MSC/NASTRAN

RELEASE GUIDE

Version 70.5

The MacNeal-Schwendler Corporation

Corporate Headquarters
The MacNeal-Schwendler Corporation
815 Colorado Boulevard
Los Angeles, CA 90041-1777 U.S.A.
Tel: (213) 258-9111 or (800) 336-4858
FAX: (213) 259-3838

Headquarters, European Operations
MacNeal-Schwendler GmbH
Innsbrucker Ring 15
Postfach 80 12 40
81612 München, GERMANY
Tel: (89) 431 9870
Telex: 523 784 MSG D
FAX: (89) 436 1716

Headquarters, Asia-Pacific Operations
MSC Japan Ltd.
Entsuji-Gadelius Building
2-39, Akasaka 5-chome
Minato-ku, Tokyo 107, JAPAN
Tel: (03) 3505-0266
Telex: J23363 MSCWATA
FAX: (03) 3505-0914

DISCLAIMER

The concepts, methods, and examples presented in this text are for illustrative and educational purposes only, and are not intended to be exhaustive or to apply to any particular engineering problem or design. The MacNeal-Schwendler Corporation assumes no liability or responsibility to any person or company for direct or indirect damages resulting from the use of any information contained herein.

©1998 by The MacNeal-Schwendler Corporation
First Printing April 1998
Printed in U.S.A.
All rights reserved.

Printed on recycled paper.

MSC and MSC/ are registered trademarks and service marks of The MacNeal-Schwendler Corporation. NASTRAN is a registered trademark of the National Aeronautics and Space Administration. MSC/NASTRAN is an enhanced, proprietary version developed and maintained by The MacNeal-Schwendler Corporation. MSC/PATRAN and MSC/ARIES are registered trademarks of The MacNeal-Schwendler Corporation. I-DEAS is a registered trademark of the Structural Dynamics Research Corporation. PostScript is a registered trademark of Adobe Systems International. Other product names and trademarks are the property of their respective owners.

MN ∗ V70.5 ∗ Z ∗ Z ∗ Z ∗ DC-REL

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Version 70.5 of MSC/NASTRAN offers important new capabilities and enhancements in performance, quality, aeroelasticity, nonlinear analysis, and finite element modeling. MSC/NASTRAN optimization pre- and post-processing is now integrated with MSC/PATRAN Version 7.5. MSC/PATRAN can now access MSC/NASTRAN results for efficient post-processing without importing the results into the MSC/PATRAN database.

This MSC/NASTRAN release has been tested extensively on numerous engineering problems across all of its supported computer platforms to meet MSC's stringent quality assurance standards. Version 70.5 also contains over 200 resolutions to existing errors or limitations. A major goal of Version 70.5 is to resolve as many client problems as possible.

This chapter covers the following topics:

1. Version 70.5 at a Glance, a summary of features and capabilities
2. New MSC/NASTRAN support in MSC/PATRAN Version 7.5
3. Licensing and Installation
4. Version 70.5 Product Documentation
5. Version 70.5 Error List
6. Sample Input Files
7. MSC Corporate Web Site

1.1 Version 70.5 at a Glance

MSC/NASTRAN Version 70.5 includes all the capabilities provided in Version 70 and offers the following new features and enhancements to extend the power and performance of MSC/NASTRAN.

New Features

- Geometric and material nonlinear spring and damper element.
- Generic control systems for Aeroelastic trim analysis.
- A modeling tool to model the gap between two coaxial cylinders.
- Interface elements that allow for 2-D and 3-D non-conforming meshes.
- Gap constraints in linear statics solution sequence (SOL 101).

Enhancements

- Externalization of error messages.
- R-type (rigid) element shape sensitivities and removal of problem size limit for optimization.
- Several enhancements for Aeroelasticity: (a) improvements to 2-D spline technology, (b) allowing aero models to be in superelements, (c) multi-disciplinary optimization with static analysis, and (d) performance improvements.
- Partitioned external superelement applicable for all solution sequences.
- MPC force output in .OP2 and .XDB files.
- Adjoint sensitivity with multiple boundary conditions.
- Inclusion of MPC forces in grid point force balance.
- Allowing interface degrees-of-freedom in superelements for coupled fluid-structural analysis.
- Large file support on IBM RISC System/6000 AIX 4.2 and Sun SPARC Solaris 2.6.
- Summary of bulk data entries in .F04 file.
- Beam library server on CRAY.
- More user-friendly estimate utility.
- MSC/ACCESS libraries on HP systems conforming to Unix standard.
- Removal of MEM size limit on Intel Windows NT.
- More numerical range for NORM module.

Performance Improvements

- Performance improvements for large solid models with both the direct and iterative solvers.
- Performance improvements for data recovery of composite elements on 64-bit machines.
- Performance improvements for frequency dependent acoustic absorber element.
- Performance improvements for data recovery of (a) normal modes for very large models and (b) transient and frequency response with multiple subcases.
- Performance improvements for STDCON and STRSRT modules.
- For restarts (a) with multi-master and (b) in cyclic normal modes analysis.

Quality and Reliability

- A consistent use of initial temperatures for all the element types.
- Over 200 error corrections, including corrections to defects in the FLEXlm licensing, the ability to run external programs, the adjoint sensitivity method, the new superelement methodology, etc.
- More robust normal modes, buckling, and complex eigenvalue analyses.

1.2 New MSC/NASTRAN Support in MSC/PATRAN Version 7.5

MSC/NASTRAN Sizing Optimization Support

Design Studies and Design Optimization preprocessing support are new in Version 7.5 of MSC/PATRAN to support and enhance the capabilities of MSC/NASTRAN Solution 200, the solution sequence which performs multidisciplinary design sensitivity and optimization.

In addition to providing a design study environment, Version 7.5 provides an initial release of sizing optimization support in the MSC/PATRAN environment. Subsequent versions will focus on adding to these basic capabilities.

The following capabilities are supported:

- Linear statics and normal modes analysis.
- Multiple linear statics subcases.
- Linear design variable-to-property relations (the model variables which can be selected as design variables are limited to those that satisfy a one-to-one relationship).
- Design variable linking (useful for enforcing symmetry, for example).
- Parameterized beam element library support.
- Weight design objective.
- Design responses (and thus constraints) can be any linear static analysis response (including composite stresses and strains), or normal modes.
- Normal modes responses in units of Hz, rather than eigenvalues.
- Limited, though automatic, generation of DRESP2-type responses (maximum element stresses and total displacements at grid points).
- Automated xy plots of design objective, design variable, and maximum constraint history.
- Full postprocessing capabilities within a design study.

MSC/NASTRAN Direct Access of Results

Direct Results Access is a new method of working directly with the MSC/NASTRAN results database (the .XDB file). Rather than importing all of the results into the MSC/PATRAN database prior to performing any results postprocessing, direct results access only imports “metadata” (i.e. the data that describes the data in the .XDB file). As results plots are created or results variables are manipulated, the data is supplied by the MSC/NASTRAN database, rather than the MSC/PATRAN database.

The direct results access capability has two major advantages:

- The time to read the results into the MSC/PATRAN database is essentially eliminated.
- Since the .XDB data is not duplicated in the MSC/PATRAN database, the size of the MSC/PATRAN database is greatly reduced.

Multiple .XDB files can be accessed simultaneously from a single MSC/PATRAN database. Up to 20 .XDB attachments can be made at any one time. Results can still exist in the MSC/PATRAN database as well. The direct results access capability is available under the Analysis application using the MSC/NASTRAN Preference (see Figure 1-1).

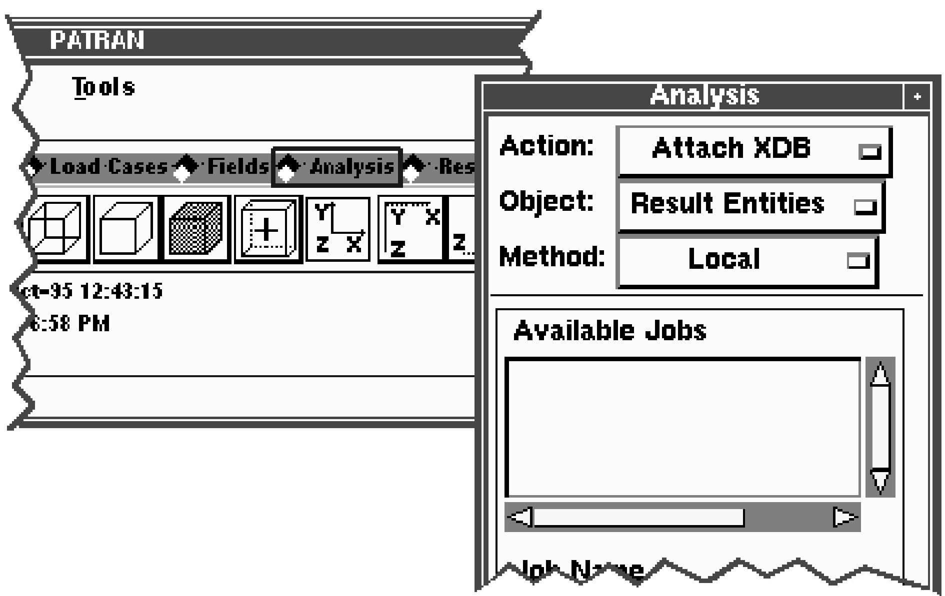


Figure 1-1. Analysis Dialog Box With "Attach .XDB" Action Selected.

Limitations

- Only results are supported. Model data must be imported separately and must exist before attaching a .XDB file.
- Grid point force, grid point stress, thermal, and p-element results are not currently supported.

- Design optimization global variables are not currently supported.
- The .XDB files can only be accessed when they are located on the same machine that MSC/PATRAN is running on or an NFS mounted disk.
- If the file is moved or deleted, the connection is lost and it must be reattached.
- Deleting a single Result Case from an attached .XDB file results in deleting all “metadata” associated with that attached .XDB file. Deleting one Result Case deletes all Result Cases.

.XDB files are created from MSC/NASTRAN by placing a PARAM, POST, 0 in the bulk data portion of the input file. Preprocessing of this PARAM entry is supported under the Translation Parameters form by the .XDB options for Data Output when setting up an analysis in the Analysis application (see Figure 1-2).

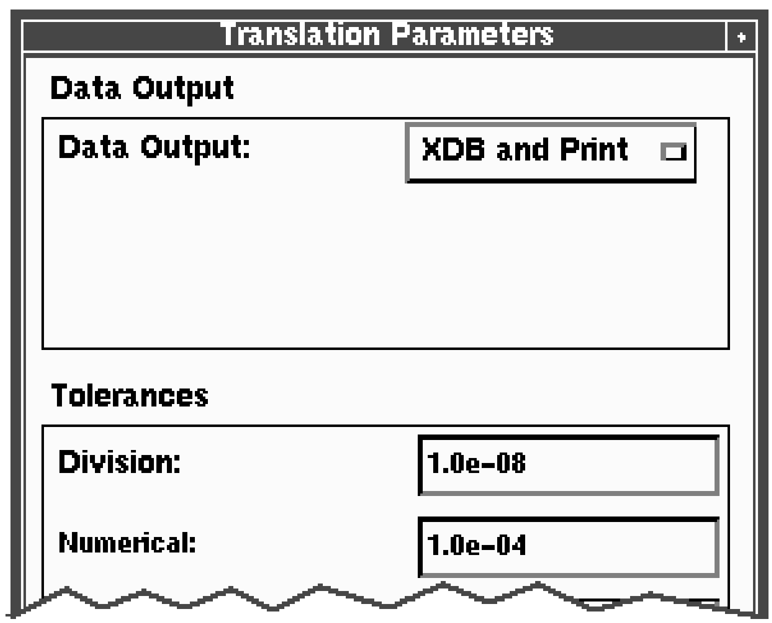


Figure 1-2. Translation Parameters Window with “.XDB and Print” Option Selected for Data Output.

See Section 4 of the *MSC/PATRAN Release Guide*, Version 7.5 for more details on MSC/NASTRAN Support and Enhancements.

1.3 Licensing and Installation

All MSC products use (or will use) a common licensing system, FLEXlm. MSC product licenses are now easier to set-up, configure, and maintain. FLEXlm offers you powerful administration options for distributing of licenses and monitoring usage. FLEXlm is available on the following platforms for MSC/NASTRAN:

- Cray Research J90, Y-MP
- Digital Alpha UNIX
- Hewlett Packard 9000
- IBM RISC System/6000
- Silicon Graphics
- Sun SPARC Solaris
- Intel Windows NT

All but the Cray systems may also host a network license server. Systems not listed above must continue to use the node-locked authorization codes. The Node-locked authorization codes are usable on all systems.

A common installation script, *mscsetup*, has been implemented for both MSC/NASTRAN and MSC/PATRAN. The common installation procedure is designed for improved flexibility and efficient use of computer disk space. Please refer to the installation instructions that accompanies this release as well as the *MSC/NASTRAN Configuration and Operations Guide*, Version 70.5.

Upward Compatibility

The license-controlled limit of problem size (GRID limit) in MSC/NASTRAN V70.5 no longer depends on VUGRID size for models that contain p-elements.

MSC/NASTRAN V70.5 converts the additional degrees-of-freedom (DOFs) contributed by p-elements and other entities with non-GRID DOFs such as SPOINTs to an equivalent number of GRIDs. The problem size is then determined by the sum of this equivalent number of GRIDs and the number of h-element GRIDs.

The equivalent number of GRIDs is calculated by dividing the number of DOFs by DOFs-per-GRID, which is six for structural analysis and one for thermal analysis.

To summarize what you can expect from GRID limits in V70.5:

- You don't need new licenses just to update GRID limit specifications.
- The new GRID limit behavior is practically the same for h-element models. If your model runs with MSC/NASTRAN V70, it will run with MSC/NASTRAN V70.5.
- The size of a p-elements model depends on the number of DOFs instead of VUGRIDs. If your model employs adaptive p-elements, MSC/NASTRAN V70.5 may terminate the run prematurely if the higher p-levels create too many DOFs during the adaptive p-level iteration process.

1.4 Version 70.5 Product Documentation

MSC/NASTRAN Version 70.5 includes the following new documents that describe Version 70.5 capabilities and enhancements.

- *MSC/NASTRAN Release Guide*
- *MSC/NASTRAN Configuration and Operations Guide*
- *MSC/NASTRAN Quick Reference Guide*
- *MSC/NASTRAN Numerical Methods User's Guide*

1.5 Version 70.5 Current Error List

The MacNeal-Schwendler Corporation maintains a *Current Error List* for MSC/NASTRAN. The *Current Error List* describes errors and limitations as well as ways to avoid them.

The *Current Error List* is delivered in text form on the V70.5 delivery media. This list describes the errors that have not yet been corrected and general limitations for Version 70.5. The file also contains the errors corrected in Version 70.5, designated with the notation "70.5" located next to the error number. You can find the error list file, error.lis, in the following directory:

```
<install_dir>/msc705/nast/misc/doc
```

1.6 Sample Input Files

This release guide contains several examples documenting the use of Version 70.5 capabilities. The MSC/NASTRAN input files used to create these examples are included with the MSC/NASTRAN Version 70.5 delivery in the following directory:

```
<install_dir>/msc705/nast/msc/doc/relnotes
```

This directory also includes a README file that lists additional sample input files. These test problem library (TPL) files illustrate various features in Version 70.5.

1.7 MSC Corporate Web Site

The MacNeal-Schwendler Corporation's Web site provides several rich sources of information that can assist you in solving your engineering problems using the MSC product line. MSC's Web address is:

http://www.macsch.com

Here you can read white papers on the use of MSC products, download technical papers from World Users' Conferences, order documentation from the MSC Bookstore, obtain a schedule of training courses, share feedback and suggestions interactively with other users, subscribe to MSC's corporate newsletter, and download software patches and utilities. Some highlights of MSC's Web page are described below.

Technical Support through the World Wide Web

If you need help installing or running MSC/NASTRAN, we encourage you first to refer to the extensive technical documentation available for MSC/NASTRAN. If you need further assistance, please contact your local MSC sales office for support. These offices are your primary source for authorization codes, technical information, and sales information about MSC/NASTRAN, documentation, training, and other MSC products.

In addition, MSC now offers automated support services through its corporate Web page. Our Customer Support page offers an e-mail support interface, technical applications notes on-line, and a "Fast-Facts Automated Fax Back System" for obtaining information on MSC products. It also gives the telephone numbers and addresses for the MSC/NASTRAN support offices by geographic location.

The internet Web address for Customer Support is:

http://www.macsch.com

Training Seminars

The MSC Institute of Technology offers a wide variety of training seminars. We provide classes at our facilities throughout the world on a regular schedule. We also provide classes at client sites and, if needed, can tailor our classes to meet your specific needs.

For more information about the MSC Institute of Technology, training facilities, course offerings, on-site or customized training, course descriptions, the current course schedule and related training topics, please visit:

http:/www.macsch.com

Browsing and Ordering Books from the *MSC Bookstore*

A complete list of documentation titles for the MSC product line is available through the Web-based *MSC Bookstore* at:

http://www.macsch.com/bookstore

You can browse tables of contents for documents of interest. A form is also provided to order books directly through the Internet. We encourage you to visit the *MSC Bookstore* to stay current with MSC's documentation set.

Subscribing to MSC/WORLD

MSC/WORLD, our newsletter about MSC and its products, is published three times a year and is free of charge for MSC customers. If you are not currently receiving *MSC/WORLD* and would like to be added to the mailing list, please notify us through MSC's Web site at:

http://www.macsch.com

Please send your name and mailing address. To order the newsletter by telephone, call 1-800-336-4858 or (213) 258-9111.

MSC User Groups

Also accessible from MSC's corporate Web page are user discussion forums devoted to our product line. You can post your comments, questions, requests for new capabilities, user experiences, and other feedback interactively. Customers can benefit from what their colleagues around the world, in a wide range of engineering disciplines, have to share. MSC, in turn, can better understand emerging technical needs and trends and promptly respond to them.

The Web address for MSC User Groups is:

http://www.macsch.com

CHAPTER 2

PERFORMANCE, QUALITY AND CUSTOMIZATION

2.1 Direct Solver – Reordering Schemes

MSC/NASTRAN provides several matrix reordering schemes for the direct solver. These schemes can be used to greatly reduce the CPU and Elapsed time as well as disk usage. The reordering method may be selected through SYSTEM(206). Below is a list of reordering options that a user may request (see the *MSC/NASTRAN Quick Reference Guide* for more details):

System 206	Reordering Option
0	EXTREME (3-D), METIS or MMD (mixed, 2-D)
1	MMD definite matrices
2	MMD indefinite matrices
3	no reordering
4	EXTREME (Default)
8	METIS
9	better of METIS and MMD

Synopsis of Reordering Methods

EXTREME reordering has been found to be very efficient in restructuring matrices for models with mostly 3-D elements. Availability of EXTREME reordering is as follows:

- V69.1 - EXTREME is available on SGI/CRAY systems. Not the default.
- V70 - EXTREME is the default on SGI/CRAY systems.

- V70.5 - EXTREME is available on ALL systems.

METIS reordering has been found to be efficient in restructuring matrices for models with both 2-D and 3-D elements. Availability of METIS reordering follows:

- V70.0.3 - METIS is available on Sun UltraSPARCs and HP V-Class systems (not the default).
- V70.5 - METIS is available on all systems.

Below is a list of models used in testing:

Test Name*	DOF	Model Description	SOL	N 2-D	N 3-D	mem
`mddst`	32,184	Wheel	SOL 101	6,840	7,632	5m
`mdest`	30,934	Crank Shaft	SOL 101	0	8,888	5m
`mdemd`	30,934	Crank Shaft	SOL 103	0	8,888	5m
`lgast`	68,000	Engine Block (SE)	SOL 101	8,696	2,469	7m
`lgbbk`	66,000	Airplane Part	SOL 105	11,552	0	8m
`lgkmd`	65,904	Satellite	SOL 103	15,218	0	6m
`lgmst`	99,191	Transmission Part	SOL 101	0	18,681	10m
`vlbst`	140,000	Airplane Bracket	SOL 101	7,424	21,508	9m
`vlgst`	134,333	T-Joint	SOL 101	0	27,081	32m
`vllst`	139,649	Transmission	SOL 101	0	36,041	16m
`xldst`	606,231	Transmission	SOL 101	0	122,284	50m
`xlnmd`	331,468	Car Body	SOL 103	71,914	428	24m

*Names have the following convention:

size - letter - SOL

where size = `md` (medium)
`lg` (large)
`vl` (very large)
`xl` (extra large)

letter = alpha numerical sequence number

SOL = `st` (static runs - SOL 101)
`md` (normal modes - SOL 103)
`bk` (buckling - SOL 105)

The following is a table listing ratios of CPU times for MMD reordering/EXTREME reordering and MMD reordering/METIS reordering for several computers for the above listed models. Please note that the performance ratios in the following tables are preliminary and not all of them are available at this time.

Test \ Ratio	CRAY T90		DIGITAL 2300 5/300		HP 9000/889/ SPP 2000		IBM 990 Power 2	
	MMD/ EXT	MMD/ METIS	MMD/ EXT	MMD/ METIS	MMD/ EXT	MMD/ METIS	MMD/ EXT	MMD/ METIS
mddst	0.97	0.98	1.06	0.66	1.03		1.03	1.05
mdest	0.99	0.94	1.19	0.53	1.17		1.18	1.07
mdemd	1.01	0.97	1.27	0.90	1.18		1.21	1.10
lgast	0.99	0.99	1.09	0.99	1.09		1.05	0.94
lgbbk	1.02	1.02	1.15	1.05	1.00		1.13	1.15
lgkmd	2.43	2.37	2.76	2.62	0.96		1.14	1.09
lgmst	0.98	0.97	1.32	0.58	1.06		1.11	1.12
vlbst	0.96	0.95	1.07	1.01	1.06		1.05	1.04
vlgst	1.06	1.03	1.33	1.11	1.51		1.52	1.53
vllst	1.00	0.98	1.40	1.35	1.27		1.33	1.30
xldst	1.56				3.26		2.89	2.59
xlnmd	0.98	0.95					1.07	0.99

	INTEL PII 300 MHz		NEC SX-4		SGI R10,000		SUN UltraSPARC1	
Test \ Ratio	MMD/ EXT	MMD/ METIS	MMD/ EXT	MMD/ METIS	MMD/ EXT	MMD/ METIS	MMD/ EXT	MMD/ METIS
mddst	1.07	1.17	0.97	0.99	1.05		1.14	1.15
mdest	1.35	1.21	1.02	1.03	1.21		1.48	1.21
mdemd	1.28	1.17	1.03	1.01	1.23		1.33	1.14
lgast	1.03	0.98	1.00	0.98	1.04		1.16	1.05
lgbbk	1.21	1.23	.96	0.90	1.00		1.21	1.20
lgkmd	1.10	1.09	1.07	1.05	1.00		1.37	1.40
lgmst	1.32	1.34	0.96	0.98	1.10		1.25	1.25
vlbst	1.33	1.23	0.91	0.97	1.07		1.19	1.13
vlgst	1.95	1.95	0.99	1.00	1.49		1.95	2.00
vllst	1.73	1.64	0.96	0.99	1.30		1.62	1.47
xldst	3.34	3.23	1.34	1.34			3.62	3.30
xlnmd	1.08	0.96	0.97	0.98			1.00	0.94

2.2 Iterative Solver Improvements

For MSC/NASTRAN Version 70.5, the memory management and spill logic for the Block Incomplete Cholesky (BIC) preconditioner (default) of the iterative solver have been improved significantly, especially for jobs where the incore memory requirements exceed the available memory.

The effect of the improved memory management is that the iterative solver requires less memory to run a job in Version 70.5, compared to Version 70.

In previous versions of MSC/NASTRAN, the spill logic for the iterative solver with BIC preconditioned was not optimized. The improved and optimized spill logic introduced in Version 70.5 speeds up out-of-core jobs by up to a factor of 5 in the solver part.

The interface for BIC preconditioning has been simplified. Instead of having to choose between BICWELL (for well-conditioned problems) and BICILL (for ill-conditioned problems), the simple specification of PRECOND=BIC (default) on the ITER Bulk Data entry invokes BIC preconditioning. The decision of whether a problem is well- or ill-conditioned is done inside the solver based on an estimated condition number of the matrix. Since BIC is the default, no ITER Bulk Data entry is required to choose BIC. The iterative solver is selected by ITER=YES on the NASTRAN statement.

The following table shows the improvements of BIC preconditioning and the advantages over the direct method for solid models. Note that the improvement for the direct solver from V70 to V70.5 is due to the EXTREME reordering, which is now available on all platforms (see the previous section).

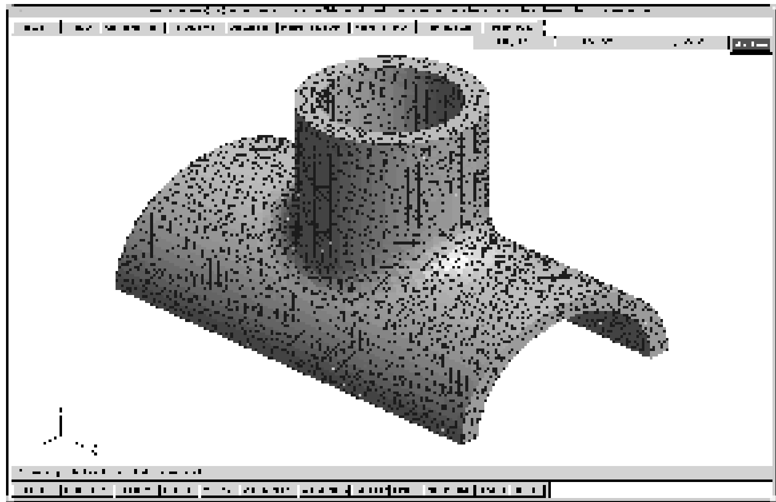


Figure 2-1. Half-Tube, IBM RS/6000 (590).

Specifications of Half-Tube

Elements:	27,081 TETRAs
DOFs:	134,333
Analysis:	linear statics
Computer:	RS/6000 (590)

Results of Half-Tube

Method	Disk Space (MB)		Memory (MB)	Total CPU(s)		Solver CPU(s)	
	V70	V70.5		V70	V70.5	V70	V70.5
iter	585.6	585.6	144	865.6	775.1	388.0	283.8
iter	645.8	585.8	96	2,039.5	777.9	1,562.5	289.2
iter	645.8	585.8	60	2,035.6	917.1	1,561.9	429.1
direct	1,066.2	902.8	144	1,219.3	892.3	824.6	482.6
direct	1,067.5	904.0	60	1,291.2	912.4	895.0	501.2

The following proprietary model from the automotive industry illustrates the performance of the iterative solver on a CRAY C90 computer and compares it to the performance of the direct solution.

Specifications of Crankshaft

Elements:	86,500 HEXAs, 23,350 PENTAs, 850 TETRAs, 650 BARs
DOFs:	388,488
Analysis:	linear statics
Computer:	CRAY C90

Results of Crankshaft

# of Loads	Method	Disk Space (GB)	Memory (MB)	CPU(s)
1	iter	1.7	916	2,593.3
1	iter	2.2	400	2,612.1
1	iter	2.2	200	2,790.7
1	direct	5.4	200	2,826.1
6	iter	2.6	400	5,948.0
6	direct	5.8	200	4,807.0

2.3 Specific Performance Improvements

1. **Eigenvalue Analysis**: PARAM,LMODES may be used in SOL 103 to reduce the number of modes to be processed in data recovery. This may improve performance for large problems with a large number of eigenvectors. LMODES is recommended only when the desired number of eigenvectors is significantly less than the number of computed eigenvectors. It should be noted that LFREQ and HFREQ parameters are not applicable to SOL 103.

2. **Dynamic Analysis**: Data recovery is now more efficient when there are multiple TSTEP/DLOAD/FREQ subcases in modal transient and frequency response analysis.

3. **Multi-Master Restarts**: NASTRAN LOCBULK=2 processing option for Multi-MASTER. If there are no changes to the Bulk Data, except PARAM entries, in a Multi-MASTER restart, then NASTRAN LOCBULK=2 may be specified to improve performance.

4. **Cyclic Normal Modes**: Restarts are more efficient in cyclic normal modes analysis (SOL 115).

5. **Composite Data Recovery**: The data recovery for composites (SDRCOMP module) has been optimized on 64-bit machines (CRAY and NEC) that may result in a four-times performance improvement for the module on a T90, IEEE machine. This enhancement reduces the CPU time for a particular example from 735 to 435 seconds. It should be noted that overall performance gain is problem dependent.

6. **Frequency Dependent Elements**: The logic of frequency dependent elements introduced in Version 69 has been optimized in Version 70.5 to significantly increase the performance for frequency response analysis. This results in as much as a three-times performance increase for the frequency dependent loop for models where frequency dependent elements are equal in number to nonfrequency dependent elements.

7. **STDCON and STRSRT Modules**: The logic of STDCON and STRSRT modules has been modified. This has resulted in significant reduction of CPU times in these two modules. For a user’s problem (CSR 7463), the total CPU time was reduced from 865 seconds to about 270 seconds, because of the improved logic in the STDCON module. Similarly, for another client’s problem (CSR 29829) with 10,000 bar elements and six static subcases, the total CPU time was reduced from about 340 seconds to 140 seconds, because of the improved logic in STRSRT module.

8. **Dense Matrix or Multiplication**: Dense matrix multiplication (Method 1) has been further optimized in Version 70.5 to take advantage of triple loop kernels. See Section 2.4 of the *MSC/NASTRAN Numerical Methods User’s Guide*, Version 70.5 for more details.

2.4 Real Eigenvalue Analysis

The shift logic of the real Lanzcos algorithm has been enhanced and improved to make it more reliable for both normal modes and buckling analyses. Specifically, the following improvements have been made:

- Provide better diagnostic messages when the user input is inconsistent with the actual eigenvalue distribution. For example, there may be no modes, or the number of modes desired exceeds the number of modes between the user-specified frequency bounds.

- Provides all the multiple roots if they happen to be at the boundary of your requests. For example, if you ask for five modes and the fifth mode is a repeated root that has two other modes with the same frequency, then the information for seven modes will be output.
- Several known user related problems for both normal modes and buckling analysis with unusual eigenvalue distribution have been fixed. (See CSRs 5467, 6080, 6270, and 27686.) You no longer need to use the avoidance specified in the error reports.
- Provides an efficient solution even when the algorithm shifts onto an eigenvalue.
- Provides solution even when the lower and upper frequency bounds happen to be eigenvalues.

The functionality of the REIGL module (Real Lanczos algorithm) has been merged into the READ module (Real eigenvalue analysis). The new READ module has been integrated into all the applicable solution sequences. See the *MSC/NASTRAN Numerical Methods User’s Guide*, Version 70.5 for the details on the new DMAP interface for the new READ module.

Benefits

- More robust normal modes analysis.
- Better diagnostic messages.
- Ease of use. You do not have to use avoidances to solve your normal modes problem.

Input

There are no input changes required from Version 70.

Examples

Several example problems from automobile companies were analyzed to ensure the performance of the new shift logic and the new READ module. We are still working with hardware vendors to further improve the performance, so the performance numbers given below are preliminary.

Example 1

C90, 35,000 DOF, 500 modes

	CPU sec	
Version 70	351.7	reigl
Version 70.5	292.3	read

Example 2

C90, 132,000 DOF, 670 modes

	I/O(MB)	CPU sec	
Version 70	86267.1	1374.6	reigl
Version 70.5	75858.1	1206.3	read

Example 3

T90, 650K DOF, 814 modes

	I/O(GB)	CPU sec	
Version 70	406.1	5123.6	reigl
Version 70.5	391.0	5195.6	read

Example 4

T90, 1.5M DOF, 1074 modes, 50 MW, blocksize=7

	I/O(GB)	CPU sec	
Version 70	2244.8	17942.0	reigl
Version 70.5	2140.5	16201.4	read

Example 5

C90, 1.9M DOF, 17 modes, solid element

	CPU sec	
Version 70	20170	reigl
Version 70.5	15162	read

Example 6

T90, 2.9M DOF, 889 modes

	Elapsed	I/O	CPU sec	
Version 70	809:56	2398.8G	27342	MAXSET=12 mem=150m Disk=74GB
Version 70.5	676:24	2587.2G	25479	MAXSET=9 mem=115m Disk=70GB

From the above examples, you can see that the majority of the example problems with the exception of Example 3 (650 K DOF job), performed better in Version 70.5 compared to Version 70.

Upward Compatibility

- All previous jobs set up to run the REIGL module will now run the new READ module and the new shift logic.
- For the majority of your problems the new shift logic is expected to be more efficient. However, there may be some problems where the new shift logic is slower. Therefore, the old shift logic is still available.
- You may run the old shift logic, explicitly, by setting the system cell 273 to 1 (i.e., SYSTEM(273)=1 in RC file or NASTRAN statement).
- The old REIGL module will be executed if you are using DMAP alters that specifically have a call to the REIGL module. It is recommended that you change your alters to call the new READ module instead of the old REIGL module.

2.5 Complex Eigenvalue Analysis

A new adaptive block complex Lanczos algorithm is implemented in Version 70.5. This new method is more robust than the previous methods.

Benefits

- Ease of use in solving brake squeal problems. Brake squeal problems typically consist of 70,000 to 100,000 degrees-of-freedom. This size problem can now be solved directly (SOL 107) without resorting to superelement or modal reduction techniques to reduce the size of the problem.
- More stringent acceptance criteria for eigenvalue. It will not provide you with a bad eigenvalue in contrast to the earlier version of complex Lanczos. However, it may sometimes reject a valid root because of the stringent criteria.
- It provides a correct solution even if you have massless degrees-of-freedom. This is in contrast to the available Hessenberg method which can provide an incorrect solution if you have massless degrees-of-freedom.

Input

You need to set SYSTEM(108) = 4 in RC file or NASTRAN statement to execute the new block complex Lanczos algorithm. This is still not the default method because the performance of the new method needs to improve.

The EIGC Bulk Data entry has been updated to provide you more options for the new method. You can specify a shift (as in the CLAN method). The default shift is (0,i) or (0.1,i) if viscous damping is present. The continuation entry's fifth, sixth, and seventh fields also enable you to input maximum and starting block sizes, as well as a stepsize. See the *MSC/NASTRAN Numerical Methods User's Guide*, Version 70.5 for the meaning and effect of these parameters.

Output

The output of the new method is the same as the previous Lanczos method. Additional output may be requested by setting DIAG 12. See the *MSC/NASTRAN Numerical Methods User's Guide*, Version 70.5 for more details on the additional output.

Examples

To demonstrate the numerical advantages of the new method, we compared two smaller application examples. The results are as follows:

Application	DOF	# of Modes Found		Average Accuracy	
		New	Old	New	Old
Brake	834	18	8	10^{-5}	10^{-4}
Acoustic	538	10	5	10^{-6}	10^{-3}

Both of these runs were SOL 107, run on an IBM RS/6000.

To analyze the performance of the new method on large platforms, two complete brake models (with frame structures) were run on our CRAY J90. These runs were unsolvable by the old method. The results are:

Job	DOF	# of Modes	Accuracy	CPU sec	I/O GB
Brake 1	63,420	21	10^{-5}	6,590.8	166.6
Brake 2	73,676	32	10^{-6}	16,199.0	447.7

These results indicate the feasibility of solving large direct complex eigenvalue jobs, without the added burden of a superelement formulation or the accuracy impact of modal solutions.

Although the new method can solve larger problems, the performance (CPU time) is not as good as the old Lanczos method. For some problems it may be considerably slower, therefore, it is not the default method.

Upward Compatibility

- Although there are additional parameters that may be set on the EIGC entry, the user interface is simplified and upward compatible.
- The new method may be slower (more CPU time) then the previous Lanczos method.
- The use of system cell 108 to select the new Hessenberg method (SYSTEM(108)=1) and the previous Lanczos method (CLAN, SYSTEM(108)=2) are still valid.

2.6 Thermal Loads User Interface

For Version 70.5, the initial temperature (for all the elements (except CONEAX) capable of thermal loading) may be specified using the TEMP (INIT) Case Control command in conjunction with TEMP, TEMPD, TEMPP1, TEMPRB, and TEMPF Bulk Data entries. Some older linear elements such as the CSHEAR and CTRIAX6 in previous versions did not recognize TEMP(INIT) and would utilize the reference temperature from the material properties (TREF value off the MATi entries) without informing the user. The consistent use of TEMP(INIT) by all the elements makes the user interface simpler and less prone to errors.

Upward Compatibility

Initial temperature for thermal loading will be based on the TEMP(INIT) Case Control command. This may cause the answers to be different from previous versions if both TEMP(INIT) and material property entries are used to specify initial temperatures.

Limitation

For the CONEAX, any TEMP(INIT) pointing to a TEMPAX will be ignored.

2.7 Externalization of Error Message

Error message handling in MSC/NASTRAN has been greatly improved. A message catalog source file is provided to the user that can be customized according to the site specific requirements. The message catalog source file contains message text used for many MSC/NASTRAN error messages except DMAP and DIAG messages. A user can modify the message catalog source file to include more appropriate message text to reflect their operations, compile the new source file using the MSGCMP utility, and invoke the new message catalog using the “msgcat” keyword during the runtime.

A typical message in the message catalog source file looks like:

```
# Comment:
#
#
#
1049.0
#
# used by: XSPAC
#
#0*** USER FATAL MESSAGE 1049 (XSPAC)
  AMOUNT OF USER OPENCORE(HICORE), MASTER(RAM)
  SCRATCH(MEM) AND BUFFER POOL SIZE HAVE EXCEEDED THE MSC/NASTRAN MEMORY LIMIT:
           MSC/NASTRAN MEMORY LIMIT  =%1  WORDS
           USER OPENCORE(HICORE)     =%2
           MASTER(RAM)  SIZE         =%3
           BUFFER POOL  SIZE         =%6          (%7 BUFFERS)
           SCRATCH(MEM) SIZE         =%4          (%5 BUFFERS)
      USER ACTION:
            DECREASE THE SUM OF HICORE, MASTER(RAM), SCRATCH(MEM)
            AND BUFFER POOL SIZE TO FIT WITHIN THE MSC/NASTRAN MEMORY LIMIT.
```

```
Remarks:  1. Comment line is preceded by a # sign in the first column.
          2. The message number begins from the first column.
          3. The text of the message begins from the second column.
          4. The variables are indicated by the numerals preceded by a % sign.
          5. The number of variables are predetermined, however,
             variables can be printed in any order.
          6. The message catalog source file provided to the user is
              in English only.
```

Please refer to the Version 70.5 *MSC/NASTRAN Configuration and Operations Guide,* Sections 3.8 and 6.4 for more details.

2.8 nastran Command Enhancements

Several modifications have been made to the nastran command to improve ease of use. Specifically,

- RC file lines can now be split using a backslash, “\”, as the last character on the line to be continued.
- A new keyword, whence, allows you to determine where a keyword value is assigned.
- The meaning of the keywords authorize, batch, etc. have been modified.
- See Sections 4.1 and 4.2 for more details in the Version 70.5 *MSC/NASTRAN Configuration and Operations Guide*.

2.9 Modified Utilities

The following utilities have been changed. The changes are briefly described below. For more details see Section 6 of the *MSC/NASTRAN Configuration and Operations Guide*, Version 70.5.

Beam Server (BMSRV)

The Beam Server is now available on all Cray systems.

ESTIMATE

- The ESTIMATE utility will now read the standard RC files.
- An error in CTRIAX6 processing has been corrected. ESTIMATE now correctly assigns the “1” and “3” DOF.
- An error in memory estimation has been corrected. As a result, many jobs now report a 200KW higher memory estimate.
- Setting a value on the command line, or in an RC file will now automatically suppress any rule that would estimate the value. For example, setting buffsize in an RC file will now suppress rule 1, the rule that would calculate a new BUFFSIZE value.
- The meaning of keywords buffsize, bpool, memory, smemory, and version have been modified and three new keywords enable, nastran, and real have been added.
- See Section 6.1 of the *MSC/NASTRAN Configuration and Operations Guide*, Version 70.5 for more details.

Accounting (MSCACT)

- An accounting file generated by previous versions of MSC/NASTRAN is now fully compatible.
- New keywords sortby and summary have been added to increase the flexibility for report generation.
- See Section 6.3 of the *MSC/NASTRAN Configuration and Operations Guide*, Version 70.5 for additional details.

External Message Compiler (MSGCMP)

- The MSGCMP utility will now validate a binary message catalog before attempting to convert it to text form.
- See Section 6.4 of the Version 70.5 *MSC/NASTRAN Configuration and Operations Guide*.

