

The New External Superelements in MSC/NASTRAN and a DMAP Alter to Create and Use OTM

by

Ted Rose

Manager, MSC/NASTRAN Training and Support

The MacNeal-Schwendler Corporation

Presented at the 1999 MSC Aerospace Conference

Abstract:

It often seems to be several years after the introduction of a new feature before it gains recognition among the user community. This paper is intended to provide an introduction to the new external superelement features introduced in V69 and also provide a DMAP alter which allows the creation and usage of OTM for them, allowing the creation and usage of "deliverable" models.

Introduction:

The idea of external superelements¹ is an approach used to simplify the model you are working with when you have a complicated structure and it would be inconvenient (high cpu and/or disk usage) to work with the entire model. When using external superelements, individual component models are processed independently and the reduced matrices are saved for use in the 'assembly' run. The advantage of this approach is that components, which may not be of interest to you, but may be important components for loading or response) do not have to be processed every time you make a change to the model. Only a small component (or several components), for which the physical model exists in the final assembly model, needs to be processed for changes, creating a very efficient approach to optimization, or nonlinear, or any 'repetitive' analysis where changes are occurring.

Other approaches include 'coupled dynamic analysis'², where component models are processed by independent contractors and the reduced matrices along with OTM (Output Transformation Matrices) are passed to a 'system integrator', who creates the assembly model and solves the problem. In the paper referenced, DMAP is provided which allows complete data recovery for the component models, along with loading changes on them at the 'system' level.

This paper shows how to use the new external superelements, along with a DMAP alter (available for V70.5 from MSC field offices or in the sssalter library starting with V71) which creates and uses OTM for the external superelements. In the referenced paper, a series of three DMAP alters were used. In this paper, a single alter, which is used in all runs (external superelement creation, assembly, and data recovery) is presented.

Overview:

In versions prior to V70 of MSC/NASTRAN, there was only one way to define external superelements and that involved the use of the CSUPER entry and manually attaching the matrices to your model. This approach did allow you to bring in matrices from an external program, or a separate MSC/NASTRAN run, but it did not allow you to perform data recovery on the component model used to make the external superelement. Also, getting the matrices into MSC/NASTRAN was often difficult to say the least.

Starting with V69 (fully implemented in V70), there is a new way to define external superelements (you might even say that there are four new ways to define external superelements). This new approach allows you to create the matrices for the external superelement from within MSC/NASTRAN, use these matrices in an assembly run, and then perform data recovery on the external superelement using the original database files for the component.

This approach is documented in the Release Guide for V70¹. However, this approach does not use OTM to allow for data recovery on the external superelements in the 'assembly' run. It only allows data recovery on the original component models using results from the assembly run. Both are valid approaches, this paper shows both methods (including a DMAP alter to create and use OTM) for comparison.

Procedure:

Step 1 - Create reduced matrices

The new external superelements are created by separate runs. Each external superelement is created from the residual structure of the component model (which may consist of a model with superelements). The user interface for this is:

- 1) ASET (or ASET1), BSET (or BSET1), and/or CSET (or CSET1) entries identifying the interface dof (note: it is not necessary to use all 6 dof on the interface GRID points).

- 2) A QSET (or QSET1) entry selecting the dof which will be used to identify the component modes (only if component modal synthesis is being performed) and SPOINTS creating the dof.
- 3) A METHOD request in the CASE CONTROL section and an associated EIGRL entry (only if CMS is being performed).
- 4) PARAM,EXTOUT - set to an appropriate value:
 - a) DMIGDB = store the reduced matrices in the database using DMIG format.
 - b) MATRIXDB = store the matrices in the database (as "plain" matrices with no dof identifiers).
 - c) DMIGOP2 = write the reduced matrices into an OUTPUT2 file (selected by PARAM,EXTOUT,I -where i is the unit number with a default is 30)
 - d) DMIGPCH = write DMIG into the punch file.
- 5) If the DMAP alter is used;
 - a) PARAM,DRMH,YES = instructs the alter to write OTM matrices in the selected form for displacements, stresses, element forces, and spcforces as requested by the CASE CONTROL.
 - b) CASE CONTROL output requests for the information you wish in the OTM.
- 6) The model of the component (as mentioned before, this model may include superelements)

Step 2 - assemble components together

Once the individual components have been processed, an assembly run (or series of runs) is done to put it all together. In this run, you may have the physical model of the residual structure (if one exists) plus the physical models any other component models (as superelements or as a part of the residual structure), which are not being represented by external superelements.

For each of the reduced component models, you need to have the following:

- 1) If EXTOUT was DMIGOP2, MATRIXDB, or DMIGDB, you need to use FMS ASSIGN statements to attach the file containing the reduced matrices for each superelement you are bringing in as an external superelement.

