Creation and Automatic Attachment of Reduced Component Models for Dynamic Analysis

by

Ted Rose

Manager, MSC/NASTRAN Training and Support

The MacNeal-Schwendler Corporation

ABSTRACT:

In most applications, different groups or even different companies model structures as a series of individual components. At some point in the analysis process, these component models need to be assembled to get a “complete” model of the structure being considered. This paper presents a set of DMAP alters for MSC/NASTRAN which allow each component to be processed independently (using the superelement enhancements implemented in V69) and replaced by a set of reduced matrices representing the properties of the component as seen by adjacent structure. The alters then allow the creation of assemblies using any combination of the components, with a special feature, automatic connection at the boundaries. The alters allow the creation of OTM (Output Transformation Matrices) for use in data recovery and also allow for assembly plots showing all elements in all components (even without the bulk data defining the components).

Introduction:

The approach implemented in this paper uses DMAP and the new superelement features introduced in V69 to allow component models to be ‘automatically’ put together to create an assembly. The alter requires that each component handled in this way be modeled as a ‘PART’ superelement using the partitioned bulk data features. Reduced matrices representing its properties as seen by adjacent structure replace each superelement and the data is stored in a file using OUTPUT2 (either binary or neutral format). The alters then allow for assemblies to be created using these reduced superelements with superelements which are physically present in the bulk data. The alters use the automatic attachment features of the new partitioned bulk data superelements to attach the reduced models to the rest of the structure. Special features of the alters are that OTM (Output Transformation Matrices) may be created to allow selected data recovery for the components, and also assembly plots (including the reduced components) are possible. In fact, deformed plots of the assembly are possible.

1. Overview

The goal of this paper is to show an approach using MSC/NASTRAN, which can be used to create assembly models using a series of components. A “one-shot” approach to perform this was introduced in V69 of MSC/NASTRAN[1] and this paper builds on that approach. MSC/NASTRAN superelements now include the ability to create an assembly using separate component models (which may even have duplicate GRID point or element numbers). This new feature is called Partitioned Bulk Data Superelements (or PARTs for short). Since the procedure presented in this paper uses these PARTs, this paper starts with a brief description of the partitioned input and then provides an explanation of the process used by the DMAP alters. The use of PARTs is described in the Release Guide[1] and will be described in the Superelement User’s Guide when it is published.

The concept of PARTs is simply described as a “building block” approach to finite elements. Each piece of the structure is modeled separately and a self-contained set of Bulk Data is created. Once all of the components are modeled and checked, the new PART superelements may be used to create the assembly model. The procedure to do this follows.

Starting with V69 of MSC/NASTRAN, the Bulk Data section of the input file may be partitioned. This is accomplished by using a new “BEGIN SUPER=I” (where I is the superelement number being defined) entry. Any input that occurs between the BEGIN BULK entry and the first BEGIN SUPER entry is called the Main Bulk Data Section (if there is a physical model for the residual structure, it must be in this section). This area of the input file is treated as in previous versions and conventional superelements (those defined using the “pre-V69 method) may be defined in this section. As soon as a BEGIN SUPER entry occurs, any input between that entry and the next BEGIN SUPER (or ENDDATA) entry is for that superelement. This section is “self-contained” and may not have a reference to entries (such as materials and section properties) in any other section of the input data. The exception to this rule is that attachment information may be provided in the main bulk data section.

One of the important features of these new PARTs is automatic attachment. That is, MSC/NASTRAN will search for any coincident GRID points and when they are found, the PARTs will be connected at those points. The automatic attachment also accounts for any “mis-alignment” of coordinate systems. That is, if one PART has a cylindrical displacement coordinate system and the point that is being connected in another PART has a rectangular system which is rotated 45 degrees from the cylindrical one, the program automatically accounts for this when attaching the PARTs.

