DD3/D3/'A'/'R'/'L' $
ENDIF            $
OUTPUT4 D1,D2,D3,,//0/21   $
CALL DBSTORE D1,D2,D3,,//0/1/'     '/0 $
$
$  GENERATE ELEMENT FORCES LTM
$  OUTPUT - LTM L1, L2, L3
$
COMPILE SEDISP, SOUIN=MSCSOU, NOLIST, NOREF $
ALTER 37
TYPE   PARM,,I,N,SUCCESS        $
TYPE   PARM,,I,Y,STATDET,APPLOAD  $
TYPE   PARM,,CHAR8,Y,LTM        $
CALL DBFETCH /D1,D2,D3,,/0/1/0/0/S,SUCCESS $
IF (LTM = 'MODEACCE') THEN      $
SDR1  USET,,D1,,,GOA,GM,,KFS,KSS,/U1,,Q1/1/APP1/NOQG  $
IF (STATDET = 0) THEN      $
SDR1  USET,,D2,,,GOA,GM,,KFS,KSS,/U2,,Q2/1/APP1/NOQG  $
ENDIF  $
IF (APPLOAD = 1) THEN       $
SDR1  USET,,D3,,,GOA,GM,,KFS,KSS,/U3,,Q3/1/APP1/NOQG  $
ENDIF  $
ELSE IF (LTM = 'MODEDISP') THEN       $
SDR1  USET,,D2,,,GOA,GM,,KFS,KSS,/U2,,Q2/1/APP1/NOQG  $
ENDIF  $
CALL DBSTORE U1,U2,U3,,//0/1/'     '/0 $
RETURN
COMPILE SEDRCVR, SOUIN=MSCSOU, NOLIST, NOREF $
ALTER 43
TYPE   PARM,,I,N,SUCCESS,NOLSET,NORSET $
TYPE   PARM,,I,Y,STATDET,APPLOAD      $
TYPE   PARM,,CHAR8,Y,LTM              $
CALL DBFETCH /U1,U2,U3,,/0/1/0/0/S,SUCCESS $
CALL DBFETCH /LAM1,LAM2,LAM3,,/0/1/0/0/S,SUCCESS $
MATGEN, /PVLAM/6/3/2/1          $
TRNSP  LAM2/LAM2T             $
MERGE, ,LAM2T,,,,PVLAM/LAMA2T/1   $
LAMX   LAMA2T,/LAMA2           $
IF (LTM = 'MODEACCE') THEN      $
TRNSP  LAM1/LAM1T             $
MERGE, ,LAM1T,,,,PVLAM/LAMA1T/1   $
LAMX   LAMA1T,/LAMA1           $
```

```
SDR2   CASEDR,CSTMS,MPTS,DIT,EQEXINS,,ETT,LAMA1,BGPDP,,,U1,EST,XYCDBDR/
         ,,,LES1,LEF1,/APP1/S,N,NOSORT2/NOCOMPS//ACOUT/PREFDB $
DRMS1, ,,,LEF1/,,,,,,TELF1,L1/   $
IF (STATDET = 0) THEN      $
SDR2   CASEDR,CSTMS,MPTS,DIT,EQEXINS,,ETT,LAMA2,BGPDP,,,U2,EST,XYCDBDR/
         ,,,LES2,LEF2,/APP1/S,N,NOSORT2/NOCOMPS//ACOUT/PREFDB $
DRMS1, ,,,LEF2/,,,,,,TELF2,L2/   $
ENDIF $
IF (APPLOAD = 1) THEN      $
TRNSP  LAM3/LAM3T              $
MERGE, ,LAM3T,,,,PVLAM/LAMA3T/1   $
LAMX   LAMA3T,/LAMA3            $
SDR2   CASEDR,CSTMS,MPTS,DIT,EQEXINS,,ETT,LAMA3,BGPDP,,,U3,EST,XYCDBDR/
         ,,,LES3,LEF3,/APP1/S,N,NOSORT2/NOCOMPS//ACOUT/PREFDB $
DRMS1, ,,,LEF3/,,,,,,TELF3,L3/   $
ENDIF $
ELSE IF (LTM = 'MODEDISP') THEN      $
SDR2   CASEDR,CSTMS,MPTS,DIT,EQEXINS,,ETT,LAMA2,BGPDP,,,U2,EST,XYCDBDR/
         ,,,LES2,LEF2,/APP1/S,N,NOSORT2/NOCOMPS//ACOUT/PREFDB $
DRMS1, ,,,LEF2/,,,,,,TELF2,L2/   $
ENDIF $
OUTPUT4 L1,L2,L3,,//0/21  $
TABPT  TELF1,TELF2,TELF3,,// $
EXIT  $
```

APPENDIX 2

```
ID  CB,LTM
APP DISP
SOL 103
TIME 500
DIAG  8
INCLUDE DMAP
CEND
TITLE= BEAM EXAMPLE -- STATICALLY INDETERMINANT LTM
$ INPUT DMAP FROM APPENDIX 1
$ OUTPUT WRITTEN ON UNIT 21
ECHO = NONE
METHOD = 100
ELFORCE=ALL
DISP=ALL
BEGIN BULK
PARAM,USETPRT,1
PARAM,STATDET,0
PARAM,APPLOAD,1
PARAM,LTM,MODEACCE
DEFUSET,U1,U1
USET1,U1,123456,105
GRID   100          0.     0.     0.
GRID   101          10.    0.     0.
GRID   102          20.    0.     0.
GRID   103          30.    0.     0.
GRID   104          40.    0.     0.
GRID   105          50.    0.     0.
GRID   106          60.    0.     0.
GRID   107          70.    0.     0.
GRID   108          80.    0.     0.
GRID   109          90.    0.     0.
GRID   110          100.   0.     0.
GRID   200          0.     0.    -10.0          123456
CBAR   1     10     100    101    200
CBAR   2     10     101    102    200
CBAR   3     10     102    103    200
CBAR   4     10     103    104    200
CBAR   5     10     104    105    200
CBAR   6     10     105    106    200
CBAR   7     10     106    107    200
CBAR   8     10     107    108    200
CBAR   9     10     108    109    200
CBAR   10    10     109    110    200
PBAR   10    1      0.5    1.     1.     1.4
MAT1   1     1.0+7         .3     1.
EIGRL,100,,100.
SUPORT,100,123456
SUPORT,110,123456
ENDDATA
```