- a) If EXTOUT was DMIGOP2, you need to assign the OUTPUT2 file using a statement similar to the following"

assign inputt2='ext1.op2', unit=31, form=formatted

In this statement, the file"ext1.op" (created in step 1) is attached as unit 31, and the file was written using the "FORMATTED" option.

- b) If EXTOUT was MATRIXDB or DMIGDB, you need to attach the database files from the initial run using FMS similar to the following:

**ASSIGN SE1DB='exse1.MASTER'
ASSIGN DBAL1='exse1.DBALL'
dblocate datablk=(extdb),convert(SEID=1), logical=SE1DB**

In the above, "extse1.MASTER" and "extse1.DBALL" are the database files created in the initial (external superelement creation) run.

- c) If EXTOUT was DMIGPCH, there is no special FMS required, the DMIG matrices from the punch file may be included in the input section of any superelement, along with CASE CONTROL requests K2GG, M2GG, and B2GG pointing to the matrices.

- 2) In the Main Bulk Data section (before the first BEGIN SUPER), you need to have a

SEBULK,id,EXTERNAL

entry (where id is the number of the superelement being brought in.) for each external superelement.

- 3) For each external superelement, you need to create a separate BEGIN SUPER=I (where I is the superelement id being created using the matrices from the first step) section in your input file. In these sections, you need to provide the GRID points, SPOINTS, and attachment information for the external superelement matrices. This is done by:

- a) Enter copies of all GRID points and SPOINTS used as interface points and QSET dof in the original (reduced matrix creation) run (it is not required to use exact copies, but it is recommended to use them, as this makes the process easier with less chance for error). These can all be identified in the printout generated in step 1 by including PARAM,USETPRT,0 and PARAM,USETSEL,128 in the input file. An example of this output is shown below:

```

U S E T   D E F I N I T I O N   T A B L E   (   I N T E R N A L   S E Q U E N C E ,   R O W   S O R T )
          A   D I S P L A C E M E N T   S E T
          -1-   -2-   -3-   -4-   -5-   -6-   -7-   -8-   -9-   -10-
1=   33-1   33-2   33-3   33-4   33-5   33-6   35-1   35-2   35-3   35-4
11=  35-5   35-6   36-1   36-2   36-3   36-4   36-5   36-6   38-1   38-2
21=  38-3   38-4   38-5   38-6   93-1   93-2   93-3   93-4   93-5   93-6
31=  98-1   98-2   98-3   98-4   98-5   98-6   11001-0   11002-0   11003-0   11004-0
41= 11005-0   11006-0   11007-0   11008-0   11009-0

```

- b) You will need to have an EXTRN entry in the partitioned data section. This entry serves different purposes depending on the method used to bring the data in, but always has the same format. As long as you are using copies of the GRID and SPOINTS used to create the external superelement, the EXTRN can be created from the above printout. On the EXTRN, you list all connecting dof in the order they occur in the matrix (the order is not important for the DMIGDB or DMIGOP2 or DMIGPCH options, but it is a good idea have it correct anyway). An example for the above table is:

```

EXTRN, 33,123456, 35,123456, 36,123456, 38,123456,
, 93,123456, 98,123456, 11001,0, 11002,0,
, 11003,0, 11004,0, 11005,0, 11006,0

```

- c) QSET entries identifying the QSET used in the original run (if used)
- d) If you are using DMIGOP2, then you must point to the file containing the data for the superelement. This is done using PARAM,EXTUNIT pointing to the file used as the OUTPUT2 unit in the original (creation run). This requires using FMS to attach the file to the current run and PARAM,EXTUNIT in the partitioned data for the superelement.

For each external superelement (which was created using PARAM,EXTOUT,DMIGOP2) brought in using this method you would have a statement similar to the following:

```

assign inputt2='ext1.op2', unit=31, form=formatted

```

and in the partitioned data section you would have PARAM,EXTUNIT, similar to:

```

param,extunit,31

```

e) If you wish to perform data recovery on the external superelement using the original database, you need to include PARAM,EXTDROUT set to the correct value:

- 1) If you wish to use the database to pass the boundary solution to the component model, set PARAM,EXTDROUT, MATRIXDB (or DMIGDB - I recommend that you use the same value used for EXTOUT in the creation run).

Or:

- 2) If you wish to use an OUTPUT2 file to pass the interface solution back to the component model, set PARAM,EXTOUT,DMIGOP2. Also assign a file to be used for the interface solution using an ASSIGN statement similar to:

```
assign output2='extlout103.op2', unit=41, form=formatted, delete
```

and set PARAM,EXTUNIT,41 (the unit on the ASSIGN statement) in the partitioned input. This will create an OUTPUT2 file (named extlout103.op2 in this example FMS), which contains the boundary solution and other information necessary to perform data recovery on the component.

