USE OF MSC/NASTRAN GENERAL ELEMENTS IN COMPLEX STATIC PROBLEMS

Mitchell L. Greenberg

Spar Aerospace Limited
Spar Space Systems
9445 Airport Road
Brampton, Ontario, L6S 4J3

ABSTRACT

This paper shall explore uses of MSC/NASTRAN general (GENEL) elements. Basic stiffness matrix concepts as pertaining to GENEL formulation will be discussed, and techniques to reduce large linear static structures to a single GENEL connected at the boundary GRIDs shall be presented. The methods herein provide an alternative to and supplement the capabilities of SuperElements for manipulating and modelling sub-components.

1.0 INTRODUCTION

General elements have a wide variety of applications. They allow the user to arbitrarily connect any number of Degrees of Freedom. Although GENEL's are not simple, a fair understanding of the structural matrices is beneficial, they are a powerful tool. Their basic use is to allow a complete sub-structure to be represented by a single element which connects the sub-structure's boundary grids. They may be used to duplicate structural test stiffness results of actual components, or to incorporate model results from external solvers. This paper however will focus on reducing large NASTRAN models to a single GENEL element.

2.0 GENERAL ELEMENTS CONNECTING ONLY TWO GRIDS

2.1 WHY GENEL's? A SIMPLE EXAMPLE

A common modelling technique is to model a detailed part, obtain its principal stiffness terms, and represent the part as a BAR element in a larger scale model. BAR's have a serious limitation however: Their properties are defined by A, I1, I2, and J. The inertia terms (I1,I2) set several stiffness terms each: The diagonal terms, i.e. Kry,ry (rotation about Y due to moment about Y) and Kz,z (deflection in Z due to force in Z) and the off-diagonal term Kry,z=Kz,ry (rotation about Y due to force in Z = deflection in Z due to moment about Y) are all set by I2. Unfortunately the relationship between these stiffness terms and the inertia terms only hold true for uniform section beams. When representing a more complex structure with a BAR element, one must choose one of the bending stiffness terms, they cannot all be matched. There may also be off-diagonal terms which are important for a specific component, and more compromises must be made.

GENEL's provide the solution, with it, two GRID's may be connected with a full 6x6 stiffness matrix, and the user may set each term at will.

EXAMPLE:

Consider the following Run results for a 1-D type structure which we would like to represent as a BAR:

DIRECTION	STIFFNESS
Linear axis - X	1.21e7 lb/in
Linear axis -Y	2.92e6 lb/in
Linear axis - Z	1.17e7 lb/in
Torsion axis - RX	1.42e6 lb/in
Bending axis - RY	1.14e7 lb/in
Bending axis - RZ	4.71e6 lb/in

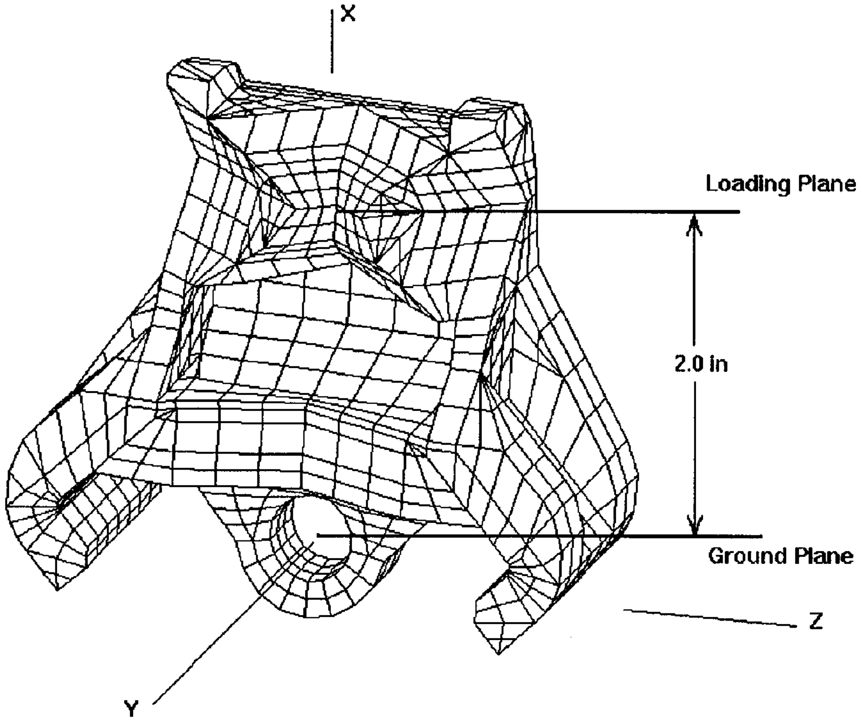

Figure 1 - Example 1D Structure

$$\mathit{ElasticModulus},\ E{=}3 \times 10^7 \quad \mathit{ShearModulus},\ G{=}1.15 \times 10^7 \quad \mathit{Length},\ L{=}2.0$$

$$Kx = 1.21 \times 10^7 = \frac{A \cdot E}{L} \rightarrow A = \frac{Kx \cdot L}{E} = 0.807 \leftarrow A \quad (1)$$

$$Ky = 2.92 \times 10^6 = \frac{3 \cdot E \cdot I1}{L^3} \rightarrow I1 = \frac{Ky \cdot L^3}{3 \cdot E} = 0.260 \leftarrow I1 \quad (2)$$

$$Kz = 1.17 \times 10^7 = \frac{3 \cdot E \cdot I2}{L^3} \rightarrow I2 = \frac{Kz \cdot L^3}{3 \cdot E} = 1.04 \leftarrow I2 \quad (3)$$

$$Krx = 1.42 \times 10^6 = \frac{J \cdot G}{L} \rightarrow J = \frac{Krx \cdot L}{G} = 0.247 \leftarrow J \quad (4)$$

$$Kry = 1.14 \times 10^7 = \frac{2 \cdot E \cdot I2}{L^2} \rightarrow I2 = \frac{Kry \cdot L^2}{2 \cdot E} = 0.76 \leftarrow I2 \quad (5)$$

$$Krz = 4.71 \times 10^6 = \frac{2 \cdot E \cdot I1}{L^2} \rightarrow I1 = \frac{Krz \cdot L^2}{2 \cdot E} = 0.314 \leftarrow I1 \quad (6)$$

Equations (2) & (6) for I1 and (3) & (5) for I2, clearly demonstrate the limitation of BAR elements. In this case the values are close, one can be picked or an average taken. None the less, the true stiffness of the structure cannot be matched. Furthermore, these are only the diagonal terms.