There are several options available to determine how the automatic attachment logic works. By default, coincident (within a tolerance) GRID points are connected, however, you may specify points which “may be” connected, points which “may not be” connected, or even manually specify the points to connect. For a description of this, see the V69 Release Guide[1]).

The MSC/NASTRAN input file now looks like the following”

```
SOL 103
TIME 10
CEND
TITLE = SAMPLE OF PARTITONED INPUT
.
. STANDARD SUPERELEMENT CASE CONTROL
.
BEGIN BULK
.
$  MAIN BULK DATA SECTION
.
GRID,1,,10.,10.
.
BEGIN SUPER=1
.
$ INPUT FOR SUPERELEMENT 1
GRID,10,,10.,10.
.
BEGIN SUPER=2
.
$ INPUT FOR SUPERELEMENT 2
GRID,2,,10.,10.
.
ENDDATA
```

In this input file, GRID point 1 in the residual structure (defined in the Main Bulk Data) is coincident with GRID point 10 in superelement 1 and GRID point 2 in superelement 2, so they will be connected. In this case, each point has a different id, but there is no requirement for this, a copy of GRID point 1 from the Main Bulk Data could have been used in each partitioned section as well as using different id's.

This new feature allows you to easily create assemblies using component models. If a component model is not modeled in the correct location or orientation, there are entries that allow you to translate and rotate it into the desired location.

Once the automatic attachment logic is performed, these new PARTs are treated the same as conventional superelements internally. That is, you may perform static reduction, component modal synthesis, and may even use residual vectors (new in V70) to improve accuracy in dynamic solutions.

2. What the Alters Do

For many years now, groups have been passing models or reduced matrices (mass, stiffness, damping, loading, and Output Transformation) between themselves. If matrices are used, these matrices represent the properties of a component as seen at its attachment locations. There are a number of problems and limitations involved in this approach. Among these is the possibility that terms in the matrices may not be in the expected order (a very common problem, which results in wrong answers) and there is no way to plot the assembly. This paper attempts to correct both of these problems.

The alters take advantage of the partitioned input and process each component separately. In the first step, a copy of the SORTED BULK DATA for each component is written to an OUTPUT2 file, along with the reduced matrices and the Output Transformation Matrices (if requested). In the second step, the data is read in, automatically attached to the rest of the model, and the problem is solved. At this point, you may use the OTM (Output Transformation Matrices) from the initial run, or may perform standard data recovery. The alters allow for undeformed and deformed plots of the assembly, including stress and deformation contours.

These alters are an enhancement on the alter1g-series[2] in the sssalter[3] directory and copies of the new alters will be placed in this directory.

3. User Interface

The process requires two steps. The initial step uses alter1ha.v70 and creates the reduced models (this is usually the sub-contractor's part of the job). The second step uses alter2ha.v70 to assemble the PARTs together (this is usually the system integrator's job).

Step 1 - Creating the Reduced Matrices

The first requirement is that each component, which you wish to replace with reduced matrices, must be modeled as a PART. One or more PARTs may be processed in a run. For each PART, all interface (connection) points must be identified either by using ASET entries in the partitioned bulk data section, or by placing coincident GRID points (simply copies of the GRID points from the partitioned data) in the Main Bulk Data Section. **This is a requirement! The current version of the alters do not perform checks to verify that the attachment points found during step 2 match those used in step 1.**

If you wish to perform component modal synthesis, include QSET (or QSET1) entries and SPOINT entries (to define the degrees of freedom {dof} to be used to represent the component modes) in the partitioned bulk data. Also, include an EIGRL (or EIGR) in the partitioned bulk data section and a METHOD command in the case control for the superelement.

Be sure to include the DMAP alter (alter1ha.v70) in the Executive Control section of the input file.

