C O N T E N T S

MSC.Nastran Common Questions and Answers

C O N T E N T S

MSC.Nastran Common Questions and Answers

C O N T E N T S

MSC.Nastran Common Questions and Answers

C O N T E N T S

MSC.Nastran Common Questions and Answers

C O N T E N T S

MSC.Nastran Common Questions and Answers

C O N T E N T S

MSC.Nastran Common Questions and Answers

Editor's Notes for Third Edition

This document describes answers to commonly asked technical questions about MSC/NASTRAN. It is organized into topical areas to make it easy to find questions and answers of interest. Because many questions and answers can be grouped into multiple topical areas, an index is included to facilitate finding answers to specific questions. This document is the second edition of the *MSC.Nastran Common Questions and Answers*.

While this document is written to coincide with Version 70, it is also applicable to earlier versions. Where necessary, version dependence is noted. As a topic is incorporated into its appropriate user's guide, that particular subject will eventually be removed from the *MSC.Nastran Common Questions and Answers Guide*.

When you have a technical question, this document may be used as a starting point. If the answer contained herein is insufficient, you should look in the other MSC.Nastran documentation referenced by the answer. If you do not have an answer after taking these steps, you should contact your MSC technical support representative.

Special thanks to Ted Rose and John Halcomb for providing various valuable suggestions. The assistance of Wendy Webb in preparing and Darryl Tewes in editing this manuscript is also gratefully acknowledged.

John Lee
October 1997

Editor's Notes for Second Edition

This document describes answers to commonly asked technical questions about MSC/NASTRAN. It is organized into topical areas to make it easy to find questions and answers of interest. Because many questions and answers can be grouped into multiple topical areas, an index is included to facilitate finding answers to specific questions. This document is the second edition of the MSC.Nastran Common Questions and Answers.

While this document is written to coincide with Version 68, it is also applicable to earlier versions. Where necessary, version dependence is noted. As a topic is incorporated into its appropriate user's guide, that particular subject will eventually be removed from the *MSC/NASTRAN Common Questions and Answers Guide.*

When you have a technical question, this document may be used as a starting point. If the answer contained herein is insufficient, you should look in the other MSC/NASTRAN documentation referenced by the answer. If you do not have an answer after taking these steps, you should contact your MSC technical support representative.

Suggestions for improvements to this book are most welcome and may be sent to MSC using the form in the back of this book. Our intent is to make this a good first step in answering many frequently asked technical questions.

Primary technical contributors included Nima Bakhtiary, Ken Blakely, John Caffrey, Brian Casey, Dan Chu, John Furno, Mike Gockel, John Halcomb, Dave Herting, Wai Ho, Louis Komzsik, David Lombard, Greg Moore, Ken Ranger, Ted Rose, and Candace Sumner. Sherry Anglum, Mark Miller, and Julie Suva provided the overall review and editing. The assistance of Lori Lampert, Joanne Oka-Benites, and Wendy Webb in preparing this manuscript is also gratefully acknowledged.

John Lee
December 1993

Glossary

Several terms are used repeatedly throughout this document and are listed in this section.

.	Used to denote multiplication.
.DAT	.dat file that contains the MSC.Nastran input.
DBset	A database set (an MSC.Nastran logical name, such as .MASTER, .DBALL, etc.) that refers to an entity within the MSC.Nastran database. This entity is used to store MSC.Nastran data and may consist of one or more physical files.
DMAP	Direct Matrix Abstraction Program.
DOF	Degree of freedom.
Error	Error Report Number in the Current Error List.
.F04	.f04 file that contains performance information and modules that were executed (the Execution Summary Table).
.F06	.f06 file that contains the printed output.
.LOG	.log file that contains the system messages.
NDDL	NASTRAN Data Definition Language.
.OP2	.op2 file that contains MSC.Nastran results for graphical postprocessing in MSC.Patran.
.PCH	.pch file that contains punched output.
SOL	Solution Sequence.
SFM	System Fatal Message.
SSS	Structured Solution Sequence (SOLs 100 through 200).
TPL	Test Problem Library.
UFM	User Fatal Message.
UIM	User Information Message.
UWM	User Warning Message.
.XDB	.xdb file that contains MSC.Nastran results for graphical postprocessing in MSC/XL, MSC/ARIES, or MSC.Nastran for Windows.

Getting Started

What Are the MSC.Nastran Analysis Capabilities?

MSC.Nastran is a comprehensive, general-purpose finite element analysis program that contains the following analysis capabilities:

- Linear statics (including inertia relief)
- Normal modes and buckling
- Heat transfer (steady-state and transient)
- Transient response
- Frequency response
- Response spectrum and random response
- Geometric and material nonlinear static and transient response
- Design optimization and sensitivity (including dynamic and shape optimization)
- Composite materials
- Acoustic response
- Aeroelasticity
- Superelements
- Complex eigenanalysis
- Cyclic symmetry
- p-elements

MSC.Nastran also includes DMAP (Direct Matrix Abstraction Programming), a high-level programming language that allows you to build custom solution sequences or modify existing ones.

Each capability uses similar input, making it easy to switch from one analysis capability to another.

What MSC.Nastran User Documentation Is Available?

This section briefly describes the MSC.Nastran documentation. A quick overview of these documents is shown in **Figure 1**.

MSC.Nastran Quick Reference Guide
: Contains a complete description of the NASTRAN statements, File Management statements, Executive Control statements, Case Control commands, Bulk Data entries, and parameters. This book is necessary for all MSC.Nastran users.

MSC.Nastran Release Guide
: Describes version-dependent capabilities and presents illustrative examples. This document is required to use the new capabilities in each new version of MSC.Nastran.

MSC.Nastran Common Questions and Answers
Contains answers to commonly asked questions on a wide variety of analysis topics.

MSC.Nastran Reference Manual
Contains material that is primarily reference oriented, non-solution sequence dependent material and is highly subdivided for ease of use. This manual is independent of machine type or operating system. This document is typically not required for the day-to-day analysis activities of most analysts and can therefore be shared by a group or department.

MSC.Nastran Configuration and Operations Guide
Contains machine-dependent information for installing, customizing, and using MSC.Nastran.

MSC.Nastran DMAP Programmer's Guide
Replaces Section 5 of the old *MSC.Nastran User's Manual*. This manual is required if you are using MSC.Nastran DMAP features.

MSC.Nastran User's Guides and Other Books. User's Guides describe in detail all aspects of MSC.Nastran input, output, and modeling guidelines used for specific analysis capabilities. These documents are recommended for the various capabilities that you use most often. The following books are available:

Getting Started with MSC.Nastran User's Guide

MSC.Nastran Linear Analysis Static User's Guide,

MSC.Nastran Basic Dynamic Analysis User's Guide

MSC.Nastran Advanced Dynamic Analysis User's Guide

MSC.Nastran Design Sensitivity and Optimization User's Guide.

MSC.Nastran Thermal Analysis User's Guide

MSC.Nastran Numerical Methods User's Guide

MSC.Nastran Aeroelastic User's Guide

MSC.Nastran Bibliography
Lists approximately 1900 technical papers pertaining to MSC.Nastran arranged by author and topic.

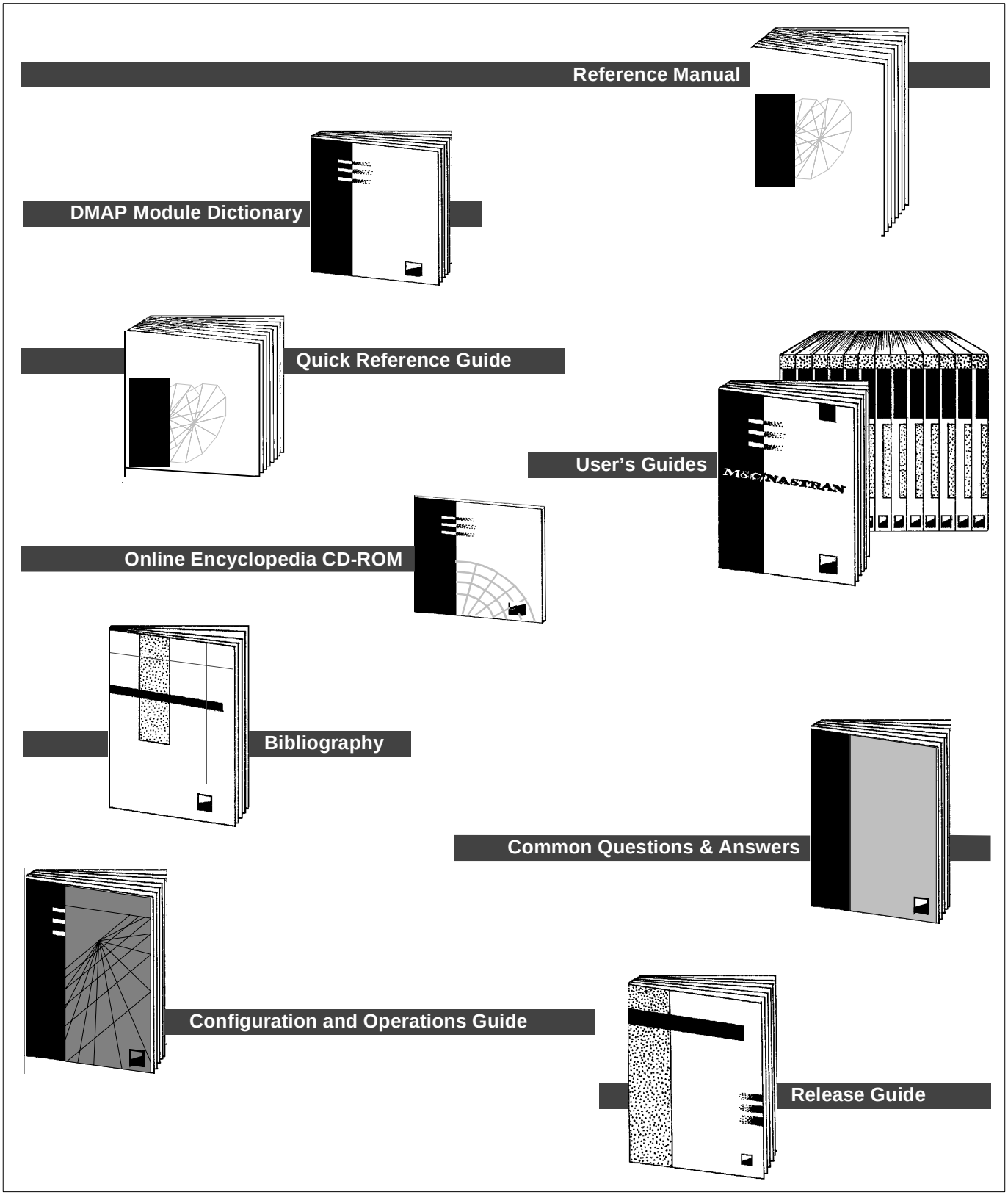


Figure 1 MSC.Nastran Documentation System

What Other Older User Documentation Is Available?