- f) If you wish to perform data recovery on the component in the solution run, you need to have set PARAM,DRMH,YES and included alteri.v705 in the creation run, and you need to place the desired output requests for this component in the CASE CONTROL section for it, and add 'include alteri.v705' in the EXECUTIVE CONTROL section.

STEP 3 Solve the Assembly Problem

This is a standard MSC/NASTRAN run (and may be done as a part of step 2). Include output requests for the desired output quantities for any superelements which physically exist in the current model, plus any external superelements created using PARAM,DRMH,YES with alteri.v705 in the creation run.

STEP 4 - Data Recovery on The External Superelement

If you did not use PARAM,DRMH,YES and include alteri.v705 in the creation run, data recovery is still available for external superelements. It is performed using the original database (the one created when generating the reduced matrices) and either the database from the assembly run (if PARAM,EXTOUT was set to MATRIXDB or DMIGDB) or the OUTPUT2 file created in the assembly run (if PARAM,EXTOUT was set to DMIGOP2). It is done by performing a RESTART run using the creation database and the results data at the interface.

This run will look like a typical restart run for data recovery, with the addition of statements informing MSC/NASTRAN where to find the boundary solution.

In the FMS, you will need to ASSIGN the original MASTER database and inform the program that you are performing a RESTART:

```
ASSIGN EXT10='run1_se10.MASTER'  
RESTART, LOGI=EXT10
```

The above FMS statements instruct MSC/NASTRAN to attach the database files from the run named run1_se10 and perform a read-only restart (not necessary, a regular restart is also fine - this is just a way to avoid any possible contamination of the original database if a mistake is made).

Always set **PARAM,EXTDR,YES** in the file to perform data recovery on an external superelement using the boundary solution.

- a) If you used **PARAM,EXTDROUT,MATRIXDB** or **DMIGDB** in the assembly run, **FMS** similar to the following must be added:

```
ASSIGN SYSTEM='run2_se10in.MASTER'  
DBLOCATE DB=(EXTDB), WHERE(SEID=10),  
LOGI=SYSTEM
```

In the above, the assembly database is attached as the logical file set, "SYSTEM" (the word "SYSTEM" can be any non-reserved name you wish) is attached and the program is instructed to locate the boundary solution for superelement 10 (in the system run) and use it as the boundary solution for the current component. In addition to this, you need to specify **PARAM,EXTDR,YES** as mentioned earlier.

Or;

- b) If you used **PARAM,EXTOUT,DMIGDB** in the assembly run, use **FMS** to attach the **OUTPUT2** file created and specify **PARAM,EXTDRUNIT,xx**, pointing to the file. For this, the **FMS** statement would be similar to:

```
assign inputt2='se10bdry.op2', unit=31
```

and the following parameter would be in the input file (along with **PARAM,EXTDR,YES**);

```
param,extdrunt,31
```

The rest of the input file is a normal data recovery file. Be careful to be sure that the **SOL** number is correct (that it matches the **SOL** used in the assembly run).

Sample Problem

The sample model is the "two-headed flyswatter" model used in the **MSC/NASTRAN Superelement seminar notes**². For demonstration purposes, I am calculating the assembly modes in **SOL 103**, and performing a restart to perform a modal transient solution using the modes from **SOL 03**. The model is divided into 7 superelements and a residual structure as shown below in figure 1.