The alter uses the following input to control it's operations:

PARAM,NOOTM – integer -

-1 = default = Create OTM as requested in Case Control Section

+1 = Do not create OTM

USET,U2 - if used, the "U2" set of dof are the only ones which will be stored in the boundary transformation matrix (GOA). If not used, all interior dof are used in the transformation matrix. When using the reduced matrices in any runs, loads may only be applied on dof which have been specified in the U2 set during step 1. If deformed plots are requested in subsequent runs, the only dof which will have non-zero deformations in the plots are the U2-set (once again, if not defined, all dof are retained for this purpose)

PARAM, MATFILE – integer – **mandatory** - this parameter indicates the FORTRAN unit number for the OUTPUT2 file, which the alter will be writing information to. Default value = -1, which indicates that the reduced matrices should be 'punched' as DMIG entries in the "PUNCH" file. If this is used, the automatic attachment will not work.

PARAM,FIXEDB,-1 – must be set to this value if OTM are desired.

PARAM,NEUTRAL – integer – controls the writing of the tables which describe what is in the OTM (only necessary when creating OTM)
-1 => unformatted (DRMH1 tables are written to OUTPUT file)
1 => formatted (DRMH1 tables are written to the punch file using DTI format and
should be included in the partitioned bulk data in the run reading the data for this
superelement)

SDAMP – if modal damping is desired on the superelement component modes, this should be used in the Case Control with a TABDMP1 in the partitioned bulk data section.

Step 2 - Reading the reduced Matrices into a new model using Automatic Attachment

This is the second step (often referred to as the “system integrator” run) in the solution process. In this run, you have the structure, to which you wish to attach the reduced matrices, modeled as the residual structure and/or other superelements. The reduced matrices are read in by creating a section in the input file for a PART (by using a BEGIN SUPER = I , where I is the superelement id to be read in as reduced matrices). In the section of the input file following the BEGIN SUPER, you have:

PARAM,INFILE, integer, default = 0, which implies that the data is in the section using DMIG entries.

A positive value is the unit number of the file containing the reduced matrices from the first step. This file needs to be available and (if possible) attached using an ASSIGN statement in the FMS.

PARAM,NEUTRAL integer, default = -1 , the default implies that the tables describing the OTM are on the OUTPUT2 file created in the previous run. Used only if OTM were created during step 1.

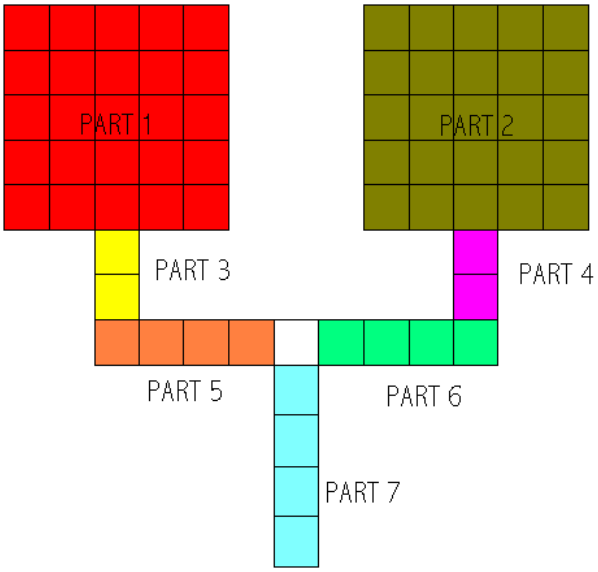
>0 then the tables are defined using DTI entries in the partitioned input data

(should have the same value as was used in the run which created the reduced matrices)

As many sets of matrices as are desired may be read in during one run. The limit is the number of files that may be opened on your computer at one time.

Once steps 1 and 2 are complete, the alters are no longer needed, and the resulting databases should support the standard restart logic of MSC/NASTRAN. If you are using the OTM or if you want boundary forces on the superelements, then the third alter (alter9ia.v70) will be needed in any transient or frequency response restarts.