2.2 PROCEDURE TO CREATE A GENEL BETWEEN TWO GRIDS

Now that we have demonstrated the potential usefulness of connecting two grid points with a full 6x6 stiffness matrix, a straight forward procedure for creating a GENEL that will accomplish this will be outlined. Later, we shall expand the method such that any number of degrees of freedom can be dealt with.

The following is a step by step procedure, the theory behind the method is not included in this section, please refer to the section 'Stiffness Matrix and GENEL Theory' for a detailed discussion on how this technique works.

STEP 1:

Construct the detailed model of a part you wish to eventually reduce to a GENEL. Keep in mind that two grids are critical, the grounding grid and the load application grid. They should be properly connected to the model in all degrees of freedom. In this procedure we shall number the ground as grid 1 and the load point as grid 2.

STEP 2:

Run the model with GRID 1 grounded in 1-6, apply unit displacements one at a time to GRID 2 while constraining the other directions there. Use SPCFORCE output to obtain the stiffness matrix terms. Here is what your input deck should resemble:

```
ID GENEL,Example
SOL 101
TIME 999
CEND
SET 1 = 2  $ DEFINE OUTPUT AT GRID 2
DISP=1
SPCFORCE=1
SPC=1
SUBCASE 1
LOAD=1
SUBCASE 2
LOAD=2
SUBCASE 3
LOAD=3
SUBCASE 4
LOAD=4
SUBCASE 5
LOAD=5
SUBCASE 6
LOAD=6
BEGIN BULK
SPC     1       1       123456      $ CONSTRAIN GROUND GRID 1-6
SPC     1       2       123456      $ CONSTRAIN LOAD GRID, NECESSARY FOR
$                                   $ USE OF SPCD, ALSO CONSTRAINS OTHER
$                                   $ DofF's NOT SUBJECT TO UNIT DISP.
$
SPCD    1       2       1       1.0 $ ENFORCED UNIT DISPLACEMENTS
SPCD    2       2       2       1.0 $ THEY ARE CALLED UP AS LOADS
SPCD    3       2       3       1.0
SPCD    4       2       4       1.0
SPCD    5       2       5       1.0
SPCD    6       2       6       1.0
$
```

```
FORCE   1       2               0.0     1.      1.      1.
FORCE   2       2               0.0     1.      1.      1.
FORCE   3       2               0.0     1.      1.      1.
FORCE   4       2               0.0     1.      1.      1.
FORCE   5       2               0.0     1.      1.      1.
FORCE   6       2               0.0     1.      1.      1.
$
$  THE ABOVE ARE DUMMY FORCES, A LOAD SET MUST HAVE AT LEAST ONE FORCE
$  PRESENT... SPCD ALONE WILL CAUSE AN ERROR
.
.
.

$  MODEL   DATA

.
.
.
ENDDATA
```

STEP 3:

After the Run, the SPCFORCE output will look as follows, it represents the required force applied to the model to cause the desired (unit) displacement:

```
1                                                                          OCTOBER  28, 1994  MSC/NASTRAN   7/ 5/94   PAGE   465

0                                                                                                              SUBCASE 1

                                 F O R C E S   O F   S I N G L E - P O I N T   C O N S T R A I N T

      POINT ID.   TYPE         T1             T2             T3             R1             R2             R3
             2      G     1.210612E+07   2.78222E+04  -6.176900E-08   7.105250E-07   9.711800E-09   3.929989E+05
0                                                                                                              SUBCASE 2

                                 F O R C E S   O F   S I N G L E - P O I N T   C O N S T R A I N T

      POINT ID.   TYPE         T1             T2             T3             R1             R2             R3
             2      G     2.814489E+04   2.958222E+06  -3.176900E-06   2.105250E-09  -4.711800E-07  -3.229989E+06
0                                                                                                              SUBCASE 3

                                 F O R C E S   O F   S I N G L E - P O I N T   C O N S T R A I N T

      POINT ID.   TYPE         T1             T2             T3             R1             R2             R3
             2      G     6.614489E-08  -1.348222E-06   1.171900E+07  -2.105250E+03   9.891800E+06   7.929989E+02
0                                                                                                              SUBCASE 4

                                 F O R C E S   O F   S I N G L E - P O I N T   C O N S T R A I N T

      POINT ID.   TYPE         T1             T2             T3             R1             R2             R3
             2      G    -7.614489E-07   2.348222E-09  -2.126900E+03   1.421250E+06   4.211800E+03   5.329989E-04
0                                                                                                              SUBCASE 5

                                 F O R C E S   O F   S I N G L E - P O I N T   C O N S T R A I N T

      POINT ID.   TYPE         T1             T2             T3             R1             R2             R3
             2      G     9.614489E-09  -4.348222E-07   9.892900E+06   4.205250E+03   1.141800E+07  -3.729989E-02
0                                                                                                              SUBCASE 6

                                 F O R C E S   O F   S I N G L E - P O I N T   C O N S T R A I N T

      POINT ID.   TYPE         T1             T2             T3             R1             R2             R3
             2      G     3.914489E+05  -3.248222E+06   7.876900E+02   5.705250E-04  -3.711800E-02   4.729989E+06
```

The Diagonal terms are in **BOLD**.
The matrix is symmetric (within error tolerances).

The GENEL card has as input, the list of degrees of freedom (GRID #, and Component Direction) to be connected, and the lower triangular portion of the stiffness matrix. As well, for our two GRID GENEL, we will include the 'Ud' set which defines the rigid body behaviour of the element.

The format for the GENEL card given the preceding results is [Ref.1, p.484]:
(Notation: S1/T2 = Subcase 1 T2 SPCFORCE result)

```
GENEL    1                 2        1        2        2        2        3        +
+        2        4        2        5        2        6                          +
+        UD                1        1        1        2        1        3        +
+        1        4        1        5        1        6                          +
+        K        S1/T1    S1/T2    S1/T3    S1/R1    S1/R2    S1/R3    S2/T2    +
+        S2/T3    S2/R1    S2/R2    S2/R3    S3/T3    S3/R1    S3/R2    S3/R3    +
+        S4/R1    S4/R2    S4/R3    S5/R2    S5/R3    S6/R3
```

Substituting the symbol for values from the preceding SPCFORCE output gives:

```
GENEL    1                 2        1        2        2        2        3        +
+        2        4        2        5        2        6                          +
+        UD                1        1        1        2        1        3        +
+        1        4        1        5        1        6                          +
+        K        1.21+7   2.8+4    0.       0.       0.       3.9+5    2.92+6   +
+        0.       0.       0.       -3.2+6   1.17+7   -2.1+3   9.89+6   7.9+2    +
+        1.42+6   4.2+3    0.       1.14+7   -3.7-2   4.71+6
```

STEP 4:

The GENEL can now be included into a model, the only other model information required are the boundary GRID cards (and any referenced coordinate systems). It is good practice to test the GENEL, it can be subjected to the same load cases used on the original model. Compare the output they should be identical (within error tolerances).