MSC/ACCESS

The MSC/ACCESS library on Hewlett-Packard systems now exports names with a trailing underscore character. This change brings MSC/ACCESS on HP systems in line with standard UNIX practice for inter-language calling conventions.

The old name formats, i.e., without the trailing underscores, are available in:

install-dir/msc705/hpux/libdbio.old.a

and

install-dir/msc705/sppux/libdbio.old.a

You may need these libraries if you have programs that must be linked using the old naming conventions. These programs should be converted to use the trailing-underscore naming convention.

CHAPTER 3

MODELING ENHANCEMENTS

3.1 Geometric and Material Nonlinear Spring and Damper Element

Introduction

A new rod-type spring and damper element has been developed in MSC/NASTRAN Version 70.5. The new element (BUSH1D) is a one dimensional version of the BUSH element (without the rigid offsets). The new element supports large displacements.

The BUSH1D element has axial stiffness and axial damping. The element includes the effects of large deformation. The elastic forces and the damping forces follow the deformation of the element axis if there is no element coordinate system defined. The forces stay fixed in the x-direction of the element coordinate system if the user defines such a system. Arbitrary nonlinear force-displacement and force-velocity functions are defined with tables and equations. A special input format is provided to model shock absorbers.

Benefits

For the first time in MSC/NASTRAN, you have an element where damping follows large deformation. Arbitrary force deflection functions can now be modeled conveniently. When the same components of two grid points must be connected, we recommend using force-deflection functions with the new BUSH1D element instead of using NOLINi entries. The BUSH1D element produces tangent stiffness and tangent damping matrices, whereas the NOLINi entries do not produce tangent matrices. Therefore, BUSH1D elements are expected to converge better than NOLINi forces.

Input

The new element is defined with a new connectivity entry CBUSH1D and a new property entry PBUSH1D described below (see the Bulk Data Section of the *MSC/NASTRAN Quick Reference Guide* Version 70.5 for a detailed description of the input). The user may define several spring or damping values on the PBUSH1D property entry. It is assumed that springs and dampers work in parallel. The element force is the sum of all springs and dampers.

CBUSH1D

Defines the connectivity of a one-dimensional spring and viscous damper element.

Format:

1	2	3	4	5	6	7	8	9	10
CBUSH1D	EID	PID	GA	GB	CID				

Example:

CBUSH1D	35	102	108	112					

Field	Contents	Default Values
EID	Element identification number. (Integer > 0).	Required
PID	Property identification number of a PBUSH1 entry. (Integer > 0).	EID
GA	Grid point id of first grid.	Required
GB	Grid point id of second grid.	blank
CID	Coordinate system id. (Integer ≥ 0)	blank

PBUSH1D

Defines linear and nonlinear properties of a one-dimensional spring and damper element (CBUSH1D entry).

Format:

1	2	3	4	5	6	7	8	9	10
PBUSH1D	PID	K	C	M		SA	SE		
	"SHOCKA"	TYPE	CVT	CVC	EXPVT	EXPVC	IDTS		
			IDETS	IDECS	IDETSD	IDECSD			
	"SPRING"	TYPE	IDT	IDC	IDTDU	IDCDU			
	"DAMPER"	TYPE	IDT	IDC	IDTDV	IDCDV			
	"GENER"		IDT	IDC	IDTDU	IDCDU	IDTDV	IDCDV	

Example:

PBUSH1D	35	3000.	200.	300.					
	SHOCKA	TABLE	2.2	1.2	1.		200		

Field	Contents	Default
PID	Property identification number. (Integer>0).	Required
K	Stiffness. (Real ≥ 0).	Required
C	Viscous damping. (Real ≥ 0).	Blank
M	Total mass of the element. (Real ≥ 0).	Blank
SA	Stress recovery coefficient [1/area]. (Real ≥ 0).	Blank
SE	Strain recovery coefficient [1/length]. (Real ≥ 0).	Blank
SHOCKA	Character string specifying that the next 10 fields are coefficients of the following force versus velocity/displacement relationship. (Character). $F(u,v) = C_V \cdot S(u) \cdot \mathrm{sign}(v) \cdot \lvert v \rvert^{EXPV}$ The force F, the displacement u, and the velocity v, are in the axial direction of the damper. The axis of the damper is defined by the two connecting grid points GA and GB on the CBUSH1D Bulk Data entry. The displacement u and the velocity v, are the relative displacement and the relative velocity with respect to the grid point GA. The scale factor S(u) must be defined with a table or with an equation.	Required
TYPE	Character string indicating the type of definition. (Character). For TYPE=EQUAT, the fields IDETS, IDECS, IDETSD, and IDECSD are identification numbers of DEQATN entries. For TYPE=TABLE the field IDTS is an identification number of a TABLEDi entry. If no character string is provided (blanks), TYPE=TABLE is set.	TABLE
CVT	Viscous damping coefficient C_V for tension v>0, force per unit velocity. (Real).	Required

Field	Contents	Default
CVC	Viscous damping coefficient C_V for compression v>0, force per unit velocity. (Real).	CVT
EXPVT	Exponent of velocity EXPV for tension v>0. (Real).	1.
EXPVC	Exponent of velocity EXPV for compression v<0. (Real).	EXPVT
IDTS	Identification number of a TABLEDi entry for tension and compression if TYPE=TABLE. The TABLEDi entry defines the scale factor S, versus displacement u.	Required if TYPE=TABLE
IDETS	Identification number of a DEQATN entry for tension if TYPE=EQUAT. The DEQATN entry defines the scale factor S, versus displacement u, for tension u>0.	Required if TYPE=EQUAT
IDECS	Identification number of a DEQATN entry for compression if TYPE=EQUAT. The DEQATN entry defines the scale factor S, versus displacement u, for compression u<0.	IDETS
IDETSD	Identification number of a DEQATN entry for tension if TYPE=EQUAT. The DEQATN entry defines the derivative of the scale factor S, with respect to the displacement u, for tension u>0.	Required if TYPE=EQUAT
IDECSD	Identification number of a DEQATN entry for compression if TYPE=EQUAT. The DEQATN entry defines the derivative of the scale factor S, with respect to the displacement u, for compression u<0.	IDETSD
SPRING	Character string specifying that the next 5 fields define a nonlinear elastic spring element in terms of a force versus displacement relationship. (Character). $F(u) = F_T(u)$ Tension is u>0 and compression is u<0.	
DAMPER	Character string specifying that the next 5 fields define a nonlinear viscous element in terms of a force versus velocity relationship. (Character). $F(v) = F_T(v)$ Tension is v>0 and compression is v<0.	
GENER	Character string specifying that the next 7 fields define a general nonlinear elastic spring and viscous damper element in terms of a force versus displacement and velocity relationship. (Character). For this element, the relationship can only be defined with TYPE=EQUAT. $F(u,v) = F_T(u,v)$ Tension is u>0 and compression is u<0. For SPRING, DAMPER, and GENER, the remaining fields are	

Field	Contents	Default
TYPE	Character string indicating the type of definition. (Character). For TYPE=EQUAT the following fields are identification numbers of DEQATN entries. For TYPE=TABLE the following field is an identification number of a TABLEDi entry. TYPE is ignored for GENER.	Required
IDT	Identification number of a DEQATN entry for tension if TYPE=EQUAT. Identification number of a TABLEDi entry for tension and compression if TYPE=TABLE.	Required
IDC	Identification number of a DEQATN entry for compression if TYPE=EQUAT. Is ignored for TYPE=TABLE.	IDT
IDTDU	Identification number of a DEQATN entry for tension if TYPE=EQUAT. The DEQATN entry defines the derivative of the force F with respect to the displacement u, for tension u>0. For SPRING and GENER only.	Required
IDCDU	Identification number of a DEQATN entry for compression if TYPE=EQUAT. The DEQATN entry defines the derivative of the force F with respect to the displacement u, for compression u<0. For SPRING and GENER only.	IDTDU
IDTDV	Identification number of a DEQATN entry for tension if TYPE=EQUAT. The DEQATN entry defines the derivative of the force F with respect to the velocity v, for tension v>0. For DAMPER and GENER only.	Required
IDCDV	Identification number of a DEQATN entry for compression if TYPE=EQUAT. The DEQATN entry defines the derivative of the force F with respect to the velocity v, for compression v<0. For DAMPER and GENER only.	IDCDT

Output

The BUSH1D element puts out axial force, relative axial displacement and relative axial velocity. It also puts out stress and strain if stress and strain coefficients are defined. All element related output (forces, displacements, stresses) is requested with the STRESS Case Control command.

Guidelines

The element is available in all solution sequences. In static and normal modes solution sequences, the damping is ignored.

In linear dynamic solution sequences, the linear stiffness and damping is used. In linear dynamic solution sequences, the BUSH1D damping forces are not included in the element force output.

In nonlinear solution sequences, the linear stiffness and damping is used for the initial tangent stiffness and damping. When nonlinear force functions are defined and the stiffness needs to be updated, the tangents of the force-displacement and force-velocity curves are used for stiffness and damping. The BUSH1D element is considered to be nonlinear if a nonlinear force function is defined or if large deformation is turned on (PARAM, LGDISP, 1). For a nonlinear BUSH1D element, the element force output is the sum of the elastic forces and the damping forces. The element is considered to be a linear element if only a linear stiffness and a linear damping are defined and large deformation is turned off.

Limitations

1. The BUSH1D element nonlinear forces are defined with table look ups and equations. In Version 70.5, equations are only available if the default option ADAPT on the TSTEPNL entry is used, equations are not available for the options AUTO and TSTEP.
2. The table look ups are all single precision in MSC/NASTRAN. In nonlinear, round-off errors may accumulate due to single precision table look ups.
3. For linear dynamic solution sequences, the damping forces are not included in the element force output.
4. The “LOG” option on the TABLED1 is not supported with the BUSH1D.

Example 1: Swinging Pendulum (TPL: ar29src1.dat)

The solution to a swinging pendulum is calculated using SOL 129 of MSC/NASTRAN Version 70.5. The pendulum has a length of 1.0 [m]. and has a concentrated mass of M = 1,000.0 [kg] at the free end, see Figure 3-1. We start the analysis with the pendulum at rest in horizontal position. The free end is loaded with a gravity induced load of G = 10,000.0 [N]. The leg of the pendulum is very stiff, K = 1.e+7 [N/m], so that the relative axial deformation is small compared to the overall motion of the pendulum. Large deformation effects are turned on with PARAM, LGDISP, 1. The MSC/NASTRAN input file is partially shown in Listing 3-1 and is available in the TPL as ar29src1.dat.

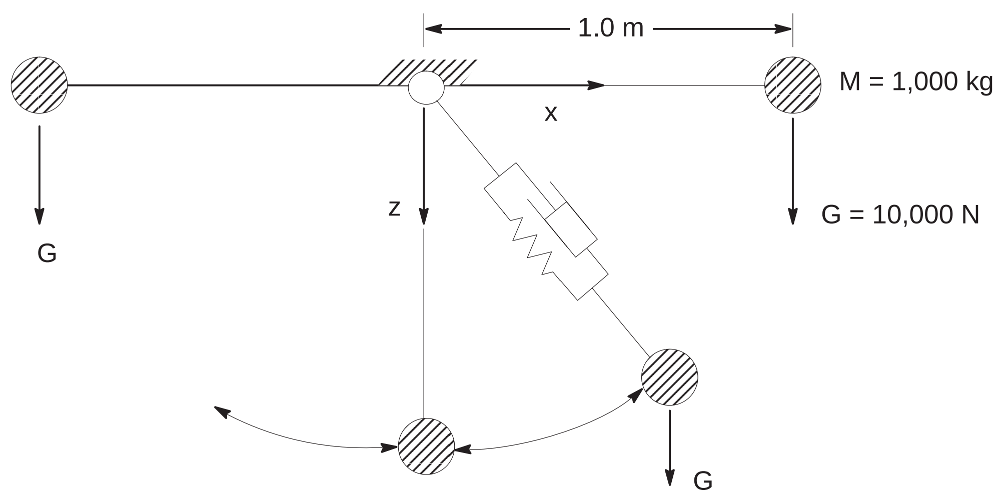
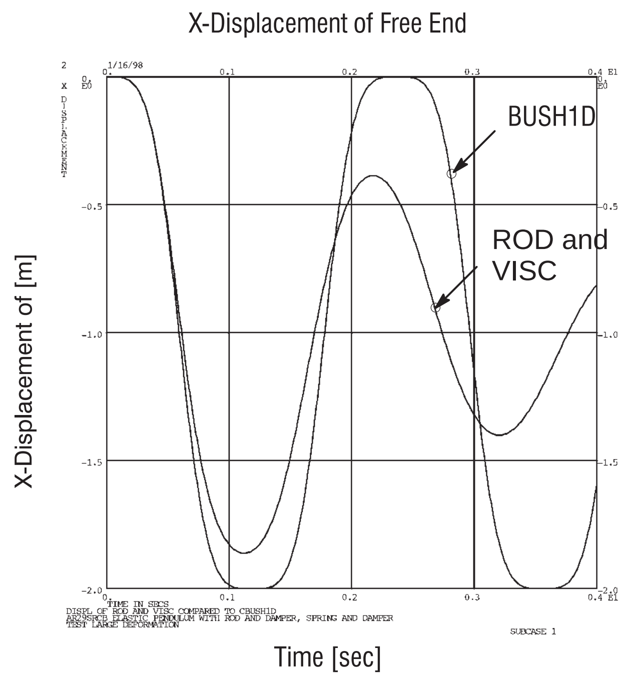


Figure 3-1. Swinging Pendulum.

Listing 3-1. MSC/NASTRAN Input File for the Swinging Pendulum.

```
$
$ Name: ar29src1
$
$ A swinging pendulum is modeled. The pendulum is loaded with
$ gravity load. The analysis goes through one and a half cycles,
$ starting in horizontal position. Large deformation is turned on.
$
$ The pendulum is modeled with two separate systems.
$ 1. CROD and CVISC, CROD rotates, CVISC stays in horizontal
$    orientation.
$ 2. CBUSH1D with SPRING and DAMPER, spring and damper rotate.
$
TIME 10  $
SOL 129 $
CEND
$
TITLE= ar29src1 swinging elastic pendulum
SUBTITLE= demonstrate rotating damping
.
.
.
$
BEGIN BULK
$
param, lgdisp, 1
$
$ GEOMETRY
$
.
.
.
$ CBUSH1D spring and rotating viscous damper
$
$CBUSH1D,EID,GA,GB,CID
cbush1d, 108, 208, 41, 42
$PBUSH1D,PID,K,C
pbush1d, 208, 1.e+7, 1000.
$
$       $       $       $       $       $       $       $       $       $
$ SOLUTION STRATEGY
$
.
.
.
$
ENDDATA
```

We investigate two different models of the pendulum in one run. In the first model, we use a Rod element for the stiffness of the leg, K = 1.e+7 [N/m], and a Visc element for 5% equivalent viscous damping, C = 1,000.0 [Ns/m]. The x- and z-displacements of the free end are shown in Figure 3-2. The Rod rotates while the viscous damper Visc stays fixed in its initial horizontal position. The Visc element is linear and does not follow large deformations. The displacements are damped out because the viscous damper is acting in the x-direction during the whole motion.



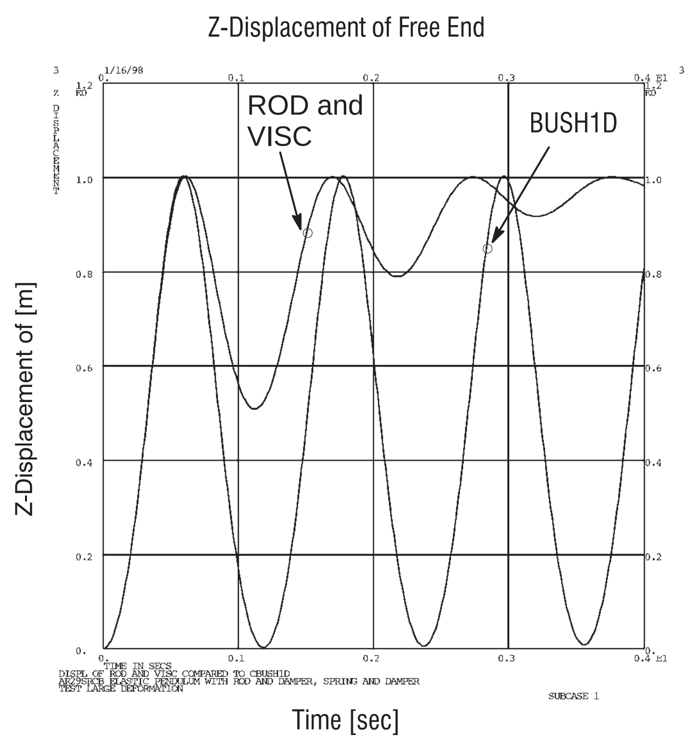


Figure 3-2. Displacement of Swinging Pendulum, Model with Rod/Visc Element Versus Model with BUSH1D Element.

In the second model, we use a BUSH1D element. The element has a linear stiffness (K) and a viscous damper (C) with the same values as in the first model. In the BUSH1D element, the spring and damper rotate. The relative axial displacements and velocities are small because of the high axial stiffness of the element. The rotating damper has no noticeable effect on the overall motion of the pendulum because it damps only the small relative deformations. The x- and z-displacements of the free end are not damped, see Figure 3-2.

The example demonstrates how the answers can change if dampers rotate with the deformation compared to dampers which stay fixed in space.

Example 2: Simple Model of an Automotive Suspension (TPL: ar29be01.dat)

A simplified model of an automotive suspension is shown in Figure 3-3. The suspension is modeled with a stiff beam which has a pinned support at the left end and a shock absorber support in the middle. The shock absorber is modeled with a BUSH1D element. The SHOCKA option on the PBUSH1D property entry is used to model nonlinear damping in tension and compression. A sinusoidal pulse load is applied at the tip of the overhang. The MSC/NASTRAN input file is partially shown in Listing 3-2 and is available in TPL as ar29be01.dat. The beam responds with a rotation of about 45^0 before the motion is damped out. The load and the z-displacement at the tip of the overhang are shown in Figure 3-4. The example demonstrates that the new BUSH1D element can undergo large deformation and can simulate complex nonlinear force deflection laws.

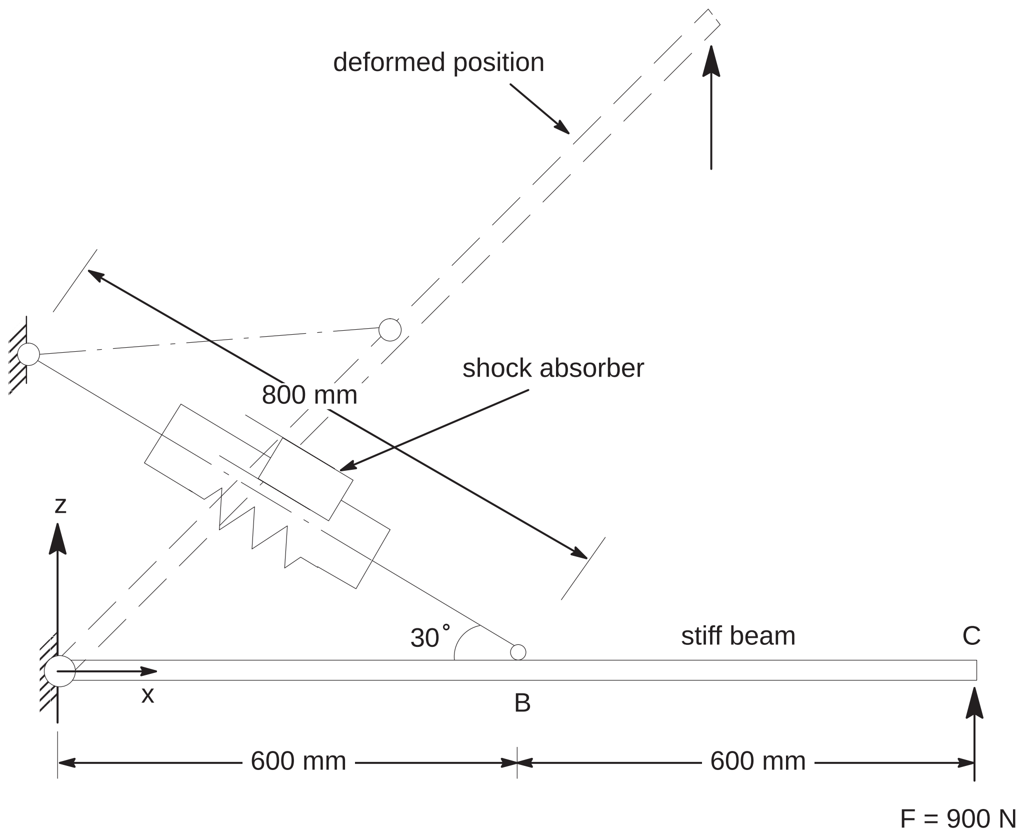


Figure 3-3. Simple Model of an Automotive Suspension.

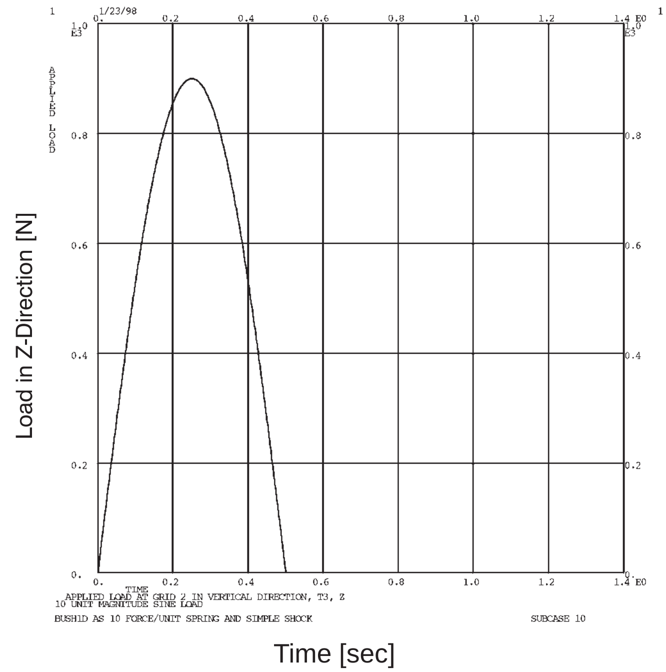
Listing 3-2. MSC/NASTRAN Input File for the Automotive Suspension.

```
$
$*********************************************
$   Test of nonlinear shock absorber
$   under large deformation
$********************************************
TIME 1000
SOL  129
$********************************************
CEND
$
TITLE = Shock absorber and spring loaded with sine pulse
LABEL = CBUSH1D with SHOCKA option and SPRING option
.
.
.
$
$**********************************************************************
*
BEGIN BULK
$
```

Listing 3-2. MSC/NASTRAN Input File for the Automotive Suspension. (Cont.)

```
PARAM, POST, –1
PARAM,AUTOSPC,YES
PARAM,MAXRATIO,1.E+8
PARAM   LGDISP  1
PARAM   LANGLE  2
$
$  CBUSH1D element (SHOCKA option with TABLE for S(u)
$                   and TABLE for SPRING constant)
$
CBUSH1D, 100, 101, 8, 19
$
$  CVT =  0.2   damping for tension
$  CVC =  0.4   damping for compression
$  S(u) = 1.0   constant scale factor
$  EVT =  1.0   exponent of velocity for tension
$  EVC =  0.5   exponent of velocity for compression
$
PBUSH1D, 101,  10.0, 0.4,    ,    ,    ,    , ,+PB1
+PB1, SHOCKA, TABLE, 0.2, 0.4, 1.0, 0.5, 999, , +PB2
+PB2, SPRING, TABLE, 998
$
TABLED1,    999,  ,  ,  ,  ,  ,  ,  , +TB999
+TB999, –1000., 1.0, 0.0, 1.0, 1000., 1.0, ENDT
$
TABLED1,   998,  ,  ,  ,  ,  ,  ,  , +TB998
+TB998, –1000., –10000., 0.0, 0.0, 1000., 10000., ENDT
$
.
.
.
$
ENDDATA
```

Sinusoidal Pulse Load at Tip C



Displacement at Tip C in the Z-Direction

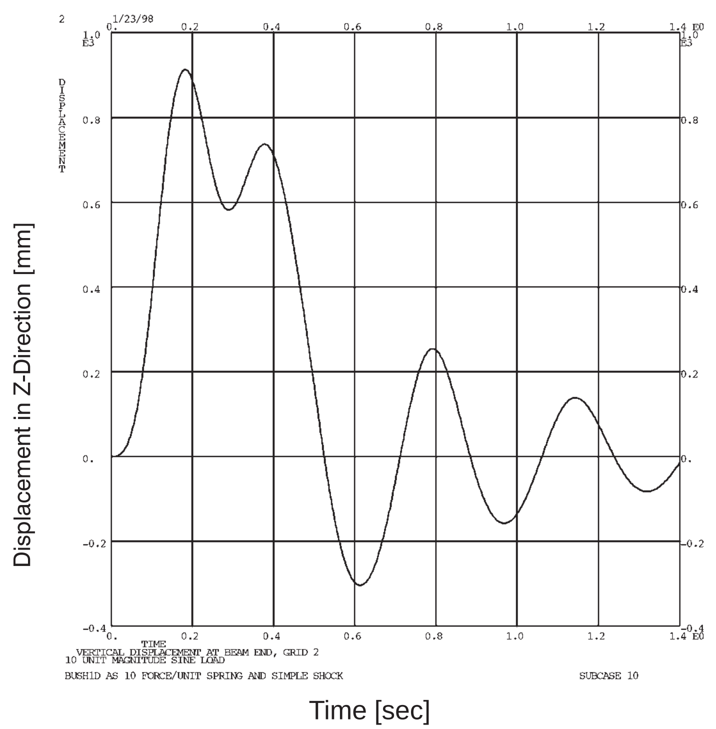


Figure 3-4. Load and Displacement at Tip C for the Automotive Suspension Model.

3.2 Generic Control System

Introduction

MSC/NASTRAN's capability to include control surfaces in a static aeroelastic trim analysis has been greatly expanded in V70.5. Among the features added for this release are:

1. Aerodynamic hinge moment coefficients.
2. Aerodynamic hinge moments for the trimmed vehicle.
3. The ability to schedule a control surface position as a function of angle of attack and/or Mach number.
4. The ability to impose limits on a control surface's position and hinge moment.
5. The ability to specify redundant control surfaces and allow MSC/NASTRAN to blend the amount of deflection of each based on a newly implemented generic control law.

The addition of features 3, 4 and 5 in the above list requires a substantial revision in the method used to trim the aircraft. Figure 3-5 provides a flow diagram of the new algorithm. First a check is made to see if there are scheduled or redundant surfaces. If no scheduled or redundant surfaces are present, the trim solution reverts back to the linear trim solution that has been used in previous versions. If there are redundant control surfaces, but no scheduling, a newly developed blending algorithm is used to create a single controller for each axis that requires control. A linear trim analysis is then performed on the reduced problem.

Figure 3-5 shows two situations where trim is performed using an optimization technique. If scheduled control surfaces are present, an iterative solution minimizes the trim equation given in Eq. 2-77 in the *MSC/NASTRAN Aeroelastic Analysis User's Guide*:

$$\begin{bmatrix} ZZX \\ IP \\ AEL \end{bmatrix} \{u_x\} = \begin{Bmatrix} PZ \\ Y \\ O \end{Bmatrix} \qquad (3\text{-}1)$$

In order to simplify the discussion, this write-up replaces this equation with an equivalent form of:

$$[A]\{u_F\} = \{B\} \tag{3-2}$$

where u_F contains the portion of the u_X vector that is free to vary and the B vector includes the effects of the fixed, linked and scheduled variables. Matrix A is of size *nr* by *nf*, where *nr* is the number of supported degrees-of-freedom and *nf* is the number of free variables.

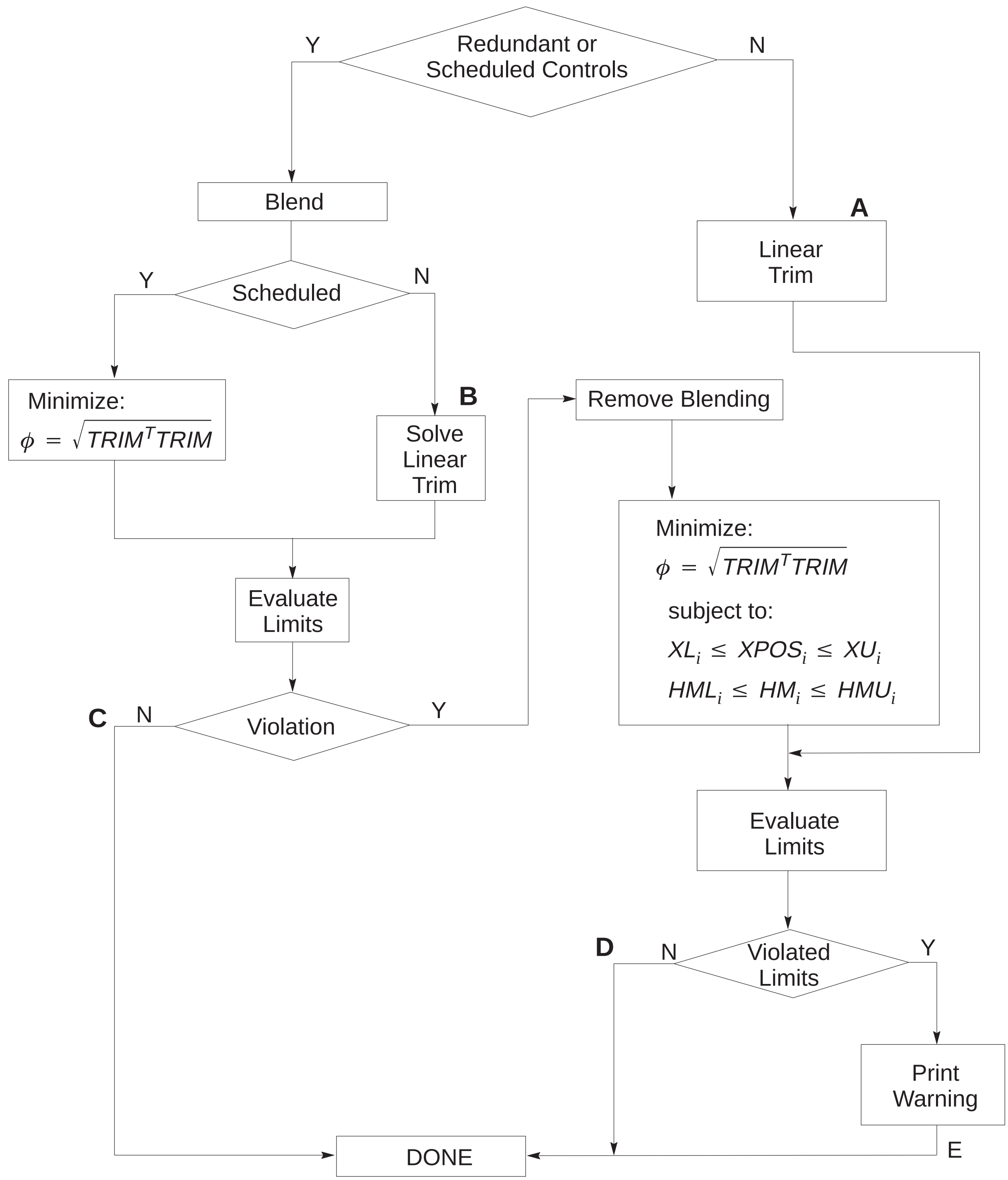


Figure 3-5. Generic Control System Trim Algorithm.

The optimization algorithm solves these equations by minimizing an objective function that is made up of terms that are the difference between the right and left hand sides of the above equation:

$$\phi = \sqrt{\sum_{i=1}^{nr} (Trim_i)^2} \quad (3\text{-}3)$$

where

$$Trim_i = -b_i + \sum_{j=1}^{nf} a_{ij} u_{F_j} \quad (3\text{-}4)$$

The first optimization shown in Figure 3-5 essentially performs an unconstrained optimization with the scheduled control surfaces being a function of the unknown angle of attack. The second optimization is performed if either the linear blended solution or the unconstrained scheduled optimization solution results in violated position or hinge moment limits. In this case, the blending algorithm is removed and a constrained optimization task is performed to produce trim variables that both satisfy the imposed limits and Eq. (3-2). The discussion of the output for this new capability that is given below helps you in determining which path the trim algorithm took and allows you to determine if the algorithm was successful.

Benefits

This provides the aeroelastic analyst with a greater insight into the derived results. It is also a tool for the preliminary design of a flight control system. The increased insight comes from the availability of the hinge moment data that quantify the loads produced on the flight control surfaces.

Input

The Generic Control System Package has added one new Case Control command (CSSCHD), one new Bulk Data entry (CSSCHD) and has extended the AESURF entry. The CSSCHD Case Control command simply invokes control surface scheduling data that are input on the CSSCHD Bulk Data entry. This new command can be at the subcase level so that you can experiment with different schedules in the same run.

The CSSCHD Bulk Data entry invokes two or three AEFACT Bulk Data entries that provide the actual scheduling data. The format of the CSSCHD Bulk Data entry is as follows (see the Version 70.5 *MSC/NASTRAN Quick Reference Guide* for more details):

CSSCHD

Format:

1	2	3	4	5	6	7	8	9	10
CSSCHD	SID	AESID	LALPHA	LMACH	LSCHD				

Example:

CSSCHD	5	ELEV	12	15	25				

Field	**Contents**
SID	Set identification number (Integer > 0)
AESID	ID of an AESURF Bulk Data entry to which the schedule is being applied.
LALPHA	ID of an AEFACT Bulk Data entry containing a list of angles of attack (in radians) at which schedule information is provided. (Integer > 0, default=no angle information provided.)
LMACH	ID of an AEFACT Bulk Data entry containing a list of Mach numbers at which schedule information is provided. (Integer > 0, default=no Mach information provided.)
LSCHD	ID of an AEFACT Bulk Data entry which contains the scheduling information.

The angle of attack and control surface rotation values are input in *radians.* If both Mach number and angle of attack data are given, the schedule for the first Mach number and all the angles of attack is given first.

You need to be aware of several subtleties associated with this entry. If the scheduled data are only a function of Mach number, the trim algorithm can simply compute the control surface position based on the schedule and on the Mach number for the subcase given on the TRIM Bulk Data entry. If the scheduled data are a function of angle of attack and the angle of attack is part of the trim solution, the trim needs to be carried out in an iterative fashion. A special case occurs when the schedule is a function of angle of attack but the angle of attack is fixed on the TRIM Bulk Data entry. In this case, it is again possible to directly compute the control surface position with no requirement for an iterative solution.

The format of the revised AESURF entry is as follows (see the Version 70.5 *MSC/NASTRAN Quick Reference Guide* for more details):

AESURF

Format:

1	2	3	4	5	6	7	8	9	10
AESURF	ID	LABEL	CID1	ALID1	CID2	ALID2	EFF		
	CREFC	CREFS	PLLIM	PULIM	HMLLIM	HMULIM	TQLLIM	TQULIM	

Example:

AESURF	6001	ELEV	100	100	200	200	0.6		
	10.0	180.0			-1.2E4	1.2E4	20	30	

Field	**Contents**
ID	Identification number of an aerodynamic trim variable degree-of-freedom. (Integer > 0)
LABEL	A character string of up to eight characters used to identify the control surface. (Character)
CIDi	Identification number of a rectangular coordinate system with y-axis that defines the hinge line of the control surface component. (Integer > 0)
ALIDi	Identification of an AELIST Bulk Data entry that identifies all aerodynamic elements that make up the control surface component. (Integer > 0)
EFF	Control surface effectiveness. (Real $\neq 0.0$; Default $= 1.0$)
CREFC	Reference chord length for the control surface. (Real > 0.0, Default=1.0)
CREFS	Reference surface area for the control surface. (Real > 0.0, Default=1.0)
PLLIM,PULIM	Lower and upper deflection limits for the control surface in radians. (Real, Default= $\pm \pi/2$)
HMLLIM,HMULIM	Lower and upper hinge moment limits for the control surface in force-length units. (Real, Default=no limit)
TQLLIM,TQULIM	Set identification numbers of TABLEDi entries that provide lower and upper deflection limits for the control surface as a function of the dynamic pressure. (Integer > 0, Default=no limit)

The revisions are confined to the new continuation portion of the entry so that existing decks do not require modification in order to execute. You may wish to add the two attributes, CREFC and CREFS, that define the reference chord and area of the control surface so that the new hinge moment coefficients are nondimensionalized in a consistent way with your definition for your vehicle.

The position and hinge moment limit inputs are optional and are of practical use only when redundant control surfaces are specified. This is because the trim algorithm needs to have enough free degrees-of-freedom (i.e, unknown trim variables) both to trim the airplane and satisfy the imposed limits. For example, if the only control surface that controls pitch is the elevator, the position of the elevator required for trim is determined so that specifying a limit does not affect the solution. On the other hand, if two controllers, such as an elevator and an all moveable stabilizer are present, it may be possible to trade off the amount each deflects so to satisfy the imposed limits.

Output

The output for static aeroelastic analyses has been modified in two areas, and additional printouts have been provided to provide insight into the new trim capabilities. Section 4.3 of this document discusses further changes in the output from aeroelastic analyses that were caused by a reformulation of the data structures used for aerodynamic geometry.

The first modification is in the printing of the stability derivatives. Section 7.1 of the *MSC/NASTRAN Aeroelastic Analysis User's Guide* provides a description of the stability derivative prints for previous versions and this description is still valid. The change is that there is now additional output that provides the hinge moment coefficient data. Figure 3-6 provides an example of the new output that is obtained when running the forward swept wing example of Section 7.1 of the *MSC/NASTRAN Aeroelastic Analysis User's Guide* with CREFC for the single ELEV control surface set to 10.0 and CREFS set to 200.0. Each AESTAT and AESURF Bulk Data entry produces its own row of hinge moment coefficients. The "Intercept" data represents the hinge moment coefficients that are produced by user input downwashes and/or pressures. The hinge moments are computed directly from the aerodynamic model, rather than from the structural model that is used for the splined stability derivative data. For this reason, there is no comparison of the splined and unsplined hinge moment coefficients as there is for the stability derivatives. There are values for the restrained and unrestrained coefficients.

Repetitive checks of the printed unrestrained hinge moment coefficients have indicated their correctness, but the final check on them to predict the total hinge moment in the trim condition has not been successful. The reason for this discrepancy is being investigated further.

```
N O N - D I M E N S I O N A L   H I N G E   M O M E N T   D E R I V A T I V E   C O E F F I C I E N T S

CONTROL SURFACE = ELEV        REFERENCE CHORD LENGTH =  1.000000E+01     REFERENCE AREA =  2.000000E+02

                     TRIM VARIABLE            RIGID                        ELASTIC
                                                                    RESTRAINED      UNRESTRAINED
                       INTERCEPT          -7.706106E-05             -7.767072E-05   -7.760999E-05
                       ANGLEA              1.551577E-02              1.510351E-02    1.494753E-02
                       PITCH              -3.062173E-01             -3.066317E-01   -3.070683E-01
                       URDD3               0.000000E+00              2.263822E-05    0.000000E+00
                       URDD5               0.000000E+00              2.893892E-04    0.000000E+00
                       ELEV                5.966925E-02              5.960630E-02    5.961495E-02
```

Figure 3-6. Hinge Moment Coefficients.

The second modification in the output is in the “Aerostatic Data Recovery.” The first new piece of information is on the trim solution used. There are five possible trim solutions that can occur and these are given here in Figure 3-7 with the letter corresponding to that given on Figure 3-5:

```
A.  TRIM ALGORITHM USED:  LINEAR TRIM SOLUTION WITHOUT REDUNDANT CONTROL SURFACES
B.  TRIM ALGORITHM USED:  LINEAR TRIM SOLUTION WITH REDUNDANT CONTROL SURFACES
C.  TRIM ALGORITHM USED:  SUCCESSFUL NONLINEAR TRIM SOLUTION WITHOUT EXPLICIT CONSIDERATION OF CONTROL SURFACE LIMITS
D.  TRIM ALGORITHM USED:  SUCCESSFUL NONLINEAR TRIM SOLUTION WITH EXPLICIT CONSIDERATION OF CONTROL SURFACE LIMITS
E.  TRIM ALGORITHM USED:  UNSUCCESSFUL NONLINEAR TRIM SOLUTION WITH EXPLICIT CONSIDERATION OF CONTROL SURFACE LIMITS
```

Figure 3-7. Trim Algorithm Selection.