Sample Problem



The sample model is the “two-headed flyswatter” model used in the MSC/NASTRAN Superelement seminar notes[4]. The model is ‘broken’ into 7 PARTs and a residual structure as shown below. Each PART has a separate input file containing the BULK DATA for it. The model is run using three different approaches for comparison.

First the model is run using standard PARTs with no DMAP. This run is the reference run for comparison.

The second set of runs uses the DMAP alter described in this paper. The first run (run1.dat) processes all of the PARTs and writes the reduced data to a different file for each one. The second run (run2.dat) reads in these files and creates and solves the assembly. In this run, the automatic attachment feature of PARTs is used. The input data for each PART consists only of a parameter, INFILE, which points to the file containing the reduced data for that PART. The only section that is modeled in this file is the residual structure (and if desired, that could have been modeled as a PART and replaced by reduced matrices).

The third set of runs uses the DMAP alter and takes advantage of the fact that there are repeated components. The first run is identical to that in the second set of samples, but the second sample uses only the reduced data for superelements 1, 3, 5, 6, and 7, since superelements 2 and 4 are identical copies of superelements 1 and 3 respectively. The SELOC entry is used in the Main Bulk Data section to ‘relocate’ the copies to the correct locations. In the partitioned sections of the file, the input sections for superelements 1 and 3 are identical, as are the sections for superelements 2 and 4.

Reference run – allparts.dat

```
$ allparts.dat
$
$  Superelement dynamic reduction - using PARTs
$
ID SE, SAMPLE PROBLEM DYNAMICS - SHOW DIFFERENT REDUCTION
METHODS
SOL 103
TIME 15
CEND
TITLE = S.E. SAMPLE PROBLEM 1 - NORMAL MODES
SUBTITLE = S.E. DYNAMICS
DISP = ALL
PARAM,GRDPNT,0
$
SUBCASE 1
SET 12 = 1,2
SUPER = 12
LABEL = MODAL SYNTHESIS FOR S.E. 1 & 2
METHOD = 1
$
SUBCASE 2
SET 13 = 3,4
SUPER = 13
LABEL = CMS FOR S.E. 3 & 4
method = 1
$
SUBCASE 3
SET 15 = 5,6
SUPER = 15
LABEL = CMS for S.E. 5 & 6
method = 1
$
SUBCASE 4
SUPER = 7
LABEL = COMPONENT MODE REDUCTION ONLY FOR S.E. 7
stress = all
spcforce = all
METHOD = 1
$
SUBCASE 5
LABEL = RES STR - SOLVE PROBLEM FOR SYSTEM MODES
METHOD = 2
$
OUTPUT(PLOT)
```

```
$
$   PLOT SYSTEM MODES
$
SEUPPLOT 0
SET 1 = ALL
AXES Z,X,Y
VIEW 0.,0.,0.
FIND SCALE, ORIGIN 1, SET 1
PLOT MODAL DEFORMATION, SET 1, ORIGIN 1
$
AXES X,MZ,Y
FIND SCALE, ORIGIN 1, SET 1
PLOT MODAL DEFORMATION, SET 1, ORIGIN 1
$
BEGIN BULK
$ main bulk data section
$
EIGRL,2,,200.
CQUAD4  5       1       13      14      24      23
GRDSET                                          6
GRID    13              -.4     3.6     0.
GRID    14              .4      3.6     0.
GRID    23              -.4     4.4     0.
GRID    24              .4      4.4     0.
MAT1,1,30.+6,,.3,.283
PSHELL,1,1,.05,1,,1
$
begin super=1
EIGRl,1,,1000.
spoint,10001,thru,10020
qset1,0,10001,thru,10020
include ’loadprt1.dat’
include ’part1.dat’
$
begin super=2
EIGRl,1,,1000.
spoint,10001,thru,10020
qset1,0,10001,thru,10020
include ’loadprt2.dat’
include ’part2.dat’
$
begin super=3
eigrl,1,,,5
spoint,3001,thru,3010
qset1,0,3001,thru,3010
include ’part3.dat’
```

```
$
begin super=4
include ’part4.dat’
eigrl,1,,,5
spoint,4001,thru,4010
qset1,0,4001,thru,4010
$
begin super=5
eigrl,1,,,5
spoint,10001,thru,10020
qset1,0,10001,thru,10020
include ’part5.dat’
$
begin super=6
eigrl,1,,,5
include ’part6.dat’
spoint,10001,thru,10020
qset1,0,10001,thru,10020
$
begin super=7
$ define Q-set
spoint,10001,thru,10020
qset1,0,10001,thru,10020
EIGRL,1,,,5
include ’part7.dat’
ENDDATA
```