3.0 Stiffness Matrix and GENEL Theory

3.1 Stiffness Matrix Formulation

The basic equation of motion for linear static problems is:

$\{ F \} = [K] \{ \delta \}$, where $\{ F \}$ is the applied force vector,
$[K]$ is the stiffness matrix,
and $\{ \delta \}$ is the resulting displacement vector.

A conceptual technique, from basic finite element theory [Ref. 2], of determining [K] for an element is to find the { F } that would produce a { δ } with a unit displacement for one degree of freedom, and zero displacement for all the rest. This { F } becomes one column of [K]. This procedure is repeated for every degree of freedom and [K } is built column by column. Consider the nth degree of freedom:

$$\begin{bmatrix} F_1 \\ F_2 \\ \cdot \\ \cdot \\ \cdot \\ F_m \end{bmatrix} = \begin{bmatrix} K_{11} & K_{12} & \cdots & K_{1n} & \cdots & K_{1m} \\ K_{21} & K_{22} & \cdots & K_{2n} & \cdots & K_{2m} \\ \cdot & \cdot & & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & & \cdot \\ K_{ml} & K_{m2} & \cdots & K_{mn} & \cdots & K_{mm} \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \\ \cdot \\ \cdot \\ \cdot \\ \delta_n \\ \cdot \\ \cdot \\ \cdot \\ \delta_m \end{bmatrix} \qquad \textit{where} \quad \delta_1, \delta_2, \ldots, \delta_{n-1}, \delta_{n+1}, \ldots, \delta_m = 0 \quad \textit{and} \quad \delta_n = 1$$

$$\begin{bmatrix} F_1 \\ F_2 \\ \cdot \\ \cdot \\ \cdot \\ F_m \end{bmatrix} = \begin{bmatrix} K_{11}\delta_1 + K_{12}\delta_2 + \cdots + K_{1n}\delta_n + \cdots + K_{1m}\delta_m \\ K_{21}\delta_1 + K_{22}\delta_2 + \cdots + K_{2n}\delta_n + \cdots + K_{2m}\delta_m \\ \cdot \\ \cdot \\ \cdot \\ K_{ml}\delta_1 + K_{m2}\delta_2 + \cdots + K_{mn}\delta_n + \cdots + K_{mm}\delta_m \end{bmatrix} = \begin{bmatrix} K_{1n} \\ K_{2n} \\ \cdot \\ \cdot \\ \cdot \\ K_{mm} \end{bmatrix}$$

Thus, it can be seen that the force vector required to cause a unit displacement of the nth degree of freedom, while constraining the rest, equals the nth column of the stiffness matrix.

The effort is to find the { F }'s. This can be done analytically for small elements such as beams, or in our work with GENEL's we use MSC/NASTRAN to determine the force vectors.

3.2 Stiffness Matrix Reduction

Recall that {F} = [K]{δ}. If we partition the matrices by separation of the boundary degree of freedom terms from the internal degree of freedom terms the following may be obtained:

$$\left[\begin{array}{c} F_i \\ \hline F_b \end{array}\right] = \left[\begin{array}{c|c} K_{ii} & K_{ib} \\ \hline K_{bi} & K_{bb} \end{array}\right] \left[\begin{array}{c} \delta_i \\ \hline \delta_b \end{array}\right]$$

Where subscript 'i' denotes internal DofF terms
and subscript 'b' denotes boundary DofF terms

$$\{ F_i \} = [K_{ii}]\{ \delta_i \} + [K_{ib}]\{ \delta_b \} \qquad (1)$$

$$\{ F_b \} = [K_{bi}]\{ \delta_i \} + [K_{bb}]\{ \delta_b \} \qquad (2)$$

No external loads may be applied to internal DofFs $\therefore \quad \{ F_i \} = 0 \qquad (3)$

$$\rightarrow [K_{ii}]\{ \delta_i \} = -[K_{ib}]\{ \delta_b \} \qquad (4)$$

$$\rightarrow \{ \delta_i \} = -[K_{ii}]^{-1}[K_{ib}]\{ \delta_b \} \qquad (5)$$

Substitute (5) *into* (2)

$$\rightarrow \{ F_b \} = \left[-[K_{bi}][K_{ii}]^{-1}[k_{ib}] + [K_{bb}] \right]\{ \delta_b \} \qquad (6)$$

$$\text{let} \quad K_b = -[K_{bi}][K_{ii}]^{-1}[k_{ib}] + [K_{bb}]$$

$$\{ F_b \} = [K_b]\{ \delta_b \} \qquad (7)$$

What is important to note from the preceding derivation is that given the stiffness matrix for the entire structure, a smaller matrix which defines the motion of the boundary degrees of freedom can be determined.

Equation (7) also matches the basic equation of motion for linear statics described in section 3.1, therefore it's terms can be found by the Unit Displacement technique.

Thus, an entire structure can be reduced to a stiffness matrix at its boundary, this matrix may be represented on a GENEL card.

Another important outcome can be seen from equation (5): The displacements for the entire structure can be obtained from it's stiffness matrix and the displacements of the boundary. This result is used to perform data recovery on the complete structure.

3.3 GENEL Creation

The format required for an arbitrary number of degrees of freedom GENEL card is [Ref.1, p.484]:

1	2	3	4	5	6	7	8	9	10
GENEL	EID	▨	UI1	CI1	UI2	CI2	UI3	CI3	+
+	UI4	CI4	UI5	CI5	-etc.-				

+*	K	K11	K21	K31	+
+*	-etc.-	K22	K32	-etc.-	+
+*	K33	K43	-etc.-		

(Double precision fields are used for 'K' to increase accuracy)

UI_i and CI_i are the grid numbers and component of each degree of freedom connected to the GENEL.

K_{ij} are the stiffness matrix terms. Note that only the lower triangular portion of the matrix (including the diagonal) is specified; the upper diagonal portion is redundant due to symmetry.

UI and CI are chosen by the user. Note that you need not connect all six components of a GRID to the GENEL; Only directions which can react load need to be included. In fact, degrees of freedom may be released (i.e. pinned connections) by simply not including them on the GENEL.