MSC.Nastran Handbook for Superelement Analysis

MSC.Nastran Handbook for Nonlinear Analysis

MSC.Nastran Demonstration Problem Manual
Shows illustrative examples with MSC.Nastran input and output.

MSC.Nastran Verification Problem Manual
Provides examples for which there are known solutions and compares MSC.Nastran output with theory. Examples using most of the Unstructured Solution Sequences are included.

The NASTRAN Theoretical Manual
Explains the theoretical basis of the elements and algorithms. The numerical algorithm material is superseded by the *MSC.Nastran Handbook for Numerical Methods* and the *MSC.Nastran Numerical Methods User's Guide.*

MSC.Nastran Programmer's Manual
Describes the input and output for the modules. This manual is not recommended for general use but is useful for advanced DMAP development.

All of the documents listed above can be ordered from your local MSC office or representative.

MSC Corporate Web Site

The MacNeal-Schwendler Corporation's Web site provides several sources of information that can assist you in running MSC.Nastran and our other products. MSC's Web address is

http://www.macsch.com

Here you can find out what is new with MSC, read white papers on the use of MSC products, download technical papers from previous Users' Conference, review the minutes of the Technical Forum, 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 even download software patches and utilities. The sssalter library and error list will also be posted on the Web in the future. 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.

Training Seminars. The MSC Institute of Technology offers a wide variety of training seminars on MSC.Nastran, MSC.Patran, and other MSC products. Public seminars are open to all and are offered at various locations throughout the world. Private classes,

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 the MSC website at http://www.macsch.com.

Browsing and Ordering Books from the *MSC Bookstore*. A complete list of documentation titles for MSC.Nastran, MSC.Patran, and our other products is available through the Web-based *MSC 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 and to order books electronically. The MSC Bookstore Web address is:

http://www.mscsoftware.com/bookstore/

Subscribing to MSC/WORLD. *MSC/WORLD* is a newsletter about MSC and its products that 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.

What Information Should I Have Available When I Contact MSC with a Technical Question?

When you contact MSC with a technical question, such as an error you encountered, you should have the following information available:

- The version level of MSC.Nastran being used (e.g., Version 70)
- Your computer platform (i.e., manufacturer, model, and operating system)
- Output files (.F06, .F04, and .LOG files)
- Your user documentation
- Description of the error encountered (if applicable)
- Description of your model

Handy Directories

Starting with Version 67.5, the /misc directory was added to the MSC.Nastran delivery tape. Refer to the *MSC.Nastran Configuration and Operations Guide* for the exact location of the directory. This directory contains information that you will find useful in your work, such as the Current Error List, expanded listings (and descriptions) of the diagnostic messages, useful alters, and examples. All files are in 80-column text format, and they can be copied and modified for use as you see fit.

This directory contains the following subdirectories:

Subdirectory	Contents
/sssalter	Miscellaneous alters, primarily for the Structured Solution Sequences (SSS). These alters provide new capabilities and error corrections. The alters were tested on several, but not all platforms, so you are cautioned about their use. Table 1 briefly describes these alters. Note that these alters may not be maintained for future versions of MSC.Nastran.

These alters are provided for special applications. They have not been subjected to extensive testing.

The alters in this directory are provided for illustrative purposes and have not been fully tested for all applications. Please exercise caution in their use.

These DMAP Alters were developed either in response to client requests or as prototypes to demonstrate the feasibility of proposed new features. Each of the DMAP Alters in the /misc/sssalter directory was tested for a number of possible applications, but they have not undergone the extensive testing required to be included in MSC.Nastran. It is possible that your application may use a path through the DMAP Alter that was not tested, so always review your results carefully. These DMAP Alters were developed independently and it is likely that a conflict will occur if more than one is used in a run. A separate readme file (xxxr.rdm) accompanies each alter.

The following table lists the DMAP Alters for Version 70 and provides a description for each alter.

Table 1 DMAP Alters

Alter Name	Purpose
addstata.v70	Adds a static solution onto a transient solution.
alter1ga.v70, alter2ga.v70, alter9ga.v70	Perform coupled load analysis.
appenda.v70	Appends eigenvalue solutions.
bra.v70	Estimates first buckling mode.
cda.v70	Simulates gap-contact type problem using a linear static analysis.
checka.v70	Performs rigid body checks, kinetic energy calculation, and modal effective weight calculation.
cova.v70	Computes the variance and standard deviation in response given the user-specified variance in design variables.
cygyroa.v70	Adds gyroscopic terms to cyclic symmetry analyses.

Table 1 DMAP Alters (continued)

Alter Name	Purpose
ddama.v70	Performs dynamic analysis of shipboard equipment using the dynamic design analysis method.
delmodea.v70	Allows the user to delete individual modes from the set of system (residual) level modes.
dmigtrna.v70	Transforms the dmig matrix to the basic coordinate system.
dtranra.v70	Allows “NOAP” restarts in the Structured Solution Sequences (SOL 109).
effmassa.v70	Calculates modal effective mass and participation factors.
evdsa.v70, evds103a.v70, evds200a.v70	Compute eigenvector derivatives in design sensitivity analysis.
gpftrna.v70	Calculates grid point force output in a transient analysis.
genela.v70	Calculates the forces resulting from a GENEL element in SOL 112.
glforcea.v70	Calculates superelement interface loads.
gridloca.v70	Transforms the location of grid points to the basic coordinate system.
irsa.v70	Incorporates dynamic reduction using the Improved Reduced System (IRS) method.
mfreqea.v70	Calculates the contribution of each mode to the strain energy of the solution.
mica.v70	Allows the use of initial conditions in a modal transient analysis.
mm44a.v70	Allows multiple mass configuration in a SOL 144 run.
modevala.v70	Determines whether the modes obtained are capable of representing the solution to a set of applied loadings.
mtacca.v70	Allows the use of modal transient analysis to solve for a structure which has nonzero initial accelerations.
mtranea.v70	Calculates the contribution of each mode to the strain energy and kinetic energy of the solution.
mtranra.v70	Allows “NOAP” restarts in the Structured Solution Sequences (SOL 112).
nlgyroa.v70	Adds gyroscopic terms to nonlinear analysis.
norma.v70	Normalizes the output eigenvectors to a maximum of 1.0.
oloadcda.v70	Transforms OLOAD resultant into user-defined coordinate system.
ortx7a.v70 ortx10a.v70	Scales complex eigenvector to unit generalized mass matrix.

Table 1 DMAP Alters (continued)

Alter Name	Purpose
pchdispa.v70	Creates DMIG entries containing the solution from a static run.
postmaca.v70	Computes the cross-orthogonality between test modes and modes of an MSC.Nastran model.
premaca.v70	Computes the cross-orthogonality between modes of a full model and modes of a reduced model.
prldbcka.v70	Includes the effects of preload in a linear buckling solution.
propa.v70, propf.for	Provides propeller aerodynamic coefficients and wing/nacelle aerodynamic interference loads for aircraft flutter analysis.
rflagb.v70	Provides an enforced motion capability and load-based initial condition using Lagrange Multiplier techniques. This alter also allows fluid-structure interface points to be placed in the superelement.
ridgyroa.v70	Adds gyroscopic terms to dynamic analysis using the structured solution sequences.
seaspa.v70	Applies aeroelastic splines to superelements.
sebloada.v70	Calculates dynamic superelement interface loads.
sedampa.v70	Includes modal damping on superelement component modes.
segyroa.v70	Adds gyroscopic terms to superelement analysis.
solchka.v70, dcchka.v70, dfchka.v70, dtchka.v70, mfchka.v70, mcchka.v70, mochka.v70, mtchka.v70, sechka.v70	Performs load epsilon checks in dynamic analysis similar to the one used in static analysis.
spc101a.v70	Separates the spc forces defined by the user versus those constrained by param,autospc.
tempmata.v70	Applies temperature-dependent material to higher-order solid linear elements in a nonlinear solution (SOL 106).
trncompa.v70	Calculates the ply-by-ply composite stresses and failure indices in a transient dynamic solution.
vmea.v70	Provides an efficient method of approximating the virtual fluid mass capability

Subdirectory	Contents	
/misc/doc	Miscellaneous documentation files:	
	error.lis	This is a text file that contains the errors corrected in this version, as well as the errors that are not corrected.

Which Solution Sequences Are no Longer Available?

MSC.Nastran solution sequences have been consolidated and restructured in Version 69. There are now only two sets of solution sequences: the basic solutions and the structured solutions.

