USING SUPERELEMENTS FOR RESPONSE SPECTRUM AND OTHER HANDY ALTERS

Ted L. Rose, P.E., S.E.*

March, 1989

Abstract

Several DMAP ALTERS are presented to allow the user to use Superelements in MSC/NASTRAN to obtain additional information from their analysis. These ALTERS include using Superelement data recovery in Response Spectrum analysis, performing Response Spectrum analysis without using a 'SUPORT', performing Superelement level buckling analysis, and using Image Superelements in a Global-Local force method analysis.

INTRODUCTION

The purpose of this paper is to share several DMAP ALTERS developed during the past year. Being a dedicated Superelement user, I recommend them to every engineer I come into contact with, both as efficient tools and for the extra information a user can get out of an analysis by using them (as evidenced by references 1 and 2 and the four ALTERS presented here).

Each of the four ALTERS is used to modify an existing solution sequence to get additional information from an analysis. As best I can tell, the response spectrum ALTERS(1 and 2) precisely match the results from a conventional, residual only, analysis. The third ALTER, which performs

*Staff Scientist, MacNeal-Schwendler Corp.,Los Angeles, Ca.

approximate Superelement buckling, is intended to allow the user to analyze areas of a model for buckling using the displacements obtained from the system-level analysis. The final ALTER, Image Superelements in the Global-Local force method, is a modification of the ALTER presented in the September, 1987 application note (see ref. 2) which allows calculation of the boundary forces on an image Superelement for use in further (local) analysis.

NOTE: THE ALTERS IN THIS PAPER ARE SET UP FOR VERSION 65

1. *Using Superelements in Response Spectrum*

As currently implemented in Sol 63, data recovery for a Response Spectrum analysis can only be performed on the residual structure. Superelements can be used in creating the model, but can not have data recovery performed using the results of the Response Spectrum analysis.

When the ALTER in appendix A is inserted in the data recovery loop for Response Spectrum analysis, it allows full data recovery to be performed on the upstream Superelements, as well as the residual structure. There are no special parameters required for implemention, only case control requests for the desired output are required.

2. *Performing Response Spectrum Analysis without a SUPORT*

The current implementation of Response Spectrum in MSC/NASTRAN requires that a SUPORT entry be included for the excitation degrees of freedom and that a large mass be placed on those dof to simulate a fixed boundary. The ALTER in appendix B allows Response Spectrum using fixed-base modes obtained without a SUPORT entry. This allows the user to perform a Response Spectrum analsis on a restart from a standard normal modes analysis in Sol 63 (Note that it is not necessary that the run be a restart).

The ALTER uses a reference point, n, identified by the parameter entry:

PARAM,GROUND,n

This causes the ALTER to calculate geometric rigid-body modes about grid point n. It is best to set n to 0, which causes the basic origin to be used for a reference. Any other value will cause the selected grid point to be used, which can cause problems at the Superelement level, where the Grid point must belong to the Superelement in order to be used. One way to use a single Grid point as a reference for all Superelements is to select a grid point which belongs to the residual structure and make it external to all Superelements by using the CSUPEXT entry.

Note that the 'rigid-body' modes don't include the effects of any constraints on the model, since they are based purely on the geometry. If the value, n, is set to 0, the rigid-body modes are calculated about the basic origin. If the parameter is set to -1 or is not included, the ALTER has no effect on the analysis and a standard Response Spectrum analysis will be performed, using the SUPORT.

In addition, a partitioning vector, DIR, must be entered by the use of DMI. In appendix B, there is a BULK DATA section with a sample of the entries required to implement this ALTER. In this case, the DMI vector DIR is entered to direct the shock loading in the 1 and 2 directions of the basic coordinates, since PARAM,GROUND is 0 (This is identical to having a SUPORT entry referencing dof 1 and 2 of a grid point located at the basic origin with all of the constraints to ground and a large mass rigidly attached to it). All of the other BULK DATA entries would be identical to those used in a standard Response Spectrum run. The advantage now is that the SUPORT entry is not required.