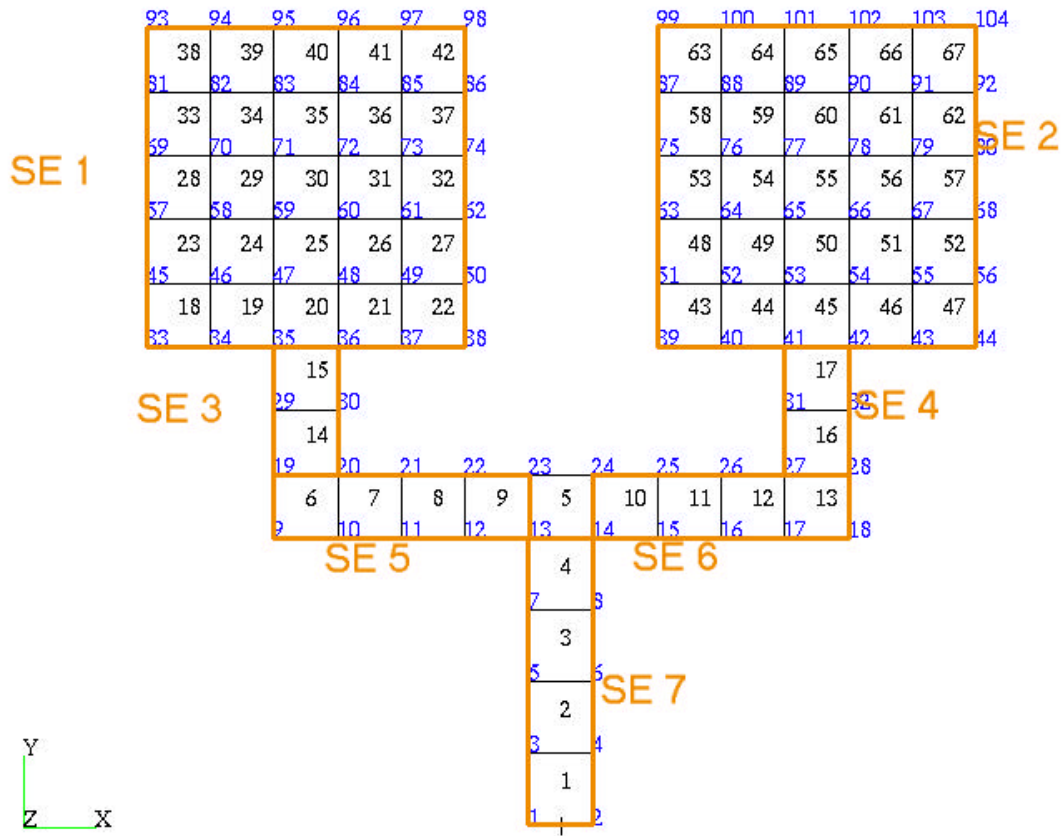


Figure 1 - the model and superelement definitions.

Separate models are created for each superelement and each one is processed to create reduced matrices, which will be used as an external superelement in the assembly run. In this example, I used PARAM,EXTOUT,DMIGOP2 for all of the components. This is not a requirement, in fact, different methods could be used for each component. I simply used a single approach for clarity. I suspect that this will be the most commonly used approach, in that there is no need to transfer database files or to be sure that the matrix terms are in the correct order. Also, when using the OUTPUT2 options, the files can be written using the neutral format, and easily transferred across computer types.

Runs 1-7 - create reduced matrices for each component

A separate model was created for each component and reduced to boundary matrices. For purposes of clarity, each model was reduced to the interface dof with the rest of the model, plus selected GRID points, which were on the corners of the component, so that PLOTTEL elements could be used to provide an outline in the final assembly. Without these additional GRID points, there would only be the single element (element 5) and the interface points in the residual structure, which would make it hard to visualize what is happening in plots.

I believe that by including these additional points, the results are easier to visualize and confirm. (Note that only the interface points are needed)

The following input file (runpart1.dat) is an example of this with the appropriate input items in bold.

```

assign output2='ext1.op2', unit=30, delete
id runpart1.dat $
SOL 103
include 'alteria.v705'
CEND
TITLE = 1999 Aero Conference example problem -      runpart1.dat
SUBTITLE = Create external superelement model - reduced matrices
DISP = ALL
$
PARAM,WTMASS,.00259
loadset = 99
subcase 1
method = 1
stress = all
$
$
BEGIN BULK
param,resvec,yes
eigr1,1,,,6
lseq,99,1001,101
lseq,99,1002,201
lseq,99,1003,301
$
$ main bulk data section
$
param,extout,dmigop2
$ create OTM
$
param,drmh,yes
$
$ define modal coordinates for CMS
$
$ define q-set for component modes and residual vectors
SPOINT 11001 THRU 11009
QSET1 0 11001 THRU 11009
$
$ define which dofs will be retained (i.e. which dofs will form the
$ attachment to the system model)
$
ASET1 123456 93 98 33 35 36 38
$
$ print dof map for connecting the external superelement to the
$ system model, with EXTRN entry.
$ usetsel=128 will print only ASET dof
$
PARAM USETPRT 0
PARAM USETSEL 128
$
include 'loadprt1.dat'
include 'part1.dat'
$
enddata

```

In the above file, the "ASSIGN" statement is creating a file (ext1.op2) which will be used to store the reduced matrices and OTM.

The "include" statement is inserting the dmap alter which will create OTM for the items requested in the CASE CONTROL (STRESSes and DISplacements).

The METHOD command selects EIGRL number 1, which requests the first 6 modes.

PARAM,EXTOUT,DMIGOP2 is the controlling parameter for the creation of an external superelement. In this case we wish to write the reduced matrices (and OTM) into an OUTPUT2 file using DMIG (and DMI) format.

PARAM,DRMH,YES instructs the alter to create the OTM.

The SPOINT and QSET1 entries define the dof which will be used to represent the component modes.