Figure 3-8 shows an example of the revised output, including the new trim selection method. There are two tables in Figure 3-8. The first gives the aeroelastic trim variables and is similar to an existing table that simply listed the variable name and its value. The new table adds columns which give the user defined ID, the type (where AESTAT entries are type “RIGID BODY” and AESURF entries are type “CONTROL SURFACE”) and the trim status. There are four types of trim status: FREE, FIXED, SCHEDULED and LINKED and they indicate that the variable value was free to vary, fixed by the user on the TRIM Bulk Data entry, scheduled using the CSSCHD entry or linked via the AELINK entry.

```
                    A E R O S T A T I C   D A T A   R E C O V E R Y   O U T P U T   T A B L E S

                              MACH = 9.000000E-01              Q = 2.400000E+03

     TRIM ALGORITHM USED: LINEAR TRIM SOLUTION WITHOUT REDUNDANT CONTROL SURFACES.

                                         AEROELASTIC TRIM VARIABLES

                          ID     LABEL              TYPE        TRIM STATUS      VALUE OF UX

                         501     ANGLEA          RIGID BODY          FIXED      3.151200E-03
                         502     PITCH           RIGID BODY          FIXED     -2.511685E-19
                         503     URDD3           RIGID BODY           FREE     -1.382048E+01
                         504     URDD5           RIGID BODY           FREE      2.109222E-01
                         505     CANARD       CONTROL SURFACE        FIXED      4.363320E-01
                         510     ELEV         CONTROL SURFACE        FIXED      0.000000E+00
                         511     SIDES           RIGID BODY          FIXED     -5.592523E-18
                         512     YAW             RIGID BODY          FIXED      1.983739E-17
                         513     ROLL            RIGID BODY          FIXED     -6.398960E-19
                         514     URDD2           RIGID BODY           FREE      3.563613E+00
                         515     URDD4           RIGID BODY           FREE      5.422129E-01
                         516     URDD6           RIGID BODY           FREE     -7.055259E-01
                         517     INBDAIL      CONTROL SURFACE        FIXED      0.000000E+00
                         518     OUTBDAIL     CONTROL SURFACE        FIXED      0.000000E+00
                         519     INBDRUD      CONTROL SURFACE        FIXED      4.363320E-01
                         520     OUTBDRUD     CONTROL SURFACE        FIXED      4.363320E-01

                              CONTROL SURFACE POSITION AND HINGE MOMENT RESULTS

                    ACTIVE LIMITS ARE FLAGGED WITH AN (A),  VIOLATED LIMITS ARE FLAGGED WITH A (V).

                                     POSITION                                          HINGE MOMENT
  CONTROL SURFACE      LOWER LIMIT      VALUE        UPPER LIMIT        LOWER LIMIT       VALUE        UPPER LIMIT
    CANARD          -1.700000E-01   4.363320E-01   1.700000E-01 V   -1.000000E+04   3.161594E+05   1.000000E+04 V
    ELEV            -1.570796E+00   0.000000E+00   1.570796E+00          N/A        -1.002511E+03        N/A
    INBDAIL         -1.570796E+00   0.000000E+00   1.570796E+00          N/A         1.534836E+02        N/A
    OUTBDAIL        -1.570796E+00   0.000000E+00   1.570796E+00          N/A         1.534625E+02        N/A
    INBDRUD         -5.000000E-01   4.363320E-01   5.000000E-01     -2.000000E+06  -2.151824E+04   2.000000E+06
    OUTBDRUD        -8.760000E-01   4.363320E-01   8.760000E-01          N/A        -2.165500E+03        N/A
```

Figure 3-8. Revised Trim Output and Control Surface Limits.

The second table provides control surface limit results. Each control surface has an upper and lower limit on its position and hinge moment value. The actual values are also listed and flags are provided to quickly identify any active limit (i.e., a response that is not violated but is within 3% of the prescribed limit marked with an (A)) while any violated constraint (i.e., a response that exceeds its limit by more than 0.3% marked with a (V)). Units of the output are radians for the positions and physically consistent units (e.g., newton-meters) for the hinge moments.

An additional output is presented that if optimization is invoked to force the trim equation to zero as stated in the introduction of this section. This table shows how well the optimizer performed. An example of this new output is given in Figure 3-9.

RESULTS OF TRIM OPTIMIZATION

			THRESHOLD	RELATIVE UNBALANCE		
SUPPORT EQUATION	TRIMMED LOAD	RIGHT HAND SIDE LOAD	LOAD	MAGNITUDE	TARGET	SATISFACTION
1	-1.974761D-04	-1.974761D-04	1.000000D-03	-1.000731D-10	7.500000D-03	SATISFIED
2	-9.377154D+03	-9.377154D+03	1.000000D-03	7.177296D-15	7.500000D-03	SATISFIED
3	2.563925D-05	2.563925D-05	1.000000D-03	2.868366D-08	7.500000D-03	SATISFIED
4	-7.444643D+03	-7.444643D+03	1.000000D-03	-9.020860D-13	7.500000D-03	SATISFIED
5	-2.206294D-03	-2.206294D-03	1.000000D-03	-1.665211D-10	7.500000D-03	SATISFIED

Figure 3-9. Report from Trim Optimization.

For each of the trim equations, the table first lists the computed load ($[A]\{u_F\}$ in Eq. (3-2)) and then the corresponding b_i. Constraints are placed on the difference between the two loads and used in the optimization. The trim is considered satisfied whenever the relative difference between the trimmed load and the right hand side load is less than 0.75%. The relative difference is computed from

$$RD_i = \frac{Trim_i}{b_i} \tag{3-5}$$

If RD_i is greater than the 0.75% target, the equation is marked FAILED, and you must decide whether you are willing to accept the discrepancy or need to do further analyses to see why the trim has failed. A special case occurs when the right hand side is small. In this case, a large percentage difference in the trim is probably not significant. Therefore, the trim equation is marked satisfied if the absolute difference between the trimmed and right hand side load is less than the threshold of 0.001.

If you want to obtain additional information on the trim process, you can set DIAG 50 around the ASG module. This produces a large quantity of diagnostic output, and you may be able to discern why the optimization algorithm is having trouble. As with all diagnostic output, this is primarily for the developers at MSC and may be of limited value to a new user.

Limitations

There are several warnings that need to be stated when applying this new capability. The first is that the specification of redundant control surfaces with constraints and/or scheduling necessitates the use of some type of iterative solution of the trim equations. As discussed in the introduction to this section, the method chosen was to pose the trim solution as an optimization task and to use the same optimization algorithm that is currently applied for structural optimization to this new task. This will not always converge to a trim solution, and the obtained solution will sometimes have violated constraints. MSC/NASTRAN will always attempt a solution and print a result. You must seek out messages that indicate what trim solution was used and whether it was successful. The control surface limits table can be scanned to reveal if there are any violated limits.

A second warning is that the use of redundant or scheduled control surfaces is not available for optimization (SOL 200). Obtaining the sensitivities of trim quantities that are determined in an iterative fashion was beyond the scope of the current project. The hinge moment calculations are made in SOL 200, but cannot be used to drive the design process. Constraining trim variables in a design optimization task is a long time SOL 200 capability and continues to be supported.

Example (TPL: ha144fhm.dat)

The following example extends the ha144fhm.dat test case given in Section 6 of the *MSC/NASTRAN Aeroelastic Analysis User's Guide* to demonstrate features of the new capability. The Reference example has three control surfaces: a canard that acts as pitch controller, an aileron and a rudder. The current example has six controllers. An elevator was added to the rear of the canard to provide an additional pitch controller, and the aileron and rudders were split to give inboard and outboard components for each.

The input file for this problem can be found in the test problem library (TPL), provided on your MSC/NASTRAN delivery media as test problem ha144fhm.dat. Figure 3-10 contains extracts from the test deck that invoke the new features. The CANARD control surface (AESURF ID = 505) has a continuation entry that provides the reference chord and surface area for the control surface plus a position limit of $\pm$0.17 radians and a hinge moment limit of $\pm$10,000 in-lbs. The outboard rudder (AESURF ID = 520) has position limits that are a function of dynamic pressure. The associated TABLED1 data shows that the limits vary linearly from $\pm$0.9 radians at q=0.0 down to $\pm$0.8 radians at $\bar{q}$=10,000. The CSSCHD data in the model indicate that the elevator (AESURF=510) is being scheduled, and that the schedule is a function of both Mach number and angle of attack. The schedule is applied at three angles of attack and two Mach numbers, with a total of six position limits provided. It is seen that the elevator deflects 0.01734 radians at Mach=0.0 and zero angle of attack. The results in Figure 3-9 show the results of applying this schedule. The Mach number is 0.90 and the trim angle of attack is 0.01 radians. It is seen that an interpolation of the input data results in an elevator setting of 0.01665 radians (0.954 degrees).

```
$                                                                         $
$                   * CONTROL SURFACE DEFINITION *                        $
$                                                                         $
$        THE AESURF ENTRY DEFINES AN AERODYNAMIC CONTROL SURFACE.         $
$        LISTED ARE THE ALPHANUMERIC NAME OF THE SURFACE, THE ID          $
$        OF A COORDINATE SYSTEM THAT DEFINES THE HINGE LINE AND           $
$        THE ID OF AN AELIST ENTRY.                                       $
$                                                                         $
$       ID      LABEL   CID1    ALID1   CID2    ALID2                     $
AESURF  505     CANARD  90      1000    90      2000
        2.5     15.     -.17    .17     -1.0E4  1.0E4
AESURF  510     ELEV    510     510     510     610
AESURF  517     INBDAIL 110     1100    210     2200
AESURF  518     OUTBDAIL110     1200    210     2100
AESURF  519     INBDRUD 301     3000
        2.5     9.625   -.50    .50     -2.0E6  2.0E6
AESURF  520     OUTBDRUD301     3100
        2.5     9.625                                   1520    1530
TABLED1 1520
        0.0     -0.9    10000.  -0.8    ENDT
TABLED1 1530
        0.0     0.9     10000.  0.8     ENDT

CSSCHD  3       510     10      20      30
AEFACT  10      0.0     0.017   0.173
AEFACT  20      0.0     1.0
AEFACT  30      0.01734 0.0100  0.005   0.0200  0.01500 0.005
CSSCHD  20      510     100     20      300
AEFACT  100     0.0     0.017   0.173
AEFACT  300     0.173   0.150   0.120   0.05    0.02    0.017
```

Figure 3-10. Examples of Revised AESURF Input and New CSSCHD Input.

3.3 Nonlinear Transient Radial Gap - NLRGAP

Introduction

MSC/NASTRAN has had the ability to input nonlinear loads in transient response via the NOLINi entries for many years. The NOLINis have allowed users to model axial contact, gyroscopic forces and many other effects. A new nonlinear load entry called the NLRGAP has been created specifically for the purpose of modeling circular or radial gap contact of a shaft in a housing.

The NLRGAP is an idealization of the contact between any circular shaft enclosed by a circular housing (Figure 3-11). In this idealization, grid point A represents the outer surface of the shaft while grid point B represents the inner surface of the housing. Grid points A and B are coincident and must have a parallel displacement coordinate system. A rudimentary friction capability is included to account for the friction-induced torque and friction-induced lateral loads that occur with a spinning shaft. However, friction can be neglected, in which case the solution is consistent with that for a nonrotating shaft in a frictionless housing.

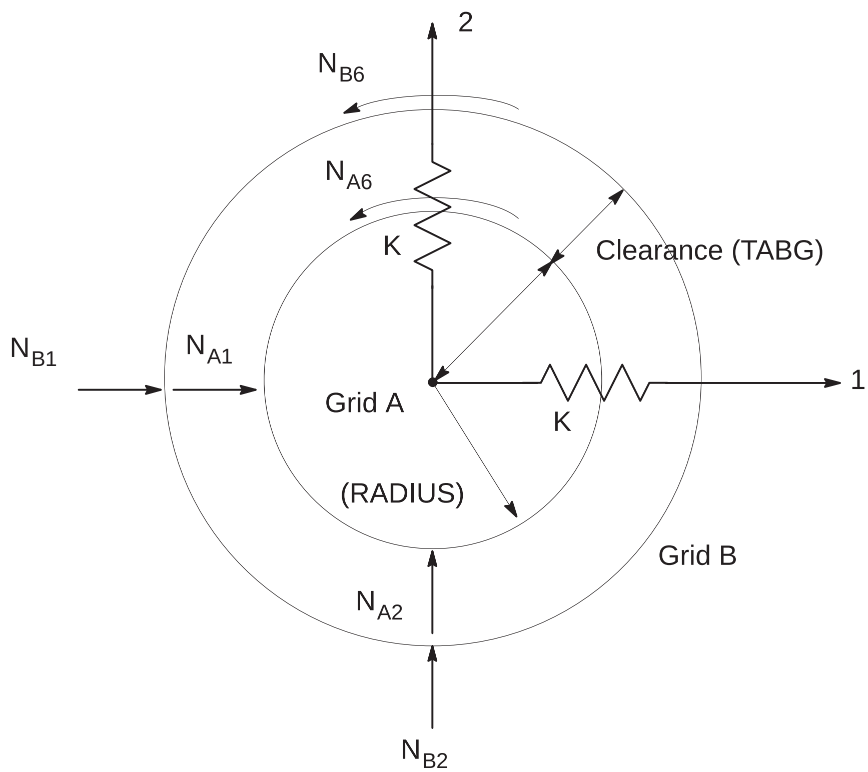


Figure 3-11. Radial Gap Orientations and Nonlinear Load Sign Conventions.

The equations underlying the NLRGAP measure the relative radial displacement between the shaft grid and the housing grid in the displacement coordinate system of grid points A and B. When the relative radial displacement is greater than the clearance (i.e. when contact has occurred), contact forces as well as frictional forces are automatically applied to the grids. The contact and frictional forces are dependent on the contact stiffness and the coefficient of friction.

NLRGAP Underlying Equations

Definitions

$K(t,u)$	Contact stiffness, function of time or displacement
$G(t)$	Radial clearance, function of time
$\mu(t)$	Coefficient of friction, function of time. Positive value implies shaft rotation from axis 1 towards axis 2.
R	Shaft Radius for friction induced torque calculation

Equations

The NLRGAP is internally composed of the following equations. It is helpful to visualize Grid A as belonging to the shaft and Grid B as belonging to the housing. Consider each grid to have DOF 1 and 2 in the plane of action (e.g. DOF 1 = X and DOF 2 = Y for XY orientation).

The element relative displacement in directions 1 and 2 are:

$$\Delta U_1 = U_{A1} - U_{B1} \tag{3-6}$$

$$\Delta U_2 = U_{A2} - U_{B2} \tag{3-7}$$

The relative radial displacement is:

$$\Delta r = \sqrt{\Delta U_1^2 + \Delta U_2^2} \tag{3-8}$$

The force in the gap when it is open $(\Delta r \le G(t))$ is zero. When the gap is closed $(\Delta r > G(t))$, the penetration is defined as:

$$P = \Delta r - G(t) \tag{3-9}$$

The nonlinear gap forces are:

$$S = K(t,u)\left(1 - \frac{G(t)}{\Delta r}\right) \tag{3-10}$$

$$N_{A1} = -\Delta U_1 S + \Delta U_2 S\mu(t) \tag{3-11}$$

$$N_{B1} = \Delta U_1 S - \Delta U_2 S\mu(t) \tag{3-12}$$

$$N_{A2} = -\Delta U_2 S - \Delta U_1 S\mu(t) \tag{3-13}$$

$$N_{B2} = \Delta U_2 S + \Delta U_1 S\mu(t) \tag{3-14}$$

If the shaft radius R is input, then the following friction induced torque loads are generated.

$$N_{A6} = -R\Delta r S\mu(t) \tag{3-15}$$

$$N_{B6} = R\Delta r S\mu(t) \tag{3-16}$$

Benefit

Allows you to model contact between two coaxial cylinders, for example shaft and housing.

Input

A new Bulk Data entry, NLRGAP, is used to model the radial gap. A brief description of NLRGAP Bulk Data entry follows (see the *MSC/NASTRAN Quick Reference Guide* for more details).

NLRGAP Input Data Summary

NLRGAP

1	2	3	4	5	6	7	8	9	10
NLRGAP	SID	GA	GB	PLANE	TABK	TABG	TABU	RADIUS	

Field	Contents
SID	Nonlinear load set identification number. (Integer >0)
GA	Inner (e.g., shaft) grid for radial gap. (Integer >0)
GB	Outer (e.g., housing) grid for radial gap. (Integer >0)
PLANE	Radial gap orientation plane: XY, YZ, or ZX in grid point A coordinate system. (Character, Default = XY)
TABK	Table ID of gap stiffness vs. time. (Integer >0) Table ID of gap stiffness vs. penetration. (Integer <0)
TABG	Table ID for radial gap clearance as function of time. (Integer >0)
TABU	Table ID for radial coefficient of friction as function of time. (Integer >0)
RADIUS	Shaft radius. (Real ≥ 0.0, Default = 0.0)

The contact stiffness can be input either as a function of time or as a function of penetration. In both cases, the stiffness curves are input on TABLEDi Bulk Data entries which are referenced by the NLRGAP entry. Inputting stiffness as a function of time allows the user to model special situations where structural changes over time cause the contact stiffness to vary over time. Inputting stiffness as a function of penetration allows the user to input a nonlinear stiffness curve which might be required, for example, if the shaft or housing were covered by some nonlinear-elastic material.

As with stiffness, both the coefficient of friction and the clearance can be functions of time. The time-varying coefficient of friction might be used where the quality of lubrication varied with time, or if experimental data showed that the surface characteristics of the shaft and housing varied with time. The clearance can be input as a function of time to model situations where an accelerating, spinning rotor expands due to centripetal loads.

The NLRGAP entry can also approximate the torque on the shaft and housing due to friction. If this effect is desired, the RADIUS field of the NLRGAP must be specified. If not specified, then friction effects will still be included in the lateral loads as shown in equations 6 through 9 below. The torque loads, equations 10 and 11, will not be computed nor applied if RADIUS is input as 0.0 or left blank.

As with the NOLINs, the NLRGAP is selected by the NONLINEAR Case Control command.

Output

The output is the nonlinear loads applied to the NLRGAP grids to simulate the contact. The loads can be recovered with the NLLOAD Case Control command.

Guidelines

1. There should be mass on both grids of the NLRGAP. Otherwise erratic results and divergence can occur. A small amount of damping (e.g. CDAMPi) between the two grids may improve stability for some problems.
2. The two grids listed on the NLRGAP should be coincident and have parallel displacement coordinate systems.
3. The friction capability is somewhat arbitrary in sign. Referring to Figure 3-11, a positive coefficient of friction is consistent with shaft rotation from axis 1 towards axis 2 (counter-clockwise). If the shaft is rotating in the opposite direction, then a negative coefficient of friction must be input in order for the friction induced loads to have the proper signs.
4. In SOL 129, setting ADJUST=0 on the TSTEPNL entry will turn off the adaptive time step. This is useful if large contact stiffness is causing an excessively small time step.
5. As with the NOLINs, the forces applied to the NLRGAP grids to simulate the radial contact are based upon the relative displacements of the grids from the previous time step. Therefore, the larger the contact stiffness, the smaller the necessary time steps. The smaller the contact stiffness, the larger the penetration. Use the smallest contact stiffness possible to model the contact to the accuracy desired. The time step may need to be decreased by 10 times or more compared to the same model without NLRGAPs and NOLINs.
6. The NLRGAP is based upon small displacement theory. Therefore, for most accurate results, especially with respect to friction induced torque, the clearance should be small relative to the housing diameter. The friction induced torque applied to grids A and B will be equal and opposite in direction.

Limitations

1. The NLRGAP can only be positioned in the XY, YZ or ZX plane of the displacement coordinates of the referenced grids.
2. The DOF in the plane of the NLRGAP must be in the d-set. This means that these DOF cannot be constrained by SPCs or be dependent DOF on MPCs and RBE type elements.
3. The NLRGAP can only be used in solution sequences 109, 112 and 129.

Example: Analysis of an Unbalanced Rotor in a Rigid Casing (TPL: nlrgap.dat)

Figure 3-12 shows a 40-inch shaft with 2-inch diameter. One end of the shaft is fixed while the other end is free with an attached 10-inch diameter rotor. The clearance between the rotor and the rigid housing is .002 inches, and is constant over time. The contact stiffness is estimated at 1.E+6 pounds/inch and is set constant. The coefficient of friction is set constant at .1. The shaft is rotating at 15 Hz, and an imbalance in the rotor causes a rotating centripetal load of 3 lb. The purpose of the analysis is to determine if the imbalance causes the shaft to make contact with the rigid housing.

Note that significant gyroscopic forces caused by the rotor will occur. However, these forces will be neglected in this example in order to keep this example simple. Alters similar to ridgyroa in the SSSALTER library provide one possible method of incorporating gyroscopic effects.

SOL 129 is used to solve this problem, although SOL 109 could just as easily have been chosen, since all elements in the model are linear. Bar elements are used to model the shaft, and a CONM2 is used to model the mass properties of the rotor.

The NLRGAP is composed of shaft grid 9 and housing grid 10. The housing is assumed rigid and grounded. Therefore, grid 10 is connected to ground by stiff springs to simulate a rigid housing. Furthermore, a concentrated mass is connected to the housing grid since mass is required at both NLRGAP grid points. Care should be taken in this case to ensure that the mass and stiffness combination used to ground the housing does not result in a natural frequency for the housing in the dynamic range of interest of the shaft. Otherwise, unintended coupling will occur.

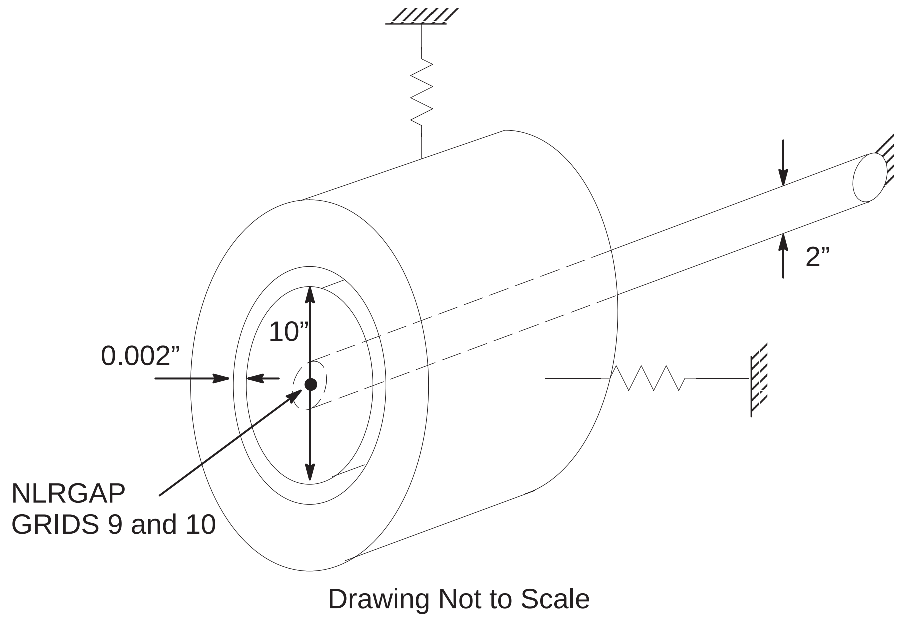


Figure 3-12. Example 1, Unbalanced Rotor in Rigid Housing.

The analysis was run up to .03 seconds with a constant time step of 0.0001 seconds. Simple transfer functions and EPOINTs were used to measure the relative displacements of the shaft and housing grids in the x and y directions. This is not required for the NLRGAP, but was inserted only to easily verify that the shaft did not penetrate the housing significantly. Figures 3-13 and 3-14 show the relative displacements between the shaft and housing grids in the x and y directions. Figures 3-15 and 3-16 show the resulting nonlinear loads applied to the shaft grid in the x and y directions. These loads are non-zero when the radial relative displacement of the shaft and housing grids is greater than the clearance. Stated more simply, these loads are non-zero when the gap is closed.

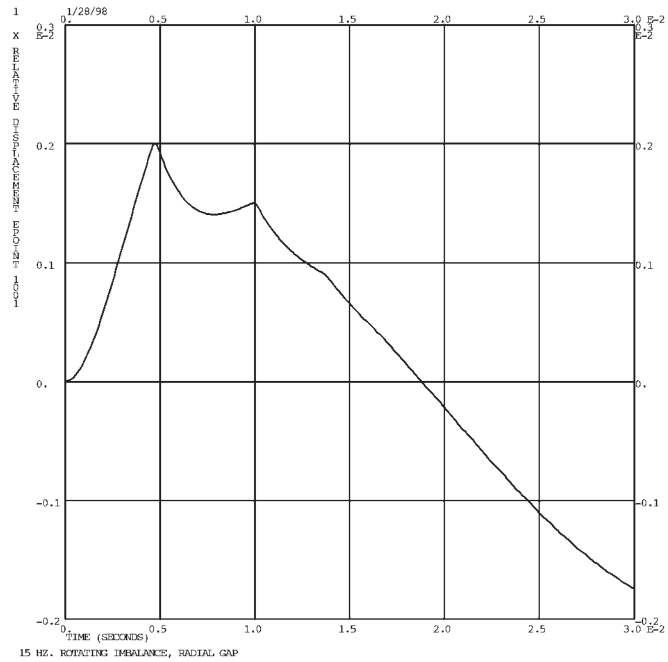


Figure 3-13. Shaft/Housing Relative X Displacement.

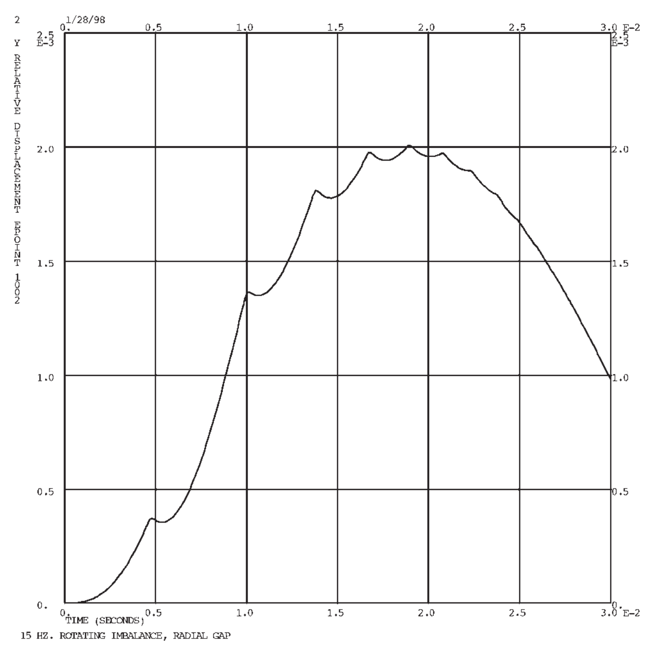


Figure 3-14. Shaft/Housing Relative Y Displacement.

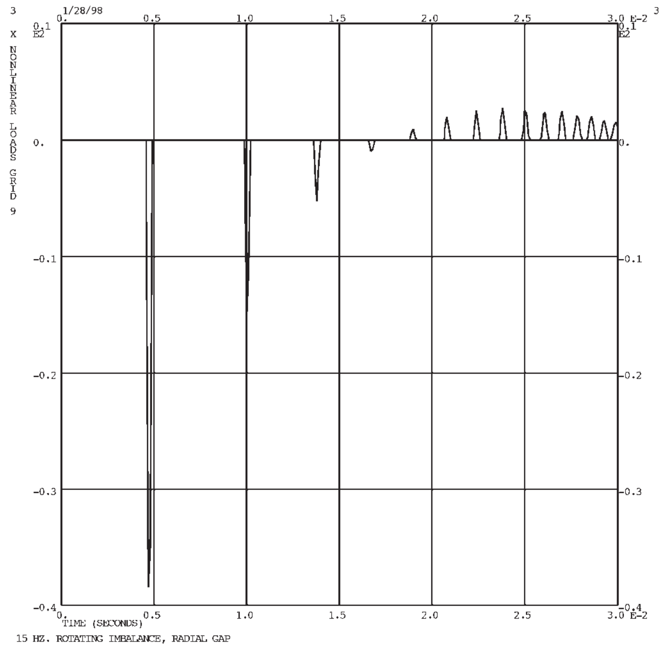


Figure 3-15. Shaft Nonlinear Loads in X Direction.

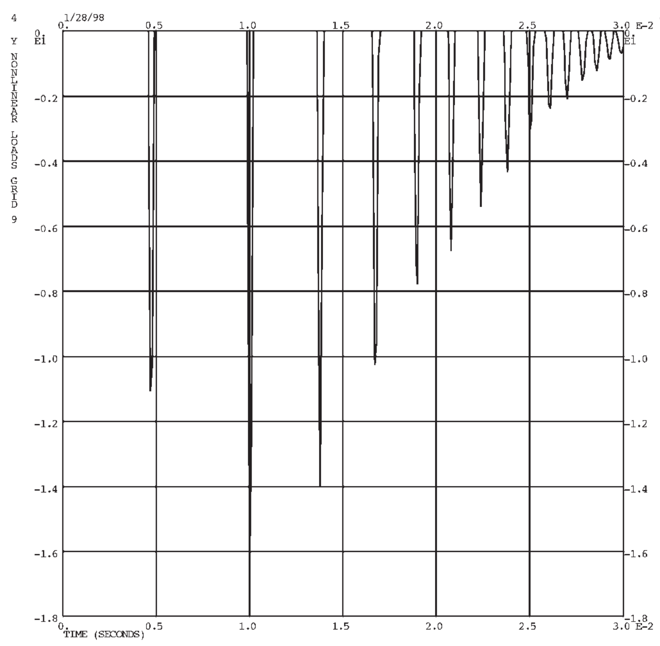


Figure 3-16. Shaft Nonlinear Loads in Y Direction.

3.4 Surface Interface Elements

Introduction

In finite element analysis the issue of connecting dissimilar meshes often arises, especially when refinement is performed. One method of connecting these dissimilar meshes is to use interface elements. An example in which patches of elements have been removed from the global model and replaced by denser patches for local details is shown in Figure 3-17, where the boundaries of the patches are bold.

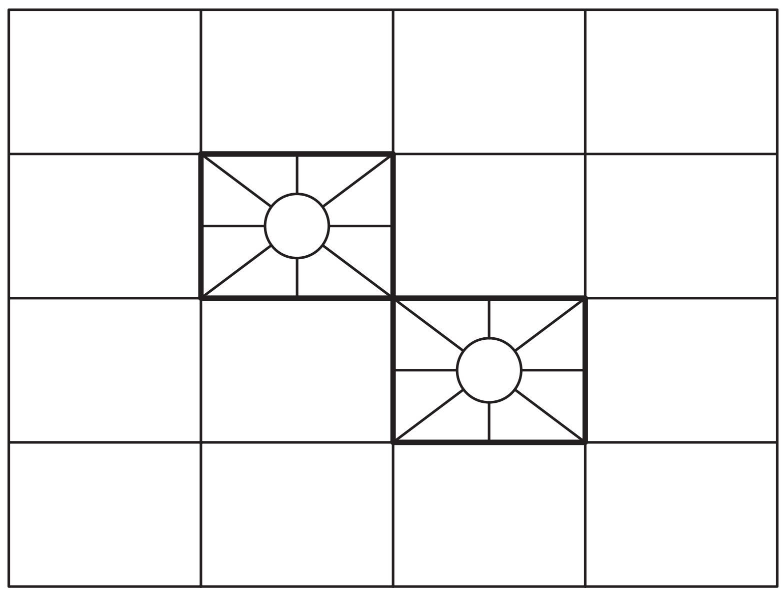


Figure 3-17. Example of Dissimilar Mesh from Global/Local Analysis.

Curve interface elements, which join p-elements along edges, were introduced in V69. See the *MSC/NASTRAN Release Guide*, Version 69 for more details on the curve interface element. Surface interface elements, which join p-elements over faces, are available in V70.5.

The interface elements use a hybrid variational formulation with Lagrange multipliers, developed by NASA Langley Research Center. There are displacement variables defined on the interface element, in order to avoid making the interface too stiff, such as a rigid element. There are also Lagrange multipliers defined on each boundary, which represent the forces between the boundaries and the interface element. This formulation is energy-based and results in a compliant interface.

Implementation

Surface interface elements have been implemented for p-elements:

- The surface interface elements connect p-element faces. These can be either faces of solid p-elements or shell p-elements, and the faces may be either quadrilateral or triangular.
- The interface elements are geometry-based. They use the same geometry as the p-element boundaries.
- The interface elements connect only corresponding displacement fields. They do not have kinematic constraints, such as connecting shell rotations to solid translations.
- The interface elements have been implemented for linear statics (SOL 101) and normal modes (SOL 103). They are not available in the other linear solution sequences, because of the absence of shell p-elements.

Benefits

This is a technology that will help perform global/local analysis and provide for mesh transitions in future versions of MSC/NASTRAN.

Input

Similar to the curve interface elements, three new Bulk Data entries, GMBNDS (Geometric Boundary – Surface), GMINTS (Geometric Interface – Surface), and PINTS (Properties of Geometric Interface – Surface), have been implemented for specifying the surface interface elements. These entries define the subdomain boundaries, the interface elements, and the interface element properties, respectively:

- There are three methods of defining the subdomain boundaries of solid or shell p-element faces (GMBNDS). For the surface interface, each boundary may be defined using the GMSURF with which the finite element faces are associated; the FEFACEs defining the finite element faces; or in the most basic form, the GRIDs over the finite element faces.

- Once the boundaries have been defined, they must be associated with the interface elements. This is accomplished by referencing the boundaries in the interface element definition (GMINTS).
- Since the interface elements consist only of the differences in displacement components weighted by the Lagrange multipliers, there are no conventional element or material properties. The property Bulk Data entry (PINTS) specifies a tolerance for the interface elements, which defines the allowable distance between the subdomain boundaries; and a scaling factor, which may improve the conditioning of the Lagrange multipliers.

Output

The interface elements have no output of their own. However, they do cause changes in the customary output:

- Internal node, edge, and face degrees-of-freedom are generated for the interface elements. Because of their formulation, use of the parameter AUTOSPC, which is the default in the allowable solution sequences, detects them as singular. This can make the Grid Point Singularity Table (GPST) larger than expected. These degrees-of-freedom will also appear in the USET table.
- The interface elements may generate high or negative matrix/factor diagonal ratios. If there are no other model errors, these messages may be ignored and PARAM, BAILOUT, –1 may be used to continue.

Guidelines

- The interface elements use the geometry of the boundary with the least number of degrees-of-freedom, which consist of cubic polynomials.
    - — If the other boundaries have widely varying geometry, poor answers may result.
    - — Warnings may be issued, but no geometrical adjustment is performed.
- Sharp corners within the interface element may degrade accuracy.
    - — The preferred alternative is to specify multiple interface elements. An example of multiple interface elements is solved as the second example.
- Connecting few elements to many will not improve the accuracy along the interface.

- The limiting factor on the accuracy will be those few elements. For example, if one boundary has one element and the other boundary has four, the accuracy will be limited to that one element.

- Interface elements will only connect common displacement fields of different element types. No kinematic constraints are enforced.
  - Shell p-elements have five fields, and solid p-elements have three. Therefore the rotational fields of the shell p-elements will not be connected to the solid p-elements.
- The sparse solver for linear statics and the Lanczos eigensolver for normal modes should be used. The sparse solver is the default solver for linear statics.
- The value of epsilon, which is the residual from the linear solution, and the shapes of modes of primary interest, which can best be evaluated graphically, should be checked to detect unstable results.
  - Plots of displacements and stresses may also indicate unstable results. This would be visible as discontinuities in the displacements or stresses across the interface, which would imply a poor solution in that area.
- Since the boundaries are physically distinct, certain functions, such as shell normals and stress discontinuities in the error estimator, will not be applied across the interface.
- Contour plots may show differences across the interface because of the different view meshes. However, this is an indication of the results processing, not the original solution. A denser view mesh would reduce the differences.

Limitations

- Constraining the boundaries may lead to unstable results in certain cases.
  - The most common constraint on a boundary is at an endpoint, such as a symmetry condition.
  - Cases with constraints include multiple interfaces at the same point, such as a sharp corner; and endpoints of different boundaries connected together, such as an interior interface.
  - Such unstable results are indicated by high epsilons or non-physical modes. They are also indicated by irregularities in the displacements or stresses.
- Superelements have not been implemented.

Examples

Several sets of example problems were analyzed, in order to test the capabilities of the interface elements with various boundary meshes. The goal of the interface element is that it should not decrease the accuracy below that obtained using the less refined boundary with a conforming mesh. However, it will not increase the accuracy above that obtained using the more refined boundary with a conforming mesh. For example, if one boundary has a single element face and the other has several element faces at a given p-level, the accuracy with the interface elements should fall between a similar problem with two conforming single-face boundaries and a similar problem with two conforming multiple-face boundaries.

Circular Shaft (TPL: ifscp88.dat and ifscp88r.dat)

The first set of example problems is a circular shaft that has exact solutions at low p-levels, as shown in Figure 3-18.

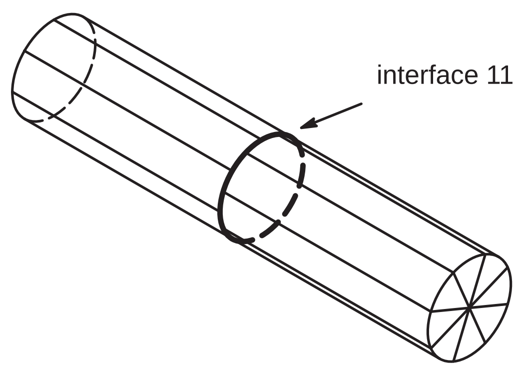


Figure 3-18. Circular Shaft

The boundaries, which consist of faces of solid elements, for the two meshes are shown in Figure 3-19. The first mesh serves as a baseline for the interface and uses an interface element, even though it is a conforming mesh.

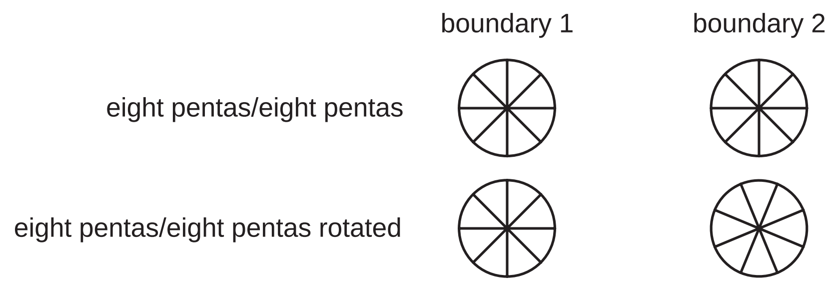


Figure 3-19. Solid Boundaries On Circular Shafts.

Tension (exact at p=1) and torsion (exact at p=3) load cases were analyzed. The grid numbering for the two boundaries, which consist of faces of solid elements, is shown in Figure 3-20.