3. Buckling Analysis for Superelements

This ALTER is different from the previous ones, in that an approximation is introduced into the analysis. The ALTER allows a buckling analysis to be performed for any superelements in Sol 65, instead of on the residual structure only.

The ALTER in appendix C is used in Sol 65 to allow buckling analysis on selected superelements using the deformations resulting from a Sol 61 run.

In a standard Sol 65 run, the differential stiffness for the Superelements is calculated based on the deformations resulting from the Sol 61 analysis of the structure, then a buckling analysis is performed on the residual structure using the accumulated effects of the differential stiffness from the Superelements.

This ALTER uses the parameter FIXEDB in much the same manner as the other Superelement solutions. In this case, if FIXEDB is -1, then a buckling analysis is performed on the selected Superelement using the differential stiffness matrix at that point and the boundaıy constraints defined by the SECSET(free) and SEBSET(fixed) (if desired, the "fixed-boundary" deformations resulting from using 'PARAM,FIXEDB,-1' in Sol 61 can be used). For collector Superelements, the analysis includes the effects of the upstream Superelements in a way similar to that done for the residual structure.

Note that there is an approximation, since the attachment to the rest of the structure is not included. However, the deformations resulting from that attachment are included.

The advantages of this ALTER are easy to see. In a normal Sol 65 run, the buckling solution is performed only at the residual level, potentially losing the local buckling which may occur in the individual (and collector) Superelements. By using this ALTER in Sol 65, the user is allowed to obtain approximate Superelement-level buckling solutions. These approximate solutions should be better than those obtained by running the areas individually, since the effects of the overall structural deformation are included in the differential stiffness calculations.

To use the ALTER, first run Sol 61 to obtain the static deformations of the model. Then perform a restart in Sol 65 similar to that done to perform buckling analysis. However, use PARAM,FIXEDB,-1 to tell the ALTERS that the run is to perform buckling analysis on the individual Superelements as requested in the CASE CONTROL section.

4. Improved GLOBAL-LOCAL Force Method - Image Superelements

A number of users are using Image Superelements to reduce the costs of their analysis. The original Global-Local Application Note (September, 1987) did not allow the force method to be used for Image Superelements. The ALTER in appendix D allows the Force Method to be applied to Image Superelements. Then, a detailed model may be used to allow further data recovery. In this way, data recovery may be obtained for an image for post-processing using any of the commercially available packages.

Perhaps a brief explanation of how the force method works in Global-Local analysis is appropriate. It works along the same lines as cutting a free-body diagram of a section of a structure. When one cuts a free-body diagram, all of the forces at the cut boundaries have to be included for any further analysis of the local area of interest. The force method does just that, it calculates the forces exerted on the boundaries of the selected Superelement by the rest of the structure. By using these values, an independent analysis may be performed on a selected area of the structure simply by modeling it as a Superelement. That is, a more detailed analysis of the area of interest may be performed using the calculated boundary forces on a refined mesh model of the same area without having to re-solve the entire model.

One obvious advantage of this method occurs on large projects, where many groups are involved, each with their own area of interest. By using this method, each group can be given boundary forces which they can, in turn, apply to their own model to either perform data recovery or obtain additional information by modifying the mesh.

The ALTER replaces the one shown on page 21 of the Application Note and works in the same manner. All the user needs to do is set PARAM,SELOCAL to the Superelement number(Primary or Image). This results in a DMIG being created in the PUNCH file which has the external forces on the Superelement's boundary. These forces can then be used in a separate run of the Superelement only (possibly a refined mesh version) to allow further data recovery. Note that the applied loads on the new model must be identical to the ones on the original. This is because we are, in essence, creating a free-body diagram of the desired area of the model.

REFERENCES

1. Rose, T. L., *Using Superelements to Identify the Dynamic Properties of a Structure*, paper presented at the 1988 MSC/NASTRAN Users Conference.

2. *Applications of Superelements in Global-Local Analysis*, September, 1987 Application Note, MSC/NASTRAN Application Manual.