The [K] terms are obtained by the unit displacement technique (Section 3.1). The SPCFORCE output for each subcase contains one column of the stiffness matrix. The [K] terms must be extracted from the SPCFORCE output. The terms appear on the GENEL card sequentially column by column of the lower triangular portion and must be ordered to match the UI,CI ordering.

For the two grid method GENEL (Section 2.2) we have included the 'Ud' set. This was done to reduce the size of the GENEL card. We could have followed the general unit displacement procedure: Each end of the 1D type structure could be subjected to unit displacements in all directions (6 each end => 12x12 matrix). However, due to symmetry we know the results from each end will be the same. The 'Ud' set provides the relationship (length) between the two ends by allowing MSC/NASTRAN to formulate the rigid body behaviour. Then, we need only find the stiffness at one of the two ends (6x6 matrix). The 'Ud' set could be used for larger number of degrees of freedom, but the GENEL card size benefit does not warrant the increased complexity.

The technique to produce the GENEL closely matches that presented in Section 2.2:

STEP 1: Create the detailed model of the structure to be reduced paying special attention to the boundary grids.

STEP 2: Run the model with 1 subcase for each boundary degree of freedom, enforcing a unit displacement at the degree of freedom while constraining all other boundary degree of freedoms.

STEP 3: Extract the stiffness terms from the SPCFORCE output to create the GENEL.

The GENEL can then be used, the only other information required is the boundary GRIDs (and any referenced coordinate systems).

4.0 Automated Procedure to Create GENEL's with an Arbitrary Number of Degrees of Freedom

Although the steps to create the GENEL (Section 3.3) are not difficult, the shear volume of data complicates the issue. Consider a model with 10 boundary GRIDs each with 3 degrees of freedom: The Unit Displacement Technique requires 30 subcases, the resulting stiffness matrix is 30 x 30 (900 terms), and the triangular portion to be included on the GENEL card contains 465 terms. That is a lot of typing!

To automate the procedure, a series of FORTRAN routines have been written (developed on VAX platforms with VAX/FORTRAN, all routines developed and tested with MSC/NASTRAN V67.5).

Appendix A contains listings for the following programs:

4.1 The Input File (GENERAL.INP)

All programs require the input file GENERAL.INP to determine the boundary data for the model. It's format consists of 8-character fields (similar to MSC/NASTRAN's input deck). It's content is:

```
Database_name      $ Filename of the NASTRAN input deck containing the
                   $ structure to be reduced (No file extension is
                   $ entered, .DAT is assumed, may be longer than 8
                     characters)
      ⋮

GRID    CID     GID         $ Single GRID (id, GID) components CID (any
                            $ combination of digits 1 through 6)
      ⋮                     $ (any number of these cards may be present)

GRID2   CID     GID1    GID2  $ GRIDs GID1 through GID2 components CID
                              $ (any number of cards may be present)
      ⋮

END                $ EOF marker
```

All boundary GRID's and their degrees of freedom which can react load must be specified on either a GRID or GRID2 card.

4.2 GENERAL.FOR

GENERAL.FOR assists in creating the data deck for the Unit Displacement run. It creates all the necessary subcases, SPC, SPCD, and FORCE cards. It creates the Executive and Case control sections as well as the beginning of the Bulk Data section. The output file has '_GEN' concatenated to the database name (.i.e. TEST.DAT database gives TEST_GEN.DAT output). To complete the Unit Displacement run deck, simply edit the output file, from the editor include (merge) the original database file, and removed duplicate cards (such as BEGIN BULK, etc..). Save the file under the same name as the output file.

4.3 CRE_GENEL.FOR

CRE_GENEL.FOR reads the SPCFORCE output from the '.F06' file of the '_GEN' run, and creates the GENEL card. The GENEL ID is arbitrarily set to 20, it can be manually changed as required. The output file is Database_name.GEN (i.e. TEST.DAT, TEST_GEN.F06 gives TEST.GEN) The '.GEN' file also contains the boundary GRID's and any defined coordinate systems.

4.4 CRE_SPC.FOR

CRE_SPC.FOR asks the user to input a filename containing displacement output (specify the '.F06' extension). The purpose is to extract from a run, containing the GENEL, the displacements at its boundaries. The program currently is set up to extract the displacements from non-linear run's '.F06' file for the final load increment of each subcase. The listing identifies the lines to remove to extract displacements from either linear runs or for all increments of a non-linear run. The output is a file with the input filename but extension '.SPC'. The output contains SPCD cards for all boundary GRID's, the loadset number is incremented for each subcase. These SPCD cards can be inserted into the original (unreduced) model to perform data recovery for stresses or whatever else may be desired. Note also that the run may be set up as a RESTART of the '_GEN' run since the stiffness matrix and boundary constraints are compatible.

4.5 Outline of Automated Procedure

STEP 1: Create the detailed model to reduce, identify the desired boundary degrees of freedom. Let us assume it is named 'MYMODEL.DAT'.

STEP 2: Create the 'GENERAL.INP' file. Database_name equals 'MYMODEL', specify the appropriate boundary.

STEP 3: Run 'GENERAL.FOR'.

STEP 4: Append 'MYMODEL.DAT' to 'MYMODEL_GEN.DAT' remove any duplicate Executive or Case control cards, save as 'MYMODEL_GEN.DAT'

STEP 5: Submit 'MYMODEL_GEN.DAT' to MSC/NASTRAN

STEP 6: Run 'CRE_GENEL.FOR'.

STEP 7: Include 'MYMODEL.GEN' (the reduced structure model) into other models as necessary. Be sure to connect the boundary GRID's appropriately to adjacent structures.

STEP 8: Run the 'other' models, requesting DISPLACEMENT output if future data recovery on the GENEL is anticipated. (Assume the other model is named 'OTHERMODEL.DAT'

STEP 9: Run 'CRE_SPC.FOR', 'OTHERMODEL.F06' is the displacement file.

STEP 10: Modify 'MYMODEL_GEN.DAT' as follows: Remove unnecessary subcase references, the unit displacement SPCD's, and dummy FORCE cards. Insert 'OTHERMODEL.SPC', and add appropriate subcases. RESTART, or re-run the new deck to recover stresses, etc..

5.0 Advantages and Disadvantages of Using GENEL Reductions

5.1 Advantages

The GENEL reduction technique discussed in this paper allows many advantages. Primarily it provides an alternative to SuperElement methods for dividing a large model into sub-components which are solved individually.