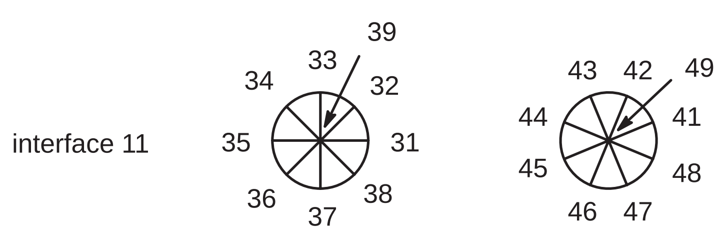


Figure 3-20. Solid Boundaries on Circular Shafts

Boundary 1 may be defined with the GMBNDS Bulk Data entry using the GMSURF option, after the appropriate surface and associated faces have been defined:

```
gmsurf, 3,mscgrp1,0,0
        v_form=periodic,u_igrid=10,v_igrid=10
        equation, 0.,1., 0.,6.283185, 31,32,33
deqatn  31      f(u,v)=u*cos(v)
deqatn  32      f(u,v)=u*sin(v)
deqatn  33      f(u,v)= 5.
feface,31, 31,32,39,  ,, 3
feface,32, 32,33,39,  ,, 3
feface,33, 33,34,39,  ,, 3
feface,34, 34,35,39,  ,, 3
feface,35, 35,36,39,  ,, 3
feface,36, 36,37,39,  ,, 3
feface,37, 37,38,39,  ,, 3
feface,38, 38,31,39,  ,, 3
$
GMBNDS,BID
,ENTITY,ID1,ID2,ID3,ID4,ID5,ID6,ID7
$
gmbnds,  1
,gmsurf,  3
```

Boundary 2 may also be defined with the GMBNDS Bulk Data entry using the GMSURF option, after the appropriate surface and associated faces have been defined:

```
gmsurf, 4,mscgrp1,0,0
        v_form=periodic,u_igrid=10,v_igrid=10
        equation, 0.,1., 0.,6.283185, 41,42,43
deqatn  41      f(u,v)=u*cos(v)
deqatn  42      f(u,v)=u*sin(v)
deqatn  43      f(u,v)= 5.
feface,41, 41,42,49,  ,, 4
feface,42, 42,43,49,  ,, 4
feface,43, 43,44,49,  ,, 4
feface,44, 44,45,49,  ,, 4
feface,45, 45,46,49,  ,, 4
feface,46, 46,47,49,  ,, 4
feface,47, 47,48,49,  ,, 4
feface,48, 48,41,49,  ,, 4
$
GMBNDS,BID
,ENTITY,ID1,ID2,ID3,ID4,ID5,ID6,ID7
$
gmbnds,  2
,gmsurf,  4
```

After the boundaries have been defined, the GMINTS Bulk Data entry is used to define the interface between those boundaries:

```
$
GMINTS,EID,PID,ID1,ID2,ID3,ID4
$
gmints, 11,  2,  1,  2
```

The interface element properties are defined on the PINTS Bulk Data entry:

```
$
PINTS ,PID,   TOL,DSCALE
$
pints ,  2,  0.01,  1.e3
```

The maximum principal stress contours for the tension case and the maximum shear stress contours for the torsion case at p=3 for the eight pentas/eight pentas rotated mesh are shown on the deformed shape in Figure 3-21. The maximum stress values are also printed at the appropriate locations. Note that the variations in the tension case are due to the very small contour range; the actual range of values is 9.995 to 10.00.

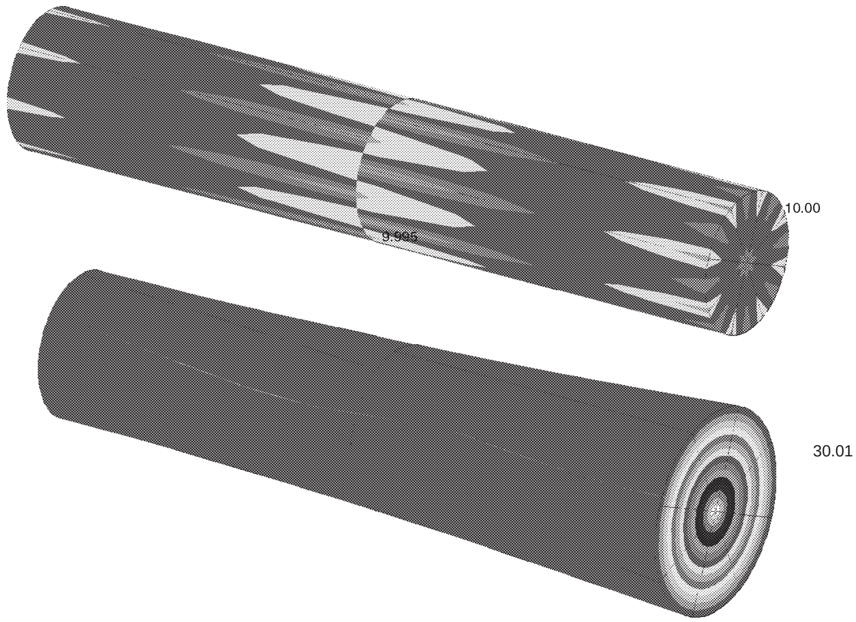


Figure 3-21. Stress Contours on Circular Shaft (p=3).

The maximum stress values at p=3 for the two load cases are listed in Table 3-1 for both meshes. The first mesh is exact, since the mesh is conforming, and the second mesh differs from the exact solution by less than 0.1%.

Table 3-1. Maximum Stress on Circular Shaft (p=3).

mesh	tension (maximum principal)	torsion (maximum shear)
eight pentas/eight pentas	10.00	30.00
eight pentas/eight pentas rotated	10.00	30.01

Square Plate with Circular Hole (TPL: ifsph22.dat, ifsph23.dat, ifsph24.dat, and ifsph44.dat)

The second example problem is a square plate with a circular hole, as shown in Figure 3-22. The hole is small enough relative to the plate that additional elements, though not necessary, greatly improve convergence. This example illustrates how a global/local problem can be modeled, since the patch of elements around the hole is being replaced without modifying the mesh away from the hole.

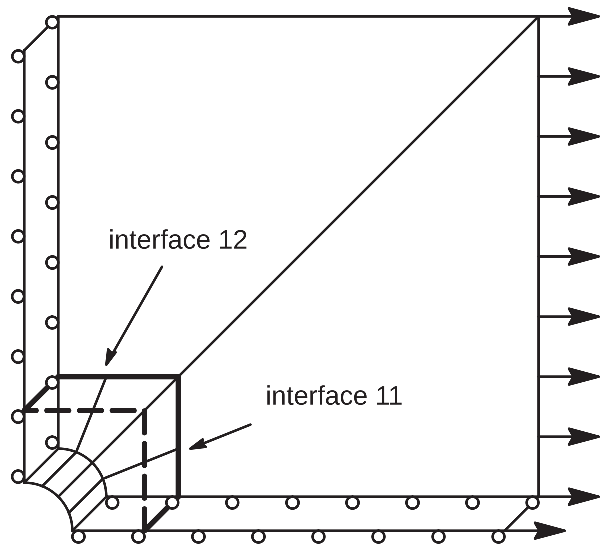


Figure 3-22. Square Plate with Circular Hole.

The square plate has a uniform tension load, so that the stress concentration factor at the hole may be calculated, and symmetry constraints. Two interface elements are used, since the interface contains a right angle. The grid numbering for the four boundaries, which consist of faces of solid elements, on the two interface elements is shown in Figure 3-23.

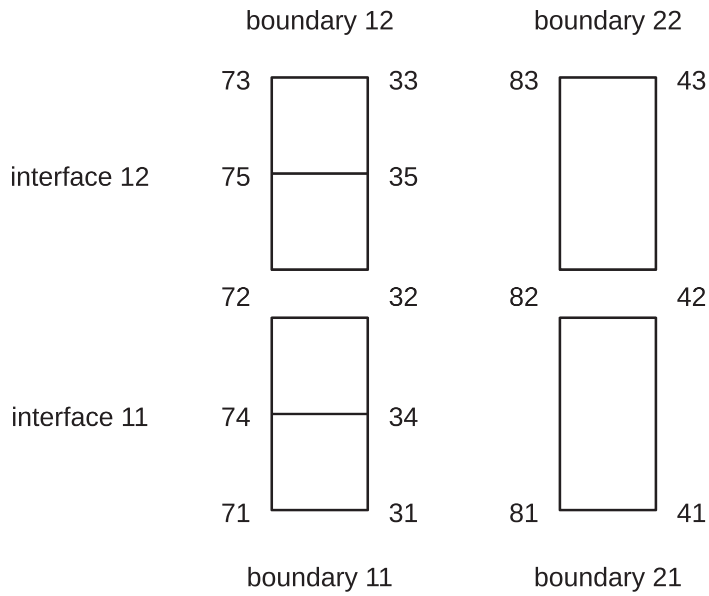


Figure 3-23. Solid Boundaries on Plate with Hole.

The interface bulk data entries are listed below, using the GMBNDS Bulk Data entry with the GRID option, since there is no need to define GMSURFs or FEFACEs on the interface:

```
$
GMBNDS,BID
,ENTITY,ID1,ID2,ID3,ID4,ID5,ID6,ID7
$
gmbnds, 11
,grid  , 31, 34, 32, 71, 74, 72
gmbnds, 12
,grid  , 32, 35, 33, 72, 75, 73
$
GMBNDS,BID
,ENTITY,ID1,ID2,ID3,ID4,ID5,ID6,ID7
$
gmbnds, 21
,grid  , 41, 42, 81, 82
gmbnds, 22
,grid  , 42, 43, 82, 83
$
GMINTS,EID,PID,ID1,ID2,ID3,ID4
$
gmints, 11,  2, 11, 21
gmints, 12,  2, 12, 22
$
PINTS ,PID,TOL,DSCALE
$
pints , 2,0.01,  1.e3
```

The von Mises stress contours for the two hexa/four hexa mesh are shown on the deformed shape in Figure 3-24. The boundary between the large light and dark areas in the figure has a contour value of exactly the applied stress of 10.00, such that any minutely small differences from the applied stress are shown.

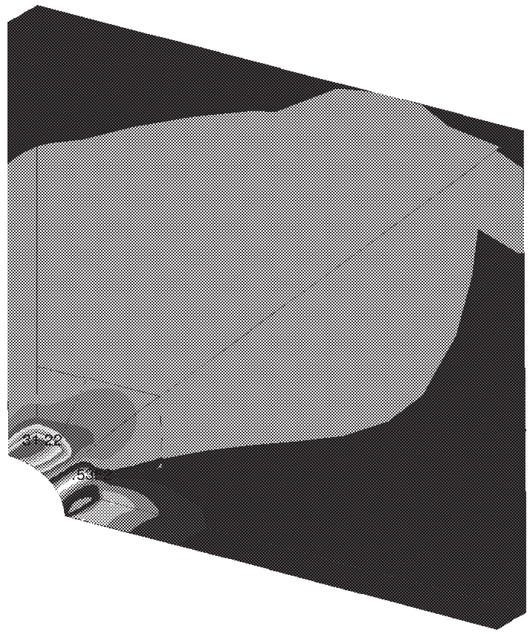


Figure 3-24. Stress Contours on Plate with Hole (p=8).

The stress concentration factors at p=8 are listed in Table 3-2 for four meshes with interface elements. Values are listed for both the middle surface and the top surface in order to show the variation through the thickness. The value calculated from the Roark handbook for a semi-infinite plate is 2.72, which is derived from curve fits to photoelastic data with a specified accuracy of much less than 5%. After the simplest mesh, the results are identical at 2.81 on the middle surface and 2.65 on the top surface. The highest factor occurs at the middle surface of the model, which is slightly higher than the plate solution, whereas the factor on the top and bottom surfaces is slightly lower, due to the Poisson effect.

Table 3-2. Stress Concentration Factors for Plate with Hole (p=8).

Mesh	Stress Concentration (middle surface)	Stress Concentration (top surface)
two hexas/two hexas	2.796	2.647
two hexas/three hexas	2.810	2.654
two hexas/four hexas	2.810	2.655
four hexas/four hexas	2.810	2.654

3.5 GAP Constraints in Linear Statics (SOL 101) Solution Sequence

Introduction

In Version 70.5, gap constraints may be modeled in linear static analysis (SOL 101). In contrast to the gap element available for nonlinear analysis (SOL 106 and SOL 129), the new gap capabilities in SOL 101 do not require a gap stiffness. However, the gap constraint is limited in that it does not consider friction effects and it can only represent a gap between a grid point and a constraint. It only allows for opening and closing of gaps. The constraint applies to a single degree-of-freedom at a grid point or spoint. The constraint ensures that:

1. The displacement cannot be negative. This is to ensure that there is no penetration. Therefore, the chosen degree-of-freedom must be perpendicular to the contact surface and positive in the opening direction.

2. The force of constraint can not be negative. This is to ensure that there is no tension.

The constraints are satisfied by an iterative technique that is built into SOL 101. The iterative process starts by a random vector. This random vector assumes certain grids to be in contact and other grids to be in an open state. A solution is obtained when all the gap constraints are satisfied, i.e., there’s no penetration and no tension forces. If a limit cycle (return to a previous state) is detected during the iterations, a new random start vector is tried.

Benefits

This provides an alternate method to the use of GAP elements in SOL 106. (Some experiments have shown that the cost of analysis will be about the same.) The advantage is that you do not have to learn how to calculate the GAP stiffnesses nor how to control SOL 106. Multiple load conditions are allowed, and each will be solved separately.

Input

You need to use the SUPORT Bulk Data entry, some new parameters and a special DMIG entry named CDSHUT.

SUPORT entry	Selects constrained degrees-of-freedom. These points must be in the a-set of the residual structure. This means they must not be dependent in an MPC equation, constrained by SPC, partitioned by OMIT, or in an upstream superelement.
PARAM CDITER	Constraints will be applied if CDITER is greater than zero. The value is the maximum number of iterations allowed. (Default = 0).
PARAM CDPRT	Controls the printing of constraint violations during the iterations. The sparse matrix printer prints UR (negative displacements) and QR (negative forces of constraint) for constrained degrees-of-freedom. (Default = 'YES')
PARAM CDPCH	Controls the PUNCH output of DMIG CDSHUT records for the final state of the constrained degrees-of-freedom. (Default = 'NO')
DMIG CDSHUT	Optional input of the vector defining the state of the constrained degrees-of-freedom. A one indicates closed and a zero means open. See PARAM CDPCH for a method to have MSC/NASTRAN create these records. (Default = all closed).

You do not need to use the CGAP and PGAP Bulk Data entries.

Output

The output is standard for SOL 101, and all existing postprocessors will work. Forces for closed degrees-of-freedom are in the SPCFORCE output. In addition there is information in the ".f06" file which shows diagnostic information for the iterations. A final state vector may be output in the ".pch" file.

Guidelines

The finite element model input looks just like input to SOL 101 with the addition of input records shown above. It is recommended that the “.f06” file be examined to ensure that the iterations have converged, since the results of the last iteration will be output. The last iteration should have zero changes.

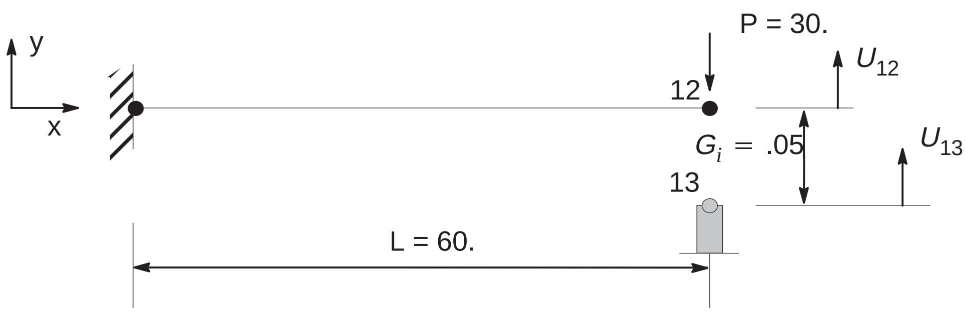
If the constraint is between a finite element model and a fixed boundary, then arrange to have one of the degrees-of-freedom at the boundary grid points represent motion perpendicular to the boundary. A positive displacement represents motion away from the boundary. If, on the other hand, the constraint represents relative motion between two bodies, MPC equations are needed to define a relative motion degree-of-freedom, which is then constrained to have a non-negative displacement.

Limitations

- The only nonlinearity allowed is the constrained displacements.
- There is no gap stiffness and no sliding friction.
- Free bodies can not be analyzed using SUPORT to define rigid body modes and have constrained degrees-of-freedom in the same model. Parameters INREL and CDITER are mutually exclusive. A fatal message is issued if both parameters are present.
- No constraint changes are allowed between subcases.
- There is no guarantee that the solution will converge or that all systems will follow the same path.

Cantilever Beam Example

A cantilever beam example with a restrained support at the end is analyzed. The purpose of this simple example problem is to demonstrate how to set the SPOINT, MPC and SUPORT entries to model the restraint.



where G_p	=	Penetrating Grid 13 (positive displacement will cause closing of the gap).
G_o	=	Opening Grid 12 (positive displacement will open gap).
G_i	=	Initial gap opening = .05.
U_g	=	Actual gap distance.
U_p	=	Displacement of penetrating grid = U_{13}.
U_o	=	Displacement of opening grid = U_{12},

then

$$U_g = U_{12} + G_i - U_{13}$$

or

$$U_g - U_{12} - G_i + U_{13} = 0$$

Since U_g can not be dependent, i.e. U_g cannot be entered as first term on MPC, rearranging the equation, we get

$$U_{13} - U_{12} + U_g - G_i = 0 \qquad (3\text{-}17)$$

U_{13} is restrained and can be removed from MPC. The resulting MPC equation is

$$-U_{12} + U_g - G_i = 0 \qquad (3\text{-}18)$$

Follow these steps to create an MSC/NASTRAN input file:

- Assign SPOINT 51 for U_g.
- Assign SPOINT 101 for G_i.

- Assign SUPORT for U_g.
- Apply initial gap as displacement to G_i.
- Create MPC as in Eq. (3-18).
- Include PARAM,CDITER (required).

The input file setup is shown:

```
$ Gap in Sol 101 :  Example 1.
Init Master(s)
Sol 101
Cend
Title = Gap Elements, Cantilever Beam.
     Mpc   = 77
     Spc    = 88
     Load  = 300
     Disp   = All
     Spcf   = All
     Mpcf  = All
Begin Bulk
$       2       3       4       5       6       7       8       9       10
Grid    11              0.0     0.0     0.0
Grid    12              60.0    0.0     0.0
Cbar    15      1       11      12      0.0     1.0     0.0
Pbar    1       2       1.0     1.0     1.0     1.0
Mat1    2       30.e6           0.3
Force   300     12              –30.0   0.0     1.0     0.0
Spoint  51      101
Suport  51      0
Spc     88      11      123456
Spc     88      101     0       0.05
Mpc     77      12      2       –1.0    51      0       1.0             +m1
+m1             101     0       –1.0
Param   Cditer  10
Param   Cdprt   Yes
Enddata
```

Results

1. Load P_g, required to close the gap is $G_i = 3 * E * \frac{I}{(L * * 3)} = 20.8333$ lb., ignoring the transverse shear flexibility.
2. The MPC force is $P - P_g = 30.0 - 20.8333 = 9.1666$. The force in the gap should show up as an SPCFORCE for point 51.

Examples (TPL: cd_1.dat and cd_2.dat)

Two simple example problems are included to illustrate the capability. The first is a thick pad on a rigid base, with a line load applied. A half model with symmetric boundary conditions is used to reduce problem size. The problem is to determine the footprint (where does lift off occur). The second problem involves concentric rings, where MPC records have been introduced to constrain penetration of the rings. The constraints are written to account for the 0.1 initial opening. A typical constraint for the radial displacement of an inner and outer pair of grid points is:

$$U_{inner} < U_{outer} + .1$$

Let U_g be the gap opening and define $U_s = 1$. Enforce the MPC:

$$-1. * U_{outer} + 1. * U_{inner} + U_g - .1 * U_s = 0$$

U_g must not be the first variable entered into the MPC equation, since the first term in a MPC relation becomes the dependent point, and it must not be U_g. The constraint equation becomes the standard form:

$$U_g > 0$$

Readers not familiar with the replication feature used in the samples may look at the introduction to Section 5 of the *MSC/NASTRAN Quick Reference Guide*, or look at the expanded records in the bulk data echo in the ".f06" file.

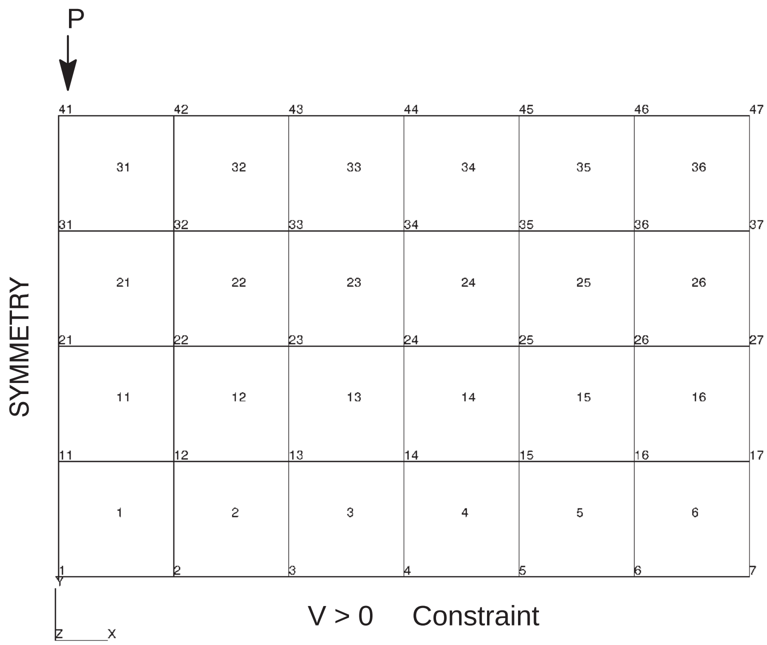


Figure 3-25. Finite Element Model, Example 1.

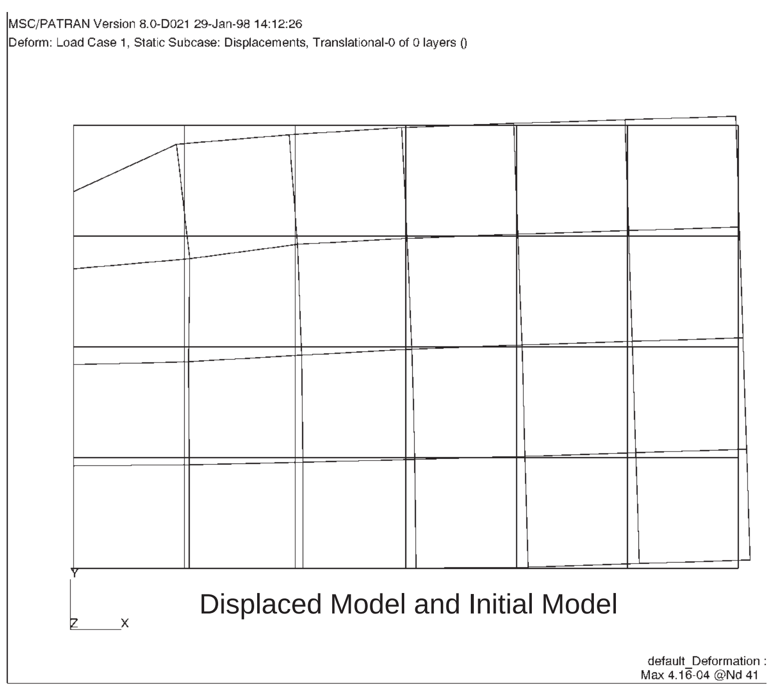


Figure 3-26. Constrained Displacement, Example 1.

Listing 3-3. Constrained Displacement Example 1, Input File.

```
TIME   10
SOL    101
CEND
TITLE = DEMONSTRATE CONSTRAINED DISPLACEMENT IN SOL 101, #1
  SPC  = 200
  LOAD = 300
  DISPL= ALL
  OLOAD= ALL
  SPCFO= ALL
BEGIN BULK
$
$ CREATE A 6 x 4 MESH USING REPLICATOR RECORDS
$
GRDSET                                          3456
GRID   1              0.     0.
=      *1      =      *.5    =
=5
.
.
.
PSHELL 100    100    .001
MAT1   100    1.+7          .3
$
$ SYMMETRIC BOUNDARY AT X=0,  DOWNWARD LOAD AT TOP CENTER
$
SPC1   200    1      1      11     21     31     41
FORCE  300    41            1.            –1.
$
$ CONSTRAIN DISPLACEMENTS ON BOTTOM EDGE, OPTIONAL CDSHUT INPUT
$             1.            –1.
SUPORT 1      2
=      *1     =
=5
PARAM  CDITER 20
PARAM  CDPRT  YES
PARAM  CDPCH  YES
DMIG   CDSHUT 0      9      1      0                     1
DMIG   CDSHUT 1      0             1      2      1.
       7      2      1.
ENDDATA
```

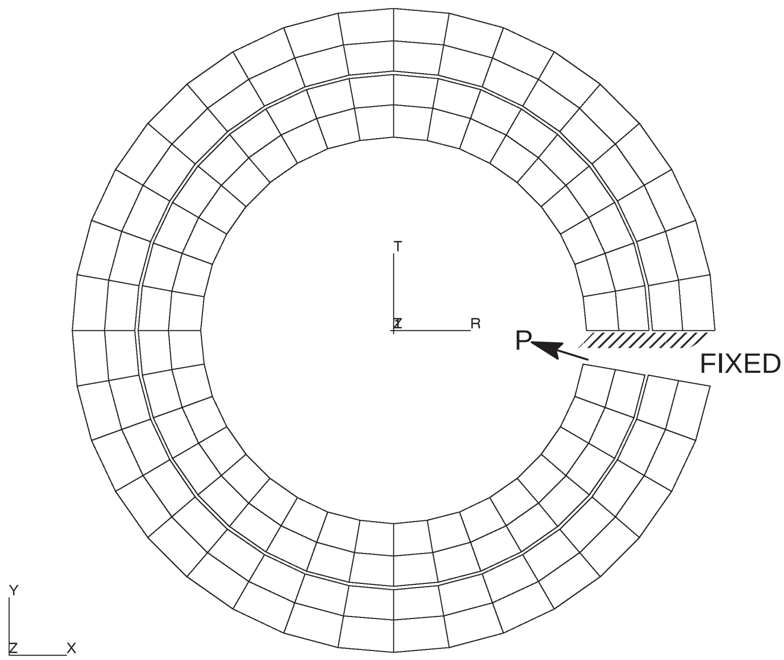


Figure 3-27. Finite Element Model, Example 2.

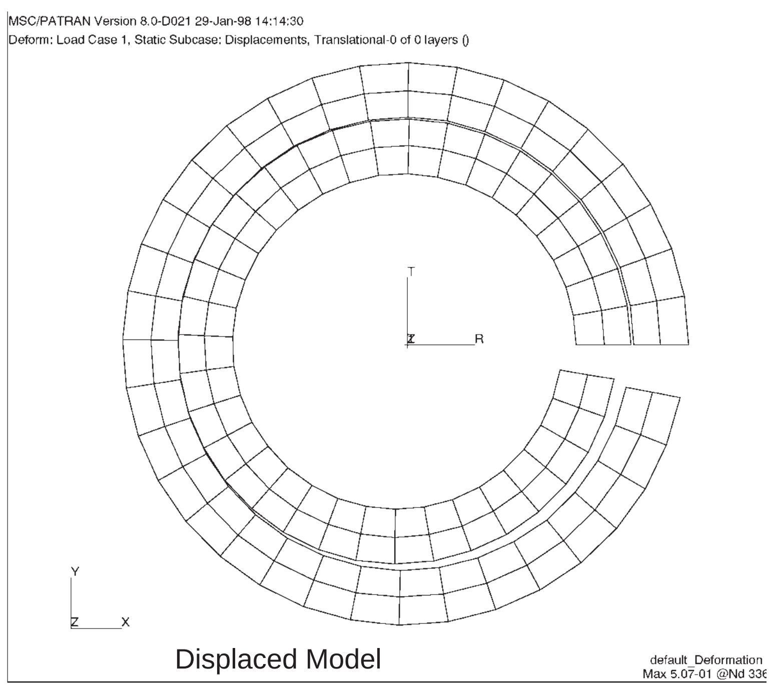


Figure 3-28. Constrained Displacement, Example 2.

Listing 3-4. Constrained Displacement Example 2, Input File.

```
TIME   100
SOL    101
CEND
TITLE =DEMONSTRATE CONSTRAINED     DISPLACEMENT IN       SOL 101, #2
  SPC  = 200
  MPC  = 200
  LOAD = 300
  DISPL= ALL
  SPCFO= ALL
BEGIN BULK
CORD2C 1                                                 1.      +C
+C     1.
GRDSET         1                             1       3456
GRID   1               10.     0.      0.
=      *1      =       =       *10.    =
=34
.
.
.
PSHELL 100     100     1.
MAT1   100     30.+6           .3
SPC1   200     12      1       101     201     301     401     501
FORCE  300     436     1       1.+4    –1.
$
$ DATA FOR GAP SIMULATION
$
PARAM  CDITER 60
PARAM  CDPRT  NO
SPOINT 902    THRU     936
SPOINT 1000
SPC    200     1000            1.
MPC    200     202     1       –1.     302     1       +1.             +M02
=      =       *1      =       =       *1      =       =       =       *1
=33
+M02           902     0       +1.     1000    0       –.1
*1,    =       *1      =       =       =       =       =
=33
SUPORT 902     0
=      *1      =
=33
ENDDATA
```

CHAPTER 4

FUNCTIONALITY ENHANCEMENTS

4.1 R-type Element Shape Sensitivities in Optimization

Introduction

For shape optimization, sensitivity analysis in previous versions ignored changes in a response due to changes in grid locations associated with R-type elements (RBAR, RROD, RBEi, RSPLINE, RTLPLT, etc.). Therefore, the resulting sensitivity information may have been inaccurate. The new capability for R-type element shape sensitivities in V70.5 removes this limitation and produces the correct sensitivity information. It should be pointed out that since multi-point constraint (MPC) equations are not a function of grid locations, changes in grid locations have no effect on those equations even with the new capability.

This capability is transparent to a user and no new user input is required. The system internally determines whether the R-type element shape sensitivities are needed. Therefore, any qualified shape optimization task will benefit from this new feature.

Benefits

The major benefit is great improvement in the accuracy by using R-type element shape sensitivities. A secondary benefit is that numerical difficulty is avoided where an optimizer terminates prematurely because the violated constraint has zero sensitivity (shown in the example later in this chapter).

Input

No new input is required.

Output

No new output is generated.

Guidelines

The shape sensitivity is calculated only for the RBAR, RBEi, RROD, RSPLINE and RTRPLT elements. Therefore, it is recommended that you use these elements rather than multipoint constraint (MPC) equations to model the physical behavior for shape optimization problems. However, when grid locations to be optimized are not associated with any R-type element, MPC equations can still be used.

Limitations

There are three limitations with the new R-type element shape sensitivity capability:

1. The changes in the grid locations associated with shell to solid connection (RSSCON), and the BUSH element, are ignored while calculating R-type element shape sensitivities.

2. The direct differentiation method (i.e., non-adjoint method) is the only option for the sensitivity analysis when R-type element shape sensitivities are required.

3. Only the forward finite difference scheme is available to calculate the perturbed constraint transformation matrix used for rigid element shape sensitivities. Therefore, the user specified value of the Bulk Data parameter, CDIF, that selects the differencing scheme, is ignored. However, calculation of other portions of the sensitivities analysis is still controlled by the CDIF parameter.

Example (TPL: rbesensa.dat)

Figure 4-1 shows a simple model where the concentrated mass 5003 is connected by three RBE2 elements. Each RBE2 connects three grounded springs in x, y, and z directions. The first natural frequency for the system is 1.771 Hz.

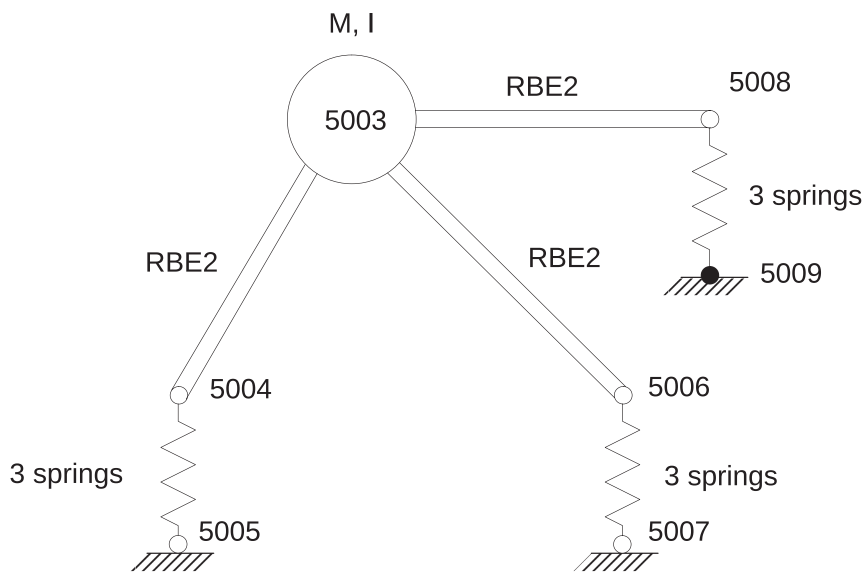


Figure 4-1. A Simple Model with Rigid Elements.

The purpose of this example problem is to illustrate the effect of changing spring location on first natural frequency. For simplification, assume the x-locations of grid points 5008 and 5009 are to be designed. Two DVGRID entries define the desired perturbations in the x-location of grid points 5008 and 5009. The sensitivity of the first natural frequency with respect to this shape design variable is calculated.

Table 4-1 lists three results: the first one from V70, the second from V70.5 and the last one from hand calculations using the finite difference scheme (F.D.). This is an example where the sensitivity of a response is solely contributed by rigid element shape sensitivities. Therefore, when the rigid element effects are ignored as in the case of V70, the resulting sensitivity is zero.

Table 4-1. Natural Frequency Sensitivity with Rigid Element Effects.

Method	dF/DX
V70	0.0
V70.5	1.034×10^2
F.D.	1.040×10^2

In addition, if the job continues with the zero sensitivity, it may terminate prematurely in the optimization phase because the optimizer cannot proceed further. The new capability implemented in V70.5 not only improves the sensitivity accuracy but also avoids certain numerical difficulties in the optimization phase.

```
ID = rigid shape
sol 200
cend
analysis = modes
desobj = 1
METHOD =  100
SPC = 100
BEGIN BULK
$
SPC1, 100, 123456, 5005, 5007, 5009
$
.
.
.
CELAS2      5017  1120.0    5008       2    5009       2     .14
CELAS       5018  1680.3    5008       1    5009       1     .14
CELAS2      5507  1604.0    5008       3    5009       3     .08
$
RBE2        5020    5001  123456    5004
RBE2        5021    5001  123456    5006
$
RBE2        5022    5001  123456    5008
$
RBE2        5023    5001  123456    5003
eigrl   100                     1
$
$ design model
$
desvar  1       x-loc   1.      -2.0    2.0
dvgrid  1       5008            1.      1.
dvgrid  1       5009            1.      1.
dresp1  1       freq    freq                    1
param,optexit,4
ENDDATA
```

4.2 Specific Optimization Enhancements

This section briefly describes four minor enhancements and several key error corrections that were made for optimization in V70.5 of MSC/NASTRAN.

Simultaneous Consideration of Static Analysis and Static Aeroelasticity

The optimization capability in SOL 200 is multidisciplinary, as you are able to simultaneously perform different kinds analyses including statics, normal modes, frequency analysis, transient analysis and flutter. Version 68 of MSC/NASTRAN added static aeroelasticity (ANALYSIS=SAERO) to this list, but with the restriction that it could not be run with any static (ANALYSIS = STATICS) analysis. This restriction has been removed by an enhancement in V70.5 that has made fairly extensive changes in the SOL 200 DMAP dealing with boundary condition qualifiers.

Machine Precision Optimization

MSC/NASTRAN relies on the DOT algorithm supplied by Vanderplaats Research and Development for the structural optimization capability contained in SOL 200. For Version 70.5, this algorithm and the supporting software that provides it with design information were enhanced to machine precision (i.e., double precision on 32-bit word machines such as an RS/6000 and single precision on 64-bit word machines such as a Cray). For large and/or difficult optimization tasks, this added precision should allow the optimizer to reach improved designs relative to the V70 results on the short word machines.

Presentation of Weight as a Function of Material ID

A minor feature has been added to design optimization in response to a client request. The enhancement prints the weight of the structure as a function of material ID. The enhancement only deals with printing and does not provide any new design functionality.

The new print is invoked by setting P2, on the DOPTPRM entry, to 16, or including 16 in a sum of values that provides the overall print. An example of the new output follows:

----- WEIGHT AS A FUNCTION OF MATERIAL ID ----

MATERIAL ID	SEID	INPUT WEIGHT	OUTPUT WEIGHT
1	0	0.0000E+00	N/A
11	0	1.2102E+04	N/A
21	0	4.3960E+03	N/A
31	0	1.5175E+04	N/A
41	0	5.5104E+03	N/A
51	0	1.5049E+03	N/A
61	0	4.4541E+02	N/A
62	0	2.9417E+03	N/A
71	0	1.9013E+03	N/A
81	0	6.4310E+03	N/A
82	0	1.8698E+03	N/A
87	0	0.0000E+00	N/A
TOTAL		5.2278E+04	N/A

Figure 4-2.

Note that only the input value of the weight is given and the output value is marked N/A (not applicable). Since the weight values are not part of the design process, there is no predicted value for the weight as a function of material ID for the design produced in the approximate optimization task.

Simplified Design of Constant Section Beams

Prior to Version 70.5, the creation of the design model data required for the design of finite elements that reference a PBEAM entry has been complicated by the desire to support the generality of this element in terms of its ability to support tapered cross sections. Section 2.4 of the *MSC/NASTRAN Design Sensitivity and Optimization User's Guide* devotes four pages to the special requirements imposed on the user for the design of BEAM elements. Version 70.5 has relieved this burden significantly if the PBEAM has a constant cross section. In this case, it is no longer necessary to either:

- Specify a negative FID on the DVPREL1 or DVPREL2 entry that references a PBEAM property or
- Specify END B design data that are identical to those given for END A.

For Version 70.5, the user can now simply identify the FID on the PBEAM Bulk Data entry that is to be designed. This applies only to constant section beams; i.e., the parent PBEAM entry and the first continuation.

If there is a desire to design a tapered beam, the guidelines provided in the User's Guide still apply. It is also necessary to refer to the EPT location if it is desired to design the last two continuations on the PBEAM entry, i.e., properties K1 through K2.

This enhancement was done too late in the release of Version 70.5 for the changes in the DVPREL1 and DVPREL2 entries that document this enhancement to make it into the *MSC/NASTRAN Quick Reference Guide*, Version 70.5. The changes are given by Remarks 4 and 5 for the DVPREL1 entry and Remark 3 of the DVPREL2 entry in Appendix B.

Adjoint Sensitivity Analysis with Multiple Static Boundary Conditions

Adjoint sensitivity analysis was implemented in V70 of MSC/NASTRAN for static analysis and for frequency analysis. One of the limitations for the method was that it could not be applied when there were multiple static boundary conditions. The design intent was to force the direct sensitivity method in this case, but due to a design flaw it was possible to encounter an error in V70 when there were multiple boundary conditions and the Adjoint method was selected. There is a DMAP avoidance for this V70 error (see CSR 29202). For V70.5, the error has been corrected and the Adjoint method has been extended to be applicable when there are multiple boundary conditions. The Adjoint method requires less computer resources than the Direct method (see the Version 70 *MSC/NASTRAN Release Guide* for the applicability of the Adjoint method) and improves the performance of optimization tasks with multiple static boundary conditions.

Removal of Problem Size Limits

In previous versions you may not have been able to run very large optimization problems due to an error in memory requirement calculations (see CSR 29140). This error has been fixed in Version 70.5.

Improved Memory Requirement Message

An improved message is printed when the user has not provided enough memory to keep even a single constraint in the design optimization process. In this rare occasion, a message will be received from the DOM9P4 subroutine indicating how much additional memory is required to proceed with optimization. It should be stated that this is a nonconservative estimate and it is recommended that even more memory than the suggested value be provided.

Performance Improvement

A design flaw in the optimization solution that forced significant writing and reading from disk has been corrected. For large problems, V70.5 should have significantly less I/O and less CPU in the DOM9 module than was experienced in earlier versions.

4.3 Enhancements to Aerodynamic Data Structures and Data Flow

In Version 70.5, a number of significant changes are made to the aerodynamic data block formats and to the order in which they are created within the static aeroelastic, flutter and dynamic aeroelastic analysis solutions (144, 145 and 146). The changes prepare for a high order aerodynamic panel method and accomplish two goals:

1. Remove the geometric assumptions of the existing aerodynamic methods (e.g., flow aligned panel edges) from the description of the aerodynamic model.

2. Reduce the modeling restrictions between the aerodynamic and structural models to just those that one expects between structural superelements.

In particular, there are no element/grid numbering restrictions between the aerodynamic and structural models in V70.5. The only remaining requirement over superelements is that the geometry of the two models must share a common “basic” coordinate system. (Note that global, or output coordinate systems vary independently from point to point in both the aerodynamic and structural models just as before.)

The aerodynamic model has been modified to be similar to a structural model in regard to its data structures. This allows the user to take advantage of the existing tools for structural model geometry manipulation (e.g., VECPLOT). While the ACPT data block is still used to communicate the geometry of the aerodynamics of the existing MSC/NASTRAN aerodynamic methods from the geometry generator (APD) to the matrix generator (AMG), it is not used for any computation that is not associated specifically with those aerodynamic methods. In particular, the ACPT data block is no longer used to generate the load integration matrices (SRKT and HMKT), and the BGPDT with aero points "appended" (the BGPA) is no longer used to build the spline matrices.