The basic solutions (SOLs 1 through 16) are simple, easy-to-use solution sequences. These solution sequences do not contain any subDMAP calls, and they may be better suited for implementing user DMAP alters. These basic solutions, however, do not support superelements, the restart capability (i.e., no database is created), fluid elements, p-elements, or sensitivity calculation. The basic solution sequences are different from the pre-Version 69 rigid-format solution sequences. Any DMAP alters written for the rigid-format solutions will need to be updated for the basic solution sequences. All rigid-format RFALTER capabilities have been included in the basic solutions, and the RFALTER library no longer exists.

The structured solutions are more sophisticated solution sequences and support all advertised capabilities, including superelements, restarts, fluid elements, p-elements, and sensitivity calculation. These solution sequences contain many subDMAPs. You can use the string-based DMAP altering tool to simplify the maintenance of your DMAP alters. Please use the structured solution sequences since they are now the standard solution sequences.

The following rigid-format and unstructured solution sequences have been removed. If they are requested, MSC.Nastran will automatically switch to the equivalent basic or structured solution sequence.

Pre-V69 SOL Numbers	V69+ SOL Numbers	Analysis Type
24 or 38	1	Statics
25 or 39	3	Normal modes
64	4	Geometric nonlinear
5	5	Buckling
29	7	Direct complex modes
26	8	Direct frequency
27	9	Direct transient
29	10	Modal complex modes

Pre-V69 SOL Numbers	V69+ SOL Numbers	Analysis Type
30	11	Modal frequency
31	12	Modal transient
47	14	Cyclic statics
48	15	Cyclic modes
77	16	Cyclic buckling
78	118	Cyclic frequency response
21	144	Static aeroelastic response
61	101	Statics
62	101	Superelement statics
63	103	Normal modes
65	105	Buckling analysis, database
66	106	Nonlinear statics
67	107	Direct complex eigenvalue
68	108	Direct frequency
69	109	Nonlinear statics
70	110	Modal complex eigenvalue
71	111	Modal frequency-response analysis
72	112	Modal transient response
75	145	Aerodynamic flutter
76	146	Aerodynamic response
81	114	Cyclic statics
82	114	Cyclic statics
83	115	Cyclic modes
88	118	Cyclic frequency response
91	101	Statics with inertia relief
99	129	Nonlinear transient response

The following rigid-format solution sequences have been removed. Their capabilities exist in the structured solution sequences, but MSC.Nastran will not automatically switch to the equivalent solution.

V69 SOL Numbers	Analysis Type
51	Design sensitivity, statics; Use SOL 101
53	Design sensitivity, normal modes; Use SOL 103
55	Design sensitivity, buckling; Use SOL 105

The following rigid-format solution sequences have been removed and do not have equivalent basic or structured solution sequences.

V69 SOL Numbers	Analysis Type
2	Fast plots
41	Modal synthesis; flexibility method
42	Modal synthesis; hybrid method
43	Modal synthesis; stiffness method
60	Checkout

The structured solution sequence numbers remain the same as in previous versions.

What Are the Contents of the .F06, .F04, and .LOG Files?

By default, MSC.Nastran writes three text files during execution:

.F06 This file contains the results of the analysis, such as the displacements, stresses, etc., as well as any diagnostic messages. This file is called xxxx.f06 where xxxx is the filename of the analysis file (xxxx.dat) that was submitted.

.F04 This file contains the start and stop time for each module executed as well as the size of the database file (if any). This file is called xxxx.f04 where xxxx is the filename of the analysis file (xxxx.dat) that was submitted.

.LOG This file contains system information, such as the name of the computer you are running on as well as any system errors encountered. This file is called xxxx.log where xxxx is the filename of the analysis file (xxxx.dat) that was submitted.

For more information about these files, see the *MSC.Nastran Configuration and Operations Guide*.

These files are appended together if your site administrator modified the MSC.Nastran execution procedure to do so. Your site administrator may also have modified the procedure to delete the .F04 and .LOG files. If you are not getting separate .F06, .F04, and .LOG files, you should check with your site administrator.

These files can also be appended together by using the app keyword in the MSC.Nastran submittal command line. See the *MSC.Nastran Configuration and Operations Guide* for more information.

What Are the .MASTER, and .DBALL Files?

These are database files generated during the run. If the restart capability is to be used, these files must be saved. If the restart capability is not being used, these files may be deleted automatically at the end of the run. For certain types of unsuccessful runs these files are not deleted automatically, and they should be deleted by you manually. If these files are not deleted, their presence may prevent subsequent runs from being successful.

What MSC.Nastran Entities are Supported in MSC.Patran?

The entities that are supported in MSC.Patran depend on the version of MSC.Patran. See the specific version of the *MSC.Patran Preference Guide for MSC.Nastran* for further details.

What Is the .XDB File?

The .XDB file contains the MSC.Nastran results for use in some postprocessors, including MSC.Nastran for Windows, MSC/ARIES, and MSC/XL. This file is created only when PARAM,POST,0 is included in the input file. The .XDB file can be deleted if you are not using MSC.Nastran for Windows, MSC/ARIES, or another postprocessor that reads it.

Where Do I Begin on an Analysis Capability That I Have Not Used Before?

The sources of documentation listed on pages six through nine provide the best place to begin learning about a particular analysis capability. You can start with the appropriate user's guide that best fits your particular topic of interest.

The TPL (Test Problem Library), supplied on the delivery media for each system, contains numerous example problems. The file tplidx.dat is an index to the TPL examples. The exact location of the TPL is documented in the *Configuration and Operations Guide.* If you cannot locate the TPL, contact your site administrator.

A word of caution: When using a new analysis capability for the first time, use small models to help you become familiar with the capability. Small models can help you understand the fundamentals without creating an unnecessary burden in generating or debugging the model.

What Is the .OP2 File?

The .OP2 file contains the MSC.Nastran results for use in some postprocessors, including MSC.Patran, SDRC / I-DEAS, MSC_NF interface by LMS International, Dynamic Design Solutions / Fem Tools, and EDS / Unigraphics. For further details, refer to the *MSC.Nastran Quick Reference Guide.*

Diagnostic Messages

What Does “User Warning Message 6080 Timing Constants Not Found” Mean?

MSC.Nastran uses timing constants to estimate runtimes to select the most efficient approach for potentially long numerical operations. Timing constants reflect each computer’s speed for basic numerical operations. MSC.Nastran runs most efficiently if timing constants are available for your specific machine model. If these constants are not available, MSC.Nastran uses its default constants (in which case excessive run times can result). For a more detailed description on the timing constants and how to generate them, see the *MSC.Nastran Configuration and Operations Guide.*

MSC attempts to include timing constants in the delivery database for every machine model for which MSC has access at the time a new version is released.

What Does “User Fatal Message 3060 ... Option xxx Not in Approved List” Mean?

MSC.Nastran uses an authorization file to enable you to run MSC.Nastran and utilize some of its optional capabilities. UFM 3060 is issued if you invoke an option that is not authorized for the particular computer you are running on. Contact your MSC representative for a list of options available for your site.

This message can occur if the content of the authorization file is incorrect (wrong authorization code, numbered dataset, etc.). This message can also occur if the authorization file itself is correct, but it has the wrong privileges. The authorization file must have read privileges.

What Does “System Fatal Message 4276 Error Code xxx” Mean?

This error occurs when MSC.Nastran encounters errors that it does not otherwise catch (including system errors). Each error code has a different meaning; for example, Error Code 923 means that your system resources (disk space or file size limit) were exceeded. Further information is usually found in the .LOG file for that run. In addition, check the Current Error List because in many cases these are errors that should have been caught (and a better error message issued). If that does not help solve the problem, then increasing the memory, BUFFSIZE, or disk space will avoid the problem in many cases.

Further information can be found in the *MSC.Nastran Reference Manual.*

Elements

What Are the Accuracy Checks for Static Stress Analysis?

MSC.Nastran makes two kinds of checks on the accuracy of your static stress solution: (1) geometry checks for element shapes and connections, which are made prior to the analysis; and (2) stress discontinuity checks, which are made after the analysis. Note that each of these checks is important only in regions of large stresses or high stress gradients; for constant and/or low stresses, the element inaccuracies are relatively unimportant. A graphical pre- and postprocessor, such as MSC.Patran, can also make the geometric checks and display the results.

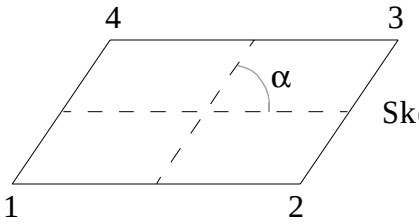
Accuracy checks for the CQUAD4 and CHEXA elements are described below.

Element Geometry Checks for CQUAD4. MSC.Nastran makes two element geometry checks for the CQUAD4 elements. The CQUAD4 elements are checked for skew and taper, both of which represent the amount that the CQUAD4 deviates from being a rectangle (see **Figure 2**). Excessive skew and taper can potentially give rise to inaccurate stress answers since the CQUAD4 accuracy degrades as the element deviates from being a rectangle. The status of the skew and taper is printed in User Information Message 5491.

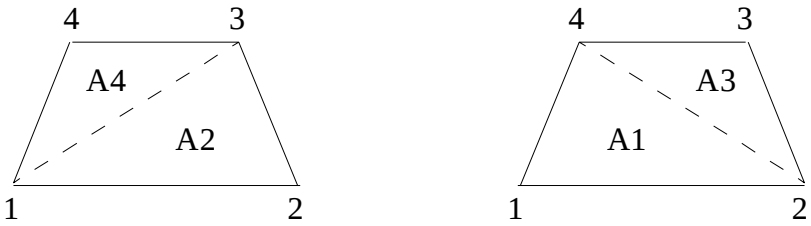
Skew. Skew is the angle between the lines that join opposite midsides. When this angle is less than 30 degrees, the message is issued. For no skew, this angle is 90 degrees.

