SUPERELEMENT ANALYSIS WITH
A USER DEVELOPED
MSC.Patran INTERFACE

2001-130

Behzad Bigdeli
Sr. Analytical Engineer
Hamilton Sundstrand
M/S 1-3-BC34
One Hamilton Road
Windsor Locks, CT 06096

(860) 654-6316
behzad.bigdeli@hs.utc.com

3rd Worldwide Aerospace Conference

April 2002

ABSTRACT

A superelement tool has been developed using MSC.Patran COMMAND LANGUAGE (PCL). This interface allows the user to readily create a MSC.Nastran input deck necessary for any superelement analysis. The capability includes a direct superelement export utility as well as an automatic boundary nodes’ determination ability. The motivation for the development has been the absence of any superelement utility within MSC.Patran. This tool has been successfully deployed in the superelement analysis of a heat exchanger. The paper will introduce the capability, discuss the gained efficiencies in this project, outline the applicability to any large finite element analysis and supply a set of recommendations for future improvements to the superelement tool.
INTRODUCTION

MSC.Nastran superelement analysis method has traditionally been reserved for advanced structural analysts. This is partially due to the absence of any tool to facilitate the process of setting up the input files necessary for the analysis. MSC.Patran currently does not support a partitioned (part) superelement analysis and there is no user-developed utility to assist an analyst in the process.

A tool has been developed, as shown in Appendix A, to offset the present shortcomings associated with the superelement analysis. This interface was developed using PCL and is intended to eliminate two of the challenges facing any analyst who may attempt this approach. Using this tool, one can directly export a selected part superelement to a file as a MSC.Nastran input deck. It can also obtain the boundary nodes between any two superelements and create a file containing the corresponding nodal data as MSC.Nastran GRID cards.

The primary advantage of superelement analysis has been to build finite element models of mating components independently before assembling and analyzing the complete model of the structure. However, there are other ways to take advantage of superelement analysis as well. One of the cases as explored in this study involves a single component that needs to be analyzed multiple times. The repeat analyses are not limited to changes in loading conditions and it can involve re-meshing and other fundamental changes to the model due to a redesign or various cycles of design optimizations. The study of the impact of any design change to a large structure using the conventional finite element analysis would require the decomposition of the stiffness matrix of the entire model for each run which can become very time consuming.

One way to reduce the run time and make the process more efficient is to divide the structure into fixed and variable sections for the purpose of superelement analysis based on the knowledge of the problem in hand. A case in point is the stress analysis studies in which the peak stress location is confined to a section of a very large structure. It is possible to consider the structure as a partitioned (part) superelement that remains the same through various design changes and a residual structure that is the site of the highest stress and will be redesigned. This is indeed the method that has been applied to the heat exchanger problem described below.

This paper will describe the technique, the application of the superelement interface tool that was developed for MSC.Patran as part of this process and the preservation of resources accomplished in this project. The reader has been spared the details of a heat exchanger design as well as the results of various design iterations and the final solutions. This is done in the interest of time and the lack of bearing on the gist of the discussion.
DISCUSSION

The design analysis of a heat exchanger is an excellent example of a case that can be made for using the superelement method and the developed superelement tool.

This component of the environmental control system is made of Aluminum and consists of two major parts, a core and four headers. The core is primarily made of fins and is the site of heat transfer between the hot bleed air from the aircraft engine and the cold ram air from the atmosphere. The header encases the core and provides passageways for the cold, hot and the return airflow. The finite element model of this heat exchanger is depicted in Figure 1.

![Finite Element Model of a Heat Exchanger](image)

Figure 1. Finite Element Model of a Heat Exchanger

This structure is subjected to thermal, vibration and pressure loads. In the course of structural analysis a feasibility study is routinely conducted to evaluate a new design under these loading conditions. The goal is to be able to propose a design that would optimize the performance of a heat exchanger.
The analytic efforts began by building a conventional finite element model of the structure. The headers were modeled by shell elements while each of the layers of the core was represented by solid elements that were sandwiched between two layers of shell elements. The outcome was a 225000 elements’ (130520 nodes) model that would require about 14 hours to perform a single load case, static analysis (MSC.Nastran solution sequence 101), on a Silicon Graphics model R10000/400 machine using MSC.Nastran version 70.7.