The existing subDMAP's PFAERO; AESTAT and AESTATRS for static analyses; FLUTTER for flutter analyses and SEAERO for aeroelastic response analyses have been modified to move the aerodynamic model creation (geometry) into PHASE0 and then to pass the geometry among the subDMAPs. This allows aerodynamic model input checks to occur before the reduction of the structural model to the analysis set. It also supports the various superelement options for dynamic aeroelastic analysis splining to upstream superelements (for more details see Section 4.6 of "Splining to Upstream Superelements in Dynamic Aeroelasticity").

The new aerodynamic geometry data blocks AECOMP, AEBGPDT, AEGRID, AEBOX are created in PHASE0 by the APD module. AECOMP is akin to the CAEROi information, while the AEBGPDT and AEGRID are instances of the BGPDT family (that is, they share a common data structure and that structure is common with the structural BGPDT) that represent the aerodynamic degrees-of-freeedom and the aerodynamic element ("box") connectivity grids, respectively. Finally, AEBOX is an instance of the ECTS family that describes the connectivity of the aerodynamic mesh. That connectivity is currently used only in plotting the model.

In keeping with the idea that the aerodynamic model might not be described as a collection of CAEROi Bulk Data entries, APD has been broken into a module specific to the existing aerodynamic methods (called APD) and parts that are more general in nature. For example, the SPLINE data block creation subroutines have been used to create a new module called MKSPLINE. Also, the GI module has been made very general (See "New Spline Methods and Spline Checks") and it performs its own checking for the existence of the structural and aerodynamic grids to remove the structural model from APD (allowing APD to be called in PHASE0). The SPLINE data block has been modified to provide GI with all the information it needs to stand separate from APD.

Upward Compatibility

DMAP alters involving the aerodynamic model geometry will need to be rewritten for V70.5. However, the changes are not as dramatic as they first appear. Essentially, the same data exist in V70.5 as in V70, but they are created in different subDMAPs. Some new data structures are created that describe the aerodynamic model in a manner similar to the structural model description. The existence of these data blocks may make certain alters easier to perform than before.

See Appendix A for descriptions of the AECOMP, AEBGPDT, AEGRID and AEBOX data blocks.

Structure plotting options showing the structural and the aerodynamic models in the same frame using OUTPUT(PLOT) Case Control is not supported in Version 70.5. The aerodynamic model geometry (alone) may still be requested, however, in the usual manner.

Modified Aeroelastic Output Formats

Two aeroelastic results outputs are reformatted in V70.5: the eigenvector print of combined structural and aerodynamic mode shapes (PARAM, OPPHIPA, 1) and the aerodynamic force and pressure print resulting from the AEROF and APRES Case Control request.

The force output has been changed to reflect the three dimensional nature of the upcoming high order panel method aerodynamics. To that end the output format, shown in Figure 4-3, of the trimmed forces and pressures on the aerodynamic model has been modified to print the forces as a set of results similar to the OLOAD print for structures.

```
EXAMPLE HA144C: HALF SPAN 15-DEG SWEPT UNTAPERED WING  HA144C          MARCH  19, 1998  MSC/NASTRAN  7/17/97   PAGE    27
 CANTILEVERED WIND TUNNEL MOUNT, DOUBLET-LATTICE AERO

 0.041 IN PLATE W/BEVELLED LEADING AND TRAILING EDGES

                          A E R O S T A T I C   D A T A   R E C O V E R Y   O U T P U T   T A B L E
                                     MACH = 4.500000E-01               Q = 2.000000E+00
                                                                                AERODYNAMIC FORCES
     AERODYNAMIC          AERODYNAMIC PRES.    AERODYNAMIC
            GRID   LABEL    COEFFICIENTS        PRESSURES         EXTERNAL ID  LABEL  NORMAL FORCES(T3)  MOMENTS(R2)

               1     LS      1.650689E+00       3.301379E+00             101    LS     1.573657E+00      2.036459E-01
               2     LS      7.473636E-01       1.494727E+00             102    LS     7.124864E-01      9.220242E-02
               3     LS      5.366662E-01       1.073332E+00             103    LS     5.116216E-01      6.620862E-02
               4     LS      4.333381E-01       8.666762E-01             104    LS     4.131156E-01      5.346103E-02
               5     LS      1.647448E+00       3.294896E+00             105    LS     1.570567E+00      2.032460E-01
               6     LS      7.004332E-01       1.400866E+00             106    LS     6.677461E-01      8.641259E-02
               7     LS      4.834797E-01       9.669595E-01             107    LS     4.609172E-01      5.964701E-02
               8     LS      2.633696E-01       5.267393E-01             108    LS     2.510790E-01      3.249198E-02
               9     LS      1.528240E+00       3.056479E+00             109    LS     1.456921E+00      1.885393E-01
              10     LS      6.702324E-01       1.340465E+00             110    LS     6.389547E-01      8.268672E-02
              11     LS      5.308363E-01       1.061673E+00             111    LS     5.060638E-01      6.548940E-02
              12     LS      2.525187E-01       5.050375E-01             112    LS     2.407345E-01      3.115330E-02
              13     LS      1.485955E+00       2.971910E+00             113    LS     1.416610E+00      1.833226E-01
              14     LS      6.615315E-01       1.323063E+00             114    LS     6.306598E-01      8.161328E-02
              15     LS      4.091379E-01       8.182757E-01             115    LS     3.900447E-01      5.047546E-02
              16     LS      2.007488E-01       4.014976E-01             116    LS     1.913804E-01      2.476641E-02
              17     LS      1.409623E+00       2.819246E+00             117    LS     1.343840E+00      1.739054E-01
              18     LS      5.771904E-01       1.154381E+00             118    LS     5.502545E-01      7.120807E-02
              19     LS      3.324490E-01       6.648979E-01             119    LS     3.169346E-01      4.101431E-02
              20     LS      1.479835E-01       2.959671E-01             120    LS     1.410775E-01      1.825675E-02
              21     LS      1.158140E+00       2.316279E+00             121    LS     1.104093E+00      1.428800E-01
              22     LS      4.176425E-01       8.352851E-01             122    LS     3.981526E-01      5.152468E-02
              23     LS      1.970197E-01       3.940393E-01             123    LS     1.878253E-01      2.430635E-02
              24     LS      7.479393E-02       1.495879E-01             124    LS     7.130356E-02      9.227350E-03

     *** LABEL NOTATIONS:  LS = LIFTING SURFACE,  ZIB = Z INTERFERENCE BODY ELEMENT,  ZSB = Z SLENDER BODY ELEMENT,
                                                  YIB = Y INTERFERENCE BODY ELEMENT,  YSB = Y SLENDER BODY ELEMENT.
```

Figure 4-3. V70 Aerostatic Data Recovery Output.

The forces and moments (computed at the aerodynamic grid points; e.g., centroids of the boxes in the Doublet Lattice method) have a set of nonzero components that depends on the nature of the aerodynamic theory, but the print format no longer assumes that there are less than six. The pressure data, however, remain as scalar data associated with the aerodynamic model collocation points (the j-set). The new format of the same output is shown in Figure 4-4.

```
EXAMPLE HA144C: HALF SPAN 15-DEG SWEPT UNTAPERED WING  HA144C           MARCH  19, 1998  MSC/NASTRAN  3/18/98   PAGE    26
 CANTILEVERED WIND TUNNEL MOUNT, DOUBLET-LATTICE AERO

 0.041 IN PLATE W/BEVELLED LEADING AND TRAILING EDGES

                          A E R O S T A T I C   D A T A   R E C O V E R Y   O U T P U T   T A B L E S
                                      MACH = 4.500000E-01               Q = 2.000000E+00
                                       AERODYNAMIC PRESSURES ON THE AERODYNAMIC ELEMENTS
                                                    AERODYNAMIC PRES.       AERODYNAMIC
                                    GRID   LABEL         COEFFICIENTS          PRESSURES
                                      1     LS           1.650689E+00         3.301379E+00
                                      2     LS           7.473636E-01         1.494727E+00
                                      3     LS           5.366663E-01         1.073333E+00
                                      4     LS           4.333380E-01         8.666760E-01
                                      5     LS           1.647448E+00         3.294896E+00
                                      6     LS           7.004333E-01         1.400867E+00
                                      7     LS           4.834797E-01         9.669595E-01
                                      8     LS           2.633697E-01         5.267394E-01
                                      9     LS           1.528240E+00         3.056480E+00
                                     10     LS           6.702325E-01         1.340465E+00
                                     11     LS           5.308362E-01         1.061672E+00
                                     12     LS           2.525186E-01         5.050372E-01
                                     13     LS           1.485955E+00         2.971910E+00
                                     14     LS           6.615313E-01         1.323063E+00
                                     15     LS           4.091378E-01         8.182756E-01
                                     16     LS           2.007488E-01         4.014975E-01
                                     17     LS           1.409623E+00         2.819246E+00
                                     18     LS           5.771903E-01         1.154381E+00
                                     19     LS           3.324490E-01         6.648980E-01
                                     20     LS           1.479835E-01         2.959669E-01
      *** LABEL NOTATIONS:  LS = LIFTING SURFACE,   ZIB = Z INTERFERENCE BODY ELEMENT,  ZSB = Z SLENDER BODY ELEMENT,
                                                    YIB = Y INTERFERENCE BODY ELEMENT,  YSB = Y SLENDER BODY ELEMENT.

EXAMPLE HA144C: HALF SPAN 15-DEG SWEPT UNTAPERED WING  HA144C           MARCH  19, 1998  MSC/NASTRAN  3/18/98   PAGE    27
 CANTILEVERED WIND TUNNEL MOUNT, DOUBLET-LATTICE AERO

 0.041 IN PLATE W/BEVELLED LEADING AND TRAILING EDGES

                           A E R O S T A T I C   D A T A   R E C O V E R Y   O U T P U T   T A B L E S
                                       MACH = 4.500000E-01               Q = 2.000000E+00
                                         AERODYNAMIC FORCES ON THE AERODYNAMIC ELEMENTS
GROUP  GRID ID  LABEL        T1                 T2                 T3                 R1                 R2                 R3
  1      101   LS     0.000000E+00      0.000000E+00      1.573657E+00      0.000000E+00      2.036459E-01      0.000000E+00
  1      102   LS     0.000000E+00      0.000000E+00      7.124863E-01      0.000000E+00      9.220241E-02      0.000000E+00
  1      103   LS     0.000000E+00      0.000000E+00      5.116217E-01      0.000000E+00      6.620865E-02      0.000000E+00
  1      104   LS     0.000000E+00      0.000000E+00      4.131155E-01      0.000000E+00      5.346102E-02      0.000000E+00
  1      105   LS     0.000000E+00      0.000000E+00      1.570567E+00      0.000000E+00      2.032460E-01      0.000000E+00
  1      106   LS     0.000000E+00      0.000000E+00      6.677462E-01      0.000000E+00      8.641262E-02      0.000000E+00
  1      107   LS     0.000000E+00      0.000000E+00      4.609172E-01      0.000000E+00      5.964701E-02      0.000000E+00
  1      108   LS     0.000000E+00      0.000000E+00      2.510791E-01      0.000000E+00      3.249199E-02      0.000000E+00
  1      109   LS     0.000000E+00      0.000000E+00      1.456922E+00      0.000000E+00      1.885393E-01      0.000000E+00
  1      110   LS     0.000000E+00      0.000000E+00      6.389548E-01      0.000000E+00      8.268674E-02      0.000000E+00
  1      111   LS     0.000000E+00      0.000000E+00      5.060637E-01      0.000000E+00      6.548938E-02      0.000000E+00
  1      112   LS     0.000000E+00      0.000000E+00      2.407343E-01      0.000000E+00      3.115328E-02      0.000000E+00
  1      113   LS     0.000000E+00      0.000000E+00      1.416610E+00      0.000000E+00      1.833226E-01      0.000000E+00
  1      114   LS     0.000000E+00      0.000000E+00      6.306595E-01      0.000000E+00      8.161324E-02      0.000000E+00
  1      115   LS     0.000000E+00      0.000000E+00      3.900446E-01      0.000000E+00      5.047544E-02      0.000000E+00
  1      116   LS     0.000000E+00      0.000000E+00      1.913804E-01      0.000000E+00      2.476641E-02      0.000000E+00
  1      117   LS     0.000000E+00      0.000000E+00      1.343840E+00      0.000000E+00      1.739055E-01      0.000000E+00
  1      118   LS     0.000000E+00      0.000000E+00      5.502544E-01      0.000000E+00      7.120807E-02      0.000000E+00
  1      119   LS     0.000000E+00      0.000000E+00      3.169346E-01      0.000000E+00      4.101432E-02      0.000000E+00
  1      120   LS     0.000000E+00      0.000000E+00      1.410775E-01      0.000000E+00      1.825674E-02      0.000000E+00
  1      121   LS     0.000000E+00      0.000000E+00      1.104093E+00      0.000000E+00      1.428800E-01      0.000000E+00
  1      122   LS     0.000000E+00      0.000000E+00      3.981524E-01      0.000000E+00      5.152467E-02      0.000000E+00
```

Figure 4-4. V70.5 Aerostatic Data Recovery Output.

The order of the print is first by GROUP and then, within each GROUP, by aerodynamic grid identification number for the aerodynamic grid points that are requested in Case Control.

Using PARAM, OPPHIPA, the user can request that the structural eigenvectors be computed on the aerodynamic mesh points (as a means of checking the spline quality). As pointed out earlier in this section, in V70 the aerodynamic points were merged with the structural points to form a collection of “grid points” that included the structure (GRIDs), the aerodynamic centroidal points (the aerodynamic grids) and the aerodynamic “connection” points (that describe the box geometry). PARAM, OPPHIPA caused the normal modes to be printed on the set of requested grid points, using the aerodynamic model box id’s to reference the aerodynamic results. Since the structural and aerodynamic points were seen as a single set, the eigenvectors were printed for the union of points as shown in Figure 4-5 (for the *MSC/NASTRAN Aeroelastic Analysis User’s Guide* sample problem “HA144C” using DISP=ALL in conjunction with PARAM, OPPHIPA, 1).

```
    EXAMPLE HA145E: HALF SPAN 15-DEG SWEPT UNTAPERED WING  HA145E           MARCH  19, 1998  MSC/NASTRAN  7/17/97   PAGE    36
     KE-METHOD FLUTTER ANALYSIS, DOUBLET-LATTICE AERO

     0.041 IN AL PLATE W/BEVELLED LEADING AND TRAILING EDGES

     EIGENVALUE =  4.656517E+04
         CYCLES =  3.434399E+01         R E A L   E I G E N V E C T O R   N O .         1

      POINT ID.   TYPE          T1             T2             T3             R1             R2             R3
             1      G       .0             .0         -3.320767E-03   5.275748E-02  -2.311046E-02    .0
             2      G       .0             .0          3.354167E-02   2.932303E-02  -1.964025E-02    .0
             3      G       .0             .0          1.197589E-01   1.662287E-01  -4.147353E-02    .0
             4      G       .0             .0          2.415489E-01   1.236907E-01  -2.596768E-02    .0
             5      G       .0             .0          3.895444E-01   2.266178E-01  -4.942432E-02    .0
             6      G       .0             .0          5.599717E-01   1.821056E-01  -3.271172E-02    .0
             7      G       .0             .0          7.364756E-01   2.416311E-01  -4.689491E-02    .0
             8      G       .0             .0          9.197437E-01   2.006610E-01  -3.359312E-02    .0
...
...
            38      G       .0             .0          6.353986E-01   2.007433E-01  -3.072660E-02    .0
            39      G       .0             .0          8.179795E-01   2.402105E-01  -4.210586E-02    .0
            40      G       .0             .0          1.000000E+00   2.000586E-01  -3.453320E-02    .0
           101      G       .0             .0          2.034613E-02    .0           -1.795585E-02    .0
           102      G       .0             .0          2.854189E-02    .0           -1.129299E-02    .0
           103      G       .0             .0          3.052557E-02    .0            4.810378E-03    .0
           104      G       .0             .0          2.312998E-02    .0            2.149920E-02    .0
           105      G       .0             .0          1.023022E-01    .0           -2.261905E-02    .0
           106      G       .0             .0          1.123743E-01    .0           -1.530259E-02    .0
           107      G       .0             .0          1.178079E-01    .0           -6.459678E-03    .0
           108      G       .0             .0          1.212074E-01    .0           -8.914401E-03    .0
           109      G       .0             .0          2.375906E-01    .0           -3.052108E-02    .0
           110      G       .0             .0          2.539630E-01    .0           -3.088147E-02    .0
...
...
           122      G       .0             .0          8.446766E-01    .0           -3.903546E-02    .0
           123      G       .0             .0          8.646864E-01    .0           -3.845605E-02    .0
           124      G       .0             .0          8.846442E-01    .0           -3.854457E-02    .0
     121000001      G       .0             .0             .0             .0             .0             .0
     121000002      G       .0             .0             .0             .0             .0             .0
     121000003      G       .0             .0             .0             .0             .0             .0
     121000004      G       .0             .0             .0             .0             .0             .0
...
...
     121000033      G       .0             .0             .0             .0             .0             .0
     121000034      G       .0             .0             .0             .0             .0             .0
     121000035      G       .0             .0             .0             .0             .0             .0
```

Figure 4-5. Partial V70 OPPHIPA Eigenvector Print.

Notice that, due to the DISP=ALL request, not only are all the structural (1 thru 40) and aerodynamic (101 thru 124) grid points shown, but so are “results” at the connection point grids (121000001 thru 121000035). These points are never connected to the structure, so their motion is always zero. To avoid this useless output, the user of earlier versions could limit the output displacement SET to points with identification numbers less than the fictitious numbers generated by MSC/NASTRAN for the aerodynamic box geometry connection grids.

In V70.5, with the separation of the aerodynamic model from the structural model, the printed output is reformatted to print the structural points for each mode and then the aerodynamic points for each mode. The connection points are automatically ignored. As in earlier versions, the output SET is controlled by Case Control references to aerodynamic box identification numbers. As a result, the equivalent DISP=ALL print in V70.5 would appear as shown in Figure 4-6.

```
     EXAMPLE HA145E: HALF SPAN 15-DEG SWEPT UNTAPERED WING  HA145E           MARCH  19, 1998  MSC/NASTRAN  3/18/98   PAGE    38
      KE-METHOD FLUTTER ANALYSIS, DOUBLET-LATTICE AERO

      0.041 IN AL PLATE W/BEVELLED LEADING AND TRAILING EDGES

      EIGENVALUE =  4.656517E+04
          CYCLES =  3.434399E+01         R E A L   E I G E N V E C T O R   N O .         1

       POINT ID.   TYPE          T1             T2             T3             R1             R2             R3
              1      G      .0             .0          -3.320767E-03   5.275748E-02  -2.311046E-02    .0
              2      G      .0             .0           3.354167E-02   2.932303E-02  -1.964025E-02    .0
              3      G      .0             .0           1.197589E-01   1.662287E-01  -4.147353E-02    .0
              4      G      .0             .0           2.415489E-01   1.236907E-01  -2.596768E-02    .0
              5      G      .0             .0           3.895444E-01   2.266178E-01  -4.942432E-02    .0
              6      G      .0             .0           5.599717E-01   1.821056E-01  -3.271172E-02    .0
              7      G      .0             .0           7.364756E-01   2.416311E-01  -4.689491E-02    .0
              8      G      .0             .0           9.197437E-01   2.006610E-01  -3.359312E-02    .0
...
...
             38      G      .0             .0           6.353986E-01   2.007433E-01  -3.072660E-02    .0
             39      G      .0             .0           8.179795E-01   2.402105E-01  -4.210586E-02    .0
             40      G      .0             .0           1.000000E+00   2.000586E-01  -3.453320E-02    .0
     EXAMPLE HA145E: HALF SPAN 15-DEG SWEPT UNTAPERED WING  HA145E           MARCH  19, 1998  MSC/NASTRAN  3/18/98   PAGE    54
      KE-METHOD FLUTTER ANALYSIS, DOUBLET-LATTICE AERO

      0.041 IN AL PLATE W/BEVELLED LEADING AND TRAILING EDGES

      EIGENVALUE =  4.656517E+04
          CYCLES =  3.434399E+01         R E A L   E I G E N V E C T O R   N O .         1

       POINT ID.   TYPE          T1             T2             T3             R1             R2             R3
            101      G      .0             .0           2.034613E-02    .0          -1.795585E-02    .0
            102      G      .0             .0           2.854189E-02    .0          -1.129299E-02    .0
            103      G      .0             .0           3.052557E-02    .0           4.810378E-03    .0
            104      G      .0             .0           2.312998E-02    .0           2.149920E-02    .0
            105      G      .0             .0           1.023022E-01    .0          -2.261905E-02    .0
            106      G      .0             .0           1.123743E-01    .0          -1.530259E-02    .0
            107      G      .0             .0           1.178079E-01    .0          -6.459678E-03    .0
            108      G      .0             .0           1.212074E-01    .0          -8.914401E-03    .0
            109      G      .0             .0           2.375906E-01    .0          -3.052108E-02    .0
            110      G      .0             .0           2.539630E-01    .0          -3.088147E-02    .0
...
...
            122      G      .0             .0           8.446766E-01    .0          -3.903546E-02    .0
            123      G      .0             .0           8.646864E-01    .0          -3.845605E-02    .0
            124      G      .0             .0           8.846442E-01    .0          -3.854457E-02    .0
```

Figure 4-6. Partial 70.5 OPPHIPA Eigenvector Print.

4.4 New Spline Methods and Features

A new spline method, the thin plate spline, has been implemented in the GI module for V70.5. The spline matrix, GTKG, has been split into two separate matrices: GPGK and GDGK. This allows independent spline interpolants (sets of aerodynamic and structural points) to be used for interpolating loads and displacements, respectively.

Prior to V70.5, splining methods for aeroelastic analyses included the Harder-Desmarais Infinite Plate Spline (SPLINE1 or IPS spline), the Infinite Beam Spline (SPLINE2 or Linear Spline), and an MPC-like interpolator (SPLINE3) that allows you to build an interconnection between select aerodynamic DOFs and select structural DOFs.

The new Thin Plate Spline (TPS) method is a three dimensional extension of the existing IPS spline. Just as the IPS, the TPS method uses an interpolation function based on the equations of an infinite plate, but the two dimensional "radially symmetric" solution function for a point load has been extended to a similar form in three dimensions. The accuracy of this method is similar to the IPS, but the structural geometry will not be projected to an interpolation plane. If, however, the structural geometry is planar, the TPS function becomes singular. In that case, the code will automatically compute the plane of the geometry, project all the points to that plane and use the IPS method instead. Just as with the IPS, the TPS is also singular for collinear or coincident structural points and will issue a FATAL error if either condition arises.

This enhancement provides an interpolation capability that supports aerodynamic analysis methods consisting of a general three dimensional mesh of points on the surface of the vehicle: general to the same extent that the structural mesh is general. Current aerodynamic methods in MSC/NASTRAN (Doublet Lattice, Mach Box, etc.) have restricted geometries such as planar components and regular arrays of points as well as restricted sets of degrees-of-freeedom such as normal forces and moments about a single axis. Prior to V70.5, this restricted geometry limited attempts by users to import higher fidelity data into MSC/NASTRAN.

The aeroelastic splines are currently used for computing:

- An equivalent structural force distribution given aerodynamic forces.
- An equivalent aerodynamic displacement given a set of structural displacements.

By virtual work, the transformation matrix to compute equivalent structural loads is the transpose of the transformation matrix to compute aerodynamic displacements. Prior to V70.5, the transformation matrix was computed only once, with an assumption (requirement) that the same set of aerodynamic and structural degrees-of-freeedom are coupled for both interpolation functions. This can be limiting for static aeroelastic applications where the set of structural DOFs that is appropriate for load application may not be the same set that is appropriate to represent the important deflections for the aeroelastic correction. The spline capability has been modified to allow (but not require) distinct sets of structural and aerodynamic points for the two transformations.

Benefits

The improvements to the aeroelastic splines in V70.5 can be broken into a new method, the USAGE feature and architectural improvements. The latter set of improvements lays the groundwork for the future implementation of higher order aerodynamic panel method. The benefits in V70.5 are that the spline computations are performed upstream of the structural computations so that debugging the aeroelastic model is faster: input errors are trapped earlier in the process. The new Thin Plate Spline method represents a modest improvement in spline methods. Since it is a 3D method (that is, the structural geometry does not need to be projected to a mean plane), it may be easier to choose an appropriate set of grid points for splining. Its behavior is otherwise similar to the IPS method (the SPLINE1 of earlier versions). Finally, the USAGE feature allows separate FORCE and DISPLACEMENT interpolants. This represents a potentially significant modeling flexibility that grants the modeler the opportunity to treat the force and displacement spline application separately.

Input

To control the method selection and usage (force or displacement) of each spline interpolant, the existing SPLINEi data are changed. All of the new fields are optional. All the spline entries have a METHOD field to select from the new spline methods (which default to the V70 method that had been indicated by the Bulk Data entry). All the spline entries also have a USAGE field to denote that the spline is for interpolating forces, displacements or both (defaulting to BOTH, for compatibility with V70). For more details see the *MSC/NASTRAN Quick Reference Guide*, Version 70.5.

Output

The new methods produce a pair of spline matrices from the GI module in place of the original one matrix. This allows you to treat the interpolation of forces separately from the interpolation of displacements. Further, if all the SPLINEi Bulk Data entry are USAGE=FORCE, the structural displacements are decoupled from the aerodynamic displacements, and RIGID aerodynamic analysis is performed.

Limitations

The TPS is not available for CAERO2 elements, since it represents a surface fitting technique rather than a beam-like interpolation.

Examples

Two sample problems serve to demonstrate the new spline features. Both sample problems are based on the HA144E example of the *MSC/NASTRAN Aeroelastic Analysis User's Guide*.

Thin Plate Spline (TPL: tpsp.dat, tpsnp.dat)

The tpsp.dat and tpsnp.dat files apply the TPS method to the splines that attach the forward swept wing aerodynamic model to the wing structure. This structure is represented as a beam with rigid-chord grid points attached upstream and downstream of the elastic axis at two points. The model, therefore, is amenable to surface splining methods (although it is probably more appropriate to use the beam spline for this class of problems). In tpsp.dat, the TPS method is applied, but the upstream and downstream wing structural grid points are coplanar. This represents a singular condition for the TPS method, but MSC/NASTRAN automatically identifies the singularity and "falls back" to the IPS method, using the plane of the grid points as the computational plane. Thus tpsp.dat yields results identical to the IPS.

In the second file, tpsnp.dat, the upstream and downstream wing grid points are artificially offset above and below the plane of the lifting surface. This allows the TPS method to be applied (since the set of structural points are no longer coplanar). As can be seen in the following table, using the TPS yields results identical to the IPS when the grids are planar and quite different results when the grids are nonplanar.

These differences are artificially large (and, therefore, physically unrealistic), due to the coarseness of this model and the fact that the aeroelastic correction of the entire wing is associated with displacements of just a few points. This model, however, serves to illustrate the similarities and differences between the IPS and TPS methods.

Planar Grids				Nonplanar Grids		
Parameter	Rigid	IPS Flex.	TPS Flex.	Rigid	IPS Flex.	TPS Flex.
CZ,ALPHA	-5.071	-5.986	-5.986	-5.071	-5.986	-5.628
CM,ALPHA	-2.871	-3.485	-3.485	-2.871	-3.485	-3.214

Force and Displacement Splines (TPL: ipspf.dat and ipspf2.dat)

While the majority of the V70.5 spline improvements are targeting the high order aerodynamic methods that will be supported in subsequent releases, a key improvement available now is the ability to use separate splines for force interpolation and displacement interpolation. This provides the aeroelastic modeler with new freedoms to more correctly model both the applied loading and the important aeroelastic feedback.

In the past, any displacement that was “sampled” for aeroelastic feedback also had to be a grid point with enough structural stiffness that it could be loaded with some portion of the aerodynamic load. If the grid point was “too weak” to distribute the external load, the associated displacement would be artificially high. While this behavior is not necessarily important for the internal load distribution, the large displacement was also used to compute the aeroelastic correction. Therefore, the aeroelastically corrected, applied load could well be heavily influenced by these loaded, but weak, points whose displacements are “sampled”. In V70.5, the modeler is free to sample weak points for displacement measurement while only applying aerodynamic loads to those points that have been modelled to allow load application. Each spline interpolant (SPLINEi Bulk Data entry) can be used to apply load, sample displacements or both. And the overall SPLINE model can be comprised of any combination of splines. The only requirement is that each aerodynamic point must be attached no more than once for displacement and once for force interpolation across all splines.

As an example, the HA144E model is again used (because the model’s coarseness shows dramatic differences due to slight modelling differences). In this case, the splines are modified to treat GRID 99 and GRID 100 (see the *MSC/NASTRAN Aeroelastic Analysis User’s Guide*) differently in two runs. GRID 99 in particular is an interesting point, because it really isn’t anywhere near the wing. It’s located on the beam fuselage just at the trailing edge of the canard. It shouldn’t be included at all, but its effects are dramatic because it is used (in combination with GRID 100) to measure the fuselage bending effects via the surface spline. (Note that the surface spline measures only the out-of-plane displacement, so the rotation of the fuselage GRID points is not “splined” unless a beam spline is used.)

Again, two runs are made: ipspf.dat and ipspf2.dat. In the former, wing forces are applied to GRID 99 and GRID 100, but they are omitted from the displacement sample (fuselage bending effects are mitigated in aeroelastic feedback). In the latter case, the reverse is done: fuselage bending is sampled, but GRID 99 and GRID 100 are not loaded with any forces from the wing. As seen in the following table, the aeroelastic effects are reduced significantly when wing forces are not applied to the fuselage points, but the fuselage bending is sampled. In both cases, however, the aeroelastic effects are larger than those obtained if the points are both loaded and sampled.

		Both	Loaded/No Displacements ipspf	Displacements/Unloaded ipspf2
Parameter	Rigid	IPS Flex.	IPS Flex.	IPS Flex.
CZ,ALPHA	-5.071	-5.986	-6.331	-6.199
CM,ALPHA	-2.871	-3.485	-3.744	-3.634

The aeroelastic modeler now has more freedom to choose how to couple the structural model to the aerodynamic model, since the mapping of forces can now be made independent of the mapping of displacements. In many cases this separation will not be necessary. But, as the use of built up FE models (internal loads models) increases in aeroelastic analysis, this splining option is expected to become increasingly useful.

4.5 Iterative and Reduced Basis Solution Algorithm for Static Aeroelasticity

Introduction

You can now solve one order of magnitude larger static aeroelasticity problems (SOL 144) than previous versions. This is due to the introduction of two new techniques: iterative and reduced basis solutions that take advantage of characteristics of the matrices and the physical problem. The computer resources for SOL 144 are now the same as for conventional static analysis (SOL 101), not an order of magnitude larger as they were in prior versions. The reduced basis analysis produces repeatable answers at low cost. The iterative analysis provides an accuracy comparable to the direct analysis used in the past, at somewhat higher cost than the reduced basis analysis. An automatic option starts with the reduced basis analysis, then switches to the iterative analysis, providing accurate and low cost analysis. The direct solution used in the past is also available.

The static aeroelasticity solution sequence SOL 144 assembles a matrix with a symmetric portion due to structural stiffness, K^s, and an unsymmetric portion K^u due to aeroelasticity effects. The most costly and computer resource-intensive portion of the analysis is several solutions involving these matrices, given a set of loads P, to solve for a set of displacements u,

$$[K^s + K^u]\{u\} = \{P\}$$

Three options are now provided for these solutions. The first option is a Direct method solution, where the unsymmetric sum in the brackets is decomposed and solved. This was the only option prior to Version 70.5. The unsymmetric part generally has many null rows and columns, but tends to have a larger bandwidth than the symmetric part. This fact can be used to reduce the cost and computer resources required by using alternate solution methods for the part involving the unsymmetric term.

The modified equation solved in the iterative solution (Iter method) is:

$$\{u\}_i = [K^s]^{-1}[P - K^u\{u\}_{i-1}]$$

where K^u is unsymmetric (aeroelastic effects, data block KALL), and K^s is symmetric and more tightly banded (structure, data block KLL). The loading function is P, and the unknown displacement vector is $\{u\}_i$. The bi-conjugate gradient solution used in the SOLVIT module is used in DMAP form in SOL 144. The equations for this method are given in the *MSC/NASTRAN Numerical Methods User's Guide*. This solution technique has been shown to be robust and efficient in testing. This method allows efficient restart from static analysis SOL 101 to SOL 144 and back to SOL 101 because the same matrix factor, used to perform operations involving $[K^s]^{-1}$, is used in both solution sequences. This matrix factor is required to compute the rigid body shapes $[D_m]$, so the major cost has been paid before starting the iterative solution step. The Iter method is more accurate than the new reduced basis method.

The third option is a reduced basis approximate solution (Ritz method), using Rayleigh-Ritz vectors as generalized functions to transform the equations to a reduced set in the u_y basis. A set of generalized loads P_1 is generated from user-supplied external loads (if any), appended to loads computed from inertial loads due to unit rigid body acceleration. The loads are applied to the structural stiffness matrix to compute the first set of displacement shapes U_1. These shapes are used to compute loads P_2 due to applying U_1 to the K^a matrix, $P_2 = K^a U_1$. Then, these loads are applied to K^s to compute U_2, a second set of displacement shapes. The two sets of displacement functions are appended to generate the reduction matrix ϕ_y, $U_1 U_2 = \phi_y$. This reduction matrix is used to reduce the two stiffness matrices and the load vector,

$$K_{yy} = \phi_y^T [K^s + K^a] \phi_y$$

$$P_y = \phi_y^T [P]$$

The equations in the reduced basis are solved for u_y, then back-transformed to the original basis u,

$$K_{yy} u_y = P_y$$

$$u = \phi_y u_y$$

Load epsilons similar to those provided in SOL 101 are provided for the new solution options. The Ritz method is the fastest option at present.

The Ritz and Iterative techniques are combined in a method named the “auto” method. The approximate solution computed in the Ritz phase is used as a starting vector for an iterative solution. The cost of this method is somewhere between the Ritz solution cost and an Iterative method-only solution, but it is as accurate as the Iter solution. This is the default method for problems where the analysis set is greater than 5000. The Direct method is the default method for smaller problems because the overhead associated with alternate solutions is a larger percentage of total costs for small size models.

Benefits

- The practical size of the models that can be analyzed is greatly enhanced.
- Performance of larger aeroelasticity models (DOF > 5000) is improved by orders of magnitudes.
- The computation cost, compared to SOL 101, is now 1.4 to 7 times greater, not hundreds of times greater.
- Efficient restarts from linear static analysis (SOL 101) to aeroelastic analysis (SOL 144) and visa versa.

Input

User parameter PARAM, AESMETH selects the method of solution. Its default value is SELECT, which selects the Direct method on models with less than 5000 DOF in the solution set, and the Auto method for larger size models. Other values of the parameter select a specific method regardless of problem size:

- DIRECT selects the direct solution.
- RITZ selects the reduced basis approximate solution.
- ITER selects the iterative solution.
- AUTO selects the reduced basis method for an approximate solution which is used as starting vectors for an ITER solution.

Other user parameters are:

AESMAXIT	Maximum number of iterations for the ITER method. Default=15.
AESRNDM	Number of random number vectors for generalized functions in the Ritz method. Default=2.
AESDISC	Tolerance for discarding generalized coordinates which are not linearly independent. Default= 1.E-8.
AESTOL	Convergence criteria for the iterative solver. Default=1.E-6.

A DIAG executive file entry (DIAG 50) introduced for other uses prior to Version 70.5 has been extended to provide the user information data described in the Output Section.

Output

All alternate methods specify the solution method being used, and name the user parameter that may be used to specify a specific method. For the iterative solution the number of iterations required for each pass through the iterative solver is printed for the last data block given the solution step, along with convergence data. Each trim case requires four solutions because the equation itself is solved in parts.

Diagnostics

Load epsilons (EPSILON) similar to SOL 101 are output for every invocation of the iterative solution. Each is associated with the name of the output “displacement” data block that is being solved. This allows you to measure the solution quality at each stage of the aeroelastic solution. Note, however, that each aeroelastic solution requires multiple invocations; so you will see more than an EPSILON for each subcase.

An optional diagnostic request DIAG 50 is provided to print information messages normally suppressed, because of the newness of the methods. Use of this executive deck entry is suggested when first using the alternate methods, or when a solution appears to be implausible. For production runs the default value is adequate. The following data is printed when DIAG 50 is present. The names in all capital letters are the names of matrices which appear in the output file.

- Maximum error at each iteration
- Maximum error for each loading condition at each iteration (NORMR)
- Beta, Alpha quantities (ALPHAI, BETAI)
- Retained generalized coordinates (RATIOFY). RITZ and Auto methods only. The loading functions used to generate generalized shape functions may not produce linearly independent shapes. The *U* matrix is tested for linear independence of its columns. Any which do not contain significant components orthogonal to prior vectors are discarded. The discarded DOFs have a null coefficient in the RATIOFY vector. Some are almost always discarded because many generalized loading functions are used to obtain the best possible accuracy in an approximate method of solution.

The error at each iteration can be inspected for convergence. The maximum number of iterations allowed by default is 15. If the error is observed to oscillate after the 10th iteration, iteration is stopped. The Beta and Alpha quantities are of interest mainly to the developers, and may be requested on Client Service communications. The load epsilon is the same quantity defined in the *MSC/NASTRAN User's Reference* under Static Analysis, and is a good measure of solution quality.

Limitations

No limitations are known at present.

Examples (TPL: ha144it1.dat, ha144it2.dat, ha144it3.dat, ha200it1.dat)

All test cases use the same small size model, with only the parameter Q varying between them. File ha144it1.dat has two trim subcases, and requests the direct solution method. File ha144it2.dat is the same problem requesting the iterative method. Both provide identical answers. File ha144it2d.dat is the same model as -it2 except that the Q value (see the TRIM entries) is increased by a ratio of 1000. This absurdly large value is used to demonstrate the robustness of the iterative solution. Some solution methods require positive definite matrices, but this value of Q makes the sum $[K^s + K^a]$ indefinite, a rigorous test of the solution method. File ha144it1d.dat shows the same solution using the direct solution. File ha144it3.dat uses the Ritz method, and ha144t4.dat the automatic method. File ha200it1.dat solves the *MSC/NASTRAN Aeroelastic Analysis User's Guide* problem HA200A for design iteration.

Test Results

An 86,000 DOF wing model was used to measure performance as a function of the number of structural DOFs attached to splines. Four sets of splines were used, ranging from a minimal set through moderately connected sets to one where almost every eligible DOF is attached to a spline. Results are normalized to a statics solution (SOL 101) of the same model, and are summarized in the table below. A question mark after a number indicates that either no data was available or that the data shown is an estimate, based on partial results because the model was too large for practical computation.

Table 4-2. Summary of Test Results.

1	2	3	4	5	6	7	8
				------- Density -----			
file name	method	Cpu ratio	Hiw. ratio	Ka ratio	K ratio	Front ratio	Its. Max
"Easy"range							
i0.1	iter	2.83	1.15				3
i0.2	ritz	1.38	1.14				
i0.3	direct	5.62	2.68	0.000154	0.965	1.17	
i0.4	auto	1.98	1.14				1
a.1	iter	3.70	1.20				3
a.2	ritz	2.16	1.17				
a.3	direct	8.16	2.93	0.053	1.02	1.67	
a.4	auto	3.17	1.27				2
Start of "difficult"range							
b.1	iter	6.03	1.28				9
b.2	ritz	1.61	1.28				
b.3	direct	20.(est.)	15.16	14.99	15.92	4.55	
b.4	auto	4.89	1.28				7
c.1	iter	7.50	1.93				12
c.2	ritz	1.77	1.94				
c.3	direct	900.(est.)	64.?	133.98	134.91	30.96	
c.4	auto	6.36	1.93				11

This testing was done with a 32 bit non-vectorizing machine.

The key to understanding this data is in column 8 (Its. max), which lists the number of iterations that the iterative solver required to solve the problem. As the degree of complexity goes from easy (file -i0.1) to difficult (file -c.1) the number of iterations rises from 3 to 12. When the Ritz method is used to obtain initial vectors used in the iterative solver, the number of iterations is always less (1 to 11) than starting with zero initial conditions, i.e., using the Iter method by itself. Conversely, when the Auto method requires many iterations, it is a signal that the Ritz solution is poor. This is signaled by large load epsilons in the Ritz solution.