Taper. Taper is the ratio of the areas on the two sides of a diagonal; if the ratio is greater than three, then the taper test fails. Taper is computed as follows: connect opposite grid points and compute the area of the enclosed triangles. Let Ji be 1/2 of the area associated with grid point i and the adjacent grid points of the enclosed triangle. Let Ja be 1/4 of (J1+J2+J3+J4). If the absolute value of (Ji-Ja)/Ja exceeds 0.5, then UIM 5491 is issued. Note that for no taper the ratio of (Ji-Ja)/Ja is 0.

Aspect Ratio and Warp. The accuracy of the CQUAD4 element depends primarily on the amount of skew and taper. Two other factors can also affect accuracy, although not to the same degree; these are aspect ratio and warp. Aspect ratio is the ratio of the longest side to its adjacent side. Warp is the amount that the element deviates from being planar. Aspect ratio and warp are not checked in MSC.Nastran.



Skew message is issued if $\alpha < 30°$



$$Ji = \frac{1}{2}Ai \qquad Ja = \frac{1}{4}(J1 + J2 + J3 + J4)$$

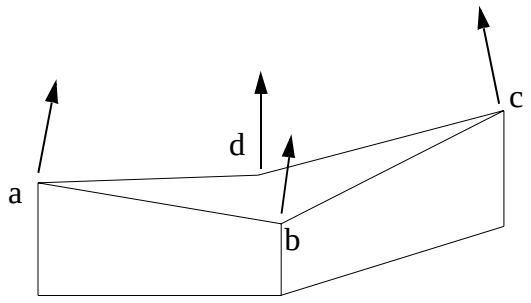
Taper message is issued if $\left|\frac{Ji - Ja}{Ja}\right| > 0.5$

Figure 2 CQUAD4 – Taper and Skew

Element Geometry Checks for CHEXA Elements. MSC.Nastran makes two element geometry checks for the CHEXA elements: aspect ratio and face warping. These checks are made because the solution accuracy for nonconstant stress can degrade if the element geometry for CHEXAs is not like a cube.

Aspect Ratio. The aspect ratio is the ratio of any two sides. If this ratio is greater than 100, indicating a very elongated CHEXA element, then UIM 4655 is issued for that element.

Warping. Warping indicates that a face of a CHEXA is not planar. This warping is indicated by UIM 4656, which indicates that a face is considerably out of plane. This message is issued if the corner normals at the opposite corners of a face deviate from each other by more than 45 degrees (see **Figure 3**). The midside nodes are neglected in the calculation.



↑ = corner normal (e.g., normal at a is normal to the plane dab)

"Out of Plane" message is issued if the angle between b and d > 45°

"Out of Plane" message is issued if the angle between a and c > 45°

Figure 3 CHEXA – Warping

Element Geometry Checks for CTETRA Elements. MSC.Nastran makes two element geometry checks for the CTETRA elements: aspect ratio and distortion. The aspect ratio measures the ratio of the longest edge to the shortest altitude for the CTETRA element. The distortion is a measure of how much the midside node deviates from the line joining the two corner nodes. If the element is highly distorted, the job may fail with a Jacobian error.

Stress Discontinuity Checks. Stress discontinuities represent the stress "jumps" across an element or at grid points and can be used as an approximate indication of the stress errors resulting from the coarseness of the finite element mesh. The regions with large stress jumps are candidates for refinement in order to reduce the magnitude of the discontinuity. Stress discontinuity checks were introduced in Version 67 and apply to plate and solid elements.

Stress discontinuities are computed after the static analysis as a postprocessing step after the grid point stress calculations. Further information can be found in the *MSC.Nastran Linear Analysis Static User's Guide*.

What Grid Point Output (Stresses, Strains, and Forces) Are Available for the CQUAD4 Elements?

Beginning with Version 67.5, corner output is directly available for CQUAD4 elements. Corner output is selected by using a corner output option with the STRESS, STRAIN, and FORCE Case Control commands. When one of these options is selected, output is computed at the center and four corners for each CQUAD4 element in a format similar to that of the CQUAD8 and CQUADR elements. For more information, see the *MSC.Nastran Release Notes* for Version 67.5.

There are four corner output options available as follows: CORNER, CUBIC, SGAGE, and BILIN. The different options provide for different approaches to the stress calculations. The BILIN option is the equivalent of CORNER, and is the default method. Based on experience, it was determined that the BILIN option produces better results for a wider range of problems. If corner stress output is desired, place the STRESS (CORNER) output request above the first subcase.

Note that corner stress output is available indirectly in all versions via the grid point stress capability.

Describe the Orientation Vector of the CBAR Elements

Purpose of a CBAR Element Coordinate System and Orientation Vector. I have just completed modeling part of the structure with a CBAR element. The next step is to consider the following questions:

- How does MSC.Nastran know where I want to place this CBAR element relative to the rest of the structure?

  This information is provided by the two grids that are specified on the CBAR entry along with the offsets from the grid points if any.

- Is this sufficient information to describe the CBAR element?

If the portion of the real structure represented by this CBAR element is capable of carrying only axial and torsional loads, then connecting this CBAR element to the two end points along with providing the appropriate cross-sectional area (A), torsional stiffness (J) and material property (MATi), in theory, sufficiently describes the contribution of this CBAR element to the overall structural stiffnesses. In MSC.Nastran, additional cross-sectional properties may be needed.

- This portion of the structure can carry a bending moment. What else do I need to provide to MSC.Nastran?

  If a CBAR element is also capable of carrying a bending moment, then the area moment of inertias, which are functions of the CBAR cross section, must also be provided. For an arbitrary cross section, you can define three inertia values IA, IB, and IAB about the CBAR's neutral axis.

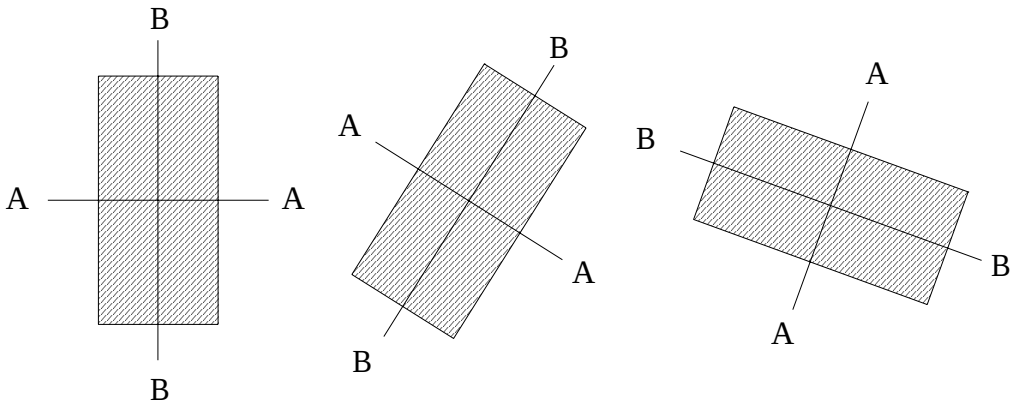


Figure 4 CBAR Element Cross Section

- Now that I have my cross-sectional properties, how does MSC.Nastran know which one of the **Figure 4** orientations to choose?

  The answer to this question is the element coordinate system. Once the element coordinate system is defined, the cross section is uniquely positioned relative to the overall structure. In MSC.Nastran, the way to construct the element coordinate system is by defining an orientation vector. Once the element coordinate system is defined, the inertia properties can then be uniquely specified relative to this coordinate system. Furthermore, this element (CBAR) coordinate system is used for the stress and force output.

This similar type of information is needed for all three-dimensional finite element codes. This is not unique to MSC.Nastran**.** The actual user interface may differ, but the idea is similar. Since it was explained why a CBAR element coordinate system and orientation vector are needed, the procedure for defining them in MSC.Nastran can now be described.

Definition of a CBAR Element Coordinate System and Orientation Vector. The following steps can be followed systematically to completely define the MSC.Nastran CBAR element.

1. Define the CBAR location.

This is accomplished by defining the two connecting end points (end A and end B). This information is provided by the GA and GB fields on the CBAR entry. End A and end B can, however, be offset from grid point A and grid point B, respectively. These components of offset vectors are measured in the displacement coordinate system of the associated grid points, and they are specified in the WA and WB fields on the CBAR entry.

2. Construct the element x-axis (x_{elem}).

   The element x-axis or neutral bending axis is defined by connecting a straight line from end A to end B of the CBAR element (including the effects of offsets). The positive direction starts at end A, and is directed toward end B.

3. Construct the CBAR orientation vector.

   The CBAR orientation vector can be constructed by specifying components of vector v in the X1, X2, and X3 fields on the CBAR entry (see **Figure 5**). These components are measured from end A in the displacement coordinate system of end A. The element x-axis along with the vector $\vec{v}$ determine a unique xv plane. This plane is also known as plane 1 in MSC.Nastran terminology.