Appendix A - ALTER to allow Superelements in Response Spectrum

```
$
$ ALTER TO ALLOW S.E. DATA RECOVERY IN RESPONSE SPECTRUM
$   TO BE USED IN VERSION 65, SOL 63
$
ALTER 1049
DBSTORE UHVR,,,,//SOLID/0/DBSET2
JUMP    RESSTR $
LABEL   SEDRTR $
SEDR    ULV,EMAP,CASECC,PCDB,DRLIST,XYCDB,SLT,,/ULVS,CASEDR,
        PCDBDR,XYCDBDR/APP/MAPS/S,N,LPFLG/S,N,SEID/S,N,PEID/0/
        S,N,NOUP/0/SOLID/S,N,NOSDR1/S,N,NOMAT/DBSET4/S,N,JPLOT/
        S,N,NOXYCDB $
$
PARAM   //ADD/RESID/-1/SEID $
PARAM   //NOT/ACON/RESID $
PARAM   //GT/NP/DMAPNO/3 $
COND    LBLNOOP,NP $
DBFETCH /GM,USET,KFS,KSS,GOAT/MODEL/PEID//DBSET3 $
DBFETCH /GOAQ,,,,/SOLID/PEID//DBSET3 $
PARAML  USET//USET////NP//C,N,Q/V,N,NOQSET/C,N,O/V,N,NOOSET/
        C,N,A/V,N,NOASET/C,N,S/V,N,NOSSET/C,N,M/V,N,NOMSET/
        C,N,T/V,N,NOTSET/C,N,R/V,N,NORSET/C,N,G/V,N,NOGSET $
COND    LBLNOOP,NP $
DBFETCH /PSS,PJ,POS,YS,QR/SOLID/SEID//DBSET3 $
PARAM   //NOT/NP/FIXEDB $
COND    COUPDR1,NP $
DBFETCH /CMLAMA,PHIAZ,PHIOZ,,/SOLID/SEID//DBSET3 $
EQUIV   CMLAMA,OLB/ALWAYS/PHIAZ,ULVS/ALWAYS/PHIOZ,UOOVS/
        ALWAYS $
PARAML  CMLAMA//PRESENCE////NP $
COND    LBLNOOP,NP $
```

```
JUMP    LNTPHICC $
DBFETCH /UOOVS,,,,/SOLID/SEID/DBSET3 $
LABEL   COUPDR1 $
COND    NOSDR1T,NOSDR1 $
EQUIV   GOAT,GOA/NOQSET $
COND    LNTPHICC,NOQSET $
EQUIV   GOAQ,GOA/NOTSET $
COND    LNTPHICC,NOTSET $
PURGE   GOA/NOOSET $
COND    LNTPHICC,NOOSET $
ADD     GOAT,GOAQ/GOA $
LABEL   LNTPHICC $
PURGE   PGD,PSSD/ACON $
COND    NOPGD1,ACON $
DBFETCH /PGD,PSSD,,,/SOLID/0//DBSET3 $
LABEL   NOPGD1 $
MODACC  CASEDR,OLB,ULVS,PGD,PSSD,/OLB1,ULVS1,PJ1,PS1,/APP1 $
DBSTORE OLB1,PJ1,ULVS1//SOLID/SEID/DBSET2 $
LABEL   NOSDR1T $
COND    LBLNOOP,NOMAT $
$
DBFETCH /GEOM2S,EQEXINS,SILS,BGPDTS,EST/MODEL/SEID/DBSET3 $
DBFETCH /ECTS,GPECT,GPLS,CSTMS,/MODEL/SEID/DBSET3 $
DBFETCH /GEOM3S,ETT,,,/SOLID/SEID/DBSET3 $
DBFETCH /UHVR,PG,,,/SOLID/0/DBSET2 $
DBFETCH /ULVS1,,,,/SOLID/SEID//DBSET3 $
$
$  ADD GKAM TO GET PHIDH
$
DPD     DYNAMICS,GPLS,SILS,USET,SLT,PG/GPLD,SILD,USETD,TFPOOL,
        DLT,PSDL,FRL,NLFT,TRL,EED,EQDYN/-1/S,N,LUSETD/0/0/0/0/
        0/DYNSTAT/NEWDYN/123/S,N,NOUE $
GKAM    USETD,ULVS1,,OLB1,DIT,,,,CASEDR/
        MHH,BHH,KHH,PHIDH/NOUE/
        C,Y,LMODES=0/C,Y,LFREQ=0.0/C,Y,HFREQ=1.E+30/-1/
        -1/-1/S,N NONCUP/S,N,FMODE $
```

```
EQUIV   PHIDH,ULVS1H/NOUE $
COND    NOUEDRMT,NOUE $
UPARTN  USETD,PHIDH/ULVS1H,,,/D/A/E/1 $
LABEL   NOUEDRMT $
SDR1    USETD,,ULVS1H,,,GOA,GM,,KFS,,/PHIGH,,QGH/1/REIGS $
SDR2    CASEDR,CSTMS,MPT,DIT,EQEXINS,,,,BGPDTS,LAMA,QGS,
        PHIGH,EST,XYCDBDR/,IQG1,IPHIG1,IES1,IEF1,/APP1/
        S,N,NP $
$
MATPRN  UHVR,,,,// $
DBSTORE IQG1,IPHIG1,IES1,IEF1,//SOLID/SEID/DBSET2/POST $
DDRMM   CASEDR,UHVR,,IPHIG1,IQG1,IES1,IEF1,XYCDBDR/
        OUPV1,OQP1,DOES1,DOEF1,/OPTION $
JUMP    TED2 $
LABEL   RESSTR $
ALTER 1051
LABEL   TED2 $
ALTER 1052
LABEL   LBLNOOP $
COND    ENDSEDTR,LPFLG $
REPT    SEDRTR,1000  $
LABEL   ENDSEDTR $
$
$ END OF ALTER TO ALLOW S.E. DATA RECOVERY
$       IN RESPONSE SPECTRUM
$
ENDALTER
```