As with SuperElements the subdivision of a model can result in substantial reduction of execution time (i.e. the solution time for two 100 x 100 matrices is less than that for one 200 x 200 matrix). Furthermore, if changes are made to one sub-component only the modified component needs to be re-solved. For non-linear solutions (prior to MSC/NASTRAN V.68) creating a 'residual' structure with only the non-linear elements and keeping the linear elements in SuperElements (or GENELs) significantly reduces run time since the matrix to be iterated and solved is much smaller. GENELs (or SuperElements) also provide a means of modelling some unique connections. For example, they can allow you to model a hinged plate since the boundary degrees of freedom can be selectively excluded (released).

GENELs provide some additional advantages over SuperElements:

They are simple to use. Once the GENEL card is created integrating it into other models only required connecting up the boundary GRIDs. Further, the GENEL card may be used in any number of separate models. All data is stored in text format, to use the GENEL one need only include the text in a model.

They provide a means of protecting confidential design features and/or trade secrets while allowing a stiffness representative model to be passed on to subcontractors.

The GENEL is completely self-contained, it has no external references to databases, GRIDs or coordinate systems. The stiffness matrix is 'translatable', that is to say, the GRIDs specified on the GENEL card need only have the same relative positions with each other as the boundary GRIDs had when the GENEL stiffness matrix was produced. Their analysis coordinate frames must also be of the same type and parallel to the systems used on the boundary GRIDs. The origins of the coordinate frames need not be in the same locations (but if more than one frame is used then the relative positions of the origins with respect to each other must remain the same. Also non-rectangular systems are less flexible since their coordinate directions are non-constant. For example a cylindrical frame's origin may only be moved in the 'Z' direction). This feature is most easily taken advantage of if the boundary GRIDs are all defined in a local coordinate frame. The GENEL may now be included at any position or orientation within another model simply by redefining the origin and orientation of the local frame. Additionally, it is easy to create duplicate GENELs (for models that have more than one of the same sub-component). Simply use a text editor to duplicate the GENEL card and boundary GRIDs, manually renumber the GRIDs specified on the GENEL, renumber the boundary GRID cards and specify a new coordinate system on them. Define the new coordinate system to place the duplicate GENEL as desired.

5.2 Disadvantages

Some disadvantages of these techniques are readily apparent:

The procedure to create the GENEL is an extra step in the modelling process.

Debugging of the models must be done prior to GENEL creation since the GENEL contains no details of the model.

This technique provides only a stiffness representation. Mass properties are not included therefore GENEL's are typically not as useful for dynamic problems.

As with SuperElements, dependent degrees of freedom cannot be part of the boundary. The boundary is subject to single point constraints, and therefore forms part of a mutually exclusive set.

Compatibility of the GENEL card is limited across different MSC/NASTRAN versions. The same version must be used for the GENEL creation and GENEL data recovery runs. This is because different versions compute slightly different stiffness matrices (due to element improvements).

6.0 Example of GENEL Usage in a Complex Model

Figure 2 illustrates a robotic grappler. Two jaws slide opened and closed along a track to grasp two different types of fixtures. The remaining structure consists of a shell, and front plate. Loads are applied by one of the two fixture types through the jaws and finally reacted out the back end of the shell. Rotational stiffness values and stress levels had to be determined for operational loads with either fixture type. The problem is complicated by the fact that the jaw's closed position is different for each fixture type. Furthermore, it was desired to model the non-linear nature of all contact surfaces (since surface separation would significantly effect stiffness values).

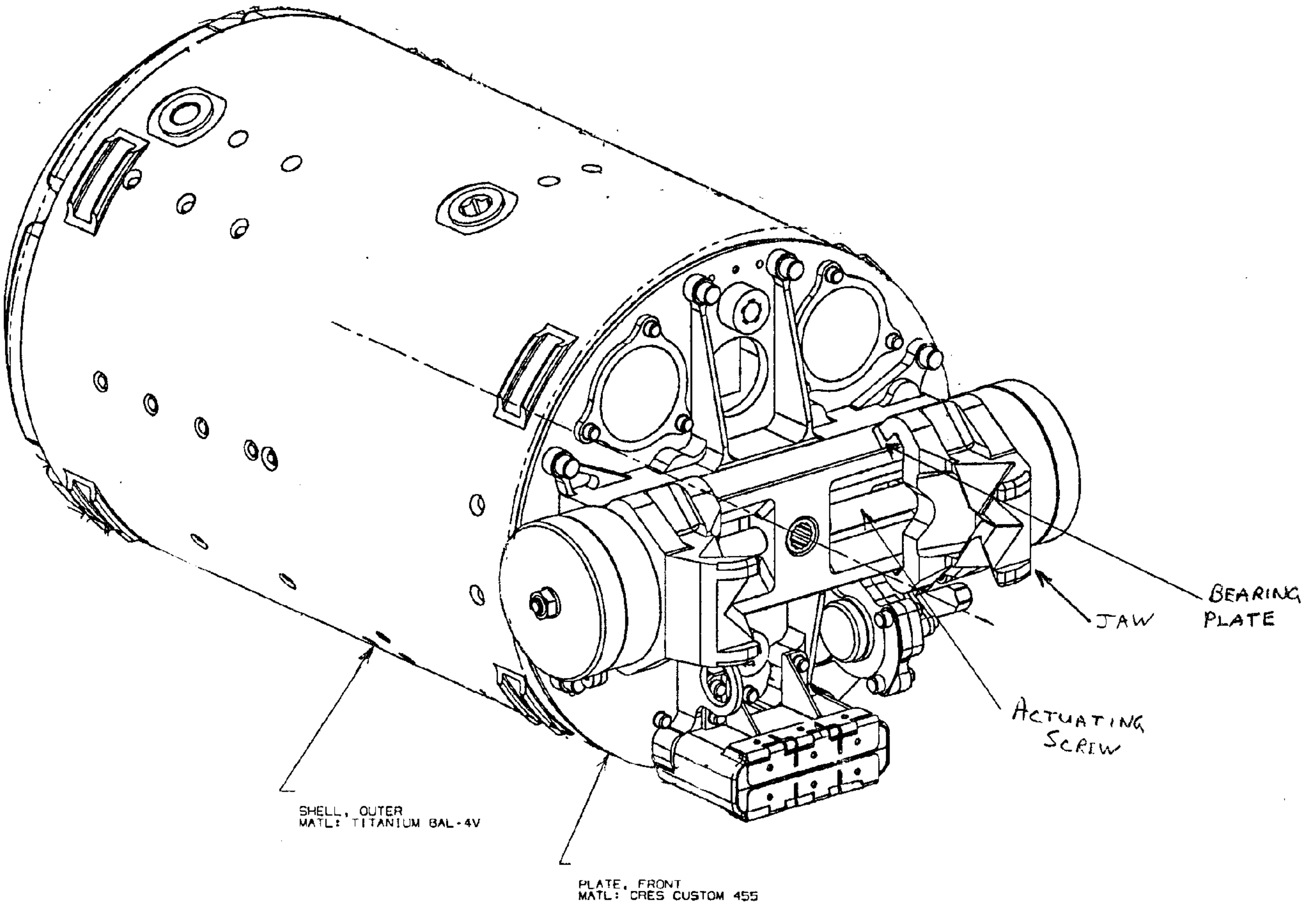

Figure 2 - Robotic Grappler Mechanism

The GENEL reduction technique was implemented to aid the analysis process. The Shell and Front Plate were modelled in detail as one unit, and a separate detailed solid model of a single jaw was created.