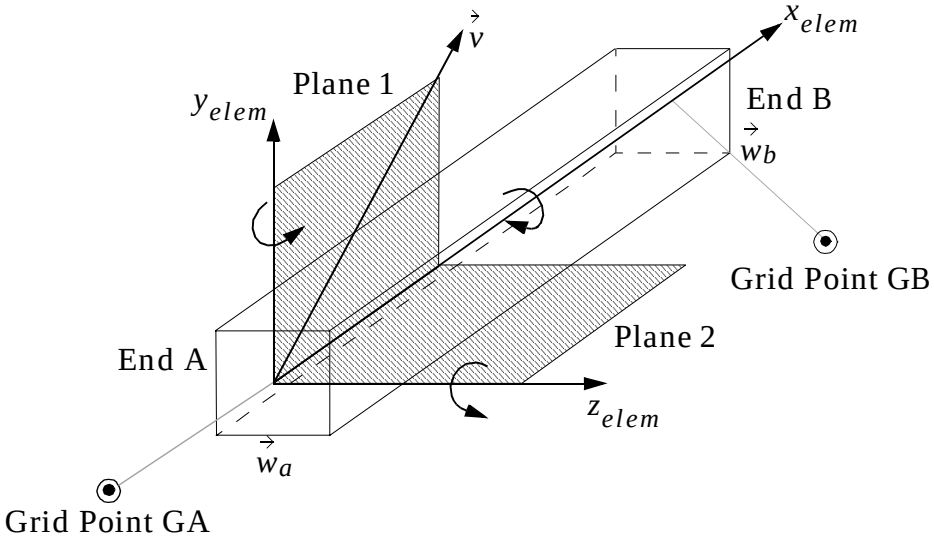
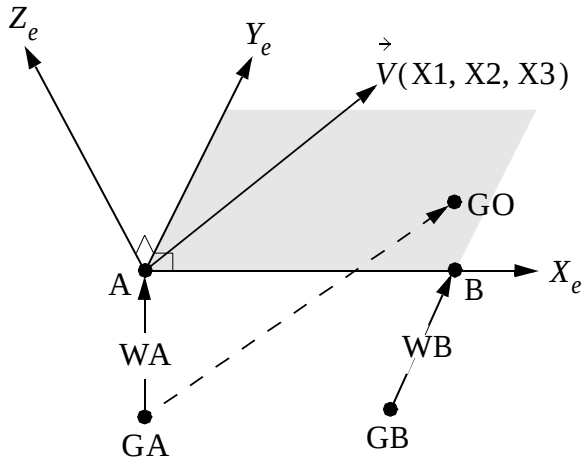


Figure 5 CBAR Element Coordinate System

The orientation vector can also be defined by introducing a grid ID in field 6 of the CBAR entry (GO) instead of the components X1, X2, and X3. However, in this case, the origin of the orientation vector is at grid point A (GA in **Figure 6**) instead of end A.



Note that the origin of $\vec{V}$ is at offset point A when $\vec{V}$ is defined by components (X1, X2, X3). The origin of $\vec{V}$ is at GA when $\vec{V}$ is defined using GO.

Figure 6 CBAR Element Orientation Vector (Alternate Form)

4. Define the element y-axis (Y_{elem}).

   The element y-axis is perpendicular to the element x-axis and lies in the xv plane with the positive direction lying in the same quadrant as the positive v-axis.

5. Define the element z-axis (Z_{elem}).

   Finally, the element z-axis is obtained by using the right-hand rule.

6. Define plane 1, plane 2, I1, I2, I_{yy} and I_{zz}.

   The xy plane formed by the element x- and y-axes is known as plane 1 (see **Figure 7**). The xz plane formed by the element x- and z-axes is known as plane 2. I1 on the PBAR entry is the inertia property that resists the bending moment in plane 1, i.e., the moment of inertia about the element z-axis, which is I_{zz}. I2 on the PBAR entry is the inertia property that resists the bending moment in plane 2, i.e., the moment of inertia about the element y-axis, which is I_{yy}.

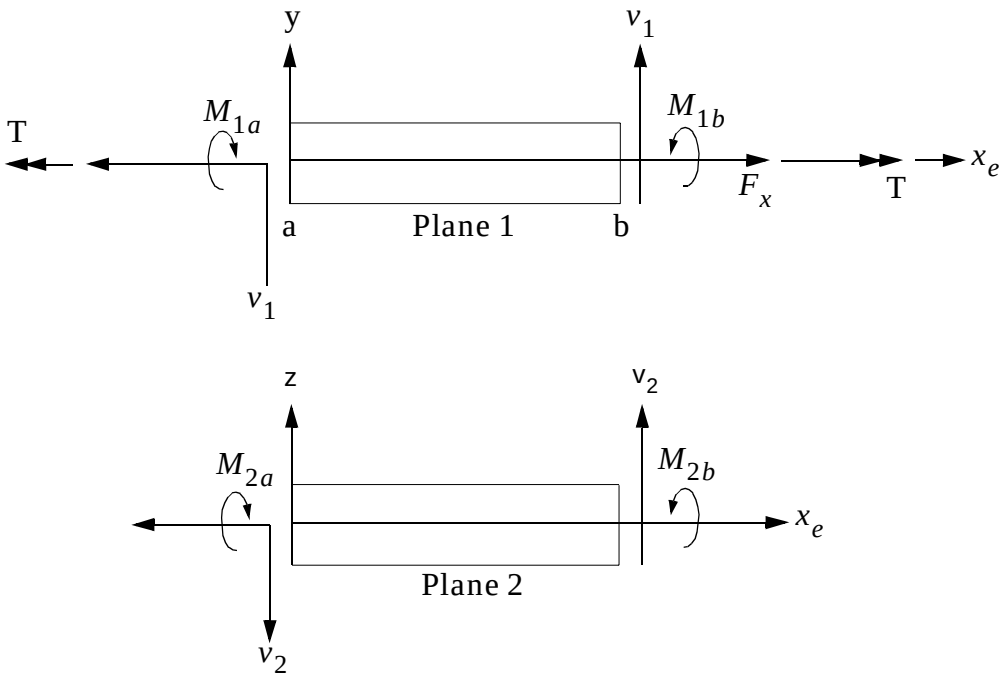


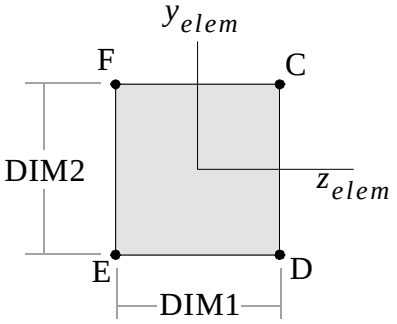
Figure 7 CBAR Element Forces in Plane 1 (top) and Plane 2 (bottom)

The CBAR element is now completely defined.

Do I Have to Calculate the Cross-Sectional Properties for My CBAR and CBEAM Elements?

Prior to Version 69, you must provide the cross-sectional properties of the beam (such as area, moments of inertia, shear center, etc.). Although this is not a particularly difficult task for standard cross sections, it is tedious and and prone to unnecessary input errors.

A new beam cross-section library was introduced in Version 69 to provide you a simple interface to input beam cross sections into MSC.Nastran. A number of common cross-section types such as bar, box, I-beam, channel, angle, etc. are described within the Bulk Data Section by the cross section's dimensions instead of the section properties. For example, the rectangular cross section shown in **Figure 7** may be defined by its height and depth (DIM1 and DIM2) rather than the area, moments of inertia, etc.



TYPE="BAR"

Figure 8 Cross Section Definition

The PBARL Bulk Data entry allows you to input cross section types along with their characteristic dimensions. You can choose from 18 different cross section shapes. These shapes are as follows: ROD, TUBE, I, CHAN (channel), T, BOX, BAR (rectangle), CROSS, H, T1, I1, CHAN1, Z, CHAN2, T2, BOX1, HEXA (hexagon), and HAT (hat section).

For some of these shapes (I, CHAN, T, and BOX), you can also select different orientations. All the 18 different shapes and orientations are shown in the *MSC.Nastran Quick Reference Guide*.

To define section attributes such as height and width on the DIMi fields, use the PBARL entry. To define section properties such as area and moment of inertia, use the PBAR Bulk Data entry. The new PBARL entries are easier to use and still retain most of the capabilities of the existing PBAR method, including non-structural mass.

An additional difference between the PBARL and the PBAR entries is that stress recovery points need not be specified to obtain stress output for the PBARL entry. The stress recovery points are automatically calculated at the locations shown in **Figure 8** to give the maximum stress for the cross section.

A similar feature is also available for the CBEAM element. To define section attributes such as height and width on the DIMi fields, use the PBEAML entry. To define section properties such as area and moment of inertia, use the PBEAM entry.

For unsymmetric sections—where the shear center and neutral axis do not coincide (e.g., a channel section)—use the CBEAM element instead of the CBAR element. See section on "Describe the Major Difference Between the CBEAM and CBAR Element" for further details.

In addition to the standard sections provided by MSC, you can add your own library of cross sections to suit your needs. To add your own library, write a few simple subroutines in FORTRAN to interface with MSC.Nastran through Inter-Process Communications. See *MSC.Nastran Release Guide for V69* for further details regarding the creation of your own library of cross sections.

This automatic cross section computation greatly simplifies the formulation of design variables for design optimization applications.

Describe the Major Differences Between the CBEAM and CBAR Elements

The CBEAM (**Figure 9** and **Figure 10**) and CBAR elements are described in detail in the *MSC.Nastran Quick Reference Guide*. This discussion is intended to point out some of the major differences between the CBEAM and CBAR elements. The *MSC.Nastran Quick Reference Guide* should be used as a reference in conjunction with this section.