Appendix B - ALTER to perform Response Spectrum without a SUPORT

```
$
$  ALTER FOR RESPONSE SPECTRUM USING GEOMETRIC RIGID BODY MODES
$   USED IN VERSION 65, SOL 63
$
$ 1. REMOVE SUPORT ENTRY AND LARGE MASS - CONSTRAIN STRUCTURE
$       AT GROUND
$ 2. ADD PARAM,GROUND,n - TO BULK DATA, WHERE n IS THE GRID ID
$       TO BE USED TO CALCULATE GEOMETRIC RIGID BODY VECTORS
$       (IF 0 IS USED, THEN THEY ARE CALCULATED ABOUT THE BASIC
$       ORIGIN
$ 3. ADD DMI MATRIX 'DIR' WHICH IS USED FOR PARTITIONING
$       SAMPLE FORMAT :
$       DMI,DIR,0,2,1,,,6,1
$       DMI,DIR,1,1,1.,2,1.
$    THIS TELLS NASTRAN THAT THE SPECTRA ARE APPLIED IN THE
$       1 AND 2 DIRECTIONS
$
ALTER 5 $ SET DEFAULTS FOR PARAMETERS
$
$ SET GROUND >-1 TO DO R.S. W/O SUPORT
$
PARAM  //NOP/V,Y,GROUND=-1 $
ALTER 334 $ JUST BEFORE SEMA
$
$ UPSTREAM Q-SET PARTITION VECTOR
$
SEMA    EQEXINS,SLIST,EMAP,/KPP/KLAA/MAPS/SOLID/
        LUSETS/SEID/V,Y,DBSET4 $
DIAGONAL KPP/VGUQ/COLUMN/1. $
$
ALTER 342 $ AFTER GP4
$
```

```
$ CREATE Q-SET PARTITION VECTOR \& COMBINE
$    W/UPSTREAM PARTITION
$
VEC      USETB/VGLQ/G/COMP/Q/ $
ADD      VGLQ,VGUQ/VGPQ/ $
DBSTORE VGPQ,VGLQ//MODEL/SEID/DBSET2 $
PARAML   VGPQ//TRAILER/5/V,N,NZWDS $
PARAM    //SUB/V,N,NOPAR/NZWDS/1 $
VECPLOT ,,BGPDTS,EQEXINS,CSTMS,,/RBTG/GROUND//4 $
COND     NOPART,NOPAR $
PARTN    RBTG,VGPQ,/RBX,,,/1 $
MERGE    RBX,,,,VGPQ,/RBTG/1 $
LABEL    NOPART $
TRNSP    RBTG/RBG/ $
DBSTORE RBG//SOLID/SEID/DBSET2 $
$
$
ALTER 1036 $ AFTER ADD BDIAG1,MDIAG......
COND     NORMRS,GROUND $ USE SUPORT IF GROUND<0
DBFETCH /LUSETD,EQEXINS,GPLS,SILS,USET/SOLID/SEID/DBSET3 $
DBFETCH /MGG,RBG,MJJ,VGPQ,/SOLID/SEID/DBSET3 $
MPYAD    UGVS,MGG,/PHITM/1//// $
MPYAD    PHITM,RBG,/PSI///// $
TRNSP    PSI/PSIT $
MATPRN   PSIT// $
JUMP     TED3 $
LABEL    NORMRS $
ALTER 1039
LABEL    TED3 $
ENDALTER
$
$ EVERYTHING FROM THIS POINT ON IS FOR INFORMATION ONLY
$
.
.   EXECUTIVE CONTROL
.
```

```
CEND
.
.   CASE CONTROL
.
BEGIN BULK
.
.   BULK DATA
$
$  ****EXAMPLE****
$
$  USE GRID POINT 7 AS THE REFERENCE GRID POINT
$
PARAM,GROUND,7
$
$ SAMPLE DMI, DIR, TO POINT SPECTRA IN 1 AND 2 DIRECTIONS
$
DMI,DIR,0,2,1,,,6,1
DMI,DIR,1,1,1.,2,1.
$
.
.
ENDDATA
```