The ASET1 entry selects the "boundary" dof. In this case, only GRID points 35 and 36 will be used for attachment, GRID points 33,38,93, and 98 are for graphical display (to outline the component in the assembly model by using PLOTELS). Any dof selected on ASET1 (ASET, BSET, or BSET1) entries will be held constrained when the component modes are calculated. Any dof selected on CSET (or CSET1) entries will be in the reduced matrices, but will not be constrained when the component modes are calculated.

PARAM,USETPRT and USETSEL are used in this model to print the A-set dof. This is a good idea so that you have a record of the dof in the reduced matrices.

Run 8 - Create assembly model

In this run, the physical model of the residual structure, plus the reduced models are combined into an assembly for the calculation of the assembly modes.

The component models are combined in a single file (readthem.dat). Excerpts from this file are shown below:

```
$ file readthem - read in external se and find system modes
$
$ attach files with reduced matrices
$
assign inputt2='ext1.op2', unit=31
assign inputt2='ext2.op2', unit=32
assign inputt2='ext3.op2', unit=33
assign inputt2='ext4.op2', unit=34
assign inputt2='ext5.op2', unit=35
assign inputt2='ext6.op2', unit=36
assign inputt2='ext7.op2', unit=37
$
$ create files for boundary solution - if wish detailed data recovery
$
assign output2='ext1out103.op2', unit=41, delete
assign output2='ext2out103.op2', unit=42, delete
assign output2='ext3out103.op2', unit=43, delete
assign output2='ext4out103.op2', unit=44, delete
assign output2='ext5out103.op2', unit=45, delete
assign output2='ext6out103.op2', unit=46, delete
assign output2='ext7out103.op2', unit=47, delete
$
SOL 103
include 'alteria.v705'
```

```

CEND
TITLE = S.E. SAMPLE PROBLEM                                readthem.dat
SUBTITLE = External superelements - read in matrices
DISP = ALL
$
$ set defaults for all se
$
PARAM,GRDPNT,0
PARAM,WTMASS,.00259
loadset = 99
subcase 1
spc = 10
method = 1
stress = all
$
output(plot)
$
$ plots
$
seupplot 0
set 1 = all
ptitle = full structure
axes x,mz,y
find scale, origin 1, set 1
plot set 1 label both
plot modal deformation set 1 origin 1
$
BEGIN BULK
$ residual structure model
include 'part0.dat'
eigr1,1,,5
lseq,99,1001,101
lseq,99,1002,201
lseq,99,1003,301
spc1,10,123456,1,2
$
$ provide a SEBULK entry for each external se
$
SEBULK   1      EXTERNAL
SEBULK   2      EXTERNAL
SEBULK   3      EXTERNAL
SEBULK   4      EXTERNAL
SEBULK   5      EXTERNAL
SEBULK   6      EXTERNAL
SEBULK   7      EXTERNAL
$
GRID     1          -0.4    0.    0.
$...GRID entries coincident with boundary points for the external
$ superelements
GRID    104          5.2    10.    0.
$
$
$ PLOTTEL'S TO OUTLINE COMPONENTS
$
plotel,101,33,35
.
$ plotel elemements for outline of assembly

```

```

.
plotel,704,14,2
$
begin super = 1
param,extunit,31
$
$ set flag for data recovery
$
PARAM EXTDROUTDMIGOP2
param,extdrunt,41
lseq,99,1001,101
lseq,99,1002,201
lseq,99,1003,301
$
$ conection points = exterior points from initial run
$
$ based on the ASET entries in the initial run for this component
$
$ ASET1      123456  93      98      33      35      36      38
$
GRID      33          -5.2    6.      0.
GRID      35          -3.6    6.      0.
GRID      36          -2.8    6.      0.
GRID      38          -1.2    6.      0.
GRID      93          -5.2   10.     0.
GRID      98          -1.2   10.     0.
SPOINT    11001    THRU    11006
QSET1     0      11001    THRU    11006
$
$ extrn entry is mandatory for external superelements
EXTRN,33,123456,35,123456,36,123456,38,123456,
,93,123456,98,123456,11001,0,11002,0,
,11003,0,11004,0,11005,0,11006,0
,11007,0,11008,0,11009,0
$
begin super = 2
.
. data for external superelements 2-7
.
enddata

```

In the above excerpts from the file, the input which is required to bring in the external superelements is shown in **BOLD**. In selected parts, the input for all external superelements is shown, as in the FMS, where the ASSIGN statements are shown for all of the external superelements. Sometimes the input for only one superelement is shown, as in the partitioned input section for superelement 1. The sections for the others are similar and would take up excessive space to show, so they are not shown.

Since the external superelements were created using EXTOUT=DMIGOP2, the OUTPUT2 files need to be attached to the run. That is the purpose of the "ASSIGN" statements attaching units 31-37. The files attached by these statements are the ones created in the "creation" runs, (runs 1-7).