The Shell and Front Plate model was reduced to a GENEL, it's boundary consisted of a single GRID at the back end as ground, two sets of contact points (one set for each of the two possible 'jaw closed' positions) on the Front Plate track, and two points which connect to the jaw actuating screws. The basic cartesian coordinate system was used for all boundary GRIDs.

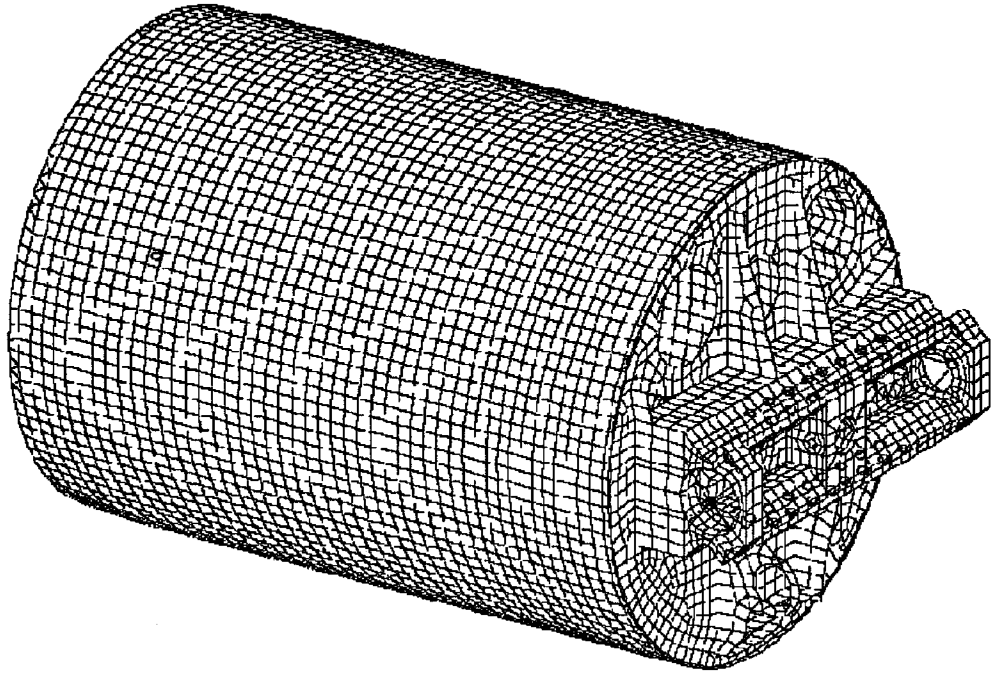

Figure 3 - Shell and Front Plate Model

The Jaw model was also reduced to a GENEL, it's boundary included the mating contact surface GRIDs at the track, two sets of contact points which would engage one or the other fixtures to be grasped, and a point to connect with the sctuating screw. A local rectangular coordinate system rotated 45 degrees from the basic system (to align it with the track contact surfaces) was used for all Jaw boundary GRIDs (Figure 1, Section 2.1, shows the Jaw Model).

Two separate non-linear models (one for each fixture type) were then created to connect the various components of the mechanism. The Shell and Front Plate GENEL was inserted as is. The Jaw GENEL was duplicated to represent the second Jaw. The local coordinate system for the first Jaw was defined to place it appropriately. The local system for the second Jaw was rotated 180 degrees and the origin placed appropriately. The only difference between the two models thus far is the origin of the Jaw coordinate systems. GAP elements and other interface features were then added to each model.

The non-linear models were executed and stiffness values obtained. Stress results were determined by back substituting the non-linear displacement results into the detailed models.

The benefits obtained by using the GENEL technique were: Separation of the sub-components from the non-linear interfaces. Re-use of the sub-components for two different configurations while requiring only one solution of the detailed models. Duplication of the detailed Jaw model while only requiring the solution for a single Jaw.

7.0 Conclusions

We have demonstrated that any linear static structure may be reduced to a stiffness matrix of it's boundary degrees of freedom. Further, it was shown that given the displacements of the boundary data recovery can be performed for the interior structure.

Converting a sub-component to a GENEL provides a means of reducing model execution times, manipulating and duplicating sub-components, reducing a detailed model with many degrees of freedom to a simpler form for inclusion in next level assembly models, and separating a large model into several smaller ones.

8.0 Acknowledgments

Special thanks to Spar Aerospace Limited for their support of this work, and for providing the continual source of new challenges that inspired it.

9.0 References

1. *MSC/NASTRAN Quick Reference Guide, Version 67*, The MacNeal-Schwendler Corporation, Los Angeles, CA, 1992.

2. Huebner, K. H., and Thornton, E. A., *The Finite Element Method For Engineers*, 2nd ed., John Wiley & Sons Inc., 1982.