The included files (part1.dat…part7.dat) are files containing physical models for each of the components.

Runs using PARTs – set 1 – all PARTs

File run1.dat – process PARTs and store reduced matrices

```
assign output2='out21.out', unit=21, delete
assign output2='out22.out', unit=22, delete
assign output2='out23.out', unit=23, delete
assign output2='out24.out', unit=24, delete
assign output2='out25.out', unit=25, delete
assign output2='out26.out', unit=26, delete
assign output2='out27.out', unit=27, delete
$
$  Superelement dynamic reduction - using PARTs
$
ID SE, SAMPLE PROBLEM DYNAMICS - SHOW DIFFERENT REDUCTION
METHODS
```

```
SOL 103
TIME 15
include ’alter1ha.v70’
CEND
TITLE = S.E. SAMPLE PROBLEM 1 - NORMAL MODES
SUBTITLE = S.E. DYNAMICS - USE DIFFERENT REDUCTION TECHNIQUES
DISP = ALL
PARAM,GRDPNT,0
$
param,fixedb,-1
$
SUBCASE 1
.
.
. same as allparts.dat
.
.
BEGIN BULK
$ main bulk data section
$
EIGRL,2,,200.
CQUAD4  5       1       13      14      24      23
GRDSET                                          6
GRID    13              -.4     3.6     0.
GRID    14              .4      3.6     0.
GRID    23              -.4     4.4     0.
GRID    24              .4      4.4     0.
MAT1,1,30.+6,,.3,.283
PSHELL,1,1,.05,1,,1
$
begin super=1
param,matfile,21
param,drmh,1
EIGRl,1,,1000.
spoint,10001,thru,10020
qset1,0,10001,thru,10020
include ’loadprt1.dat’
include ’part1.dat’
$
begin super=2
param,matfile,22
param,drmh,1
EIGRl,1,,1000.
spoint,10001,thru,10020
qset1,0,10001,thru,10020
include ’loadprt2.dat’
include ’part2.dat’
$
begin super=3
param,matfile,23
eigrl,1,,,5
spoint,3001,thru,3010
qset1,0,3001,thru,3010
param,drmh,1
include ’part3.dat’
$
begin super=4
include ’part4.dat’
param,matfile,24
eigrl,1,,,5
spoint,4001,thru,4010
qset1,0,4001,thru,4010
param,drmh,1
$
begin super=5
param,matfile,25
param,drmh,1
eigrl,1,,,5
spoint,10001,thru,10020
qset1,0,10001,thru,10020
include ’part5.dat’
$
begin super=6
param,matfile,26
param,drmh,1
eigrl,1,,,5
include ’part6.dat’
spoint,10001,thru,10020
qset1,0,10001,thru,10020
$
begin super=7
param,matfile,27
$param,neutral,1
$ define Q-set
spoint,10001,thru,10020
qset1,0,10001,thru,10020
EIGRL,1,,,5
include ’part7.dat’
uset1,u2,123,3,4,5,6
param,drmh,1
ENDDATA
```

The differences between this file and allparts.dat are shown in bold. Superelement 7 uses the option of the alter to write a “reduced” transformation matrix. In this way, the size of the files and databases are reduced. This limits data recovery to the selected points (unless the OTM are used), and if standard data recovery is used, it is possible that the results may be incorrect (any dof not listed in the U2 set are set to 0.0 during data recovery).