CPU ratios are shown in column 3. The Ritz method costs only 40 % more than the static solution for the easy configuration. Even for the complex configuration (file c.2) a low-quality Ritz answer costs only twice the cost of the static run. This type of run should be adequate for diagnosing gross modeling errors, such as connecting aero DOFs on the vertical tail to splines on the wing tips. After these errors are removed, the modeler can switch to the Auto method and obtain good quality answers for less than seven times the static solution cost. The ratio may be more like the Ritz ratio when the loading functions are smooth, that is, two to one cost growth. It was difficult to get good data for the complicated models using the Direct method, as indicated by the question marks. In any event, CPU times for the Direct method grow rapidly with complexity, leading to impractical cost and storage space requirements.

Storage cost ratios are shown in column 4 (hiwater ratio). For all but the Direct method, the storage required is essentially the same as the SOL 101 run. For the Direct method, it grows rapidly with problem complexity, and may be even higher than the estimates next to the (est.) symbols. The cause of this cost growth can be seen in columns 5 and 6, where the density of K^a and $K = [K^a + K^s]$ matrices are compared with K^s, the statics matrix. The "easy" models do not change the ratio appreciably from the statics matrix, but for the b.3 and c.3 models (Direct method) it grows rapidly. Similar effects are shown in column 7, where the maximum front size is compared to statics. Maximum front size is roughly analogous to bandwidth, with computation costs and storage requirements being quadratic on this effect. Even the "easy" models (i0.3, a.3) show a significant growth due to sprinkling spline points outside the band, and the effects on the "difficult" models is profound.

In summary, the Ritz solutions are always economical, and have a repeatable, predictable cost. They may produce low accuracy solutions but are adequate for model checkout. The Auto solutions are always correct, but more costly. On this model, the Iter method is a last choice, and that result may be general across all medium- through large-size models. The Direct method is best-suited for small, simple models and for those with very large computers.

4.6 Splining to Upstream Superelements in Dynamic Aeroelasticity

Introduction

Prior to Version 70.5 aeroelastic splines could only be specified in the residual structure, and not in upstream superelements. In Version 70.5, aeroelastic splines may be specified in both residual and upstream superelements for aerodynamic analysis (SOLs 145 and 146). For static aeroelastic analysis (SOL 144), you still can only specify splines in the residual structure.

MSC/NASTRAN has two major classes of superelements. Grid list superelements, or list s.e.'s, the only class available prior to Version 69, have all of their input in the main Bulk Data Section. Grid points are placed in superelements with several options for listing superelement interior points, such as the SESET entry. Partitioned bulk data superelements, or parts, introduced in Version 69, are defined by a separate section of the bulk data file for each superelement. Parts are discussed in the Version 69 and Version 70 *MSC/NASTRAN Release Notes*. External superelements, a sub-class of parts, also allow aeroelastic splines and are discussed in the next subsection.

Input

No new input is required. Existing SPLINEi and SETi entries can now be applied to upstream superelements. The input decks differ for the two types of superelements.

Grid List Superelements

All input data is in a main Bulk Data Section. The SPLINEi entries can now invoke grid points which are in a superelement.

Part Superelements

When there are part superelements their input file starts with a BEGIN SUPER=[seid] entry and continues until the next BEGIN entry or an ENDDATA entry, which ends the total input file. A part superelement input file to which splines are attached has all of the entries used to describe the part, including all SPLINEi and SET entries for that part superelement only. The CAEROi entries input in the main Bulk Data Section are used in part superelement reduction.

Output

No new output is provided.

Examples (TPL: ha145ss1.dat, ha145ss2.dat, ha146ss1.dat)

These three examples are variations of the standard test problems HA145E and HA146E which are described in detail in the *MSC/NASTRAN Aeroelastic Analysis User's Guide*. The modified test files are named ha145ss1.dat, ha145ss2.dat, and ha146ss1.dat. The model is based on a cantilevered swept wing made of quadrilateral elements, with a mesh of 7 elements span-wise by 4 elements chord-wise. Most of the labor described below is associated with changing the structural model into a superelement configuration. The labor involved in partitioning the aeroelastic data is minor by comparison.

TEST FILE ha145ss1.dat This demonstration problem performs flutter analysis. It is a variation from the ha145e.dat file. It is described first because grid list superelements, which it uses, require the least amount of modification for both use of the spline capability and for converting a non-superelement model to a superelement model. The wing model is cut into two superelements along a span line. This is accomplished by placing all points inboard of the span line in superelement 10 with SESET, 10 entries; and all points on the outboard in superelement 20. The boundary points on the span line are placed in the residual structure by default. SPOINT and SEQSETi entries are added to allow computation of component modes for the superelements. Figure 4-7 shows the additions to the test deck required for this example. No other changes to the input file are required. All of the answers agree well with a non-superelement equivalent model.

```
BEGIN BULK
$ DATA ADDED FOR SUPERELEMENTS
$ PLACE INBOARD SECTION IN S.E. 10
SESET, 10,    1      THRU   5
SESET, 10,    10     THRU   13
SESET, 10,    17     THRU   21
SESET, 10,    25     THRU   29
SESET, 10,    33     THRU   37
$ OUTBOARD SECTION IN S.E. 20
SESET, 20,    7      8      15     16     23     24     31
SESET, 20,    32,    39     40
```

```
$ GRID POINTS 6, 14, 22, 30, 38 ARE BOUNDARY POINTS, S.E. 0
$ GEN. COORDS FOR C.M.S.
SPOINT,      70     THRU   85 $ FOR S.E. 10
SPOINT,      86     THRU   99 $ FOR S.E. 20
SEQSET1, 10, 0,     70     THRU   85
SEQSET1, 20, 0,     86     THRU   99
```

Figure 4-7. Superelement Related Portion of Input File ha145ss1.dat.

TEST FILE ha145ss2.dat This demonstration problem starts with the same non-superelement model used above. The wing model is cut in half along its mid-chord line to make two part superelements. The elements for the leading edge part of the wing are moved after BEGIN SUPER = 10, along with any grid points to which they are attached, and mass entries (CONM2). Masses and loads on boundary points may be placed totally in the superelement or in the residual structure, or split between them, with equal validity. The trailing edge structure is moved after BEGIN SUPER=20. The elements that remain in the residual structure are then inspected to determine their connected grid points. Boundary points which were moved to a part superelement must have their GRID entries duplicated and added to other part files which have common boundary points.

The SPLINEi and SETi Bulk Data entries are placed only in the part file which has the grid points to which they attach. Other aeroelastic entries are placed in the main Bulk Data Section only. Figure 4-8 shows the fragments of the test file that are discussed here.

```
BEGIN BULK $ MAIN BULK       $      $      $      $      SE 0
$   MODES FOR TOTAL STRUCTURE
$EIGRL  MID,    F1,     F2,     NDES
EIGRL,  10,     ,       ,       10
$SENQSET, SEID,        N   GENERALIZED COORDS FOR COMPONENT MODE SYNTHESIS
SENQSET, 10,   10
SENQSET, 20,   10
..
..

BEGIN SUPER=10 $ PART STYLE SUPERELEMENT     $      $      SE 10
$   COMPONENT MODES FOR SE 10
EIGRL,10,,,10 $
GRID    1               0.0     0.0     0.0
GRID    2               .211491 .7893   0.0
GRID    3               .422983 1.5786  0.0
..
..
CQUAD4  13      1       14      15      23      22
CQUAD4  14      1       15      16      24      23
$       PID     MID1    T      MID2 12.*I/T**3  MID3    TS/T    NSM     $
..
..
$ FOR SE 10, CHANGE BOX2 TO 112
$SPLINE1 100     101     101     124     100     .0
SPLINE1 100     101     101     112     100     .0
$       SID     G1      G2      G3      G4      ETC
SET1    100     2       4       6       8       9       11      13      +S1
+S1,     15      18      20      22      24 $     25      27      29      +S2
$+S2     31      34      36      38      40
$        M1     M2      M3      ETC

BEGIN SUPER=20 $ PART STYLE SUPERELEMENT     $      $      SE 20
$   COMPONENT MODES FOR SE 20
EIGRL,10,,,10
GRID    17              1.03528 0.0     0.0
GRID    18              1.24677 .7893   0.0
..
..
CQUAD4  28      1       31      32      40      39                      +M00013
+M00013                 .041    .041    0.0     0.0
$       PID     MID1    T      MID2 12.*I/T**3  MID3    TS/T    NSM     $
..
..
SPLINE1 200     201     201     212     100     .0
$       SID     G1      G2      G3      G4      ETC
$SET1    100     2       4       6       8       9       11      13      +S1
$+S1     15
SET1,  100,   18      20      22      24      25      27      29      +S2
+S2     31      34      36      38      40
$ END OF SE 20 FILE
ENDDATA
```

Figure 4-8. Superelement Related Portion of Input Data File ha145ss2.dat.

Component mode synthesis is requested by adding SENQSET entries for each part in the main Bulk Data Section, and by placing EIGRL entries in each part of the input file. Answers agreed well with a non-superelement model which did not have any omitted DOFs.

Part superelements support non-unique coordinate system IDs in separate superelements. However, this practice should be avoided in aeroelastic applications since the CAEROi data may become confused.

TEST FILE ha146ss1.dat This test file performs gust response analysis. The same wing model cut along a span line used in the prior model is also used here. One half is in a part superelement, and the other in a grid list superelement.

Limitations

Splining to superelements is not supported in SOL 144 (static aeroelasticity).

4.7 Use of External Superelements in Aerodynamic Analysis

An **external superelement** is a model of a component or collection of components which are reduced to a set of boundary points on a **generation** run, and stored on a permanent file. The external superelement itself may have superelements included in it, or it may have only a residual structure. The a-set points of its residual structure become the boundary points of the external superelement. The boundary points are attached to another model on a subsequent **assembly** run, and the total assembled model is solved. If data recovery is desired in the external set of superelements, a third **data recovery** run is made. For more details on external superelements, see the *MSC/NASTRAN Release Guide*, Version 70.

Input

The input requirements for the three types of external superelement runs are best explained by example. The **generation** run is in ha145ss7.dat while the **assembly** data is in ha145ss8.dat. The external superelement for this example consists of an engine model, analyzed in a multilevel superelement configuration. In principle, an external superelement can be a very large, detailed model, although the demonstration model is small. The scenario is that this model is provided to the aircraft developer by the engine manufacturer. The aircraft developer attaches this engine model at several places in one aircraft model and plans to attach it to many other aircraft models over time. The aircraft modeler does not want to modify the engine model, even inadvertently. He also wants the IDs of the grid points and elements of the aircraft to be independent of the IDs used in the engine model. The latter requirement is met by making the engine model a part superelement. The former requirement is met by analyzing the engine model once in a stand-alone generation run, then attaching the boundary matrices which result from this run to many other models. The cost of reducing the engine model is paid only once, and the engine model cannot be changed inadvertently on subsequent runs.

Generation Run

The engine modeler wants to make sure that his model is thoroughly validated before it is turned over to the aircraft developer. The engine modeler also knows which structural points are capable of taking loads imposed by aerodynamic forces. It is therefore desirable for the engine modeler to define the SETi “hard points” to which aerodynamics will be attached via splines. For checkout purposes, the engine modeler may also design the aeroelastic splines which couple structural grid points to points at the center of the boxes used to define aerodynamic properties. The center of the boxes are called aero points. The geometry of aero points is described on CAEROi entries, which in turn are referenced by SPLINE entries, which reference SETi entries that list the structural points, which interact with the aero points on specific CAREOi entries. The engine modeler therefore also prepares the CAERO-type and splining data for his component. He can use this data to check the aeroelastic behavior of his model by grounding the engine attach points in a checkout run which simulates a wind tunnel experiment. After the model has passed all of the checkout and validation runs he removes the ties to ground which simulate the wind tunnel condition and passes the input file for the engine model to the aircraft developer.

The engine model is analyzed and reduced to its boundary matrices in the generation run. This run places the boundary matrices in an engine model database. This database and bulk data entries which define the geometry and connectivity of the interface points are added to a model where the total aircraft is analyzed. In this example, this engine model is attached to an aircraft model of a swept-forward wing and a canard. This model was derived from the *MSC/NASTRAN Aeroelastic Analysis User’s Guide*, model HA200A. Several product improvements were made to the model, including the addition of engine mounting brackets at each wing tip.

The input data file is shown in Figure 4-9. ASSIGN statements are placed in the File Management Section (FMS) to provide permanent storage of the database. PARAM, EXTOUT, MATRIXDB is placed in Case Control to request that the reduced matrices be stored in matrix format directly on the database, rather than on the other options of user tapes or DMIG entries. The engine is modeled about a point mid-way between its front mount points so that its grid points are symmetrical about the x-z plane which goes through this point. The engine is modeled as a box which hangs below the wing, attached by vertical links and a sway damper. Grid list superelements are used. The grid points on the lower surface are interior to superelement 2, the grid points on the upper surface are interior to superelement 1, and the attach points at the tops of the links are the only physical points in the residual structure of the component model. Generalized coordinates are attached to each superelement. They are used to transmit the dynamic characteristics of the superelements to the assembled model.

```
$ FILE ha145ss7.dat   GENERATE AN EXTERNAL SUPERELEMENT
$ ENGINE MODEL, MULTILEVEL SUPERELEMENT  MAG  27 AUG 97
$ INCLUDING AERO SPLINES   last revised 23 oct 97
ASSIGN MASTER='eng.MASTER' $ SAVE FOR ha145ss8 RUN
ASSIGN DBALL ='eng.DBALL'
DIAG 8
SOL 145 $
CEND
TITLE = ENGINE MODEL, COMPLETE TO FUSELAGE ATTACH POINTS    ENG
LABEL = USED AS EXTERNAL S.E. FOR EXTSP MODEL
ECHO   = BOTH
PARAM, EXTOUT, MATRIXDB $ SAVE REDUCED MODEL IN DATABASE
METHOD   = 20
$ DO FLUTTER ON A STAND-ALONE VERIFICATION RUN
$FMETHOD  = 30 $ RUN A FLUTTER CASE FOR COMPONENT VERIFICATION
BEGIN BULK $
$ ALL GRID POINTS MEASURED FROM ENGINE CENTERLINE PLANE
$ FUSELAGE ATTACH POINTS
GRID,   12,     ,       0.0,     -1.
GRID,   14,     ,       0.0,     1.
GRID,   22,     ,       5.0,     -1.
GRID,   24,     ,       5.0,     1.
$ THE ATTACH POINTS IN THIS MODEL AND THE ha145ss8 MODEL MAY
$ HAVE GIDS AND Y LOCATIONS WHICH DIFFER, BUT THE RELATIVE SPACING OF
$ POINTS MUST BE IDENTICAL.
$ ENGINE GRID POINTS
GRID,   112,    ,       0.0,     -1.,   -4.
GRID,   114,    ,       0.0,    -1.,    -2.
GRID,   116,    ,       0.0,     1.,    -4.
GRID,   118,    ,       0.0,    1.,     -2.
GRID,   122,    ,       5.0,    -1.,    -4.
GRID,   124,    ,       5.0,    -1.,    -2.
GRID,   126,    ,       5.0,    1.,     -4.
GRID,   128,    ,       5.0,    1.,     -2.
$ ENGINE COWL.  FRONT AND BACK PANELS OPEN
CQUAD4, 112,    112,    112,    116,    126,    122 $ BOTTOM
CQUAD4, 114,    112,    114,    118,    128,    124 $ TOP
CQUAD4, 116,    112,    116,    126,    128,    118 $ SIDE
CQUAD4, 122,    112,    122     112,    114,    124 $ OTHER SIDE
PSHELL, 112,    1,      .05,    1
$ MOUNTING LINKS
CBAR,   12,     12,     12,     114,    0.,     1.,     0.
CBAR,   14,     12,     14,     118,    0.      1.      0.
CBAR,   1418,   12,     114,    118,    0.      0.      1.
CBAR,   22,     12,     22,     124,    0.      1.      0.
CBAR,   24,     12,     24,     128,    0.      1.      0.
CBAR,   2428,   12,     124,    128,    0.,     0.,     1.
PBAR,   12,     1,      .1      .2      .3,     .4
$ ADD SWAY BRACE TO SHOW FRONT FROM BACK, LH VS. RH, TEST BAA EFFECTS
CVISC,  1218,   1218    12,     118
PVISC,  1218    .02
$MAT1   MID     E       G       NU      RHO ,       ,       GE
MAT1,   1       1.44+9  5.40+8,         ,       1.0,    ,       ,       .01
CONM2,  1180,   118,    ,       100. $ ENGINE WEIGHTS
CONM2,  1280,   128,    ,       100. $
CONM2,  1140,   114,    ,       100. $
CONM2,  1240,   124,    ,       100. $
$ PUT THE UPPER SURFACE IN S.E. 1 AND THE LOWER SURFACE IN S.E. 2
SESET,  1,      114,    118,    124,    128
SESET,  2,      112,    116,    122,    126
$ MAKE S.E. 2 THE TIP
DTI,    SETREE, 1,      2,      1,      1,      0
$ DON'T ALLOW UNUSED GEN. COORDS TO BE AUTOSPCD
SPCOFF1,        0,      701     THRU    720
SPOINT, 701,    THRU,   720 $ GEN. COORDS
SEQSET1, 1,     0,      701     THRU    710 $ FOR CMS
SEQSET1, 2,     0,      711     THRU    720
$  DATA FOR WIND TUNNEL VALIDATION RUN FOLLOWS.  NOT NEEDED FOR EXTERNAL S.E.
$MKAERO1 M1     M2      ETC                                             +MK
```

Figure 4-9. Input Data File ha145ss7.dat for the Generation Run.

```
MKAERO1 0.90    1.20                                                    +MK
$+MK     K1     K2      K3      K4      K5      ETC
+MK     0.001   0.01    0.1     0.3     0.5     1.0
$       SID     METHOD  DENS    MACH    VEL     IMETH   NVALUE  EPS     $
FLUTTER 30      PK      1       2       3       S       8
$       SID     F1      F2      F3      F4      F5      F6      F7      $
FLFACT  1       1.0                                                     DENSITY
FLFACT  2       0.90                                                    MACH
FLFACT  3       1000.0  1100.0  1200.0  1300.0  1400.0  1500.0  1600.0  VELOCITY
FLUTTER 40      PK      11      12      13      S       8
FLFACT  11      1.0
FLFACT  12      1.20
FLFACT  13      1000.0  1100.0  1200.0  1300.0  1400.0  1500.0  1600.0
$       ACSID   VELOCITY REFC   RHOREF  SYMXZ   SYMXY
AERO    1               10.0    2.378-3
CORD2R, 1,     0       0.      0.      0.      0.      0.      10.
,       20.    0.      0. $ IDENTICAL TO BASIC.  EXTERNAL BASIC IS AT
$ CENTER POINT BETWEEN FRONT MOUNTS.  THIS ENTRY CHANGED WHEN ENGINE MODEL
$ MOVED TO WING TIP IN EXTSP INPUT FILE
PARAM   WTMASS  .031081
PARAM   AUNITS  .031081
PARAM,  GRDPNT, 0
$       EID     PID     CP      NSPAN   NCHORD  LSPAN   LCHORD  IGID
CAERO1  4001    1000            2       2                       1
$      ( FWD LEFT POINT  )     CHORD ( FWD RIGHT POINT   )     CHORD
$       X1      Y1      Z1            X12     X4            Y4      Z4      X14
,      0.0,    2.,     2.     5.     0.0,    4.,     2.     5.
$       PID     B1      B2      B3      B4      B5      B6
PAERO1  1000
EIGRL,  20,     ,       ,      5
$ PRINT DOF MAP FOR HOOKING UP DOF IN STEP 2 WITH EXTRN ENTRY
$ USETSEL=128 WILL PRINT ONLY ASET DOF
PARAM,  USETPRT,       0
PARAM,  USETSEL,128 $ THIS VALUE PRINTS ONLY A–SET
$ PUT AIR ON THE TOP SURFACE
$      EID     CAERO          ID1     ID2     SETG    DZ      DTOR    CID
SPLINE2,4001   4001    4001    4004    4001    0.      1.      0      +SPRW
$       DTHX    DTHY
+SPRW   –1.     –1.
$       SID     G1             G2             G3             G4
SET1    4001    114    118     124     128
ENDDATA
```

Figure 4-9. Input Data File ha145ss7.dat for the Generation Run. (Cont.)

The analysis set (a-set) of the residual structure defines the DOFs in the boundary matrices. It is the user's responsibility to insure that the proper set is available. There are many ways to do this. A method that has been verified by testing is to turn off AUTOSPC for generalized coordinates with an SPCOFF entry listing the generalized coordinates. This avoids having the AUTOSPC function placing unused generalized DOFs on single point constraints, which makes them unavailable for assembly. Also apply SPCOFF to any boundary grid points which may have singular DOFs. Each boundary grid point must have 6 DOFs in the a-set, even when some of these DOFs are singular. The singularities are removed in the assembly run. The number of generalized coordinates in the a-set must match the number listed on SECONCT and EXTRN entries in the next run.

All CAERO-type data, that is, the CAEROi entries and other entries referenced on a CAEROi entry, and the spline data, is placed in the main Bulk Data Section because these are grid list superelements. In the superelement spline method for **internal superelements** (that is, list and part superelements in the same input file) it is necessary to place all CAEROi entries for the entire model in the main Bulk Data Section. This allows the columns of the spline matrices to be properly aligned. This requirement is relaxed for external superelements. The engine modeler inputs CAERO-type entries only for the components in his model. A device is added to the total aircraft model, described later, which performs this alignment function.

Assembly Run

Figure 4-10 shows portions of the ha145ss8.dat input data file. The assembled aircraft model of ha145ss8.dat has an FMS section with ASSIGN statements for the databases created on this run, and to be used on the external superelement data recovery run. It also has ASSIGN statements for the database created on the prior generation run, given the logical names se3db and se4db on this run. DBLOCATE statements state that the a-set matrices of the residual structure of the engine model are to be treated as external superelement numbers 3 and 4 in the assembled model. The same database (and its associated matrices) are used to model both engines, duplicating the function of identical images with the older technology. PARAM, USETPRT, 0 is added the first time this model is run to print the correlation of the external sequence numbers of the aero points with their internal sequence number.

```
$ FILE ha145ss8    TOTAL MODEL.  BRING IN ENGINES AS EXTERNAL S.E.S,
$ INCLUDING AERO SPLINES   last revised 23 oct 97
ASSIGN MASTER='extsp.MASTER' $
ASSIGN DBALL ='extsp.DBALL'  $
$ LOCATE S.E.S 3, 4 FROM S.E. 3'S DATABASE:
ASSIGN SE3DB = 'eng.MASTER'
DBLOCATE DB=(EXTDB) CONVERT(SEID=3) LOGI=SE3DB $ USE AS RH ENGINE
DBLOCATE DB=(EXTDB) CONVERT(SEID=4) LOGI=SE3DB $ USE AS LH ENGINE
SOL 145 $ CHANGED TO RUN FLUTTER ONLY
..
..
BEGIN BULK $ THE WING AND WING STORES ARE IN S.E. 0
$ ENGINE ATTACH POINTS ON WING TIPS.  ENGINES ARE EXTERNAL SUPERELEMENTS
$ RH SIDE
GRID,    12,   ,      18.83975 14.,  0.
GRID,    14,   ,              18.83975 16.,  0.
GRID,    22,   ,              23.83975 14.,  0.
GRID,    24,   ,              23.83975 16.,  0.
$ ENGINE MOUNT BRACKETS
RBE2,   1120,   121,    123456, 12      14 $ FRONT
RBE2,   1140,   122,    123456, 22,     24 $ REAR
BEGIN SUPER=3 $ RH ENGINE AS EXTERNAL S.E.   $       $       S.E. 3
PARAM,  FIRSTKI, 161 $ INTERNAL SEQUENCE NUMBER OF FIRST AERO POINT
$ SAVE EXTERNAL DATARECOVERY OUTPUT (EXDROUT)
$ BLOCKS ON THE DATABASE FOR USE IN SUBSEQUENT DATA RECOVERY:
PARAM,EXTDROUT,MATRIXDB
$ EXTRN DESCRIBES HOW TO HOOK UP THE DBLOCATED MATRICES TO
$ THE CURRENT MODEL.  NOTE THAT THE ORDER OF THE GRIDS MUST BE
$ IN ASET ASCENDING ORDER OF THE ORIGINAL PROBLEM.
$ (SEE USET PRINTOUT FROM RUN1.)
$EXTRN  GID     CID     GID     CID     GID     CID     GID     CID
$       GID     CID     (ETC)
EXTRN,  12      123456  14      123456  22      123456   24     123456
,       701,    0,      702,    0,      703,    0,      704,    0,
,       *4,     =,      *4,     =,      *4,     =,      *4,     =
=3 $ DUPLICATE THE DUPLICATOR
,       720,    0
$ DEFINE THE GEOMETRY OF THE EXTERNAL SUPERELEMENT INTERFACE POINTS
$ FUSELAGE ATTACH POINTS, MEASURED FROM AIRCRAFT CENTERLINE
$ INPUT AN AXIS SYSTEM AT CENTER OF FRONT MOUNTS, RH SIDE
$ THIS CORDI ENTRY INPUT IN .CMN DECK
$CORD2R,        3,      0       18.83795, 15., 0.      18.83795, 15., 1.
$,      20.,    15.,    0. $ MID-POINT OF FRONT MOUNTS
GRID,   12,     3,      0.0,    -1.
GRID,   14,     3,      0.0,    1.
GRID,   22,     3,      5.0,    -1.
GRID,   24,     3,      5.0,    1.
..
..
ENDDATA
```

Figure 4-10. Portions of Assembly Run Input Data File ha145ss8.dat.

The main Bulk Data Section includes the conventional data, plus GRID entries that list the points to which the engine links are attached. These points are connected to the wing grid points with rigid elements. The conventional aircraft model is described in the *MSC/NASTRAN Aeroelastic Analysis User's Guide*, and is broken into part superelements. SEBULK entries are used to state that the external superelements are numbered 3 and 4. SECONCT entries are required to attach the generalized coordinates of part superelements 3 and 4 (the external superelements) to scalar points in the residual structure.

The external superelement geometry is described in part superelement files 3 and 4, which start with the delimiters BEGIN SUPER=[SEID]. PARAM, FIRSTKI, 161 lists the location of the first k-set point of the CAEROi point of superelement 3 in the internal sequence. PARAM, FIRSTKI, 201 lists the location of the first k-set point of the CAEROi point of superelement 4. The internal sequence of the k-set DOFs is a closed set, starting with the value of 1. The lowest-numbered CAEROi entry uses the first internal number, followed by a set of numbers in the range NSPAN*NCHORD-1 for the remaining points defined on that CAEROi entry. The next lowest-numbered CAEROi entry starts with the next internal number. As it is easy to miscalculate this number, a table correlating the external (CAEROi ID, etc.) vs. internal sequence of the k-set is printed at the beginning of the run, and has the appearance:

```
USET  DEFINITION  TABLE  (INTERNAL  SEQUENCE,  ROW  SORT) K    DISPLACEMENT SET
           -1-        -2-        -3-        -4-        -5-       -6 . . .
  1=     1000-3     1000-5     1001-3     1001-5     1002-3     1002-5     . . .    . . .
161=     3001-3     3001-5     3002-3     3002-5     3003-3     3003-5     . . .. . .
201=     4001-3     4001-5     4002-3     4002-5     4003-3     4003-5     . . .
```

It is essential that the FIRSTKI values be input correctly because they are used to make a partitioning vector that inserts the spline matrices for the external superelements in the proper columns. Some checks for necessary but not sufficient attributes are made to determine that the columns have been inserted properly. One such check is for the presence of null columns in the total assembled spline matrix. All null columns are identified in terms of their k-set index. While null columns may be permissible in some circumstances, they are usually an indication of a modeling error, and should be checked out. A corollary of this discussion is that if the external superelement has more than one CAEROi entry their Ids must be numbered such that they are adjacent to each other in the sorted sequence of the assembly input file. Also note that all CAEROi entries are placed in the same interference group to couple their aerodynamic effects.

PARAM, EXTDROUT, MATRIXDB selects the database-matrix option for obtaining the boundary matrices. The EXTRN entry lists the boundary points and their connected DOFs, including the generalized coordinates. The grid points and generalized coordinates of the external superelement are entered here. The grid points are located relative to a local coordinate system which moves the centerline plane of the engine to the wing tip. This is the same coordinate system used to locate the CAEROi entry. No ASETi entries are used so that all DOFs in the part file remain in its a-set.

The CAEROi-type data for the external superelements is copied from the set used for the engine reduction run, except that a coordinate system is referenced which aligns the CAEROi entry with the engine mount points.

Data Recovery

This type of run is useful in static and normal mode analysis, and is explained in the *MSC/NASTRAN Release Guide,* Version 70. It is not supported in SOLs 145 and 146 at present.

4.8 Omitted Degrees-of-Freeedom in Static Aeroelasticity

Introduction

Aeroelastic splines that have non-zero terms for their omitted degrees-of-freeedom reduce the accuracy of the solution. Because of the addition of two new solution techniques (see Section 4.5) in Version 70.5, the practical size of a problem that can be analyzed has been increased by orders of magnitude. Therefore, there is no need to omit degrees-of-freeedom that are attached to aeroelastic splines. Further, the implementation of the static condensation method required to process omitted DOFs in static aeroelasticity (SOL 144) is incomplete, and can introduce large approximations (see CSR30017). Hence, from Version 70.5 onwards MSC/NASTRAN will exit with a fatal message if aeroelastic splines have non-zero terms for their omitted degrees-of-freeedom.

Guidelines

Errors less serious than those associated with splining to omitted DOFs occur when static loads are placed on omitted DOFs and when DOFs with mass are omitted, which leads to omitted inertial loads. The U_0^0 matrix of SOL 101 is not computed in SOL 144, which can lead to errors in stresses for elements near the omitted DOFs having significant masses and/or external loads. The nature of the error is that gross effects are calculated correctly, such as the rigid body acceleration of the aircraft, but the masses at o-set DOFs are re-lumped to a-set DOFs, thereby changing the local load distribution.

In summary, omitted DOFs should be avoided when possible in SOL 144, and used with care when omitted DOFs are essential.

User parameter PARAM, USETPRT,0 will print the set membership of the model, and will identify any DOFs in the omitted set. Remove the ASETi-type and OMITi type entries which caused an o-set in the residual structure.

Upward Compatibility

Any aeroelastic splines that have non-zero terms for their omitted degrees-of-freeedom will cause a fatal error in static aeroelastic analysis (SOL 144).

Omitted DOFs can occur from several user input entries. The most common is through use of ASETi entries and their several equivalents (BSETi, CSETi, etc.). All free DOFs not mentioned on ASETi-type entries are place in the omitted set (o-set) by default. Another input type is the OMITi class, which is a complement of the ASETi entry. Still another input type is superelements, where interior points (free non-boundary points) in superelements are automatically placed in the o-set. All of these entry types will now cause fatal errors when spline DOFs are placed directly on omitted DOFs, or when they are placed on rigid elements or MPCs which include omitted DOFs. If the omitted DOFs are not attached to splines directly or indirectly through MPC equations they will not cause this fatal error exit.

Omitted DOFs are not commonly used on models prepared for SOL 144 analysis. They typically occur on models first developed for dynamic analysis or superelement analysis, then converted to SOL 144. The easiest course here is to remove all ASETi-type and OMITi entries from the input file, for non-superelement models. This precludes an o-set from being formed. For superelement models, move any superelements connected to splines into the residual structure. New methods of solution are available, which allow solution of very large SOL 144 models at costs and computer resource requirements similar to SOL 101 analysis. See Section 4.5 for more details.

4.9 Coupled Fluid-Structure Models with Interface DOFs in Superelements

Introduction

Prior to Version 70.5 the fluid-structure analysis capability required that all fluid elements and all points (structure and fluid) at the fluid-structure interface be in the residual structure. The structural points may now be assigned to upstream superelements. This increases the practical size of models that use this capability. The fluid-structure analysis capability that was available prior to Version 70 is described in the *MSC/NASTRAN Reference Manual*. The material in that reference with regard to superelement usage is now obsolete. The rules for superelement usage in Version 70.5 are described in this section.

All fluid points may be in the residual structure, as before. Structural elements connected to the fluid may be in superelements and the residual structure. Some or all fluid points may instead be in a new special superelement reserved for fluid elements. The new superelement capability is selected with a user parameter (FLUIDSE). The new capability is supported in SOLs 103, 107 through 112, and 200.

Theory

Fluid and structural DOFs are coupled with the splining matrix A_{gg}, also known as the area factor matrix. This splining matrix is calculated for the whole model. Then the spline matrix is made into quasi-load vectors, partitioned to the superelements, reduced, then processed with the modules used to assemble load vectors. This is similar to techniques used for aeroelastic splines, as discussed in Section 4.6.

Matrices derived for the total structure are given the superscript “x”. A^x_{gg} is nominally a square matrix but the only non-zero terms are in an off-diagonal block which couples the two types of points. Let the superscript “s” symbolize the structural grid points in the model, and “f” the fluid points. A^x_{gg} is of the form

$$A^x_{gg} = \begin{bmatrix} O^{ss} & A^{sf} \\ O^{fs} & O^{ff} \end{bmatrix} \tag{4-1}$$

The structure and fluid variables undergo many types of reductions from the physical set u_g to the solution set u_h. These reductions may include constraint elimination and/or superelement and/or modal reduction, symbolized by the reduction matrix R_{gh}, shown in its partitions.

$$\begin{bmatrix} u^s_g \\ u^f_g \end{bmatrix} = \begin{bmatrix} R^s_{gh} & 0 \\ 0 & R^f_{gh} \end{bmatrix} \begin{bmatrix} u^s_h \\ u^f_h \end{bmatrix} \tag{4-2}$$

When the reduced basis h-set variables are substituted for the g-set variables an equation similar to Eq. 2 is produced, except that the reduction process eliminates the constraint forces. The term of most interest in the reduced basis is the coupling term,

$$A^{sf}_{hh} = R^{sT}_{gh} * \left[A^{sf}_{gg} * R^f_{gh} \right] \tag{4-3}$$

For the new fluid superelement option the term in brackets at the right is evaluated first. It represents the reduction of the columns of the matrix (fluid degrees-of-freeedom). A restriction of the new method is that all fluid elements must be interior to the fluid superelement. The fluid superelement will be reduced before all other superelements so that its component eigenvectors are available for reducing the number of columns of the splining matrix.

$A_{gg}^{sf} R_{gh}^{f}$ is stored in the data base and fetched for every other superelement and the residual structure. The rows of $A_{gg}^{sf} R_{gh}^{f}$ associated with interior points of a structural superelement are assembled and reduced by the same techniques used for load vectors to produce A_{aa} in the residual structure. This matrix and its transpose are merged to d-set size.

The cost savings of this approach come from several sources:

- The fluid data passed through superelement boundaries is for N_q columns of modal data rather than the typically larger N_f columns of fluid data. The number of columns which need to be reduced with the load reduction technique for splines is therefore also reduced.
- The residual structure need not include most of the fluid points nor the structural points to which they couple, resulting in a smaller residual structure compared to cases where all such points must be in the residual structure.

At present only one special fluid superelement is allowed. For models where it is necessary to have several separate fluid compartments joined only by structural elements several disjoint fluid components may be placed in the same special superelement. Frequency-dependent fluid elements, however, must be in the residual structure.

Input

If the fluid points are in the residual structure, no other special input is required. Structural grid points may be placed in upstream superelements with the SEID field of GRID Bulk Data entries and/or the SESET Bulk Data entry.

For the new special superelement option the fluid superelement is identified with PARAM, FLUIDSE, [seidf], where seidf is an integer. All fluid elements are indirectly assigned to the special superelement by assigning some of their fluid points to that superelement. The least laborious method for defining the interior and exterior DOFs of the special superelement is to place the value [seidf] in the SEID field of all fluid GRID entries, then list any fluid points to be in the boundary on SESET, 0 entries, which over-rule the GRID entries. These boundary points are automatically connected to the special superelement because they are attached to fluid elements that are in turn connected to interior points of the superelement. Specifying the superelement as of the fluid type results in it being processed first. This order is a requirement because the special superelement forms the reduced column spline matrix whose rows must be reduced by the other superelements.

The most efficient method of analyzing the fluid, and at the present the least laborious from an input preparation point of view, is by component mode synthesis of the special superelement. q-set points are assigned to this superelement. In order for these DOFs to be identified as generalized coordinates assigned to the fluid, they are assigned to fluid grid points rather than the conventional practice of assigning them to scalar points. The generalized coordinates are the only exterior points of the special superelement unless enforced pressures are used, or there are other reasons to require some fluid points to be in the residual structure. Any points which must be accessible to the residual structure such as DOFs for enforced pressure, DOFs to be used for sensitivity calculations, and DOFs connected to frequency-dependent fluid elements are placed in the residual structure. Component mode synthesis is requested for the fluid superelement.

Output

Output is conventional.

Limitations

- Only the older grid list superelements available prior to Version 69 are supported for fluid structure analysis. The newer part superelements introduced in Version 69 may not be used when fluid grid points are present.
- Frequency dependent fluid elements must be in the residual. They cannot be in the special fluid superelement.

Example 1: Assigning Fluid Points to Upstream Superelement (TPL: fsp11a.dat)

Demonstration file fsp11a.dat. This is a modification of one of the original verification problems for the fluid-structure capability. A listing of the entire input file is given in the next section. There is one fluid element. The fluid points are all placed in fluid superelement 1 by the GRID-SEID method. The only structural element, a CQUAD4, is placed in superelement 2. Although the GRID entry used for fluid generalized coordinates does not have the “thru” convenience feature of the SPOINT entry, many generalized coordinates may be generated with a few entries using the replicator feature. The only data items which need to be defined for fluid points used as generalized coordinates are the grid point id (gid) and a value of –1 in the CD field, which specifies that the grid point is of the fluid type. All other data may be left blank. The structural grid points are placed in s.e. 2, which is given component mode synthesis (CMS).

```
$ FILE FSP11A  VERIFICATION PROBLEM FOR FLUID–STRUCTURE IN SUPERELEMENTS
ID KIZ,ACOUSTIC
TIME 100
SOL 111 $ MODAL FREQ. RESP.
CEND
TITLE= RECT. VOLUME (1X1X1)– 3 SIDES FIXED, MODAL REDUCTION ON BOTH
SUBTITLE= FLUID–STRUCTURE INTERACTION, 4TH SIDE ON BEAM ON ELAST. FOUND.
LABEL = VERIF. FOR SUPERELEMENT CAPABILITY, STRUCTURE IN SE. 2 FSP11A
ECHO = BOTH
DLOAD=1100
FREQ=200
DISP(SORT2)=ALL
SPC =1313
METHOD(STRUCT)=30
METHOD(FLUID)=20
SDAMP=1000 $
SDAMP(FLUID) = 1000
LOADSET=2000
BEGIN BULK
$ CHANGES FOR –A MODEL
$                                               CD
PARAM,FLUIDSE,1 $ Fluid superelement
GRID,   2001,   ,       ,       ,       ,       –1 $ G.C.S
=,      *1,     == $
=6 $ GEN. COORDS MUST BE ON FLUID POINTS, IN S.E. 0.
SEQSET1, 1,    0,      2001    THRU    2008 $ Q–SET FOR S.E. 1
SESET,  1,     100     THRU    224 $ ALL FLUID POINTS TO S.E.1
SESET,  2,     41      THRU    65 $ ALL STRUCTURE POINTS TO S.E. 2
SPOINT, 1001   THRU    1008 $ GC.S FOR THE STRUCTURAL SUPERELEMENT
SEQSET1, 2,    0,      1001 THRU 1008
.
.
.
ENDDATA
```

Example 2: Apply Enforced Pressure to Fluid Superelement (TPL: fsp11j.dat)

This is a variation of the first problem. The fluid is represented by a hexagonal element, with one face connected to the structure. The other fluid points are coupled to one point with MPC equations so that all will be at the same pressure. One of these fluid points (120) is moved to the residual structure, and an enforced pressure (“displacement”) is applied to it. It is necessary to attach a large mass to this DOF. At present, using a scalar element such as a CMASS2 element connected to a fluid point causes a user fatal error. This error is avoided by adding another structural grid point (1200) and tying it to the fluid point with an MPC equation. The large mass is applied to the grid point, and the input point (SLOAD) is multiplied by the same factor. The input function is converted from acceleration units of measure to displacement units by referencing a table which describes the function $-((2*\pi)**2)*f**2$, where "*f*" is the excitation frequency. The GRID-SEID method is used to put fluid points in s.e. 1, with an SESET entry used to move the enforced motion point back to s.e. 0. The entire input file is listed in Listing 4-1. Both of the demonstration models described here are a variation of this model. File fsp11jx.dat, delivered in the test problem library but not shown here, performs the same analysis without superelements, using the old technology. Both $-11j*$ test files produce identical results.