Appendix C - ALTER to allow buckling analysis on Superelements

```
$
$  SOL 65 ALTER TO ALLOW BUCKLING ANALYSIS ON SUPERELEMENTS
$
$   CREATED FOR V65 - 4/88 - TLR
ALTER 263 $
$ CHECK IF FIXEDB=-1
COND      SEBUCKL,FIXEDB $
JUMP      NOSEBUKL $ SKIP S.E. BUCKLING IF PARAM FIXEDB<>-1
LABEL     SEBUCKL $
PARAM     //STSR/0/-82 $
DBFETCH   /GPLS,SILS,USET,DM,EQEXINS/MODEL/SEID//DBSET3 $
DBFETCH   /KAA,GOAT,GM,,/MODEL/SEID//DBSET3 $
DBFETCH   /BGPDTS,CSTMS,,,/MODEL/SEID//DBSET3 $
DBFETCH   /MR,MAA,BAA,K4AA,DIT/SOLID/SEID//DBSET3 $
DBFETCH   /GOAQ,KLAA,MLAA,MATPOOL,/SOLID/SEID//DBSET3 $
DBFETCH   /SLT,PG,DYNAMICS,,/SOLID/SEID//DBSET3 $
PARAML    XYCDBS//PRES////V,N,NOXYCDB $
PARAML    BAA//PRES////V,N,NOBGG $
PARAML    USET//USET////NP//C,N,Q/V,N,NOQSET/C,N,R/V,N,NORSET/
          C,N,A/V,N,NOASET/ C,N,G/V,N,NOGSET/C,N,T/V,N,NOTSET $
SETVAL    //ERRNO/4410 $
COND      RFERRM,NP $
PARAM     //EQ/V,N,NOA/NOGSET/NOASET $
EQUIV     GOAT,GOA/NOQSET/KAA,MKAA/NOQSET $
COND      D11NOQST,NOQSET $
EQUIV     GOAQ,GOA/NOTSET/KLAA,MKAA/NOTSET $
COND      D11NOQST,NOTSET $
ADD       GOAT,GOAQ/GOA $
ADD       KLAA,KAA/MKAA $
LABEL     D11NOQST $
ADD       MLAA,MAA/MMAA $
PARAML    MMAA//PRES////V,N,NOMGG $
```

```
DPD       DYNAMICS ,GPLS,SILS,USET,SLT,PG/GPLD,SILD,USETD,
          TFPOOL,DLT,PSDL,FRL,NLFT,TRL,EED,EQDYN/-1/S,N,LUSETD/
          0/0/0/0/0/V,N,DYNSTAT=0/V,Y,NEWDYN=1/123/S,N,NOUE $
DBSTORE   PSDL,FRL,DLT//SOLID/SEID/DBSET2 $
SETVAL    ////////////V,N,READAPP/BUCK $
DBFETCH   /KDAA,,,,/SOLID/SEID//DBSET3 $
ADD       KDAA,/MMAA/(-1.,0.0) $
COND      TOSAVEM1,NOMETH $
SETVAL    //V,N,NOARED/-1 $
MATMOD    CASES,DYNAMICS,,,,/,/23/S,N,NP/S,N,LANCZOS $
PARAM     //EQ/V,N,INVPOW/NP/1 $
PARAM     //EQ/NP/NP/0 $
PARAM     //ADD/NP/2/NP $
MATMOD    MMAA,MKAA,,,,/VAXW1,MATAA/12/S,N,NOARED/NP $
COND      NOARED11,NOARED $
EQUIV     VAXW1,VAXW/NORSET $