APPENDIX A
PROGRAM LISTINGS

```
$**************************************************************************
$       GENERAL.FOR, creates necessary subcases for GENEL reduction
$       Developed by Mitch Greenberg, SPAR AEROSPACE LIMITED
$**************************************************************************

        character db*40,line*79,cdof*8
        integer dofs(2,1200),dofl(2,1200)

        ndofs=0
        ndofl=0
        open(1,file='general.inp',status='old')

        read(1,'(a40)')db
10      read(1,'(a80)',END=20)line
        if (line(1:8).eq.'GRID    '.or.line(1:8).eq.'grid    ')
     +  then

        read(line,'(bn,8x,a8,i8)')cdof,igrid

        do 1 i=1,8

        if (cdof(i:i).ne.' ') then
        ndofs=ndofs+1
        dofs(1,ndofs)=igrid
        dofs(2,ndofs)=ichar(cdof(i:i))-ichar('0')
        endif

1       CONTINUE
        endif

        if (line(1:8).eq.'GRID2   '.or.line(1:8).eq.'grid2   ')
     +  then

        read(line,'(bn,8x,a8,2i8)')cdof,igrids,igridf
        print*,igrids,igridf

        do 2 i=1,8
        if (cdof(i:i).ne.' ') then
        do 3 j=igrids,igridf
        ndofs=ndofs+1
        print*,ndofs
        dofs(1,ndofs)=j
        dofs(2,ndofs)=ichar(cdof(i:i))-ichar('0')
3       CONTINUE
        endif

2       CONTINUE
        endif

        if (line(1:8).ne.'END     '.or.line(1:8).ne.'end     ')
     +  goto 10
20      close(1)

        do i=1,ndofs-1
        do j=1,ndofs-i
        if (dofs(1,j+1)*10+dofs(2,j+1).lt.dofs(1,j)*10+dofs(2,j)) then
        it1=dofs(1,j)
        it2=dofs(2,j)
        dofs(1,j)=dofs(1,j+1)
        dofs(2,j)=dofs(2,j+1)
        dofs(1,j+1)=it1
        dofs(2,j+1)=it2
        ENDIF
```

```
enddo
enddo

dofs(1,ndofs+1)=0
dofs(2,ndofs+1)=0
do j=1,ndofs
if (dofs(1,j+1)*10+dofs(2,j+1).ne.dofs(1,j)*10+dofs(2,j)) then
ndofl=ndofl+1
dofl(1,j)=dofs(1,j)
dofl(2,j)=dofs(2,j)
endif
enddo
LINE=DB(1:INDEX(DB,' ')-1)//'.DAT'
open(1,file=LINE,status='old')
LINE=DB(1:INDEX(DB,' ')-1)//'_GEN.DAT'
open(2,file=LINE,status='new',carriagecontrol='list')
line='ID  GENEL,'//DB
write(2,'(A80)')LINE
WRITE(2,'(''SOL 61'')')
WRITE(2,'(''TIME 10000.'')')
WRITE(2,'(''CEND'')')
WRITE(2,'(''SEALL=ALL'')')
WRITE(2,'(''SUPER=ALL'')')
WRITE(2,'(''SPCFORCE=ALL'')')
WRITE(2,'(''ECHO=NONE'')')
WRITE(2,'(''ECHOOFF'')')
WRITE(2,'(''SPC=1234'')')

do i=1,ndofl
write(2,'(''SUBCASE '',I3)')I
WRITE(2,'(''LOAD = '',I3)')I
ENDDO
WRITE(2,'(''BEGIN BULK'')')

DO I=1,NDOFL
WRITE(2,'(''SPC     1234    '',2I8)')DOFL(1,I),DOFL(2,I)
ENDDO

DO I=1,NDOFL
WRITE(2,'(''FORCE   '',3I8,4F8.4)')I,DOFL(1,I),0,.001,
1.,1.,1.
WRITE(2,'(''SPCD    '',3I8,F8.4)')I,DOFL(1,I),DOFL(2,I),1.
ENDDO

CLOSE(1)
CLOSE(2)
STOP
END
```

```
$***********************************************************************
$       CRE_GENEL.FOR, creates GENEL card from SPCFORCE output
$       Developed by Mitch Greenberg, SPAR AEROSPACE LIMITED
$***********************************************************************

        character db*40,line*79,cdof*8,linea*132,lineb*132,linec*132
        character conv*8
        integer dofs(2,1200),dofl(2,1200)
        real mat(1200,1200),spcf(6),mat1(720600)
        data mat/1440000*0./
        ndofs=0
        ndofl=0
        open(1,file='general.inp',status='old')

        read(1,'(a40)')db
10      read(1,'(a80)',END=20)line
        if (line(1:8).eq.'GRID    '.or.line(1:8).eq.'grid    ')
     +  then

        read(line,'(bn,8x,a8,i8)')cdof,igrid
        do 1 i=1,8

        if (cdof(i:i).ne.' ') then
        ndofs=ndofs+1
        dofs(1,ndofs)=igrid
        dofs(2,ndofs)=ichar(cdof(i:i))-ichar('0')
        endif

1       CONTINUE
        endif

        if (line(1:8).eq.'GRID2   '.or.line(1:8).eq.'grid2   ')
     +  then

        read(line,'(bn,8x,a8,2i8)')cdof,igrids,igridf
        do 2 i=1,8
        if (cdof(i:i).ne.' ') then
        do 3 j=igrids,igridf
        ndofs=ndofs+1
        dofs(1,ndofs)=j
        dofs(2,ndofs)=ichar(cdof(i:i))-ichar('0')
3       CONTINUE
        endif

2       CONTINUE
        endif

        if (line(1:8).ne.'END     '.or.line(1:8).ne.'end     ')
     +  goto 10
20      close(1)

        do i=1,ndofs-1
        do j=1,ndofs-i
        if (dofs(1,j+1)*10+dofs(2,j+1).lt.dofs(1,j)*10+dofs(2,j)) then
        it1=dofs(1,j)
        it2=dofs(2,j)
        dofs(1,j)=dofs(1,j+1)
        dofs(2,j)=dofs(2,j+1)
        dofs(1,j+1)=it1
        dofs(2,j+1)=it2
        ENDIF
        enddo
        enddo


        dofs(1,ndofs+1)=0
        dofs(2,ndofs+1)=0
        do j=1,ndofs
        if (dofs(1,j+1)*10+dofs(2,j+1).ne.dofs(1,j)*10+dofs(2,j)) then
        ndofl=ndofl+1
        dofl(1,j)=dofs(1,j)
        dofl(2,j)=dofs(2,j)
        endif
        enddo
        LINE=DB(1:INDEX(DB,' ')-1)//'.DAT'
        open(1,file=LINE,status='old')
        LINE=DB(1:INDEX(DB,' ')-1)//'_GEN.F06'
        open(2,file=LINE,status='old')
        LINE=DB(1:INDEX(DB,' ')-1)//'.GEN'
        open(3,file=LINE,status='NEW',CARRIAGECONTROL='LIST')
```

```
100     READ(1,'(A79)',end=109)LINE
        IF (LINE(1:8).EQ.'GRID    '.OR.LINE(1:8).EQ.'grid    ')
     +  then
        read(line,'(bn,8x,i8)')igrd