The additional "ASSIGN" statements are optional - they are to create files for the boundary solution for each external superelement. These files can be sent back to the person who created the reduced matrices and can be used to perform data recovery for the components (shown in run9). These files will be used as units 41-47 and each will contain the boundary solution information for a different component. Which information is written in each of these is controlled by PARAM,EXTDRUNT,ii (where "ii" is the unit number for the file used). And PARAM EXTDROUT, which selects the format of the data written to the

file. In this case, EXTDROUT is set to DMIGOP2, so the boundary solutions will be written to the OUTPUT2 files using DMIG format.

The "include alteria.v705" statement is used to add the alter referenced in this paper. When this alter is included and the OTM were created in the runs which created the reduced matrices, the program will honor output requests for the external superelements in this run as if the models were available (with the exception of plot requests, as the geometry is not available). Any results, which are contained in the OTM (requested in the creation run) and are requested in the current case control will be printed in the output file. The case control has no SUPER command, so it applies to all superelements.(expanded case control with different output requests for each superelement can be used if desired). Requests are made for displacements and stresses. If the alter were not included, results for only the points and element, which belong to the residual structure would be printed. Since the alter is included, displacements will be printed for all points which exist in the residual structure, plus any in the OTM, and stresses will be printed for the elements in the residual structure, plus any which exist in the OTM. The plot requests are for an undeformed plot of the model (recommended for checking), shown in figure 1, and deformed plots of the mode shapes (a selected one shown in figure 2). These plots are the reason for keeping the corner points on each component as exterior points in this model. By doing this, PLOTTEL elements can be used to draw an outline of the model in the assembly run, so that the assembly modes may be viewed. If the corner points were not included, only the interface points would be available for plotting and the plots would not be nearly as informative.

The bulk data is divided into the Main Bulk Data section and the BEGIN SUPER sections, defining the external superelements (note: Conventional PART and Main Bulk Data superelements may also be defined in the input in addition to the external superelements. This model contains only external superelements and a residual structure). I

In the Main Bulk Data section, there is the model for the residual structure. This section contains the physical model of the residual structure, plus (for this demonstration problem) the interface and corner points for the external superelements, along with PLOTTEL elements to provide an outline plot of the assembly.

The physical model of the residual structure is brought in by the "include part0.dat" command. It consists of a single CQUAD4 entry with its associated GRID points and properties.

The additional GRID points in the Main Bulk Data are copies of the exterior points of the external superelements (once again, this is not a requirement, it is done only for the plots in this model) and PLOTTELS connecting them. Figure 1 shows the assembly model, with a single CQUAD4 element and the additional GRID points connected by PLOTTELS.

There is a SEBULK entry for each external superelement. On this entry, you simply provide the superelement id and the word EXTERNAL, indicating that it is an external superelement.

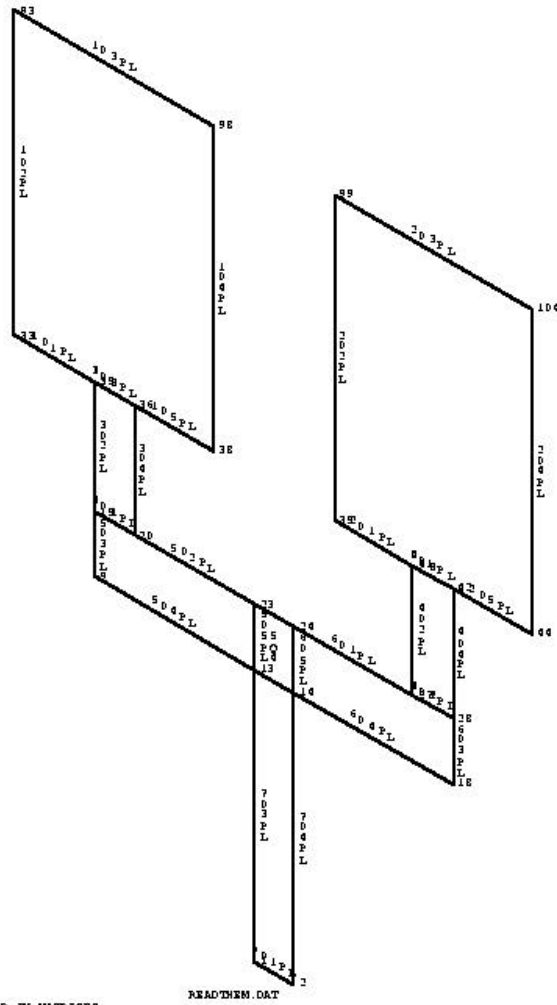
In the partitioned superelement input (beginning with BEGIN SUPER=1) we have the information for each of the external superelements.