Judging from the anticipated number of loading cases and possible redesign cycles the duration of the analysis was far from optimized. Therefore, the superelement analysis was considered as a viable option. The approach was to include the entire structure except the area of interest in a single part superelement and assign the remainder to the residual structure. Figure 2 depicts the portion of the model that constitutes the end-plate structure which will form the residual structure.

![Figure 2. End-Plate Structure](image)

The developed superelement tool as described in Appendix A was used to export the residual structure and the superelement for this problem. A portion of the input deck for this run is shown below:
The original superelement run was conducted without applying any load to the structure. This run would provide the .DBALL and .MASTER files necessary for the subsequent restart runs which would include re-meshing and application of loads to the partitioned structure. This run under identical conditions as stated for the conventional model took about 24 hours. At first glance such a run time is less efficient than the previously mentioned 14 hours for the conventional analysis. However, the conceived efficiencies became apparent in the subsequent runs. For example, the restart run of the superelement model to compute the stresses due to a single mechanical point load, applied to one of the residual structure’s grids took only approximately 10 minutes.

The first proposed improvement to the design of the heat exchanger was to eliminate some of the welds in the headers. The changes as shown in figure 3 were limited to the highlighted portion of the end-plate, leaving the other superelement intact. Thus, the restart run would have to replace at least the portion of the residual structure’s mesh that was affected by the changes. This could be accomplished by deleting the affected grids and elements and including the new ones, which could be a tedious task for a large model. However, since the residual structure is processed at its entirety in each superelement run one can replace the entire residual structure without any adverse impact on the restart run time. This is especially advantageous since one can easily export a new residual structure from a MSC.Patran database (Appendix A). The following is a section of the restart input deck for this case:

... 
BEGIN BULK

\/,1,7588
INCLUDE 'redsgn_residual_SE0.dat'

BEGIN SUPER = 1
ENDDATA
Unfortunately, it turned out that MSC.Nastran reprocesses the superelement and the residual structure for such a run which results in no savings despite the expectations. Further investigation indicated that any time a GRID card appears as part of a restart of a superelement run, that superelement is reprocessed as is the case above. This is true even if the specified grid is not a boundary grid. Thus, the only time that the above approach is advantageous is in a problem where re-meshing constitutes a change of elements or properties other than the GRID card’s information.

An alternative way to overcome the above mentioned problem is to break the model in three parts: a residual structure that contains the boundary grids only, a superelement part that remains the same through the various design cycles and another superelement which has the variable part of the model. The “Get Boundary Grid” (Appendix A) feature of the developed superelement tool can be used to obtain the boundary grids separating the superelements and storing them in a file. For example, the initial run of the heat exchanger problem can be set up as follows:

```
SOL 101
CEND
SEALL = ALL
SUPER = ALL
ECHO = SORT
...
SUBCASE 1
   SPC = 2
...
```
The following script shows a portion of the restart run for this case which replaces superelement 1 (figure 2) with the re-meshed part (figure 3). There were no changes to the case control during the restart and both the initial and restart runs were conducted without application of any loads. This was intentional as the goal is to create a master set of databases that could be rerun for any future loading condition.

The 4-minute duration of this run was much shorter than the 24-hour initial superelement analysis time and the recorded 14 hours conventional analysis duration for the same model. This was a reasonable comparison of the two runs with different end-plates, as the number of elements during re-meshing of the end-plate did not change significantly compared to the original design. This was certainly a significant improvement in the time necessary to process such a large model.

The method was further revised by simply assuming that the initial superelement run could be conducted without any residual structure at all. After all, if the redesign cycle would mean that a number of different residual structures should be analyzed in conjunction with the same superelement it would only be enough to analyze the same superelement once. This idea for this case would mean a superelement run with one superelement and no residual structure at all. Unfortunately, such a MSC.Nastran run could not be completed, as the code would require at least the inclusion of the boundary grids in place of the residual structure. The superelement tool (Appendix A) was employed once more to obtain the boundary grids between the superelement and the modified residual structure. A section of the input deck for this case is:

```plaintext
SET 1 = 2
SEALL = 1
BEGIN BULK
```
INCLUDE 'boundary_grids.dat'

BEGIN SUPER = 1
INCLUDE 'boundary_grids.dat'

BEGIN SUPER = 2
INCLUDE 'superelement_SE1.dat'

END DATA

Note that the SEALL forces the superelement generation and assembly to take place for the superelement (SUPER = 2) only. Failure to specify this will result in a fatal message stating that “MATRIX XAAV FOR UPSTREAM SUPERELEMENT 1 DOES NOT EXIST..” as the MSC.Nastran default (SEALL = ALL) is assumed.