        do 101 i=1,ndofl
101     if (igrd.le.dofl(1,i)) goto 102

102     if (igrd.eq.dofl(1,i)) write(3,'(A79)')line
        endif

        if (line(1:4).eq.'CORD'.or.line(1:4).eq.'cord') then
        write(3,'(A79)')line
        READ(1,'(A79)')LINE
        write(3,'(A79)')line
        endif
        goto 100
109     close(1)

201     linea=' '
        lineb=' '
        linec=' '

200     linec=lineb
        lineb=linea
        read(2,'(a132)',end=299)linea
        if (index(linea,'F O R C E S   O F   S I N G L E').eq.0)
     +  goto 200

        linec=linec(100:)
        read(linec(index(linec,'SUBCASE')+7:),'(bn,i6)')isub

        read(2,'(a132)')linea
        read(2,'(a132)')linea
230     read(2,'(a132)')linea
        in=index(linea,'G')
        if (in.eq.0) goto 201

        linea(in:in)=' '
        read(linea,*,err=201)igrd,spcf
        do 210 id=1,6
        do 205 i=1,ndofl
205     if (igrd.eq.dofl(1,i).and.id.eq.dofl(2,i)) goto 220
220     mat(i,isub)=spcf(id)
210     continue
        goto 230

299     ILT=1


        DO 250 J=1,ndofl
        DO 250 I=J,ndofl
        mat1(ILT)=mat(I,J)
250     ILT=ILT+1

        iel=20
        WRITE(3,'(A8,I8,8X,6I8,''+'',<int((ndofl*2-6)/8.)>
     +  (/''+       '',8I8,''+''),
     +  /''+       '',<ndofl*2-6-int((ndofl*2-6)/8.)*8>I8,
     +  T73,''+'')')'GENEL   ',iel,((dofl(j,k),j=1,2),k=1,ndofl)

        WRITE(3,'(''+*      K                       '',3E16.6,''+'',/,
     +  <int((ndofl**2/2.+ndofl/2.-3)/4.)>(''+*       '',4E16.6,''+'',/),
     +  ''+*      '',<INT(ndofl**2/2.+ndofl/2.-3-int((ndofl**2/2.+
     +  ndofl/2.-3)/4.)*4)>E16.6)')
     +  (mat1(k),k=1,int(ndofl**2/2.+ndofl/2.))

        CLOSE(1)
        CLOSE(2)
        CLOSE(3)
        STOP
        END
```

```
$**********************************************************************
$       CRE_SPC.FOR, creates SPCD card from displacements for GENEL
$       datārecovery
$       Developed by Mitch Greenberg, SPAR AEROSPACE LIMITED
$**********************************************************************

        character db*40,line*79,cdof*8,linea*132,lineb*132,linec*132
        character conv*8,fname*40
        integer dofs(2,1200),dofl(2,1200)
        real spcf(6)
        ndofs=0
        ndofl=0
        print*,'enter displacement file'
        read(*,'(a40)')fname
        open(1,file='general.inp',status='old')
        open(2,file=fname,status='old')
        fname=fname(1:index(fname,'.'))//'spc'
        open(3,FILE=fname,status='new')

        read(1,'(a40)')db
10      read(1,'(a80)',END=20)line
        if (line(1:8).eq.'GRID    '.or.line(1:8).eq.'grid    ')
     +  then

        read(line,'(bn,8x,a8,i8)')cdof,igrid
        do 1 i=1,8

        if (cdof(i:i).ne.' ') then
        ndofs=ndofs+1
        dofs(1,ndofs)=igrid
        dofs(2,ndofs)=ichar(cdof(i:i))-ichar('0')
        endif

1       CONTINUE
        endif

        if (line(1:8).eq.'GRID2   '.or.line(1:8).eq.'grid2   ')
     +  then

        read(line,'(bn,8x,a8,2i8)')cdof,igrids,igridf
        do 2 i=1,8
        if (cdof(i:i).ne.' ') then
        do 3 j=igrids,igridf
        ndofs=ndofs+1
        dofs(1,ndofs)=j
        dofs(2,ndofs)=ichar(cdof(i:i))-ichar('0')
3       CONTINUE
        endif

2       CONTINUE
        endif

        if (line(1:8).ne.'END     '.or.line(1:8).ne.'end     ')
     +  goto 10
20      close(1)

        do i=1,ndofs-1
        do j=1,ndofs-i
        if (dofs(1,j+1)*10+dofs(2,j+1).lt.dofs(1,j)*10+dofs(2,j)) then
        it1=dofs(1,j)
        it2=dofs(2,j)
        dofs(1,j)=dofs(1,j+1)
        dofs(2,j)=dofs(2,j+1)
        dofs(1,j+1)=it1
        dofs(2,j+1)=it2
        ENDIF
        enddo
        enddo

        dofs(1,ndofs+1)=0
        dofs(2,ndofs+1)=0
        do j=1,ndofs
        if (dofs(1,j+1)*10+dofs(2,j+1).ne.dofs(1,j)*10+dofs(2,j)) then
        ndofl=ndofl+1
        dofl(1,j)=dofs(1,j)
        dofl(2,j)=dofs(2,j)
        endif
        enddo

        ISCT=0
201     linea=' '
        lineb=' '
        linec=' '

200     linec=lineb
        lineb=linea
        read(2,'(a132)',end=299)linea
        if (index(linea,'D I S P L A C E M E N T   V E C T O R').eq.0)
     +  goto 200

        linec=linec(100:)
        read(linec(index(linec,'SUBCASE')+7:),'(bn,i6)')isub
        ILS= INDEX(LINEB,'LOAD STEP =')
        IF (ILS.EQ.0) GOTO 201
        READ(LINEB(ILS+11:),*)XLS
        IF (FLOAT(INT(XLS)).NE.XLS) GOTO 201

        read(2,'(a132)')linea
        read(2,'(a132)')linea
230     read(2,'(a132)')linea
        in=index(linea,'G')
        if (in.eq.0) goto 201

        linea(in:in)=' '
        read(linea,*,err=201)igrd,spcf
        do 210 id=1,6
        do 205 i=1,ndofl
        if (igrd.eq.dofl(1,i).and.id.eq.dofl(2,i)) THEN
220     IF (ISCT.NE.ISUB)
     +  WRITE(3,'(''FORCE   '',3I8,4F8.4)')ISUB,IGRD,0,.001,1.,1.,1.
        ISCT=ISUB
        WRITE(3,'(''SPCD*   '',3I16,E16.6)')ISUB,IGRD,ID,SPCF(ID)
        GOTO 210
        ENDIF
205     CONTINUE
210     continue
        goto 230

299     CLOSE(1)
        CLOSE(2)
        CLOSE(3)
        STOP
        END
```

Remove for Linear (See section 4.4)