The files created (out21.dat through out27.dat) contain the reduced matrices for the superelements.

File run2.dat – read reduced matrices for the PARTs and store in the database

```
assign inputt2=’out21.out’, unit=21
assign inputt2=’out22.out’, unit=22
assign inputt2=’out23.out’, unit=23
assign inputt2=’out24.out’, unit=24
assign inputt2=’out25.out’, unit=25
assign inputt2=’out26.out’, unit=26
assign inputt2=’out27.out’, unit=27
$
$ bring in PARTs from previous run
$
ID SE, SAMPLE PROBLEM DYNAMICS - SHOW DIFFERENT REDUCTION
METHODS
SOL 103
TIME 15
include ’alter2ha.v70’ $ alter to allow reading in matrices and data
CEND
TITLE = S.E. SAMPLE PROBLEM 1 - NORMAL MODES
.
. CASE CONTROL identical to run1.dat
.
.
BEGIN BULK
$ main bulk data section
$
param,wtmass,.00259
EIGRL,2,,200.
CQUAD4  5       1       13      14      24      23
GRDSET                                          6
GRID    13              -.4     3.6     0.
GRID    14              .4      3.6     0.
GRID    23              -.4     4.4     0.
GRID    24              .4      4.4     0.
MAT1,1,30.+6,,.3,.283
PSHELL,1,1,.05,1,,1
$
```

```
begin super=1
param,infile,21
$
begin super=2
param,infile,22
$
begin super=3
param,infile,23
$
begin super=4
param,infile,24
$
begin super=5
param,infile,25
$
begin super=6
param,infile,26
$
begin super=7
param,infile,27
ENDDATA
```

Notice that the partitioned input sections contain only the unit number for the files containing the reduced matrices. This is all that is needed when using the alter, as the alter provides for automatic attachment using the standard logic of the program.

File run2image.dat – read reduced matrices for the PARTs and store in the database. Superelements 2 and 4 use data from superelements 1 and 3

```
assign inputt2='out21.out', unit=21
assign inputt2='out23.out', unit=23
assign inputt2='out25.out', unit=25
assign inputt2='out26.out', unit=26
assign inputt2='out27.out', unit=27
$ superlements 2,4, and 6 are copies of 1,3, and 5 - use SELOC to
$  re-position them properly
$
ID SE, SAMPLE PROBLEM
$
$  SUPERLEMENT SAMPLE USING IMAGE S.E. FOR 2, 4, AND 6
$   PARTITIONED BULK DATA EXTERNAL SUPERELEMENTS
$
SOL 103
include ’alter2ha.v70’
CEND
TITLE = S.E. SAMPLE PROBLEM 2 - USING IMAGE S.E.
```