The alter package rflagb.f705 provides a Lagrange multiplier method technique (LMT) for applying enforced pressure while referencing a simpler table which produces a constant function. The only extra input which is required is to list the variables where pressure is to be enforced on USET, U1 entries. Any SLOADs applied to these points are then applied pressures rather than the applied second derivative of pressure used in the large mass approximation. The demonstration problems described below include an example of an enforced pressure problem using the LMT technique.

Listing 4-1.

```
$ FILE FSP11J  VERIFICATION PROBLEM FOR FLUID SUPERELEMENTS, V70.5
ID KIZ,ACOUSTIC $ MODDED BY MAG, FEBRUARY 20, 1998
SOL 111 $ MODAL FREQ. RESP.
CEND
TITLE= RECT. VOLUME ( 1X1X1)– 3 SIDES FIXED,MODAL REDUCTION ON BOTH
SUBTITLE= FLUID–STRUCTURE INTERACTION,4TH SIDE ON BEAM ON ELAST. FOUND.
$LABEL= RESIDUAL ONLY– DYNAMIC LOADING–MODAL FREQ. RESP.
LABEL = VERIF. FOR FLUID S.E.S, ENF MOTION(PRESSURE)   FSP11J
ECHO = BOTH
DLOAD=1100
FREQ=200
DISP(SORT2)=ALL
SET 120 = 120,1200 $ THE LOADED POINTS
ACCE = 120
OLOAD(SORT2) = 120
SPC = 1313
MPC = 1 $ USED TO AVOID ERROR, AND COUPLE FLUID FACE FOR ENF. MOTION
METHOD(STRUCT)=30
METHOD(FLUID)=20
$SDAMP=1000 $
$SDAMP(FLUID) = 1000
LOADSET=2000
```

Listing 4-1. (Cont.)

```
BEGIN BULK
$ USE THE LARGE MASS METHOD TO ENFORCE PRESSURE (DISPLACEMENT)
CMASS2, 1200,   1.E6    1200,   1 $ THE LARGE MASS
GRID,   1200,   ,       0.      0.,     1. $ COINCIDENT WITH FLUID
$ POINT 120.  LARGE MASS PLACED ON STRUCTURE POINT
MPC,    1,      1200,   1,      -1.,    120,    0       1. $ AVOID ERROR WHICH
$ PREVENTS USING SCALAR ELEMENTS SUCH AS CMASS2 ON FLUID POINTS
SPCOFF, 1200,   1 $ TURN OFF AUTOSPC ON THIS DOF
MPC,    1,      124,    0,      -1.     120,    0,      1.
MPC,    1,      220,    0,      -1.     120,    0,      1.
MPC,    1,      224,    0,      -1.     120,    0,      1.
$ THE POINTS ABOVE ARE ALL AWAY FROM THE FLUID-STRUCTURE INTERFACE.
$ THEY ARE TIED TOGETHER SO THAT THE SAME PRESSURE ACTS AT EACH.
$ FLUID POINTS FOLLOW. GIVE ALL AN SEID OF 1
$ GRIDS 200 THRU 204 AND 100 THRU 104 ARE CONTACT GRIDS.
$ THE OTHERS ARE ON THE OPPOSITE FACE        -1 MEANS FLUID SEID
GRID    200             0.0     0.2     0.0     -1              1
GRID    204             1.0     0.2     .0      -1              1
GRID    220             0.0     0.2     1.0     -1              1
GRID    224             1.0     0.2     1.0     -1              1
GRID    100             0.0     0.0     0.0     -1              1
GRID    104             1.0     0.0     .0      -1              1
GRID    120             0.0     0.0     1.0     -1              1
GRID    124             1.0     0.0     1.0     -1              1
$                                               CD
GRID,   2001,   ,       ,       ,       ,       -1 $ G.C.S FOR FLUID
=,       *1,     == $
=6 $ GEN. COORDS MUST BE ON FLUID POINTS.
SEQSET1, 1,     0,      2001    THRU    2008 $ Q-SET FOR S.E. 1, FLUID G.C.S
SESET,  0,      120     $ ALL FLUID POINTS TO S.E.1 EXCEPT
$ FLUID GRID 120, MOVE IT BACK TO S.E. 0.  USED FOR ENF. PRESSURE INPUT
.
.
.
ENDDATA
```

4.10 Miscellaneous Enhancements

This section describes the following miscellaneous enhancements in MSC/NASTRAN Version 70.5:

Postprocessing

1. PARAM,PATPLUS,YES may be used to request .op2 file output of GPFORCE, GPSTRESS when PARAM,POST,0 is requested.

2. PARAM,POST,-1 is now available in SOL 4.

3. MPCForce output has been added to PARAM,POST,-1 (.op2 file) and PARAM,POST,0 (.xdb file) for MSC/PATRAN.

4. Random response output has been added to PARAM,POST,-1 (.op2 file) and PARAM,POST,0 (.xdb file) for MSC/PATRAN.

Superelement Analysis

1. Partitioned external superelements (SEBULK Bulk Data entry with METHOD=EXTERNAL), first introduced in Version 69.1 for SOLs 101 and 103, may now be assembled in SOLs 101 through 159. Data recovery for partitioned external superelements is extended to SOLs 107 through 112.

2. CSUPER-type external superelements may now be processed the same way as partitioned external superelements using parameters, EXTDR, EXTDROUT, EXTDRUNT, EXTOUT, EXTUNIT. See Section 6 of the *MSC/NASTRAN Quick Reference Guide*, Version 70.5 for more details.

3. Virtual mass may be included at the g-set level by specifying PARAM,VMOPT,1 (Default is at the a-set level). The benefit is that Grid Point Weight Generator Output and the component modes include the effects of virtual mass. However, this also increases the CPU and storage cost because the mass matrix is very dense.

4. For component modes, the INREL module has been replaced by DMAP which performs better orthogonalization and filtering.

Data Recovery

1. Strains at the grid points may be requested with the new GPSTRAIN Case Control command. It is similar to the GPSTRESS command.

2. RESULTANT and MAXIMUM output for SPCForce, MPCForce, and displacement results is now available in SOL 106 and can be controlled with PARAMs PRTRESLT and PRTMAXIM.

3. GPFORCE output now includes MPCForces.

Nonlinear Analysis

Damping is now included in the calculation of nonlinear complex modes (see PARAM,NMLOOP and PARAM,UNSYMF,YES) in SOL 106.

Access to .XDB Database

The DBC modules convention for formulating access keys for the .XDB database has been modified. This modification allows you to process more qualifiers for a given number of grids or elements than previous versions. Qualifiers are output time steps, number of frequencies, etc. In the previous versions, the maximum number of qualifiers was 9,999. For example, you could not process a transient analysis output with more than 9,999 time steps. This limit constraint was hard-wired in the DBC modules. On the other hand, the maximum number of grids or elements that could be processed was limited as the access key stored in a 32-bit word. Access key is a function of number of grids and its qualifiers. From Version 70.5 onwards, the number of grids and qualifiers you can process is dynamic based on problem characteristics. For example, you can now process larger numbers of grids with less than 9,999 qualifiers or you can process less numbers of grids with more than 9,999 qualifiers. The access key is still limited to a 32-bit word.

For applications in the field that do not support the new dynamic convention, a system cell is provided to specify the use of the old convention. This system cell name is DBFACT and the number is 274.

A system cell value of one will cause the old convention to be used. This value may be specified on the NASTRAN statement or in an RC file by following statement:

NASTRAN DBCFACT=1 or NASTRAN SYSTEM(274)=1

Bulk Data Metrics

In Version 70.5, a summary of your bulk data file is provided in the .F04 file. This summary is a count of each parent bulk data entry and a list of all the parameters you have specified. This summary does not include the value of the parameters, as the parameter values can be set in several different ways, such as case control command, DMAP, etc. This information is in addition to the information provided by the sequencer module in the .F06 file. The sequencer module provides information on element and grids only, which is a subset of the new summary information provided in the .F04 file. This summary information can be used to develop execution control metrics based upon model characteristics, or guide application engineers when speaking with clients concerning interactions of data structures and execution characteristics.

Inclusion of MPC Forces to Grid Point Force Balance

The Multi-Point Constraint Forces and rigid elements are now included in the Grid Point Force Balance summations to better represent an equilibrium at a joint. The following table summarizes those effects that are considered and those effects that are ignored in the calculation of grid point forces in the global coordinate system:

Contributions Included	Contributions Ignored
Applied Loads	GENEL Forces
SPC Forces	DMIG and DMI Forces
Element Elastic Forces	Forces Due to Differential or Nonlinear Stiffness
Thermal Loads	Boundary Loads from Upstream Superelements
MPC and Rigid Element Forces	

NORM Module Enhancement

The NORM module normalizes an input matrix and also sets a real single precision output parameter XNORM equal to the absolute value of the term of largest magnitude of the input matrix. Prior to Version 70.5 this parameter was stored internally in single precision. On 32 bit (short word) operating systems most matrices are stored in double precision. Details differ between operating systems, but the exponent range of double precision variables is typically more than twice that of single precision variables. The consequence was that matrices with large but acceptable terms (1.E60, for example) caused an abnormal exit due to overflow when attempting to normalize the matrix.

The maximum term is now stored internally in a double precision variable and output in two formats. This value is always output in a new, fifth parameter, XNORMRD, a real double precision variable. When it is smaller than the single precision limit it is also output in the old third parameter XNORM. When it is larger than the limit a User Warning Message is printed stating that the number is too large to fit the single precision format, that XNORM is returned with a zero value, and that the true value is output in XNORMRD. On 64 bit (long word) operating systems the action is the same, except that the majority of the matrices are in single precision, so that it is unlikely that the XNORM parameter will ever be set to a zero because the largest term is too large to fit into its format.

This enhancement is useful in iterative solutions when testing a trial solution to find if the solution is diverging. The NORM module should now be able to process any matrix that other modules produce. The XNORMRD parameter can be used to determine if the iterations should terminate or take another branch more likely to converge.

It was found in testing that it is not possible to cause abnormal exits due to overflow in NORM when real matrices were input. Complex matrices can cause overflow exits, but the range where this occurs is orders of magnitude greater in V70.5 than in Version 70.

Linear Perturbation Analysis for Geometric Nonlinear Analysis

The parameter NMLOOP allows you to restart from a nonlinear static analysis (SOL 106) into linear dynamic solution sequences (SOL 103, 107 to 112). The restart in dynamic solutions is a linear perturbation about the nonlinear state from SOL 106. This restart capability is limited and should be used with caution for the following only:

1. Restart from SOL 106 to SOL 103 to obtain current natural frequencies and mode shapes. The current natural frequencies and mode shapes may also be obtained in SOL 106.

2. Restart from geometric nonlinear analysis (SOL 106) to dynamic analyses (SOL 107 to 112). For this restart the displacement element forces and stresses are incremental (perturbed) values.

Limitations

1. You may perform linear dynamic perturbation analysis (SOLs 107 to 112) only for geometric nonlinear analysis.

2. Linear perturbation analysis with nonlinear material does not produce correct element forces and stresses. However, the incremental displacements are correct. See the error report for CSR 31145 for more details. A warning message is issued as follows:

   > “PARAM,NMLOOP requests a restart from a nonlinear solution sequence and will produce incremental displacements, forces, and stresses relative to the nonlinear solution at the loop id specified for NMLOOP. Also, in the case of elements with nonlinear material properties, the element stress and force data recovery will be based on the material properties specified on the MAT1 Bulk Data entry and not MATS1.”

Recommendations

For linear material and small deformation, we recommend you to stay in the linear solution sequences and use STATSUB to include initial stress and follower force effects.

RBE3 Element Error Checking

The error checking capability of the RBE3 element has been improved in Version 70.5. Elements with illegal geometry-connectivity combinations are detected more reliably and provide more repeatable results across all computer types. A new warning message is produced when an element is approaching numerical instability. See the error report for CSR 25891 for more details.

Ease of use for DMAP Alters

Prior to Version 70.5, if a data block is specified on a TYPE statement in a subDMAP and it was stored on the scratch DBset then it is automatically deleted at the end of the subDMAP. If you do not want the data block to be automatically deleted, you must specify the data block on the SUBDMAP statement. This makes DMAP altering very inconvenient. In Version 70.5, and if the data block is used as input only in the subDMAP, then it will not be automatically deleted. However, if the data block is output, it will still be deleted.

CHAPTER 5

UPWARD COMPATIBILITY

MSC/NASTRAN Version 70.5 has been enhanced in several areas based on the feedback and requests of our customers. Some of these enhancements may change answers as the physical behavior is modeled more accurately, input file format, and the data flow of solution sequences. This chapter discusses the following topics:

- Answer Changes
- Database Migration
- Upward Compatibility of Input Files
- Summary of DMAP Modules Changes from Version 69 to Version 70
- Summary of DMAP Modules Changes from Version 70 to Version 70.5.

5.1 Answer Changes

The enhancements in MSC/NASTRAN Version 70.5 result in improved answers compared to the prior version. Answers may change slightly due to the following general improvements:

1. The interpolation of non-uniform temperature over the element is more accurate in Version 70.5 than prior versions. See Section 2.6 for more details.
2. For static aeroelastic models, the program will exit with a fatal message if splines are attached to omitted degrees-of-freedom (o-set). It should be noted that the interior degrees-of-freedom for superelements are removed from the analysis set and are part of the o-set. See Section 4.6 for more details.
3. Due to the replacement of the INREL module the displacement, velocity, and acceleration output will be different at the generalized coordinates which represent component modes. However, the physical solution is the same.

4. Grid point force balance (GPFORCE) output now includes the effect of MPCFORCE output.

5.2 Database Migration

You cannot restart from Version 70 database into Version 70.5. This is because the migration utilities have not been updated.

Input files from previous versions of MSC/NASTRAN may be used to create Version 70.5 databases. MSC/NASTRAN input files are upward compatible except for the changes discussed in Section 5.3.

5.3 Upward Compatibility of Input Files

- Specification of initial temperature using TEMP(INIT) subcase control command is now applicable to all elements. Use of TEMP(INIT) is easier than the use of the TREF field in the MATi Bulk Data entries. Use of TEMP(INIT) overrides the use of TREF in MATi entries.

- For static aeroelastic models, the program will exit with a fatal message if splines are attached to omitted degrees-of-freedom (o-set). It should be noted that the interior degrees-of-freedom for superelement are removed from the analysis set and are part of the o-set. See Section 4.6 for more details.

- The license-controlled limit of problem size (GRID limit) no longer depends on the VUGRID size for models that contain p-elements. See Section 1.3 for more details.

- The access key formulation used for the .DBC modules (.XDB database) is now dynamic and takes into account the problem characteristics to allow for more flexibility in data recovery. See Section 4.10 for more details.

- The line numbers for all the MSC/NASTRAN solution sequences have been changed in Version 70.5. All DMAP alters should be converted to match the new Version 70.5 format (see Section 5.5 for a summary of DMAP module changes in Version 70.5). MSC recommends the use of the string-based DMAP alter capability (first introduced in Version 68.2) to convert your DMAP alters or sequences.

5.4 Summary of DMAP Module Changes from Version 69 to Version 70

This section was not included in the printed Version 70 *MSC/NASTRAN Release Guide*.

This section summarizes DMAP module changes from Version 69 and 69.1 to Version 70. This information is intended to help you convert your Version 69 and 69.1 DMAP alters or sequences to run in Version 70. The following modules have changed in Version 70. (SEP1X changed in V69.1.)

DSAD DSAH DSAL DSTAP2 DSVG1P GPSP MDCASE MODTRL SEP1X

A structured solution sequence excerpt shows how the module is actually invoked in the structured solution sequences, and the additions or deletions are shown in lowercase in the Version 69 or 70 excerpt, respectively.

DSAD Module

Add 1 data block, BGPDTS, to the end of the input list. Add 4 data blocks: CASDSN, DRDUG, DRDUTB, and CASADJ, to the end of the output data block list. These data blocks are needed for adjoint sensitivity analysis.

Version 69 in SubDMAP DESCON

```
DSAD      RSP1CT,R1TAB,RESP12,OBJTAB,CONTAB,BLAMAX,LAMASX,
          DIVTAB,AUXTAB,STBTAB,FLUTAB,OUGVS1,OESS1,OSTRS1,
          OEF1AA,EFITS,ES1CS,EA1CS,OQGS1,
          DSCREN,XINIT,COORDN,OLX,FRQRSP,
          CASEDS,CASERS,UGX,OPTPRM,PROPI/
          R1VAL,R2VAL,RSP2R,R2VALR,CVAL,CVALR,OBJTBR,CNTABR,
          R1TABR,R1VALR,DRSTBL,FRQRPR,UG1,R1MAPR,R2MAPR/
          WGTS/VOLS/S,N,OBJVAL/S,N,NR1OFFST/S,N,NR2OFFST/
          S,N,NCNOFFST/APP/DMRESD/SEID/DESITER/EIGNFREQ $
```

Version 70 in SubDMAP DESCON

```
DSAD      RSP1CT,R1TAB,RESP12,OBJTAB,CONTAB,BLAMAX,LAMASX,
          DIVTAB,AUXTAB,STBTAB,FLUTAB,OUGVS1,OESS1,OSTRS1,
          OEF1AA,EFITS,ES1CS,EA1CS,OQGS1,
          DSCREN,XINIT,COORDN,OLX,FRQRSP,
          CASEDS,CASERS,UGX,OPTPRM,PROPI,bgpdts/
          R1VAL,R2VAL,RSP2R,R2VALR,CVAL,CVALR,OBJTBR,CNTABR,
          R1TABR,R1VALR,DRSTBL,FRQRPR,UG1,R1MAPR,R2MAPR,
          casdsn,drdug,drdutb,casadj/
          WGTS/VOLS/S,N,OBJVAL/S,N,NR1OFFST/S,N,NR2OFFST/
          S,N,NCNOFFST/APP/DMRESD/SEID/DESITER/EIGNFREQ/
          s,n,adjflg/pexist/mbcflg $
```

DSAH Module

Remove data blocks UGSN and QGSN, and their subsequent commas, from the input list. Remove data blocks UGDSN and QGDSN, and their subsequent commas, from the output list. Add input parameters ADJFLG and SEID to end of the parameter list.

Structured solution sequence excerpts:

Version 69 in SubDMAP RESPSEN

```
DSAH      DRSTBL,R1TABR,CASEDS,TABECN,BLAMAX,LAMASX,
          ugsn,qgsn,OLB,DIVTAB,FRQRPR,VIEWTBDS,CASERS,
          BUGX,PHGX/
          BUDG,PHDG,CASEDSF,LBTAB,BDIAG,LFTAB,
          COGRID,COELEM,ugdsn,qgdsn,DSEDV,OINTDSF,PELSDSF/
          APP/DMRESD/NDVTOT $
```

Version 70 SubDMAP RESPSEN

```
DSAH      DRSTBL,R1TABR,CASDSN,TABECN,BLAMAX,LAMASX,
          OLB,DIVTAB,FRQRPR,VIEWTBDS,CASERS,csnmb,
          BUGX,PHGX/
          BUDG,PHDG,CASEDSF,LBTAB,BDIAG,LFTAB,
          COGRID,COELEM,DSEDV,OINTDSF,PELSDSF/
          APP/DMRESD/NDVTOT/adjflg/seid $
```

DSAL Module

Add two data blocks, DRDUTB and ADELU, to the end of the input list. Add input parameter ADJFLG to the end of the parameter list. These data blocks and parameter are needed for adjoint sensitivity analysis.

Structured solution sequence excerpts:

Version 69 in SubDMAP RESPSEN

```
DSAL     DRSTBL,DELWS,DELVS,DELB1,DELF1,COGRID,COELEM,
         OUGVDSN,OESDSN,OSTRDS1,OEFDSN,EFITDS,ES1CDS,EA1CDS,
         R1VALR,TABDEQ,OLB,DSDIV,DELX,DELS,DELFL,FRQRPR,
         DELBSH/
         DSCM/NDVTOT/DELTAB/EIGNFREQ $
```

Version 70 in SubDMAP RESPSEN

```
DSAL     DRSTBL,DELWS,DELVS,DELB1,DELF1,COGRID,COELEM,
         OUGVDSN,OESDSN,OSTRDS1,OEFDSN,EFITDS,ES1CDS,EA1CDS,
         R1VALR,TABDEQ,OLB,DSDIV,DELX,DELS,DELFL,FRQRPR,
         DELBSH,drdutb,adelu/
         DSCM/NDVTOT/DELTAB/EIGNFREQ/adjflg $
```

DSTAP2 Module

Remove the last 4 data blocks, DRSTBLG, FOL1V, TOL2V, and FRQRPRG from the end of the input list.

Structured solution sequence excerpts:

Version 69 in SubDMAP EXITOPT

```
DSTAP2   R1TABRG,RSP2RG,drstblg,fol1v,tol2v,frqrprg/DSCMCOL $
```

Version 70 SubDMAP EXITOPT

```
DSTAP2   R1TABRG,RSP2RG/DSCMCOL $
```

DSVG1P Module

Add two data blocks, GPSNTS and ESTDV2B, to the end of the input list. These data blocks are required for optimization with p-elements.

Structured solution sequence excerpts:

Version 69 in SubDMAP PSLGDV

```
DSVG1P     EST,ESTDV2F,BGPDVP,CSTMS,MPTS,DIT,DEQATN,DEQIND,UGX,,
           DSPT1/
           EGKP,/
           COUPMASS/K6ROT/ALTSHAPE/WTMASS/NOPSLG/1 $
```

Version 70 SubDMAP PSLGDV

```
DSVG1P     EST,ESTDV2F,BGPDVP,CSTMS,MPTS,DIT,DEQATN,DEQIND,UGX,,
           DSPT1,gpsnts,estdv2b/
           EGKP,/
           COUPMASS/K6ROT/ALTSHAPE/WTMASS/NOPSLG/1 $
```

GPSP Module

For correct processing of MPCs and rigid elements during PARAM AUTOSPC operations specify KMM in the second position of the input data block list and the string ’KMM’ in the tenth parameter position.

Structured solution sequence excerpts:

Version 69 in SubDMAP SEKR

```
GPSP       KNN,  ,USET0,SILS,GPLS,,GEOM4S,EQEXINS/USET,/S,N,NOSSET/
           AUTOSPC/PRGPSTX/SPCGEN/EPZERO/ACON/
           S,N,SING/EPPRT/S,N,NOSET/S,N,NGERR $
```

Version 70 SubDMAP SEKR0

```
IF ( NOMSET>=0 ) UPARTN   USET0,KGG/KMM,,,/'G'/ 'M'/'N' $

GPSP       KNN,kmm,USET0,SILS,GPLS,,GEOM4S,EQEXINS/USET,/S,N,NOSSET/
           AUTOSPC/PRGPSTX/SPCGEN/EPZERO/ACON/
           S,N,SING/EPPRT/S,N,NOSET/S,N,NGERR/'kmm' $
```

MDCASE Module

Add 1 data block, EDOM, to the end of the input data block list and 1 output parameter, WVFLG to the end of the parameter list. This data block and parameter are needed for adjoint sensitivity analysis.

Structured solution sequence excerpts:

Version 69 in SubDMAP DESOPT

```
MDCASE     CASEXX/
           CASESTAT,CASEMODE,CASEBUCK, CASEFREQ,
           CASEMTRN,CASESAER,CASEDVRG,CASEFLUT ,,,,,
           CASESADV,CASESNMB,CASEXN/
           S,N,STATCC/S,N,MODECC/
           S,N,BUCKCC/S,N,DFRQCC/S,N,MFRQCC/
           S,N,MTRNCC/S,N,SAERCC/S,N,DVRGCC/S,N,FLUTCC/
           ////S,N,DESOBJ/S,N,DESGLB/S,N,OBJSID/TRUE $
```

Version 70 SubDMAP DESOPT

```
MDCASE     CASEXX,edom/
           CASESTAT,CASEMODE,CASEBUCK,CASEFREQ, CASEMTRN,CASESAER,
           CASEDVRG,CASEFLUT,,,,,CASESADV, CASESNMB,CASEXN/
           S,N,STATCC/S,N,MODECC/S,N,BUCKCC/S,N,DFRQCC/S,N,MFRQCC/
           S,N,MTRNCC/S,N,SAERCC/S,N,DVRGCC/S,N,FLUTCC/
           ////S,N,DESOBJ/S,N,DESGLB/S,N,OBJSID/TRUE/s,n,wvflg $
```

MODTRL Module

The fourth parameter, which changes type and/or precision, has been disabled.

1. In order to change the precision of a matrix, use ADD5. If the new precision does not match the machine precision specify PUTSYS (newprecision,55) before ADD5. For example, on a double-word machine:

   a. Single to double

      Version 69:

      ```
      MODTRL    SINGLE/////2 or 4 $
      ADD       SINGLE,/DOUBLE $
      ```

      Version 70:

      ```
      ADD5      SINGLE,,,,/DOUBLE $
      ```

   b. Double to single

      Version 69:

      ```
      MODTRL    DOUBLE/////1 or 3 $
      ADD       DOUBLE,/SINGLE $
      ```

      Version 70:

      ```
      PUTSYS(1,55) $
      ADD5     DOUBLE,,,,/SINGLE $
      PUTSYS(2,55) $
      ```

      or

      ```
      MPYAD    DOUBLE,,DOUBLE/SINGLE///1 $
      ```

2. In order to change the type (complex or real) of a matrix, use ADD to convert real to complex and MATMOD(34) for complex to real. For example,

   a. Real to complex

      Version 69:

      ```
      MODTRL    REAL/////3 or 4 $
      ADD       REAL,/CMPL$
      ```

      Version 70:

      ```
      ADD       REAL,/CMPLX//(0.,1.) $
      ```

b. Complex to real

Version 69:

```
MODTRL    CMPLX/////1 or 2 $
ADD       CMPLX,/REAL $
```

Version 70:

```
MATMOD    CMPLX,,,,,/REAL,/34 $
```

SEP1X Module

In Version 69, SEP1X was executed only if partitioned superelements were present. In Versions 69.1 and 70, SEP1X is now executed for all models, regardless of the presence of superelements. SEP1X was enhanced to refine the grid point connections of RSSCON elements. Three parameters, SEP1XOVR, SEBULK, and TOLRSC, are added to control the operation of the module. SEP1XOVR and TOLRSC are described in the *MSC/NASTRAN Quick Reference Guide* Version 70. SEBULK is a logical parameter which is TRUE if partitioned superelements are present.

Structured solution sequence excerpts:

Version 69 in SubDMAP PHASE0

```
IF ( SEBULK
   ) SEP1X     SELIST,GEOM1F,GEOM2F,GEOM4F,SETREE,SGPDTF/
               EMAP,SGPDT,SCSTM/
               S,N,NOSE/CONFAC/'PEID'//S,N,PARTRS/NQSET/'XEID' $
```

Version 70 SubDMAP PHASE0

```
SEP1X      SELIST,GEOM1F,GEOM2F,GEOM4F,SETREE,SGPDTF/
           EMAP,SGPDT,SCSTM/
           S,N,NOSE/CONFAC/'PEID'//S,N,PARTRS/NQSET/'XEID'/
           sep1xovr//sebulk/tolrsc $
```

5.5 Summary Of DMAP Module Changes from Version 70 to Version 70.5

This section summarizes DMAP module changes from Version 70 to Version 70.5 which could affect your DMAP alters and solution sequences. This information is intended to help you convert your Version 70 DMAP alters and solution sequences to run in Version 70.5.

The following modules have changed in Version 70.5.

ADG	ADR	APD	ASDR	ASG	CEAD	DOM10	DOPR5
DOPR6	DSAD	DSPRM	GI	GP3	GPFDR	IFP	MODEPT
MODTRK	NLITER	NLTRD2	NORM	RANDOM	READ	REIGL	SDP
SEDRDR	WEIGHT						

The following modules are new in Version 70.5.

DSVGP4 DSVGP5 MAKETR MKSPLINE

DMAP Module Changes

This section describes how to convert DMAP module statements from Version 70 to 70.5. A structured solution sequence excerpt shows how the module is actually invoked in the structured solution sequences and the additions or deletions are shown in lowercase in the Version 70 or 70.5 excerpt, respectively. For the purpose of clarity or simplification, the excerpt may be slightly modified from the actual DMAP in Version 70 or V70.5.

ADG Module

ADG is now executed in subDMAP PHASE0. Most of the inputs are replaced or specified in a different position. Data blocks BGPA and USETA are replaced by AEUSET and AEBGPDT which are defined with DBVIEW statements. Data block

JLIST is no longer required. On the output, data block TR is now formed by the MAKETR module in subDMAP AESTATRS. Data block SRKT occupies the fourth output position and HMKT has been added to the fifth output position.

Version 70 in SubDMAP PFAERO

```
ADG       EDT,aero,CSTMA,ACPT,useta,jlist,bgpa/
          DJX,TRX,XLIST,tr,SRKT/
          AUNITS/NJ/NK/S,N,NX/CIDAP $
```

Version 70.5 in SubDMAP PHASE0

```
dbview aeuset=uset0   where (modltype='aerostrc' and wildcard) $
dbview aebgpdt=bgpdts where (modltype='aerostrc' and wildcard) $
ADG       EDT,CSTMA,aebgpdt,aeuset,aecomp,ACPT/
          DJX,TRX,XLIST,SRKT,hmkt/
          AUNITS/NJ/NK/S,N,NX/CIDAP $
```

Version 70.5 in SubDMAP AESTATRS

```
dbview stbgpdt=bgpdts where (modltype='structur') $
dbview stuset=uset    where (modltype='structur') $
MAKETR    AERO,CSTMA,STBGPDT,STUSET/TR $
```

ADR Module

The last three input data blocks; SPLINE, SILA, and USETA are replaced by AEBGPDT and AEUSET which are defined with DBVIEW statements

Version 70 in SubDMAP SEAERO

```
ADR       AUHF,CASES,QKHL,FOL,SPLINE,SILA,USETA/
          PKF/
          BOV/MACH/APP $
```

Version 70.5 in SubDMAP SEAERO

```
dbview aeuset =uset   where (modltype='aerostrc' and wildcard) $
dbview aebgpdt=bgpdts where (modltype='aerostrc' and wildcard) $
ADR       AUHF,CASES,QKHL,FOL,AEBGPDT,AEUSET/
          PKF/
          BOV/MACH/APP $
```

APD Module

APD is now executed in subDMAP PHASE0. Only two input data blocks remain: EDT and CSTM or CSTMS. The output data blocks: ECTA, BGPA, USETA, EQAERO, GPLA, and SILGA are replaced by BGPDTS and USET0 which are now qualified on MODLTYPE. JLIST is no longer required. The 3rd parameter is removed. The SPLINE table is now formed by a new module MKSPLINE.

Version 70 in SubDMAP PFAERO

```
APD       EDT,EQDYN,ECTS,BGPDTS,SILD,USETD,CSTMS,GPLD/
          EQAERO,ECTA,BGPA,SILA,USETA,SPLINE,AERO,
          ACPT,JLIST,CSTMA,GPLA,SILGA/
          S,N,NK/S,N,NJ/0/S,N,BOV/APP/S,N,CIDAP $
```

Version 70.5 in SubDMAP PHASE0

```
MODLTYPE='AEROSTRC' $
APD       EDT,CSTM/
          ECTS,BGPDTS,USET0,AECOMP,AERO,
          ACPT,CSTMA/
          S,N,NK/S,N,NJ/ S,N,BOV/AERTYP/S,N,CIDAP $
dbview aegrid=bgpdts  where (modltype='aeromesh' and wildcard) $
MKSPLINE EDT,CSTMA,AEGRID,AECOMP/SPLINE $
```

ASDR Module

APD is now executed in subDMAP PHASE0. Only two input data blocks remain: EDT and CSTM or CSTMS. The input data blocks: BGPA, USETA, EQAERO, and SILA are replaced by AEBGPDT and AEUSET which are defined with DBVIEW statements. Data block UX in the second input position is replaced by UXDAT in the fourth position. Data blocks XLIST, EDT, CSTMA, and SPLINE are no longer required.

Version 70 in SubDMAP AESTATRS

```
ASDR      CASEA,ux,FFAJ,PAK,xlist,sila,useta,edt,bgpa,
          eqaero,cstma,ACPT,spline//
          MACHNO/Q $
```

Version 70.5 in SubDMAP AESTATRS

```
dbview aebgpdt=bgpdts where (modltype='aerostrc' and wildcard) $
dbview aeuset=uset    where (modltype='aerostrc' and wildcard) $
ASDR      CASEA,uxdat,FFAJ,acpt,PAK,AEUSET,AEBGPDT,aecomp//
          MACHNO/Q $
```

ASG Module

Data blocks TR, KRZX, DIT, and ERHM are new inputs to ASG.

Version 70 in SubDMAP AESTATRS

```
ASG       CASEA,ZZX,PZ,XLIST,EDT/
          UX,UXDAT $
```

Version 70.5 in SubDMAP AESTATRS

```
ASG       CASEA,ZZX,PZ,XLIST,EDT,tr,krzx,dit,erhm/
          UX,UXDAT $
```

CEAD Module

For complex Lanczos the cross-orthogonality matrix was in the third position as OCEIGS. It is now output in the fifth position as XORTH.

Version 70 in SubDMAP DCEIGRS

```
CEAD      CKDXX,CBDXX,CMDXX,EED,CASES,VDXC,VDXR/
          CPHDX,CLAMA,OCEIGS,LPHDX     /S,N,NCEIGV $
```

Version 70.5 in SubDMAP DCEIGRS

```
CEAD      CKDXX,CBDXX,CMDXX,EED,CASES,VDXC,VDXR/
          CPHDX,CLAMA,OCEIGS,LPHDX,XORTH/S,N,NCEIGV $
```

DOM10 Module

Data block WMIDG is a new input to DOM10.

Version 70 in SubDMAP in DESOPT

```
DOM10     DESTAB,XINIT,XO,CNTABRG,CVALRG,CVALO,
          DVPTAB,PROPI,PROPO,R1TABRG,R1VALRG,R1VALO,
          RSP2RG,R2VALRG,R2VALO,OPTPRMG,OBJTBG,DRSTBLG,
          TOLV,FOLV,FRQRPRG,DBMLIB,BCON0,BCONXI//
          DESCYCLE/DESMAX/OBJIN/OBJOUT/EIGNFREQ $
```

Version 70.5 in SubDMAP in DESOPT

```
DOM10     DESTAB,XINIT,XO,CNTABRG,CVALRG,CVALO,
          DVPTAB,PROPI,PROPO,R1TABRG,R1VALRG,R1VALO,
          RSP2RG,R2VALRG,R2VALO,OPTPRMG,OBJTBG,DRSTBLG,
          TOLV,FOLV,FRQRPRG,DBMLIB,BCON0,BCONXI,wmidg//
          DESCYCLE/DESMAX/OBJIN/OBJOUT/EIGNFREQ $
```

DOPR5 Module

Data blocks GEOM4 and DESGID are new inputs. RGSENS is a new output parameter.

Version 70 in SubDMAP in DESOPT

```
DOPR5     XINIT,EPTTAB,PROPI,DESTAB,DTOS2K,DTOS4K,
          TABDEQ,DELBSH/
          DTOS2,DTOS4,DELBSX/STPSCL $
```

Version 70.5 in SubDMAP in DESOPT

```
DOPR5     XINIT,EPTTAB,PROPI,DESTAB,DTOS2K,DTOS4K,
          TABDEQ,DELBSH,geom4,desgid/
          DTOS2,DTOS4,DELBSX/STPSCL/s,n,rgsens $
```

DOPR6 Module

Data block GEOM4S is a new input. RGSENS is a new input parameter.

Version 70 in SubDMAP in PSLGDV

```
DOPR6     DTOS4,GPECT,EQEXINS,DESGID,EST/
          DGTAB,ESTDVS,TABEVS/
          RSONLY $
```

Version 70.5 in SubDMAP in PSLGDV

```
DOPR6     DTOS4,GPECT,EQEXINS,DESGID,EST,geom4s/
          DGTAB,ESTDVS,TABEVS/
          RSONLY/rgsens $
```

DSAD Module

Data blocks CASDSX and AUG are new outputs and RGSENS is a new input parameter.

Version 70 in SubDMAP DESCON

```
DSAD      RSP1CT,R1TAB,RESP12,OBJTAB,CONTAB,BLAMAX,LAMASX,
          DIVTAB,AUXTAB,STBTAB,FLUTAB,OUGVS1,OESS1,OSTRS1,
          OEF1AA,EFITS,ES1CS,EA1CS,OQGS1,
          DSCREN,XINIT,COORDN,OLX,FRQRSP,
          CASEDS,CASERS, GX,OPTPRM,PROPI,BGPDTS/
          R1VAL,R2VAL,RSP2R,R2VALR,CVAL,CVALR,OBJTBR,CNTABR,
          R1TABR,R1VALR,DRSTBL,FRQRPR,UG1,R1MAPR,R2MAPR,
          CASDSN,DRDUG,DRDUTB,CASADJ/
          WGTS/VOLS/S,N,OBJVAL/S,N,NR1OFFST/S,N,NR2OFFST/
          S,N,NCNOFFST/APP/DMRESD/SEID/DESITER/EIGNFREQ/
          S,N,ADJFLG/PEXIST/MBCFLG $
```

Version 70.5 in SubDMAP DESCON

```
DSAD      RSP1CT,R1TAB,RESP12,OBJTAB,CONTAB,BLAMAX,LAMASX,
          DIVTAB,AUXTAB,STBTAB,FLUTAB,OUGVS1,OESS1,OSTRS1,
          OEF1AA,EFITS,ES1CS,EA1CS,OQGS1,
          DSCREN,XINIT,COORDN,OLX,FRQRSP,
          CASEDS,CASERS,UGX,OPTPRM,PROPI,BGPDTS/
          R1VAL,R2VAL,RSP2R,R2VALR,CVAL,CVALR,OBJTBR,CNTABR,
          R1TABR,R1VALR,DRSTBL,FRQRPR,UG1,aug,R1MAPR,R2MAPR,
          CASDSN,casdsx,DRDUG,DRDUTB,CASADJ,lcdvec0/
          WGTS/VOLS/S,N,OBJVAL/S,N,NR1OFFST/S,N,NR2OFFST/
          S,N,NCNOFFST/APP/DMRESD/SEID/DESITER/EIGNFREQ/
          S,N,ADJFLG/PEXIST/MBCFLG/rgsens $
```

DSPRM Module

Version 70 in SubDMAP RESPSEN

```
DSPRM     DRSTBL//
          S,N,DOWTVOLD/S,N,DOBUCKLD/S,N,DOMODESD/S,N,DOSTATIC/
          S,N,FAILD/S,N,CSTRSD/S,N,CSTRND/S,N,DOFREQ/S,N,DOMTRAN/
          S,N,DODVRGD/S,N,DOSAERD/S,N,DOFLUTD//s,n,dostatid/ADJFLG  $
```

Version 70.5 in SubDMAP RESPSEN

```
DSPRM     DRSTBL//
          S,N,DOWTVOLD/S,N,DOBUCKLD/S,N,DOMODESD/S,N,DOSTATID/
          S,N,FAILD/S,N,CSTRSD/S,N,CSTRND/S,N,DOFREQ/S,N,DOMTRAN/
          S,N,DODVRGD/S,N,DOSAERD/S,N,DOFLUTD//            ADJFLG $
```

GI Module

Data block BGPA is now replaced by BGPDT, AEBGPDT, and AEUSET. AEBGPDT, and AEUSET are defined with DBVIEW statements. Data blocks AERO and AECOMP are new inputs. Data block GTKG is replaced by GPGK0 and GDGK0. Parameter NK is no longer required.

Version 70 in SubDMAP PFAERO

```
GI        SPLINE,cstma,bgpa/gtkg/nk $
```

Version 70.5 in SubDMAP PHASE0

```
dbview aeuset=uset0   where (modltype='aerostrc' and wildcard) $
dbview aebgpdt=bgpdts where (modltype='aerostrc' and wildcard) $
GI        aero,SPLINE,bgpdt,aebgpdt,aeuset,aecomp,CSTMA/
          gpgk0,gdgk0 $
```

GPFDR Module

Data block QMG is a new input.