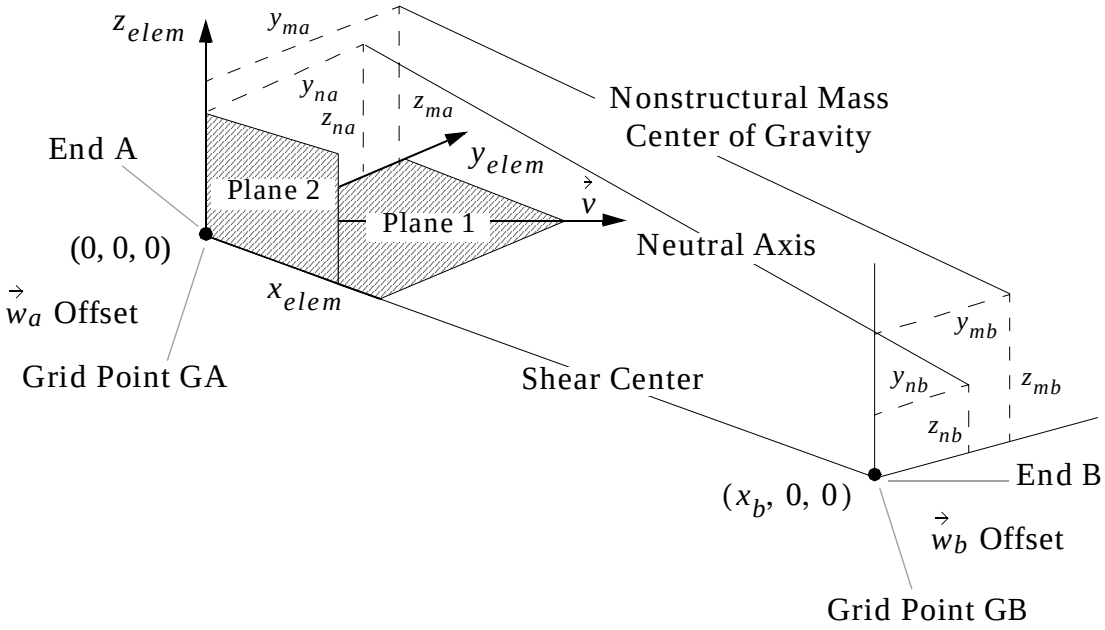


Figure 9 CBEAM Element Coordinate System

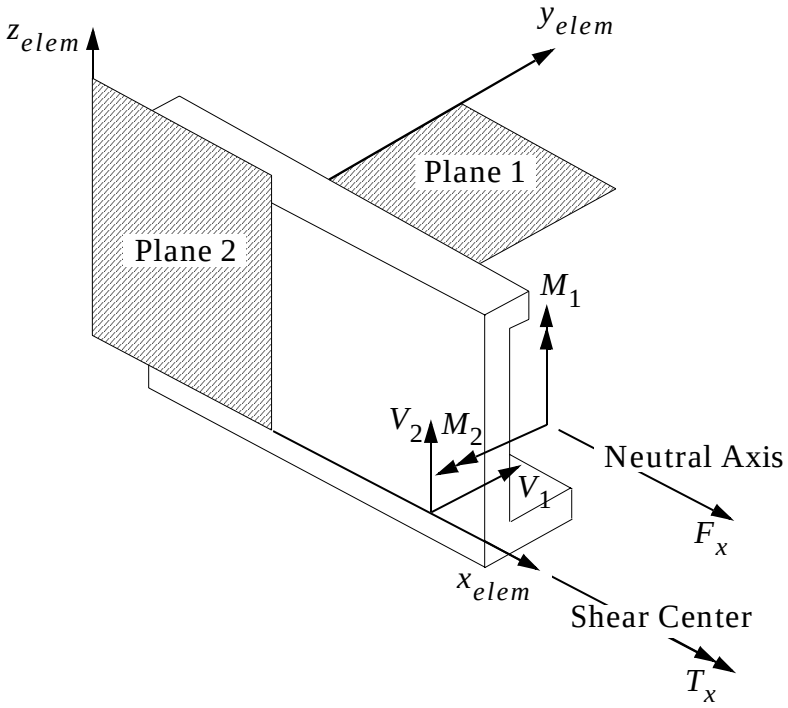


Figure 10 CBEAM Element Forces

The CBEAM element in MSC.Nastran has all the capabilities of the CBAR element plus the following additional/different features:

- Tapered cross-sectional properties are allowed. Different cross-sectional properties may be specified at the two end points plus nine additional interior points. Unlike the CBAR element, the area, moment of inertia in plane 1, and moment of inertia in plane 2 at the end point (parent entry) are required input for the CBEAM element. The CBAR element may have one or several of these properties equal to zero as long as one of them is nonzero.
- The neutral axis and shear center do not have to coincide.

- The CBEAM element can account for the effect of cross-sectional warping on the torsional stiffness. (See Timoshenko and Gere, *Theory of Elastic Stability*, McGraw Hill Book Company, 1961.)
- The CBEAM element can account for shear relief due to the effect of taper on the shear relief stiffness.
- Nonstructural mass center of gravity can be offset from the shear center.
- The CBEAM element has nonzero rotational mass moment of inertia about its neutral axis.
- The CBEAM element is based on a flexibility formulation. The element stiffness matrix is then generated by inverting the flexibility matrix. For this reason I1 and I2 may not be zero.
- The default values for the transverse shear stiffness factors (K1,K2) are 1.0. For the CBAR element, the defaults are set to infinity; in other words, the deflection due to shear is ignored if the default values are used for the CBAR element. The deflection due to shear flexibility is negligible for a long slender CBAR/CBEAM element.

The basic concept of specifying the element coordinate system, orientation vector v, offsets, and pin flags is similar to those for the CBAR element. One major difference is that the element x-axis, which connects end A to end B of the CBEAM, lies on the shear center axis rather than the neutral axis. The stress data recovery points Ci, Di, Ei, and Fi are defined relative to the shear center axis rather than the neutral axis. In the case of the CBAR element, the neutral axis and shear center axis are the same.

For a CBEAM element that has a constant symmetrical closed cross section, shear center axis coinciding with the neutral axis, and zero nonstructural mass, the input to the PBEAM entry looks very similar to a PBAR entry.

1	2	3	4	5	6	7	8	9	10
PBEAM	PID	MID	A	I1	I2		J		

What Is the Transverse Shear Stiffness in Beams?

For certain shapes of beams (I-beams, for example), and for beams that are deep relative to their length, the deformation due to shear can be significant. In MSC.Nastran, deformation due to shear is activated by setting K1 and K2, the area factors for shear, to some value on the PBAR or PBEAM entries. K1 is the area factor in plane 1, and K2 is the area factor in plane 2. For the CBAR element, default values for K1 and K2 are infinity, meaning that deformation due to shear is neglected. For the CBEAM element, default values are 1.0.

Transverse shear stiffness is given by $K \cdot A \cdot G$ where A is the cross-sectional area and G is the shear modulus. Transverse shear flexibility, a term often used in the literature, is $1/(K \cdot A \cdot G)$.

The area factor depends on the shape of the beam cross section (see Table). The area factor derivation is described in Roark and Young, *Formulas for Stress and Strain*, 5th edition, and Timoshenko and Gere, *Theory of Elastic Stability*, 2nd edition. Note that the area factor defined in the above references is the reciprocal of that defined in MSC.Nastran.

Table 2 Shape Factors for Various Cross Sections

Shape of Cross Section	Recommended Value of K in MSC.Nastran
Rectangular	5/6
Circular	9/10
Thin-wall hollow circular	1/2
Wide Flange Beams:	
Minor axis	$A_f / 1.2A$ (approximately)
Major axis	A_w / A (approximately)

A_f = the area of the flange

A_w = the area of the web

A = the area of the cross section

When Should Midside Nodes Be Used?

If the CQUAD8 element is used, the general recommendation is to include all midside nodes. A CQUAD8 with all midside nodes deleted is excessively stiff and therefore is inferior to a CQUAD4. If midside nodes are desired for a CHEXA element, the general recommendation is to include all of them. The midside nodes should be located as close to the center of the edge as possible for either element.

Deleting midside nodes is not a recommended method for mesh transitions from higher order elements to lower order elements. Other more preferred methods for transition include the use of an R-type element (MPC, RBE2, RBE3, etc.).

However, the transition areas, as always, should be away from the areas of interest. The local stresses in the area of transition may be in error. A good quick check to see if the transition is acceptable is to look at a stress contour plot. A small stress gradient (a "smooth" stress variation) may indicate that the transition is acceptable, whereas a large stress gradient may indicate that it is not. For more details, refer to the question "How Should I Model a Mesh Transition?"

What Value Should I Use for K6ROT?

Other than the CQUADR and CTRIAR elements, MSC.Nastran's plate elements do not contain rotational stiffness normal to the plane of the elements. The K6ROT parameter adds a small fictitious R3 stiffness about the axis normal to the plate element (CQUAD4

and CTRIA3 only). MSC.Nastran internally inspects the stiffness of the structure and arrives at an appropriate, small fictitious value. This stiffness is then multiplied with the value provided on PARAM,K6ROT,xx to arrive at the final R3 stiffness value. For linear solutions (all solution sequences except SOLs 106 and 129), the default value for K6ROT is equal to zero; in other words, no fictitious R3 stiffness value is added. In most instances, the default value should be used.

If K6ROT > 0.0, then a small stiffness is added to the normal rotation (R3) for the CQUAD4 and CTRIA3 elements. This is an alternate way to suppress grid point singularities and is intended primarily for geometric nonlinear analysis. For linear analysis, if two adjacent plates are slightly non-coplanar (see **Figure 11**), then the stiffness at a particular degree of freedom may be small enough that PARAM,AUTOSPC constrains this degree of freedom. Due to this action, it may yield "unacceptable" constraint values, or the model may not pass a rigid body check. In this instance, a K6ROT value of 1.0 can be used to alleviate this problem.

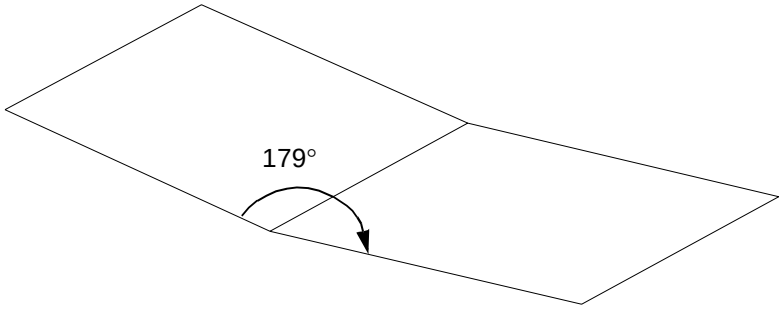


Figure 11 Two Plates Joined Together That Are Slightly Non-Coplanar

For nonlinear analysis, the default value of K6ROT is equal to 100. In most instances, the default value can be used. For highly nonlinear problems, a K6ROT value of 1.E5 has been used successfully. In all instances, K6ROT is applicable only for the CQUAD4 and CTRIA3 elements. For linear analysis beginning with Version 68.2, the use of shell normal (PARAM,SNORM) is recommended instead of PARAM,K6ROT. See the section on "*Shell Normals—A Better Alternative for K6ROT*."