Since the input matrices are on an OUTPUT2 (INPUTT2) file from the initial run, PARAM,EXTUNIT must be set to the proper value for the file (in the case of superelement 1, the file is attached as unit 31).

In this section, we have copies of the GRID points and SPOINTs which were used as exterior points in the original component models (the Ids and relative geometry should be identical to be sure of proper attachment and answers - NOTE there is no checking of geometry of coordinate systems for external superelements, so it is best to copy the GRID, SPOINT, and COORDi entries from the original model file) and the QSET entry indicating the SPOINTs used to represent the component modes.

The EXTRN entry is mandatory for external superelements. It should be a list of the A-set from the component creation run, in the correct order (NOTE: The order is not so important when DMIGPCH, DMIGDB, or DMIGOP2 is used, as the matrices contain the dof identifiers, but when MATRIXDB is used, the order on the EXTRN entry indicates the order of the terms in the matrices and is very important).

PARAM,EXTDROUT (set to DMIGOP2 for superelement 1) and PARAM,EXTDRUNIT (set to 41 for superelement 1) are to request that the boundary solution be written to a file which can be used with the database from the creation run for superelement 1 to perform stand-alone data recovery using the assembly solution. This file (ext1out103.op2) can be sent back to the people who created the component model for data recovery without the assembly model being required.



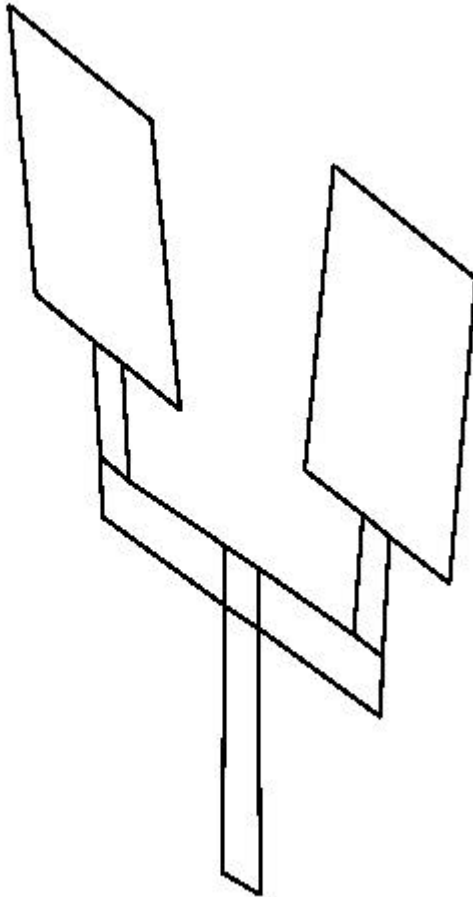
S.E. SAMPLE PROBLEM
EXTERNAL SUPPLEMENT - READ IN MATRICES

READINM.DAT

Figure 1 - undeformed plot of the assembly model showing the PLOT elements connecting the points

Figure 2 shows an example of the plots created by this run for the modes. In this figure, mode 3 is shown. Note that if the additional GRID points had not been retained, this plot would show only the single CQUAD4 element.

```
4.25.99      MAX-COR. = 00.3030300
FULL STRUCTURE
```



```
S. E. SAMPLE PROBLEM          DEFORMED.DAT
RESTARTED COMPONENTS - READ IN MATRICES
MODEL DEFORM. EXERCISE 3  MODE 3  0000. 31.70000
```

Figure 2 - Deformed plot of assembly model - mode 3

The final input file is one which shows how to perform data recovery on a component model using the boundary solution file created by this run.

Run 8 - Restart to Perform a Modal Transient Solution

This file is included to show how to perform a restart using external superelements. The original system solution was done in SOL 103 and found the modes of the assembly. Now we are going to apply the loadings in a modal transient restart. The following file performs this operation.

```
$ file restart.dat = restart from normal modes to perform modal
transient
$
```

```

$ attach master database files
$
assign oldrun='readthem.MASTER'
restart, logical=oldrun
$
$ create files for boundary solution - if wish detailed data recovery
$
assign output2='ext1out112.op2', unit=41, delete
assign output2='ext2out112.op2', unit=42, delete
assign output2='ext3out112.op2', unit=43, delete
assign output2='ext4out112.op2', unit=44, delete
assign output2='ext5out112.op2', unit=45, delete
assign output2='ext6out112.op2', unit=46, delete
assign output2='ext7out112.op2', unit=47, delete
$
$dbdict
dbdict datablk=(extdb),
      select=(name,size,xname,seid,database,vers) $
id restart.dat $
SOL 112
diag 8,15,56
include 'alteria.v705'
$ correction for error 35206....
compile phase0
alter 'INVOKE ALL RESTART DELETIONS HERE',''
if(setype<>'EXTRNA')RESTART ,,,HISTS/TLIST/TRUE $
CEND
TITLE = S.E. SAMPLE PROBLEM                      readthem.dat
SUBTITLE = External superelements - read in matrices
DISP = ALL
$
$ set defaults for all se
$
param,post,-1
PARAM,GRDPNT,0
PARAM,WTMASS,.00259
loadset = 99
subcase 1
spc = 10
sdamp = 12
dload = 15
tstep = 20
method = 1
set 999 = 43,67
stress = all
$
$ PLOT RESULTS
$
OUTPUT(XYPLOT)
$SEPLOT 1
  XAXIS = YES
  YAXIS = YES
  XGRID = YES
  YGRID = YES
  XTITLE = TIME
  YTITLE =      Z D I S P O F G R I D 9 3
XYPLOT DISP /93(T3)