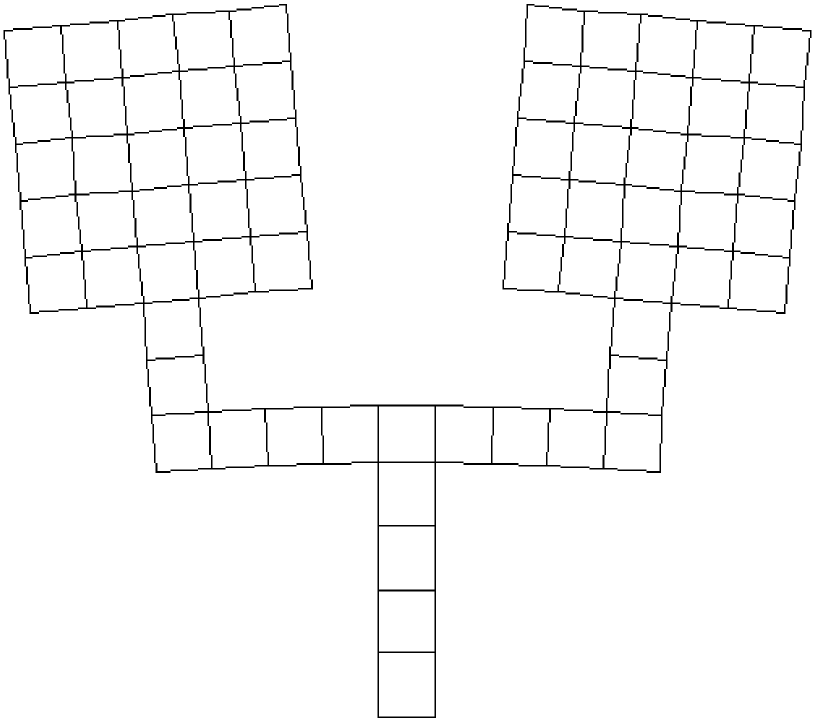
```
.
. CASE CONTROL identical to run1.dat
.
BEGIN BULK
eigrl,2,,,5
$
$ DEFINE IMAGE S.E. - USE REPEAT IMAGE S.E. FOR S.E. 2 & 4
$
seloc,2,33,38,98,233,238,298
point,233,,1.2,6.,0.
point,238,,5.2,6.,0.
point,298,,5.2,10.,0.
$
seloc,4,19,20,36,319,320,336
point,319,,2.8,4.4,0.
point,320,,3.6,4.4,0.
point,336,,3.6,6.,0.
$
INCLUDE ’part0.dat’
$
begin super = 1
param,infile,21
$
begin super=2
param,infile,21
$
begin super = 3
param,infile,23
$
begin super=4
param,infile,23
$
begin super = 5
param,infile,25
$
begin super=6
param,infile,26
$
begin super = 7
param,infile,27
$
ENDDATA
```

Superelements 2 and 4 use the identical data to superelements 1 and 3. The SELOC entries handle the translation of the copies.

Results

The three samples provide identical results. One feature of using the alters is that assembly plots can be generated for the assembly, including stress contours for the PARTs coming from external sources.

The following plot, showing mode 9, is identical for all three sets of runs.



Limitations

This alter has not been extensively tested. The following Known limitations exist:

1) If cms is used, the QSET must be defined in the partitioned bulk data section for the superelement using SPOINTs.
2) Currently there is no testing performed by the alters to verify that the connection locations found during the second step match up with those used in the first step. If they are not identical, incorrect answers will occur.
3) The alters have not been extensively tested in a production environment, and, as such, may contain errors. If you do find an error in the alters, contact me and I will attempt to correct it.

Summary and Conclusions

A set of DMAP alters is developed which allows groups to create reduced models, pass the reduced data to other groups (or companies), automatically create assembly models using the reduced matrices, and perform assembly analysis (complete with assembly plots). The resulting files allow standard restarts, including model changes (a copy of the BULK DATA for each PART is included in the reduced file). If a restart is done with a model change, components may be processed completely (to account for changes), resulting in correct results for the changed model. If this occurs, then the affected superelements will become “standard” PARTs and will be treated accordingly by MSC/NASTRAN.

References

1. MSC/NASTRAN Release Guide Version 69, The MacNeal-Schwendler Corporation, Los Angeles, Ca, 1996
2. Ted Rose, “Creation and use of Craig-Bampton Models Using MSC/NASTRAN”, Procedings of the 1991 MSC World Users’ Conference, The MacNeal-Schwendler Corporation, Los Angeles, Ca., 1991
3. MSC/NASTRAN Common Questions and Answers, Second Edition, John Lee (editor), The MacNeal-Schwendler Corporation, Los Angeles, Ca., 1993
4. “MSC/NASTRAN Superelement Seminar Notes”, The MacNeal-Schwendler Corporation, Los Angeles, Ca., Feb., 1997