It is also worth noting that if the boundary grids are available in a MSC.Patran database as a group of nodes only one can attempt to use the “Analysis/Current Group/Model Only” feature to write out the nodal information as a MSC.Nastran input deck. However, MSC.Nastran will issue a warning message for grid’s coordinates that are longer than 8 characters. This can potentially slow down the process in the analysis of a large problem. Using the superelement tool’s “Get Boundary Grid” (Appendix A) feature can curtail this problem as well as providing an efficient way to determine the boundary grids separating the superelements.

CONCLUSION

The developed superelement tool presented here certainly makes the analysis process more efficient as it allows a less advanced user to attempt an approach that would otherwise be beyond his/her reach. It would also help an advanced user to perform the analysis much faster and with less possibility of errors.

The potential savings over the long run can be enormous, in particular if one considers:

1. The elimination of the need for extensive training of the analysts,

2. The increase in the number of users that apply the method due to the reduction in the complexity of the approach,

3. The reduction in possibility of errors by all users, and

4. The applicability of the superelement tool and the method to similar projects in the future.

The following table provides a summary of the efficiencies that have been gained in this particular project based on the above discussion. This data certainly can serve as a scale
to measure the potential gain and improvements for similar superelement analysis of
large structures.

<table>
<thead>
<tr>
<th>RUN TYPE</th>
<th>DURATION (HR)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONVENTIONAL</td>
<td>14</td>
<td>NO SUPERELEMENT</td>
</tr>
<tr>
<td>INITIAL SE RUN 1</td>
<td>24</td>
<td>PART SUPERELEMENT</td>
</tr>
<tr>
<td>RESTART RUN 1</td>
<td>0.17</td>
<td>POINT LOAD</td>
</tr>
<tr>
<td>REDESIGN RESTART</td>
<td>0.07</td>
<td>END-PLATE CHANGE</td>
</tr>
</tbody>
</table>

RECOMMENDATION

The superelement tool outlined here assumes that all nodes were created in MSC.Patran. This avoids any problem with the handling of the reference coordinate systems by MSC.Patran. However, if nodes are imported from an existing MSC.Nastran input deck then a conversion of coordinate system is necessary before the nodal coordinates can be written out correctly. This can be a valuable improvement to the existing tool which the author intends to make soon.

Ultimately the hope is that MSC.Software Corporation would make the process fully automated and accessible to all users. The ideal tool would allow the users to set up superelement analysis without any need for editing the input decks.

ACKNOWLEDGEMENT

I greatly appreciate the help that I have received from Mr. Jim Leedom, Larry Pearce, Lance Proctor and Joe Baldwin of MSC.Software Corporation throughout this project. Further, I am grateful to Mr. Art Wanderlingh for his continuous guidance and encouragement in this effort. I also like to extend my gratitude to my wife, Melissa Tyson, for her editorial comments on this paper.

REFERENCES


APPENDIX A – Superlement Tool Details

The following page shows a snapshot of the developed interface in PCL. A user can choose one of the two available options:

1. **Export Superelement:** This allows the user to export an existing superelement group in any MSC.Patran database to a user defined file in MSC.Nastran card format. The choice of a filename is arbitrary while the filter associated with the superelement group list only allows the groups in the SE_groupname format to appear.

2. **Get Boundary Nodes:** The second option provides the capability of determining the boundary nodes shared by any two superelements and writing the corresponding nodal information to a file in the MSC.Nastran GRID card format. The user can select any two superelements by highlighting them on the list of available ones. Clicking on the “Get Default Boundary Nodes” button supplies the list of boundary nodes to the “Selected Boundary Nodes” box. The user can modify this list by adding or removing nodes via screen picking. The filter is set to node only once the cursor is placed in the “Select Boundary Nodes” box.

A number of additional features have been built into the code to facilitate the use and reduce the chance of any user errors. These include but are not limited to a notification if:

a. No superelement group selected for export.
b. Number of selected superelements for boundary nodes’ determination is not 2.
c. No common boundaries exist between the selected superelements.
d. No filename has been selected.
e. The specified filename already exists in the directory path.
f. No selected nodes are available in the list box.
g. A file is created for a superelement or boundary nodes.