```

```

$SEXPLOT 2
  XAXIS = YES
  YAXIS = YES
  XGRID = YES
  YGRID = YES
  XTITLE = TIME
  YTITLE =      Z D I S P O F G R I D 1 0 4
XYPLOT DISP /104(T3)
$
BEGIN BULK
$
      SID      DAREA      DELAY      TYPE      T1      T2      F      PHASE
TLOAD2      15      1001      .01      NO(1)      0.      5.      10.      90.
$
      SID      N(1)      DT(1)      NO(1)
TSTEP      20      600      .01      5
tabdmp1,12,crit
,.0,.01,100.,.01,endt
enddata

```

With the exception of the ASSIGN statements in the FMS, this is a standard restart run. The loadings were actually defined in the initial runs of the components. In the initial run for superelement one, three static loadings were defined using the LOADSET-LSEQ method. In fact, residual vectors were created for these loadings to help improve the accuracy. These loads are processed as the component model is processed and are stored along with the reduced matrices. When the reduced matrices are read in, these reduced loads are stored and are available for application in the transient solution.

The results of this run are identical to those from a standard superelement solution for this model and include displacement and stress output for all GRID points and elements in the model (as long as the OTM were created in the initial run for the component).

Since alteria.v705 is included in this run, data recovery using the OTM created earlier is available. This run produces displacements and stress results for all GRID points and elements in the component models, along with the points and elements available in this assembly model.

The ASSIGN statements are providing files to be used by the subcontractors for data recovery. These files will contain the boundary solution plus the time interval information from the assembly transient response.

Run 9 - Data Recovery on the Component Model

In this run, the boundary solution from the assembly run will be used to perform data recovery on the model for superelement 1. The OUTPUT2 file from the previous run (ext1out112.op2) will be used on a restart to perform data recovery.

The input file, which does this, is:

```

assign inputt2='ext1out112.op2', unit=31
ASSIGN EXT1='runpart1.MASTER'
RESTART, LOGI=EXT1
$
$
SOL 112
CEND
TITLE = S.E. SAMPLE PROBLEM 1
SUBTITLE = S.E. data recovery on an external superelement
DISP = ALL
stress = all
$
$ set defaults for all se
$
PARAM,GRDPNT,0
PARAM,WTMASS,.00259

```



```

loadset = 99
subcase 1
method = 1
stress = all
$
$
BEGIN BULK
param,post,-1
param,extdrunt,31
param,extdr,yes
enddata

```

This file is a restart from "runpart1.dat". The first line attaches the file containing the boundary solution for superelement 1 from the assembly run. The next two lines attach the database from the original run (creating the reduced matrices for superelement 1) and performs a "read=only" restart. The rest of the file is a standard data recovery file, with the exception of the addition of PARAM,EXTDR,YES and PARAM,EXTDRUNT,31. These entries instruct the program to perform data recovery using the boundary solution, and which file (unit 31) contains the boundary solution.

The results from this run are standard output for this part and include displacements (the eigenvectors) and stresses. Since PARAM,POST is set to -1, the results are written to an OUTPUT2 file for use in MSC/PATRAN.

Conclusions

The purpose of this paper was a simple one, to demonstrate the new external superelements in MSC/NASTRAN. These external superelements are available in all superelement solutions and allow data recovery by using restarts and the original database files.

In addition, a DMAP alter (available starting with V71 in the sssalter directory or by request from any MSC office) is discussed. This alter creates OTM and allows for data recovery during the assembly run in frequency response or transient response analysis.

References

1. MSC/NASTRAN Superelement Users' Guide, V70.5, The MacNeal-Schwendler Corporation, Los Angeles, Ca, 1999
2. "Creation and Automatic Attachment of Reduced Component Models for Dynamic Analysis", by Ted Rose, presented at the *1997 MSC Aerospace Users' Conference*
3. MSC/NASTRAN Version 70 Release Guide, The MacNeal-Schwendler Corporation, Los Angeles, Ca, 1997