Shell Normals—A Better Alternative for K6ROT in Linear Analysis

By default, the direction of the normal rotation vector for flat plate elements is assumed to be perpendicular to the plane of each element. If the model is curved, the shell bending and twist moments must change direction at the element intersection. If transverse shear flexibility is present, the deformations may be too large. (Because elements using low-order formulations ignore the edge effect, this rarely causes any problems-a large value of the parameter K6ROT partially cures the problem.) With the unique normal (SNORM) option, the rotational degrees of freedom at each corner of an element are measured relative to the specified normal vector direction. Thus, all elements connected to a grid point will use a consistent direction for defining shell bending and twisting moments.

In CQUAD4 and CTRIA3 elements, the stiffness matrices of the elements are modified to eliminate the undesirable small stiffness in the rotational motions about the shell normal vector. In effect, the transformation replaces the normal moments with in-plane forces. No changes were made to the basic element stiffness matrix, and therefore flat plate models will not be affected. The objective of the new transformation was to remove a potential weakness in curved shell models and allow the automatic constraint process to remove the true singularity in the assembled stiffness matrix.

The CQUADR and CTRIAR elements are also improved for some types of shell problems. Because of their extra degrees of freedom, these elements are more sensitive to the coupling between in-plane and out-of-plane motion due to curvature. The new formulation provides more consistency between adjacent elements in a curved shell.

Shell normals are available for CQUAD4, CQUADR, CTRIA3, and CTRIAR elements. Normals are activated with the SNORM parameter; for example:

```
PARAM,SNORM,20.0
```

where 20.0 is an angle in degree. A shell normal defines a unique direction for the rotational degrees of freedom of all adjacent elements. A shell normal vector is created by averaging the normal vectors of the attached elements. In this example, shell normals are used if the actual angle, β, between the local element normal and the unique grid point normal is less than $20.0°$. (See **Figure 12**.) A value of SNORM = 20.0 is recommended. The default is SNORM = 0.0, which means that shell normals are turned off. If p-elements exist, the default is 20.0. If the angle β is greater than the value defined on PARAM,SNORM, the edge is assumed to be a corner, and the old method is used. Shell normals improve the accuracy of the results in curved shells where in-plane shear and twisting moments act together.

In linear solution sequences, the values of “param,k6rot,0.” and “param,snorm,20.” are recommended. Transverse shear flexibility (MID3 on the PSHELL property entry) should be left on when normals are used.

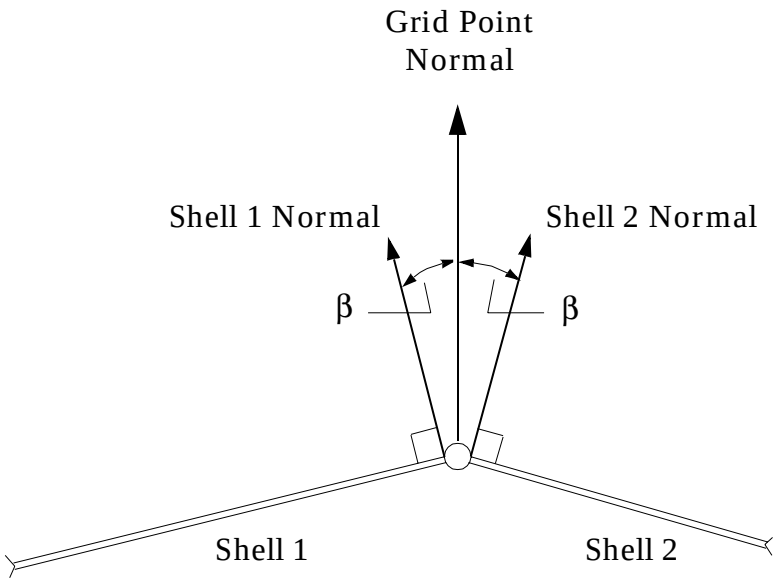


Figure 12 Unique Grid Point Normal

What Is MID4 on the PSHELL Entry?

The MID4 field on the PSHELL entry is used to input a coupling relationship between the in-plane forces and bending moments. This coupling only occurs in plates that are nonsymmetric about the neutral plane or where the neutral plane is offset from the grid points. Typical applications include reinforced skins and aluminum bonded to fiberglass.

In general the material properties can be determined using the equation:

$$G_{mid4} = -\frac{1}{T^2}\int_{-T/2}^{T/2} Gzdz$$

As an example, consider the simple two layer composites as shown in **Figure 13**. In this example, the grid points are located on the midplane of the plate (z = 0).

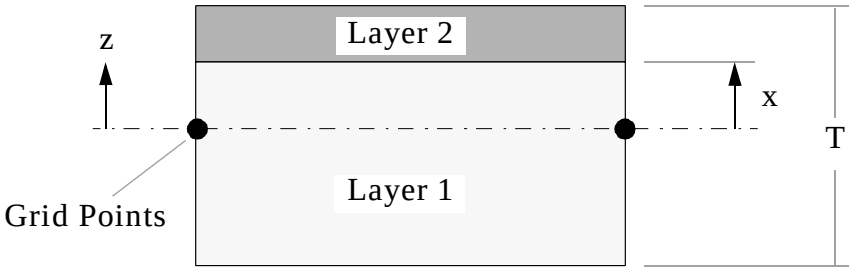


Figure 13 Simple Two-Layer Composite

As an example for simple two-layer composites as shown in **Figure 13**, the properties for MID4 are computed by integrating the above equation to give

$$G_{mid4} = \frac{(G1 - G2)}{8}\left[1 - 4\left(\frac{X}{T}\right)^2\right]$$

where:

G1, G2 = material properties for layers 1 and 2, respectively

T = total thickness of the composite

X = distance from the midplane to the layer intersection

For more complicated composite lay-ups (three layers or more) or when interlaminar stresses are needed, the equations become more complicated and the PCOMP and MAT8 entries can be used to automatically generate equivalent PSHELL and MAT2 entries. For additional information, see the *MSC.Nastran Quick Reference Guide*.

How Do I Interpret Scalar Element Force and Stress Output?

In 1-D elements such as the CBAR, a positive axial force or stress indicates that the member is in tension. This is not the case for scalar elements. Since the scalar elements do not have any geometry, the sign of the output force or stress does not necessarily indicate compression or tension in the spring. The force in the spring is recovered as follows:

$$F = K(U_1 - U_2)$$

Simply by reversing the G1 and G2 IDs on the CELAS1 and CELAS2 entries or the S1 and S2 on CELAS3 and CELAS4 entries, the sign of the force and stress output for these elements reverse. For example, consider the spring shown in Figure where D1 and D2 are scalar displacements.

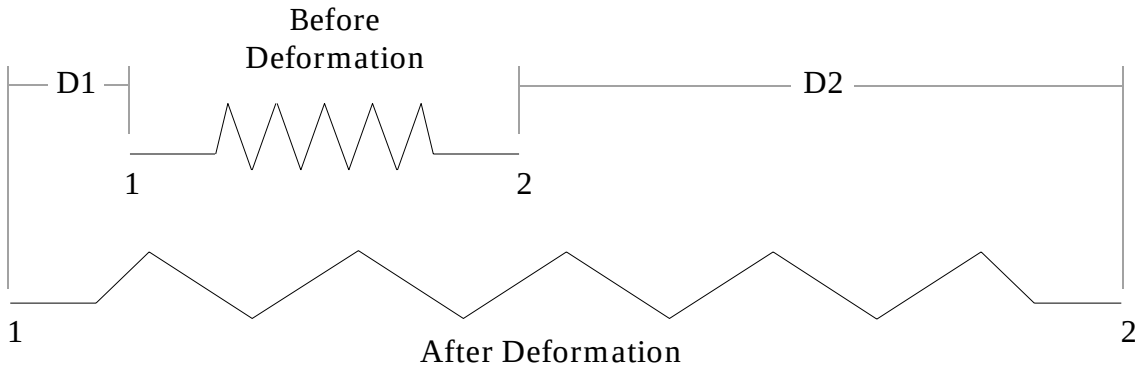


Figure 14 Spring – Before and After Deformation

For D2 > D1,

CELAS2, 1, 1., 1, 0, 2, 0, , 5.	Produces a negative axial force
CELAS2, 1, 1., 2, 0, 1, 0, , 5.	Produces a positive axial force

Use caution with the sign when using any of the scalar elements.

Why Should I Use Coincident Grid Points for Scalar Elements?

Whenever a scalar element is connected across two grid points, it is good modeling practice to make sure that the grid points are coincident and defined in the same coordinate system. If the grid points are not coincident, a difference in displacement may be caused by a rigid body rotation. This situation causes forces in the spring and a net moment on the structure. This condition is an example of an internal constraint, which is allowed because it is valid for some types of modeling. Unless it is desired to have these internal constraints, use coincident grid points. Since the length of the spring is zero when coincident grid points are used, no internal moments are created. Also, the scalar element's degrees of freedom are in the global (also known as the output or displacement) coordinate system.