Version 70 in SubDMAP SEDRCVR

```
GPFDR     CASEDR,UG,KELMX,KDICTX,ECTS,EQEXINS,GPECT,PG1,QG,
          BGPDTS,SILS,CSTMS,VELEM,PTELEM/
          ONRGY1,OGPFB1/APP2/TINY//CYCFLG $
```

Version 70.5 in SubDMAP in SEDRCVR

```
GPFDR     CASEDR,UG,KELMX,KDICTX,ECTS,EQEXINS,GPECT,PG1,QG,
          BGPDTS,SILS,CSTMS,VELEM,PTELEM,qmg/
          ONRGY1,OGPFB1/APP2/TINY//CYCFLG $
```

GP3 Module

Data block EDTS is a new input.

Version 70 in SubDMAP PHASE0

```
GP3        GEOM3Y,EQEXINS,GEOM2S/SLT,ETT/0/0/0 $
```

Version 70.5 in SubDMAP in PHASE0

```
GP3        GEOM3Y,EQEXINS,GEOM2S,edts/SLT,ETT/0/0/0 $
```

IFP and MODEPT Modules

RUNMEPT is a new output parameter of IFP used to control the execution of the MODEPT module. NOGOMEPT is a new output parameter indicating success or failure in MODEPT.

Version 70 in SubDMAP IFPS

```
IFP        IBULK/
           IGEOM1.0,IEPT.0,IMPT.0,IEDT,IDIT,IDYNAMIC,
           IGEOM2.0,IGEOM3.0,IGEOM4.0,IEPTA,UNUSED2,
           POL.0,IAXIC,PVTX,IDMI,IDMINDX,IDTI,IDTINDX,
           DEFUSET,IEDOM,DEQATNX,DEQINDX,CONTACT,OINT,UNUSED3/
           S,N,NOGOIFP0/S,N,RUNIFP3/S,N,RUNIFP4/S,N,RUNIFP5/S,N,RUNIFP6/
           S,N,RUNIFP7/S,N,RUNIFP8/S,N,RUNIFP9/SEID $
MODEPT     IEPTX,IDIT/IEPT $ MODIFY ABSORBER AND BARRIER PROPERTIES
```

Version 70.5 in SubDMAP IFPS

```
IFP       IBULK/
          IGEOM1.0,IEPT.0,IMPT.0,IEDT,IDIT,IDYNAMIC,
          IGEOM2.0,IGEOM3.0,IGEOM4.0,IEPTA,UNUSED2,
          IMTPOL.0,IAXIC,PVTX,IDMI,IDMINDX,IDTI,IDTINDX,
          DEFUSET,IEDOM,DEQATNX,DEQINDX,CONTACT,OINT,UNUSED3/
          S,N,NOGOIFP0/S,N,RUNIFP3/S,N,RUNIFP4/S,N,RUNIFP5/S,N,RUNIFP6/
          S,N,RUNIFP7/S,N,RUNIFP8/S,N,RUNIFP9/SEID/s,n,runmept $
if ( runmept ) then $
   MODEPT  IEPTX,IDIT/IEPT/s,n,nogomept $
else $
   EQUIVX  IEPTX/IEPT/-1 $
endif $
```

MODEPT Module

See IFP.

MODTRK Module

Data block R1TABR is a new input.

Version 70 in SubDMAP FEA

```
MODTRK    CASEM,EDOM,LAMAS,MGG,MAA,PHG,PHSA,
          PHGREF0,PHAREF0/
          MTRAK,NEWLAMA,NEWPHG,NEWPHA,PHGREF,PHAREF/
          DESCYCLE/S,N,NOTRACK $
```

Version 70.5 in SubDMAP FEA

```
MODTRK    CASEM,EDOM,r1tabr,LAMAS,MGG,MAA,PHG,PHSA,
          PHGREF0,PHAREF0/
          MTRAKS,NEWLAMA,NEWPHG,NEWPHA,
          PHGREF,PHAREF/
          DESCYCLE/S,N,NOTRACK $
```

NLITER Module

Data blocks DEQIND and DEQATN are new inputs.

Version 70 in SubDMAP NLSTATIC

```
NLITER      CASEB,CNVTST,PLMAT,YSMAT,KAAL,ELDATA,KELMNL,LLLT,
            GMNL,MPTS,DIT,MJJ,SLT1,CSTMS,BGPDTH,SILS,USETNL,RDEST,RECM,
            KGGNL,UUU,GPSNTS,EDTS,DITID/
            UGNIMO,FGNL,ESTNLH,CIDATA,QNV,FFGH,MUGNI,MESTNL,DUGNIMO,
            BTOPCNV,BTOPSTF/
            S,N,LOADFAC/S,N,CONV/S,N,RSTEP/S,N,NEWP/S,N,NEWK/
            S,N,POUTF/S,N,NSKIP/LGDISP/S,N,NLFLAG/S,N,ITERID/
            S,N,KMATUP/S,N,LSTEP/S,N,KTIME/S,N,SOLCUR/TABS/
            S,N,KFLAG/S,N,NBIS/NORADMAT/S,N,LASTCNMU/SIGMA/
            S,N,ARCLG/S,N,ARCSIGN/S,N,TWODIV/LANGLES/S,N,IOPT/
            S,N,ITSEPS/ITSMAX/S,N,PLSIZE/IPAD/IEXT/
            S,N,ADPCONX/PBCONT/S,N,NBCONT  $
```

Version 70.5 in SubDMAP NLSTATIC

```
NLITER      CASEB,CNVTST,PLMAT,YSMAT,KAAL,ELDATA,KELMNL,LLLT,
            GMNL,MPTS,DIT,MJJ,SLT1,CSTMS,BGPDTH,SILS,USETNL,RDEST,RECM,
            KGGNL,UUU,GPSNTS,EDTS,DITID,deqind,deqatn/
            UGNIMO,FGNL,ESTNLH,CIDATA,QNV,FFGH,MUGNI,MESTNL,DUGNIMO,
            BTOPCNV,BTOPSTF/
            S,N,LOADFAC/S,N,CONV/S,N,RSTEP/S,N,NEWP/S,N,NEWK/
            S,N,POUTF/S,N,NSKIP/LGDISP/S,N,NLFLAG/S,N,ITERID/
            S,N,KMATUP/S,N,LSTEP/S,N,KTIME/S,N,SOLCUR/TABS/
            S,N,KFLAG/S,N,NBIS/NORADMAT/S,N,LASTCNMU/SIGMA/
            S,N,ARCLG/S,N,ARCSIGN/S,N,TWODIV/LANGLES/S,N,IOPT/
            S,N,ITSEPS/ITSMAX/S,N,PLSIZE/IPAD/IEXT/
            S,N,ADPCONX/PBCONT/S,N,NBCONT  $
```

NLTRD2 Module

Data blocks DEQIND and DEQATN are new inputs.

Version 70 in SubDMAP NLTRAN

```
NLTRD2      CASESX2H,PDT,YS,ELDATAH,KELMNL,KDD0,GM0,MPTS,DIT,KBDD,
            DLT1,CSTMS,BGPDTS,SILS,USETD,BRDD,MDD1,NLFT,RDEST,RECM,
            BDD0,GPSNTS,DITID/
            ULNTH,IFSH,ESTNLH,IFDH,OESNL1H,PNLH,TELH,MULNT,MESTNL,
```

```
          BTOPCNV,BTOPSTF,OESNLBH/
          KRATIO/S,N,CONV/S,N,RSTIME/S,N,NEWP/S,N,NEWDT/S,N,OLDDT/
          S,N,NSTEP/LGDISP/S,N,CONSEC/S,N,ITERID/ITIME/
          S,N,KTIME/S,N,LASTUPD/S,N,NOGONL/S,N,NBIS/MAXLP/
          TSTATIC/LANGLES/NDAMP/TABS/SIGMA/NORADMAT/
          S,N,ADPCONX/PBCONT/S,N,NBCONT  $
```

Version 70.5 in SubDMAP NLTRAN

```
NLTRD2    CASESX2H,PDT,YS,ELDATAH,KELMNL,KDD0,GM0,MPTS,DIT,KBDD,
          DLT1,CSTMS,BGPDTS,SILS,USETD,BRDD,MDD1,NLFT,RDEST,RECM,
          BDDL,GPSNTS,DITID,deqind,deqatn/
          ULNTH,IFSH,ESTNLH,IFDH,OESNL1H,PNLH,TELH,MULNT,MESTNL,
          BTOPCNV,BTOPSTF,OESNLBH/
          KRATIO/S,N,CONV/S,N,RSTIME/S,N,NEWP/S,N,NEWDT/S,N,OLDDT/
          S,N,NSTEP/LGDISP/S,N,CONSEC/S,N,ITERID/ITIME/
          S,N,KTIME/S,N,LASTUPD/S,N,NOGONL/S,N,NBIS/MAXLP/
          TSTATIC/LANGLES/NDAMP/TABS/SIGMA/NORADMAT/
          S,N,ADPCONX/PBCONT/S,N,NBCONT  $
```

NORM Module

XNORMD is a new output parameter which is the double-precision equivalent of XNORM.

Version 70

```
NORM      A/ANORM/
          S,N,NCOL/S,N,NROW/S,N,XNORM/IOPT $
```

Version 70.5

```
NORM      A/ANORM/
          S,N,NCOL/S,N,NROW/S,N,XNORM/IOPT/s,n,xnormd $
```

RANDOM Module

Data blocks OSTR2 and OQMG2 are new inputs. Data blocks OUGPSD2, OUGATO2, OUGRMS2, OUGNO2, OPGPSD2, OPGATO2, OPGRMS2, OPGNO2, OQGPSD2, OQGATO2, OQGRMS2, OQGNO2, OESPSD2, OESATO2, OESRMS2,

OESNO2, OEFPSD2, OEFATO2, OEFRMS2, OEFNO2, OEEPSD2, OEEATO2, OEERMS2, OEENO2, OQMPSD2, OQMATO2, OQMRMS2 and OQMNO2 are new outputs.

Version 70 in SubDMAP SEDRCVR

```
RANDOM    XYCDBDR,DIT,PSDL,OUGV2,OPG2,OQG2,OES2,OEF2,CASEDR/
          PSDF,AUTO/S,N,NORAND $
```

Version 70.5 in SubDMAP SEDRCVR

```
RANDOM    XYCDBDR,DIT,PSDL,OUGV2,OPG2,OQG2,OES2,OEF2,CASEDR,
          ostr2,oqmg2/
          PSDF,AUTO,
          ougpsd2,ougato2,ougrms2,ougno2,
          opgpsd2,opgato2,opgrms2,opgno2,
          oqgpsd2,oqgato2,oqgrms2,oqgno2,
          oespsd2,oesato2,oesrms2,oesno2,
          oefpsd2,oefato2,oefrms2,oefno2,
          oeepsd2,oeeato2,oeerms2,oeeno2,
          oqmpsd2,oqmato2,oqmrms2,oqmno2/
          S,N,NORAND $
```

READ and REIGL Modules

The REIGL module is now embedded inside READ module. Data block DMX is replaced by DXR which has the same number of columns as KXX and MXX. Data block USET is moved to the sixth input position. Data block EQEXINS is a new input for the Lanczos method. Data block LLL is a new input only required for buckling analysis. Data block EIGVMAT is a new output for the Lanczos method.

Version 70 in SubDMAP MODERS

```
EQUIVX   DMS/DMLQ/NOQSET $
IF ( NOQSET>-1 ) THEN $
   VEC      USET/VALCOMP/'A'/'L'/'COMP' $
   PARTN    VALCOMP,,VACMPRS/VLQ,,,/1 $ REMOVE R-SET
   MERGE    DMS,,,,,VLQ/DMLQ/1 $ ADD NULL COLUMNS FOR Q-SET
ENDIF $ NOQSET>-1
IF ( NORSET>-1 ) PARTN    DMLQ,,VLQXW/DMX,,,/1 $
IF ( LANCZOS>-1 ) THEN $
```

```
   READ   KXX,MXX,MR,DMX,EED,,CASES,VAXF, SILS,USET/
          LAMA,PHIX,MI,OEIGS/
          READAPP/S,N,NEIGV/NSKIP/SECND $
ELSE $
   REIGL  KXX,MXX,DYNAMICS,CASES,,MR,DMX,USET,EQEXINS,SILS,VAXF/
          LAMA,PHIX,MI,EIGVMAT,/
          READAPP/S,N,NEIGV/NSKIP/SECND//////////NSUB $
ENDIF $ LANCZOS>-1
```

Version 70.5 in SubDMAP MODERS

```
if ( norset>-1 ) then $
   matgen   ,/irr/1/norset $
   umerge1  uset,dm,irr,,/dtr/'t'/'l'/'r'/1 $
   if ( noqset>-1 ) then $
      umerge1  uset,dtr,,,/dar/'a'/'t'/'q'/1 $
   else $
      equivx   dtr/dar/-1 $
   endif $
endif $ norset>-1
equivx   dar/dxr/noared $
if ( noared>-1 and norset>-1 ) partn    dar,,vaxw/dxr,,,/1 $
READ     KXX,MXX,mr,dxr,EED,USET,CASES,VAXF,SILS,,,111,EQEXINS/
         LAMA,PHIX,mix,OEIGS,eigvmat,/
         READAPP/S,N,NEIGV/NSKIP/SECND $
```

REIGL Module

See READ.

SDP Module

Data block EDT is a new input. Data blocks RHMCF, ERHM, and EUHM are new outputs related to hinge moment output.

Version 70 in SubDMAP AESTATRS

```
SDP        CASEA,XLIST,TR,AERO,CSTMA,edt,KRZX,IPZ,Z1ZX,IPZF2,RSTAB,
           RINT,KSAZX,INTZ/
           STBDER/MACHNO/Q/LPRINT $
```

Version 70.5 in SubDMAP AESTATRS

```
SDP       CASEA,XLIST,TR,AERO,CSTMA,KRZX,IPZ,Z1ZX,IPZF2,RSTAB,
          RINT,KSAZX,INTZ,rhmcf,erhm,euhm/
          STBDER/MACHNO/Q/LPRINT $
```

SEDRDR Module

SCNDRY and EXTRN are new output parameters indicating secondary and external superelements and are the same as output parameters on the SEP2DR module.

Version 70 in SubDMAP SEDRCVR

```
SEDRDR    DRLIST,EMAP//
          S,N,ENDDR/S,N,SEID/S,N,PEID/S,N,SEDWN/S,N,NODR/NOSE/
          S,N,SETYP/S,N,REID $
```

Version 70.5 in SubDMAP SEDRCVR

```
SEDRDR    DRLIST,EMAP//
          S,N,ENDDR/S,N,SEID/S,N,PEID/S,N,SEDWN/S,N,NODR/NOSE/
          S,N,SETYPE/S,N,REID/s,n,scndry/s,n,extrn $
```

WEIGHT Module

Data block WMID is a new output. SEID is a new input parameter.

Version 70 in SubDMAP DESCON

```
WEIGHT    VELEM,EST,MPTS,DIT/
          /
          WGTVOL/S,N,VOLS/S,N,WGTS $
```

Version 70.5 in SubDMAP DESCON

```
WEIGHT    VELEM,EST,MPTS,DIT/
          wmid/
          WGTVOL/S,N,VOLS/S,N,WGTS/seid $
```

A P P E N D I X A

NEW DATA BLOCK DESCRIPTIONS

DATABLK EDT

```
DATABLK EDT     TYPE=TABLE      PATH=IFP        LOCATION=IFPXOPT,
 $ DESC={AERO AND ELEMENT DEFORMATIONS                  }
  RECORD=HEADER,
         NAME(2),CHAR4,{         }
  EOR,

......Records that are unchanged relative to V70 are not shown

  RECORD=SPLINE1(3302,33,266),
         EID,I,{         }
         CAERO,I,{         }
         BOX1,I,{         }
         BOX2,I,{         }
         SETG,I,{         }
         DZ,RS,{         }
         METHOD,CHAR4, {Method: IPS|TPS|FPS}
         USAGE,CHAR8, { Usage flag: FORCE|DISP|BOTH }
         NELEM,I, {Number of elements for FPS on x-axis }
         MELEM,I, {Number of elements for FPS on y-axis }
  EOR,
  RECORD=SPLINE2(3402,34,267),
         EID,I,{         }
         CAERO,I,{         }
         BOX1,I,{         }
         BOX2,I,{         }
         SETG,I,{         }
         DZ,RS,{         }
         DTOR,RS,{         }
         CID,I,{         }
         DTHX,RS,{         }
         DTHY,RS,{         }
         USAGE,CHAR8, { Usage flag: FORCE|DISP|BOTH }
  EOR,
  RECORD=SPLINE3(4901,49,173),
         EID,I,{         }
         CAERO,I,{         }
         UKID,I,{         }
         COMP,I,{         }
         USAGE,CHAR8, { Usage flag: FORCE|DISP|BOTH }
         ENTRY,
               G1,I,{         }
               C1,I,{         }
```

```
                A1,RS,{         }
          ENDENTRY,WITH,–1,
   EOR,
   RECORD=SPLINE4(6501,65,308),
          EID,I,{ Spline element Identification }
          CAERO,I,{ Component Identifification }
          AELIST,I,{ AELIST Id for boxes }
          SETG,I,{ SETi Id for grids }
          DZ,RS,{ Smoothing Parameter }
          METHOD,CHAR4, {Method: IPS|TPS|FPS}
          USAGE,CHAR8, { Usage flag: FORCE|DISP|BOTH }
          NELEM,I, {Number of elements for FPS on x–axis }
          MELEM,I, {Number of elements for FPS on y–axis }
   EOR,
   RECORD=SPLINE5(6601,66,309),
          EID,I,{ Spline element Identification }
          CAERO,I,{ Component Identifification }
          AELIST,I,{ AELIST Id for boxes }
          SETG,I,{ SETi Id for grids }
          USAGE,CHAR8, { Usage flag: FORCE|DISP|BOTH }
          CID,I,{ Coordinate Sys. Id. for Beam CS }
          DTHX,RS,{ Smoothing/Attachment Flags for X rotations }
          DTHY,RS,{ Smoothing/Attachment Flags for Y rotations }
          DTHZ,RS,{ Smoothing/Attachment Flags for Z rotations }
          DZ,RS,{ Smoothing Parameter }
          DTORXY,RS,{ Flexibility ratio in XY Plane }
          DTORZY,RS,{ Flexibility ratio in ZY Plane }
   EOR,

......Records that are unchanged relative to V70 are not shown

  $ NOTE={                }
   FEOF
```

DATABLK AEBOX (the aerodynamic instances of the ECT family)

```
 DATABLK ECT    TYPE=TABLE      PATH=GEOM       LOCATION=DBDNX,
$ Desc={ Element Connection Table in internal ID order }
 RECORD=HEADER,
        NAME(2),CHAR4,{ Block Name        }
 EOR,
```

... Records for structural elements omitted ...

```
 RECORD=Q4AERO(3002,46,0),
        EID,I, { BOX IDENTIFICATION NUMBER }
        COMPID,I, { COMPONENT ID IN AECOMP }
        COMPTYPE(2),CHAR4, { COMPONENT TYPE }
        G(4),I,{ GRID ID'S IN AEGRID DEFINING PERIMETER }
 EOR,
 RECORD=T3AERO(2701,27,0),
        EID,I,{ BOX IDENTIFICATION NUMBER }
        COMPID,I, { COMPONENT ID IN AECOMP }
        COMPTYPE(2),CHAR4, {COMPONENT TYPE}
        G(3),I,{ GRID ID'S IN AEGRID DEFINING PERIMETER }
 EOR,
 RECORD=BEAMAERO(2601,26,0),
        EID,I,{ BOX IDENTIFICATION NUMBER }
        COMPID,I, { COMPONENT ID IN AECOMP }
        COMPTYPE(2),CHAR4, {COMPONENT TYPE: SLBD (SLENDER BODY)}
        G(2),I,{ GRID ID'S IN AEGRID DEFINING PERIMETER }
 EOR,
 RECORD=TRAILER,
        WORD1,I,{            }
```

```
       WORD2,I,{            }
       WORD3,I,{ Not defined }
       WORD4,I,{ Not defined }
       WORD5,I,{ Not defined }
       WORD6,I,{ Not defined }
 EOR,
$ Note={ When the ECT is an alias for the GEOM2VU block         }
$ Note={the following applies }
$ Note={                }
$ Note={ The VUHEXA  header word three becomes 9921 }
$ Note={ The VUPENTA header word three becomes 9922 }
$ Note={ The VUTETRA header word three becomes 9923 }
$ Note={ For each of the above, the grid id is then a VIEW grid id
}
$ Note={ The beginning value of the VIEW grids is controlled by }
$ Note={ system cell 182.                                       }
$ Note={                }
$ Note={ Internal indices:}
$ Note={ cquadp: (NGRIDS + 2*NEDGES +4*NFACES + 1 Bubble Point) }
$ Note={ ctriap: (NGRIDS+2*NEDGES+NFACES + 1BODY(for bubble) }
$ Note={ cbeamp: (NGRIDS+2*NEDGES)}
$ Note={                }
$ Note={ --------       }
$ Note={                }
$ Note={ For AERO Records the }
$ Note={ Component Types are general labels for components. }
$ Note={                }
$ Note={ SLBD are Slender Body Types and are “BEAM–LIKE” }
$ Note={      Elements appearing only in the BEAMAERO Record.}
$ Note={      }
$ Note={ The remaining Components Types can be QUAD or TRIA}
$ Note={      connections denoting various element types.}
$ Note={      }
$ Note={ INBD are Interference Body Panels. }
$ Note={      }
$ Note={ LS   are Lifting Surface Panels.}
$ Note={      }
$ Note={ WAKE are Wake Boxes.}
$ Note={      }
$ Note={ MFLO are Flow–Thru Surfaces like Inlets (Mass–Flow).}
 EOF
```

DATABLK AEBGPDT (the Aerodynamic DOF instance of BGPDTS)

```
$ Desc={Basic Substructure Grid Point Definition Table  }
$ Desc={One logical record contains a list of  }
$ Desc={all grid and scalar points in internal  }
$ Desc={ sort, with (for grid points) their }
$ Desc={x, y, z coordinates in the substructure basic system  }
$ Desc={along with a coordinate system ID  }
$ Desc={for displacement computations.  }
 RECORD=HEADER,
       NAME(2),CHAR4,{ Header record        }
 EOR,
 RECORD=DATA,
   ENTRY=GRID,
        CID,I,{Coordinate system ID   }
        SIL,I,{Internal (scalar) ID   }
        EXTID,I,{External (User)   ID   }
        DOF_TYPE,I,{Degree of Freedom/ Point Type }
        PSC,I,{Permanent Set Constraint }
        BGID,I,{Boundary Grid ID of –EXTID }
        XCOORD,RX,{x in substructure basic system     }
        YCOORD,RX,{y in substructure basic system     }
```

```
         ZCOORD,RX,{z in substructure basic system     }
   ENDENTRY,WITH,
 EOR,
 RECORD=XIDMAP,
   ENTRY=EQEXIN,
         EXTID,I,{External (User)  ID   }
         INTID,I,{Internal (    )  ID   }
   ENDENTRY,WITH,
 EOR,
 RECORD=BIDMAP,
   ENTRY=EQBGIN,
          BGID,I,{Boundary (System)  ID   }
         INTID,I,{Internal (      )  ID   }
   ENDENTRY,WITH,
 EOR,
 RECORD=NORMAL,
   ENTRY=VECTOR,
         XNORM,RX,{X NORMAL IN AERODYNAMIC SYSTEM     }
         YNORM,RX,{Y NORMAL IN AERODYNAMIC SYSTEM     }
         ZNORM,RX,{Z NORMAL IN AERODYNAMIC SYSTEM     }
   ENDENTRY,WITH,
 EOR,
 RECORD=TRAILER,
        WORD1,i,{ number  of grid points + number of scalar points }
        WORD2,i,{ number of boundary grids }
        WORD3,i,{ length of the u-set }
        WORD4,i,{ precision of the real data }
        WORD5,i,{ number of scalar points }
        WORD6,i,{ maximum external identification }
 EOR,
$ Note={ Coordinates system  ID is integer;              }
$ Note={ x, y, z and normal are machine precision, floating point.   }
$ Note={Scalar points are identified by                   }
$ Note={coordinate system ID = '–1', and x, y, z = 0.     }
$ Note={ When T(2), number of boundary is zero, record    }
$ Note={ BIDMAP does not exist and XIDMAP will be used.   }
EOF
```

DATABLK AECOMP

```
$ Desc={ AERODYNAMIC COMPONENT DEFINITION TABLE  }
 RECORD=HEADER,
        NAME(2),CHAR4,{ Block Name }
 EOR,
 RECORD=COMPNTS,
        ENTRY=COMP,
              COMPID,I,{COMPONENT IDENTIFICATION }
              COMPTYPE(2),CHAR4, {COMPONENT TYPE}
              CID,I,{ COORDINATE SYSTEM ID FOR COMPONENT}
              NXBOX,I,{NUMBER OF BOXES IN FIRST PARAMETER DIRECTION }
              NYBOX,I,{NUMBER OF BOXES IN SECOND PARAMETER DIRECTION }
              GROUP,I,{ INTERFERENCE GROUP IDENTIFICATION }
              GEOMTYPE(2),CHAR4, {GEOMETRY TYPE BEAM, TRIA or QUAD }
              GRID1,I,{INBOARD LE GRID ID FROM AEGRID }
              GRID2,I,{INBOARD TE GRID ID FROM AEGRID }
              GRID3,I,{OUTBOARD TE GRID ID FROM AEGRID }
              GRID4,I,{OUTBOARD LE GRID ID FROM AEGRID }
        ENDENTRY,WITH,
 EOR,
 RECORD=TRAILER,
   WORD1,i,{ Number of Components }
   WORD2,i,{ not defined }
   WORD3,i,{ not defined }
```

```
   WORD4,i,{ not defined }
   WORD5,i,{ not defined }
   WORD6,i,{ not defined }
 EOR,
$ Note={THESE DATA REPLACE THE DATA USED BY GI }
$ Note={ FOR SPLINE CREATION. THE DATA INCLUDE GEOMETRY }
$ Note={ DATA FOR EACH COMPONENT SO THAT THE LOCAL GEOMETRY}
$ Note={ IS PARAMETERIZED FOR SPLINING }
$ Note={    }
$ Note={ COMPONENT TYPES ARE GENERAL LABELS FOR COMPONENTS }
$ Note={    }
$ Note={ SLBD ARE SLENDER BODY BOXES}
$ Note={ SLBDY, LSBDZ, SLBDZY }
$ Note={    }
$ Note={ INTF ARE INTERFERENCE BODY BOXES }
$ Note={    }
$ Note={ BOXES ARE FLAT PLATE CAERO1,2,3,4,5 BOXES}
$ Note={    }
$ Note={ LS   ARE GENERAL LIFTING SURFACE BOXES}
$ Note={    }
$ Note={ WAKE ARE WAKE BOXES}
$ Note={    }
$ Note={ MFLO ARE FLOW–THRU BOXES LIKE INLETS (MASS–FLOW)}
EOF
```

DATABLK AEGRID (the aerodynamic mesh point instance of BGPDTS)

```
DATABLK AEGRID TYPE=TABLE     PATH=GEOM      LOCATION=DBDNX,
$ DESC={ CONTAINS AERODYNAMIC CONNECTIVITY GRID POINTS }
$ DESC={ AS OPPOSED TO AEBGPDT WHICH CONTAINS THE GEOMETRY }
$ DESC={ OF THE DEGREES OF FREEDOM (K–SET) }
        SAMEAS,BGPDT,
EOF
```

DATABLK SPLINE

```
$ Desc={     }
$ Desc={ SPLINE BULK DATA   (SEE BULK ENTRY DEFINITIONS)     }
 RECORD=HEADER,
       NAME(2),CHAR4,{ BLOCK NAME   }
 EOR,
 RECORD=SET1(3502,35,268),
       SID,I,{  SPLINE ID   }
       ENTRY,
            G1,I,{ GRID ID        }
       ENDENTRY,WITH,–1,
 EOR,
 RECORD=SET2(3602,36,269),
       SID,I,{ SPLINE ID        }
       NROWS,I,{ Number of rows in CB and B  }
       CB(18),RS,{ Prism Inequality Matrix }
       B(6),RS,{ Prism Inequality Vector }
 EOR,
 RECORD=AELIST(2302,23,341),
       AELSID,I, { AELIST ID }
       ENTRY,
             BOXID,I,{ One External Aero Box ID }
       ENDENTRY,WITH,–1,
```

```
EOR,
RECORD=SPLINE1(3302,33,266),
       EID,I,{ SPLINE ID }
       CAERO,I,{ COMPONENT ID }
       AELIST,I,{ AELIST ID for boxes }
       SETG,I,{ SET ID OF SET1 OR SET2 }
       DZ,RS,{ SMOOTHING PARAMETER }
       METHOD(2),CHAR4, {Method: IPS|TPS|FPS}
       USAGE(2),CHAR4, { Usage flag: FORCE|DISP|BOTH }
       NELEM,I, {Number of elements for FPS on x-axis }
       MELEM,I, {Number of elements for FPS on y-axis }
EOR,
RECORD=SPLINE2(3402,34,267),
       EID,I,{ SPLINE ID }
       CAERO,I,{ COMPONENT ID }
       AELIST,I,{ AELIST ID for boxes }
       SETG,I,{ SET ID OF SET1 OR SET2 }
       DZ,RS,{         }
       DTOR,RS,{         }
       CID,I,{         }
       DTHX,RS,{         }
       DTHY,RS,{         }
       USAGE(2),CHAR4, { Usage flag: FORCE|DISP|BOTH }
EOR,
RECORD=SPLINE3(4901,49,173),
       NWDS,I,{ Number of following words in the entry }
       EID,I,{ SPLINE ID }
       CAERO,I,{ COMPONENT ID }
       BOXID,I,{ BOX ID }
       COMP,I,{the component of motion to be interpolated 1<=0<=6}
       USAGE(2),CHAR4, { Usage flag: FORCE|DISP|BOTH }
       ENTRY,
           G1,I, { External Grid ID }
           C1,RS,{ Component Number (0 to 6) }
           A1,RS,{ Coefficient of the constraint relationship }
       ENDENTRY,WITH,
EOR,
RECORD=SPLINE4(6501,65,308),
       EID,I,{ Spline element Identification }
       CAERO,I,{ Component Identifification }
       AELIST,I,{ AELIST Id for boxes }
       SETG,I,{ SETi Id for grids }
       DZ,RS,{ Smoothing Parameter }
       METHOD(2),CHAR4, {Method: IPS|TPS|FPS}
       USAGE(2),CHAR4, { Usage flag: FORCE|DISP|BOTH }
       NELEM,I, {Number of elements for FPS on x-axis }
       MELEM,I, {Number of elements for FPS on y-axis }
EOR,
RECORD=SPLINE5(6601,66,309),
       EID,I,{ Spline element Identification }
       CAERO,I,{ Component Identifification }
       AELIST,I,{ AELIST Id for boxes }
       SETG,I,{ SETi Id for grids }
       DZ,RS,{ Smoothing Parameter }
       CID,I,{ Coordinate Sys. Id. for Beam CS }
       DTHX,RS,{ Smoothing/Attachment Flags for X rotations }
       DTHY,RS,{ Smoothing/Attachment Flags for Y rotations }
       DTHZ,RS,{ Smoothing/Attachment Flags for Z rotations }
       USAGE(2),CHAR4, { Usage flag: FORCE|DISP|BOTH }
       DTORXY,RS,{ Flexibility ratio in XY Plane }
       DTORZY,RS,{ Flexibility ratio in ZY Plane }
EOR,
RECORD=TRAILER,
       WORD1,I,{Locate Flags}
       WORD2,I,{Locate Flags}
       WORD3,I,{Locate Flags}
       WORD4,I,{Locate Flags}
       WORD5,I,{Locate Flags}
```

```
        WORD6,I,{Locate Flags}
 EOR,
$ Note={FOR SPLINE1 AND SPLINE2, THE BOX1,BOX2 RECTANGULAR}
$ Note={ REGION DEFINED BY INBOARD LE BOX1 AND OUTBOARD TE BOX2}
$ Note={ ARE CONVERTED INTO AN INTERNALLY GENERATED AELIST }
$ Note={      }
$ Note={ THE SPLINE RECORDS THEN POINT TO THE AELIST RECORD FOR }
$ Note={ ALL SPLINE FORMS }
$ Note={      }
$ Note={ SET2 DATA ARE CONVERTED TO CB AND B MATRICES, THE }
$ Note={ ACTUAL LIST OF ATTACHED GRIDS IS GENERATED IN GI }
EOF
```

APPENDIX B

DVPREL1 AND DVPREL2 BULK DATA ENTRIES

Enhancements were done too late in the release of Version 70.5 for the changes to the DVPREL1 and DVPREL2 entries to make it into the *MSC/NASTRAN Quick Reference Guide*, Version 70.5. The changes are noted in Remarks 4 and 5 for the DVPREL1 entry and Remark 3 of the DVPREL2 entry.

DVPREL1

Defines the relation between an analysis model property and design variables.

Format:

1	2	3	4	5	6	7	8	9	10
DVPREL1	ID	TYPE	PID	FID	PMIN	PMAX	C0		
	DVID1	COEF1	DVID2	COEF2	DVID3	-etc.-			

Example:

DVPREL1	12	PBAR	612	6	0.2	3.0			
	4	0.25	20	20.0	5	0.3			

Field	Contents
ID	Unique identification number. (Integer > 0)
TYPE	Name of a property entry, such as “PBAR”, “PBEAM”, etc. (Character)
PID	Property entry identification number. (Integer > 0)
FID	Field position of the property entry, or word position in the element property table of the analysis model. (Integer $\neq 0$)

(Continued)

PMIN	Minimum value allowed for this property. If FID references a stress recovery location, then the default value for PMIN is −1.0+35. PMIN must be explicitly set to a negative number for properties that may be less than zero (for example, field ZO on the PCOMP entry). (Real; Default=1.0E–20)
PMAX	Maximum value allowed for this property. (Real; Default=1.0E–20)
C0	Constant term of relation. (Real; Default=0.0)
DVIDi	DESVAR entry identification number. (Integer > 0)
COEFi	Coefficient of linear relation. (Real)

Remarks:

1. The relationship between the analysis model property and design variables is given by:

$$P_i = \text{C0} + \sum_i \text{COEFi} * \text{DVIDi}$$

2. The continuation entry is required.
3. PTYPE="PBEND" is not supported, either directly through FIDs or indirectly via word positions in the element property table.
4. FID may be either a positive or a negative number. If FID > 0, it identifies the field position on a property entry. If FID < 0, it identifies the word position of an entry in the element property table. For example, to specify the area of a PBAR, either FID = +4 or FID = −3 can be used.
5. Special rules apply for PTYPE=PBEAM.
   a. If non-constant cross section beams are being designed, the FID must be < 0. It is also necessary to design one more station using DVPRELi entries then there are stations on the PBEAM entry. See Section 2.4 of the *MSC/NASTRAN Design Sensitivity and Optimization User's Guide*.

   b. If constant cross section beams are being designed, the properties at end A only can be designated by FID > 0 ($4 \le \text{FID} \le 19$ and FID ≠ 20 or 21.) End B data should not be designated in this case. If the last two continuations of the PBEAM entry (K1 through N2(b)) are being designed, they must be referenced with FID < 0 to the EPT location.

   c. See the following element property table for the word position for PBEAM.

(Continued)

Table 1. EPT Section for PTYPE = “PBEAM”.

Word	Typ e	Item
1	I	Property ID
2	I	Material ID
3	I	Number of segments
4	I	Constant cross section flag (1 = yes, 0 = no)
5	I	Unused
6	I	Stress output request flag, SO (1 = yes, 0 = no)
7	R	X/XB ratio; at end A, X/XB = 0.0
8 through 13	R	A, I1, I2, I12, J, NSM
14 through 21	R	C1, C2, D1, D2, E1, E2, F1, F2
22 23 24 through 29 30 through 37	I R R R	Repeat of words 6 through 21 for the 1st intermediate station
38 39 40 through45 46 through 53	I R R R	2nd intermediate station
54 55 56 through 61 62 through 69	I R R R	3rd intermediate station
70 71 72 through 77 78 through 85	I R R R	4th intermediate station
86 87 88 through 93 94 through 101	I R R R	5th intermediate station
102 103 104 through 109 110 through 117	I R R R	6th intermediate station
118 119 120 through 125 126 through 133	I R R R	7th intermediate station

134 135 136 through 141 142 through 149	I R R R	8th intermediate station
150 151 152 through 157 158 through 165	I R R R	9th intermediate station
166 167 168 through 173 174 through 181	I R R R	End B
182 through 189	R	K1, K2, S1, S2, NSI(A), NSI(B), CW(A), CW(B)
190 through 197	R	M1(A), M2(A), M1(B), M2(B), N1(A), N2(A), N1(B), N2(B)

DVPREL2

Defines the relation between an analysis model property and design variables with a user-supplied equation.

Format:

1	2	3	4	5	6	7	8	9	10
DVPREL2	ID	TYPE	PID	FID	PMIN	PMAX	EQID		
	"DESVAR"	DVID1	DVID2	DVID3	-etc.-				
	"DTABLE"	LABL1	LABL2	LABL3	-etc.-				

Example:

DVPREL2	13	PBAR	712	5	0.2	4			
	DESVAR	4	11	13	5				
	DTABLE	PI	YM						

Field	Contents
ID	Unique identification number. (Integer > 0)
TYPE	Name of a property entry, such as PBAR, PBEAM, etc. (Character)
PID	Property entry identification number. (Integer > 0)
FID	Field position of the property in the analysis model entry. (Integer ≠ 0)
PMIN	Minimum value allowed for this property. If FID references a stress recovery location field, then the default value for PMIN is −1.0+35. PMIN must be explicitly set to a negative number for properties that may be less than zero (for example, field ZO on the PCOMP entry). (Real; Default = 1.E–20)
PMAX	Maximum value allowed for this property. (Real; Default = 1.0E20)
EQID	DEQATN entry identification number. (Integer > 0)
"DESVAR"	DESVAR flag. Indicates that the IDs of DESVAR entries follow. (Character)
DVIDi	DESVAR entry identification number. (Integer > 0)
"DTABLE"	DTABLE flag. Indicates that the IDs for the constants in a DTABLE entry follow. This field may be omitted if there are no constants involved in this relation. (Character)
LABLi	Label for a constant on the DTABLE entry. (Integer > 0)

Remarks:

1. The variables identified by DVIDi and LABLi correspond to variable names (x1, x2, etc.) listed in the left-hand side of the first equation on the DEQATN entry identified by EQID. The variable names x1 through xN (where N = m + n) are assigned in the order DVID1, DVID2, ..., DVIDn, LABL1, LABL2, ..., LABLm.

2. If both “DESVAR” and “DTABLE” are specified in field 2, “DESVAR” must appear first.

3. FID may be either a positive or a negative number. If $FID > 0$, it identifies the field position on a property entry. If $FID < 0$, it identifies the word position of an entry in EPT. For example, to specify the area of a PBAR, either $FID = +4$ or $FID = -3$ may be used. See Remark 5 on the DVPREL1 entry description for specification of the PBEAM element properties.

4. PTYPE = “PBEND” is not supported, either directly through FIDs or indirectly via word positions in the element property table.

I N D E X

Title: MSC/NASTRAN Release Guide – Version 70.5

I (we) wish to make copies of this material for distribution to co-workers. I understand that The MacNeal-Schwendler Corporation (MSC) grants permission to do this subject to the following:

1. All copyright notices must be included on all copies.
2. Copies may be made only for my fellow employees.
3. There will be no charges made for these copies (except that the costs of copying may be allocated internally).
4. I (we) will not provide any copies of this manual or excerpts thereof, to anyone who is not an employee of the company.

Signature: ______________________________ Date __________

Name: ______________________________

Title: ______________________________

Company: ______________________________

Address: ______________________________

____

____

Approved: The MacNeal-Schwendler Corporation

Signature: ______________________________

Name: ______________________________

Title: ______________________________

Date: ______________________________

Return completed form to your local MSC Office.

ERROR REPORT OR COMMENTS AND SUGGESTIONS

Title: MSC/NASTRAN Release Guide – Version 70.5

Page(s): ______________________________

Please describe error or suggestion:

Thanks for your feedback. Please FAX or Mail to:

Documentation and Customer Training
The MacNeal-Schwendler Corporation
815 Colorado Blvd.
Los Angeles, CA 90041
FAX: (213) 259-3838