COND      NOARED11,NORSET $
PARTN     VAXW1,,VACOMPR/VLQXW,,,/1 $
MERGE     VLQXW,,,,,VACOMPR/VAXW/1 $
PARAML    VAXW//NULL////NOARED $
LABEL     NOARED11 $
EQUIV     MKAA,KXX/NOARED/MMAA,MXX/NOARED/DMLQ,DMX/NOARED/
          USET,VXCOMPR/NOARED $
COND      LBLNORD,NOARED $
PRTPARM   //4415/DMAP $
MATGPR    GPLS,USET,SILS,VAXW//H/A $
COND      RFERR,ASING $
PARTN     MMAA,VAXW,/MXX,,,/-1 $
PARTN     MKAA,VAXW,/KXXBAR,KWX,,KWW1/-1 $
EQUIV     KXXBAR,KXX/INVPOW$
COND      KINV21,INVPOW $
PARAML    KWX//NULL////V,N,NOAOMIT $
EQUIV     KXXBAR,KXX/NOAOMIT/ $
COND      KINV21,NOAOMIT $
DECOMP    KWW1/LWW1,/1////////58 $
FBS       LWW1,,KWX/GWX1/1/-1/0/0 $
```

```
MPYAD     GWX1,KWX,KXXBAR/KXX/1////6 $
LABEL     KINV21 $
COND      LBLNORD,NORSET $
PARTN     VACOMPR,,VAXW/VXCOMPR,,, / 1 $
PARTN     DMLQ,,VLQXW/DMX,,,/1 $
LABEL     LBLNORD $
SETVAL    //V,N,NEIGV/-1 $
COND      LREIG1,LANCZOS $
READ      KXX,MXX,MR,DMX,EED,VXCOMPR,CASES/LAMA,PHIX,MI,OEIGS/
          V,N,READAPP=MODES/S,N,NEIGV $
JUMP      LREAD1 $
LABEL     LREIG1 $
REIGL     KXX,MXX,DYNAMICS,CASES/LAMA,PHIX,MI,EIGVMAT/
          V,N,READAPP/S,N,NEIGV $
LABEL     LREAD1 $
OFP       LAMA,OEIGS// $
SETVAL    //V,N,ERRNO/4405 $
COND      RFERRM,NEIGV $
EQUIV     PHIX,PHIA/NOARED $
COND      LBNOEXP,NOARED $
COND      KINV31,INVPOW $
MPYAD     GWX1,PHIX,/PHIW $
LABEL     KINV31 $
MERGE     PHIX,PHIW,,,,VAXW/PHIA/0 $
LABEL     LBNOEXP $
DBSTORE   PHIA,LAMA//SOLID/0/DBSET2 $
LABEL     TOSAVEM1 $
DBFETCH   /PHIA,LAMA,,,/V,Y,SOLID/0//DBSET3 $
COND      LBLSKPH1,NOMETH $
VDR       CASES,FQEXINS,USET,PHIA,LAMA,,/OPHIA,/REIG/DIRECT/0/
          S,N,NP/1/1 $
COND      LBLSKPH1,NP $
OFP       OPHIA//S,N,CARDNO $
LABEL     LBLSKPH1 $
JUMP      LSKIP11 $
PRTPARM   ///V,N,APP=REIG $
```

```
PARAM    //NOP/V,N,NSORT1=1 $
LABEL    LSKIP11 $
LABEL    NOSEBUKL $  SKIP TO THIS POINT IF NO S.E. BUCKLING
ENDALTER
```

Appendix D - Modified Global-Local Force Method ALTER

```
$
$  ALTER FOR GLOBAL-LOCAL ANALYSIS USING FORCE METHOD - CREATE
$    BOUNDARY FORCE VECTOR FOR LOCAL S.E. AND THEN CREATE DMIG
$    ENTRIES IN PUNCH FILE TO BE USED IN ANALYSIS OF LOCAL
$    MODEL IN SEPARATE RUN
$  FOR VERSION 65, SOL 61
$
$  MODIFIED  TO PERFORM IMAGE S.E.
$
$ ***USES PARAMETER SELOCAL - SET TO I.D. OF S.E. IN GLOBAL
$       MODEL WHICH GETS REPLACED BY LOCAL MODEL***
$
$ ALTER STATEMENT NUMBERS ARE FOR VERSION 65 - NUMBERS FOR
$   VERSION 64 ARE ALSO GIVEN
$
ALTER 216 $ AFTER GP1 MODULE
$
$ CREATE VECTOR INTEXT TO GO FROM INTERNAL TO EXTERNAL SORT
$
MATGEN    EQEXINS/INTEXT/9/0/LUSETS $
$ STORE INTEXT FOR FUTURE USE
DBSTORE INTEXT//MODEL/SEID/DBSET1 $
ALTER 586  $ AFTER SEDR MODULE
$ALTER 584 $ VERSION 64
$
$ CALCULATE BOUNDARY FORCES AND CREATE DMIG FOR SELOCAL
$
$ IF SELOCAL NOT SET BY USER, SET = -1 = DON'T CREATE DMIG
$
PARAM   //NOP/V,Y,SELOCAL=-1 $
COND    LOCAL1,SELOCAL $  SKIP IF SELOCAL NOT USED IN THIS RUN
PARAM    //NE/V,N,SKIPIT/SELOCAL/SEID $ CHECK IF S.E = SELOCAL
```

```
COND      LOCAL1,SKIPIT $ SKIP IF SEID NE SELOCAL
DBFETCH    /KAA,USET,INTEXT,,/MODEL/PEID/DBSET3 $ GET INFO
DBFETCH    /PL,EQEXINS,,,/SOLID/SEID/DBSET3 $ GET ASET LOADS
MPYAD      KAA,ULVS,PL/BFORCEL///-1 $ CALC BOUNDARY FORCES
UMERGE     USET,BFORCEL,/BFORCE/G/L/O $ EXPAND TO GLOBAL SIZE
MPYAD      INTEXT,BFORCE,/BFEXT/1 $ CHANGE TO EXTERNAL SORT
$
$   PUNCH DMIG
$
MATMOD     BFEXT,EQEXINS,,,,/MATPOOL,/16/1//////////BF $
LABEL   LOCAL1 $ END OF BOUNDARY FORCE DMIG ALTER
ENDALTER
```