CBUSH Elements—A Better Alternative for the CELASi Elements

As discussed in the Section "*Why Should I Use Coincident Grid Points for Scalar Elements?*," if CELASi elements are used and the geometry is not aligned properly, internal constraints may be induced. The CBUSH elements—which are used primarily as vibration control devices that have impedance values (stiffness and dampling) that are frequency dependent—can be used as replacement for the CELASi elements.

The CBUSH element contains all the features of the CELASi elements plus avoiding the potential internal constraint problem. Furthermore, several CELASi elements can be reduced to a single CBUSH element. For example, the following six CELAS2s

```
CELAS2, 1,     1.+6,     111000, 1,     210000, 1
CELAS2, 2,     1.+6,     111000, 2,     210000, 2
CELAS2, 3,     1.+6,     111000, 3,     210000, 3
CELAS2, 4,     1.+6,     111000, 4,     210000, 4
CELAS2, 5,     1.+6,     111000, 5,     210000, 5
CELAS2, 6,     1.+6,     111000, 6,     210000, 6
```

can be reduced to the CBUSH element as shown below.

```
CBUSH,  10000, 1,        111000, 210000, ,     ,     ,     0
PBUSH,  1,     K         1.+6,   1.+6    1.+6  1.+6  1.+6  1.+6
```

In order to connect noncoincident grid points properly using the CELASi elements, it is necessary to account for the offsets between the grid points. The most practical method may be to define coordinate systems, which align with a line between each pair of grid points, and then input the CELASi elements along these coordinate systems, which is a tedious, error-prone task. If the grid points are located in non-Cartesian systems or several Cartesian coordinate systems, the task is even more tedious and error prone. The CBUSH element alleviates this problem completely by internally generating rigid elements to a common location. Furthermore, if non coincident points are joined by CELASi elements, internal constraints that are generated can be difficult to diagnose since they cannot be plotted. Plots for the CBUSH elements can be requested with the nastran structure plots.

Evidence of internal constraints are apparent in static analysis when the OLOAD and SPCFORCE resultants are not in balance. The analysis does not isolate the elements with internal constraints; instead it merely states that some internal constraints must exist. When there are no internal constraints, the constraint resultants should be equal and opposite to the oload resultants. Any unbalance is due to internal constraints.

Internal constraints provide plausible results for some loading conditions but may provide unexpected results for other loadings or eigenvectors. There is no direct evidence of internal constraints in element forces, grid point forces, SPC forces or other commonly used output data. The internal constraints are hidden SPC moments, which do not appear in SPC force output.

What Are the Restrictions on the Use of the CQUADR and CTRIAR Elements?

The CQUADR and CTRIAR elements use a fictitious stiffness in the sixth degree of freedom for the in-plane rotational stiffness to improve membrane behavior. These elements are better than the CQUAD4 and CTRIA3 elements for modeling planar structures under in-plane loading (i.e., membrane behavior)—especially if the elements deviate substantially from being rectangular.

Restrictions on their use include:

- Use only with CQUADR or CTRIAR elements. Do not use in conjunction with any other elements.
- No membrane-bending coupling.
- Offsets (ZOFFS) are not supported.
- No differential stiffness

- Use only for linear models (i.e., nonlinear and buckling are not supported).
- No consistent load vector.
- No heat transfer capability.

The output for the CQUADR element is output at the center and corners. The membrane corner results are computed and printed; bending results are computed only at the center, and it is this value that is printed at each corner. Note that the corner output for CQUAD4 elements does not affect the CQUADR elements.

Describe the Rigid Elements

The name rigid element is often misleading. The R-type elements include both rigid and interpolation elements. Rigid elements consist of RBAR, RBE1, RBE2, RROD, and RTRPLT. Of these elements, the RBAR and RBE2 are the most commonly used. Interpolation elements consist of the RBE3 and RSPLINE. The RBE3 is a linear interpolation element often used to distribute either loading or mass, and the RSPLINE is an interpolation element that uses the beam equations to write MPC relationships. The RSPLINE is often used to model mesh transitions.

Each of these R-type elements generates internal MPC (multipoint constraint) equations in MSC.Nastran. The simplest description of an MPC equation is that it is used to describe the motion of one (dependent) DOF in a model as a linear combination of the motion of one or more other (independent) DOFs. Simple examples of MPCs are to describe the motion of one DOF as the average of two other DOFs or to impose a rigid connection between two points.

The following problem statement describes a model of a thick plate with bars attached. Samples are given of three methods that can be used to model a rigid connection.

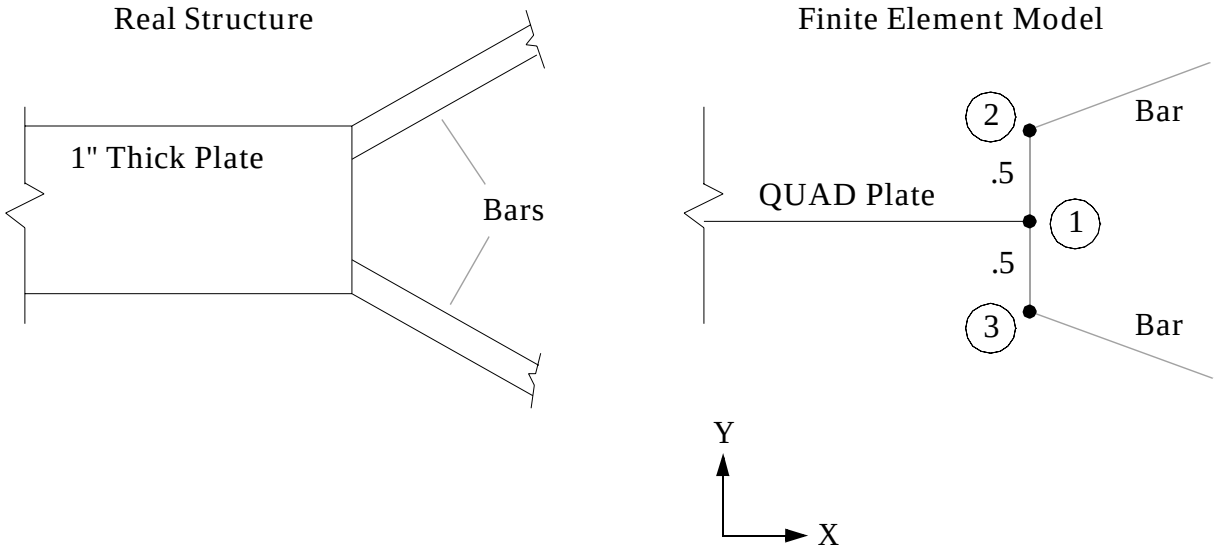


Figure 15 Model of a Thick Plate with Bars Attached

Plate theory states that plane sections remain plane. If this is the case, then grid points 2 and 3 are “slave” to grid point 1.

Therefore, write the equations for the in-plane motion of grid points 2 and 3 as a function of grid point 1.

Option 1 – MPC Equations for the Motion in the X-Y Plane. Looking only at the motion in the x-y plane,

$$u_{1_2} = u_{1_1} - .5 \cdot u_{6_1} \qquad (u_{1_2} = \text{displacement 1 at grid point 2})$$

$$u_{1_3} = u_{1_1} + .5 \cdot u_{6_1}$$

$$u_{6_3} = u_{6_1} \qquad u_{2_2} = u_{2_1}$$

$$u_{6_2} = u_{6_1} \qquad u_{2_3} = u_{2_1}$$

The MPC entries for this are as follows. (Note that MPC = 1 must appear in the Case Control Section to use these entries.)

1	2	3	4	5	6	7	8	9	10
MPC	1	2	1	1.	1	1	-1.		
		1	6	.5					
MPC	1	3	1	1.	1	1	-1.		
		1	6	-.5					
MPC	1	3	6	1.	1	6	-1.		
MPC	1	2	6	1.	1	6	-1.		
MPC	1	2	2	1.	1	2	-1.		
MPC	1	3	2	1.	1	2	-1		

Option 2 – RBAR (Rigid) Elements.

RBAR	99	1	2	123456					
RBAR	100	1	3	123456					

RBAR 99 generates MPC equations for the motion of grid point 2 as a function of grid point 1.

RBAR 100 generates MPC equations for the motion of grid point 3 as a function of grid point 1.

As written, these RBARs calculate the MPC equations for all six DOFs at grid points 2 and 3. If it is desired to have the equations generated only for the in-plane motion, the field labeled as CMB in the RBAR entry should have 126 entered.

Option 3 – RBE2 Element.

1	2	3	4	5	6	7	8	9	10
RBE2	99	1	123456	2	3				

This RBE2 generates MPC equations for the motion of grid points 2 and 3 as a function of the motion of grid point 1. Once again, this entry generates MPC equations for all six DOFs at grid points 2 and 3. If it were desired to generate only the in-plane terms, the “123456” should be changed to “126”.

All three methods produce the correct results. Methods 2 and 3 are preferable since you are not required to know the geometry in order to write the equations. Method 1 requires the knowledge of the geometry and the coordinate systems used and thus increases the probability of user error.

In summary, the rigid elements are used to model areas that are very stiff compared to the adjoining structure in order to prevent numerical difficulties and often to simplify the model. These elements (RBAR and RBE2) state that the motion of one or more grid points is dependent on the motion of one other grid point. Common uses include spacers, washers, connectors (such as bolts), and stiff components of a structure. These elements can also be used to connect two coincident grid points that have different coordinate systems. This condition can reduce the labor needed to apply dynamic loads and attach image superelements.

The interpolation elements allow you to model the motion of one or more grid points as a function of the motion of the other points. The RBE3 writes equations typical for interpolating the motion of points on a surface or in a volume to one grid point, and the RSPLINE uses the equations of a beam to write constraint equations.

Here are examples of modeling an interpolation function. Suppose that there are four grid points at the vertices of a square surface and a fifth grid point is to have the average of the motion of all four points (see **Figure 16**).

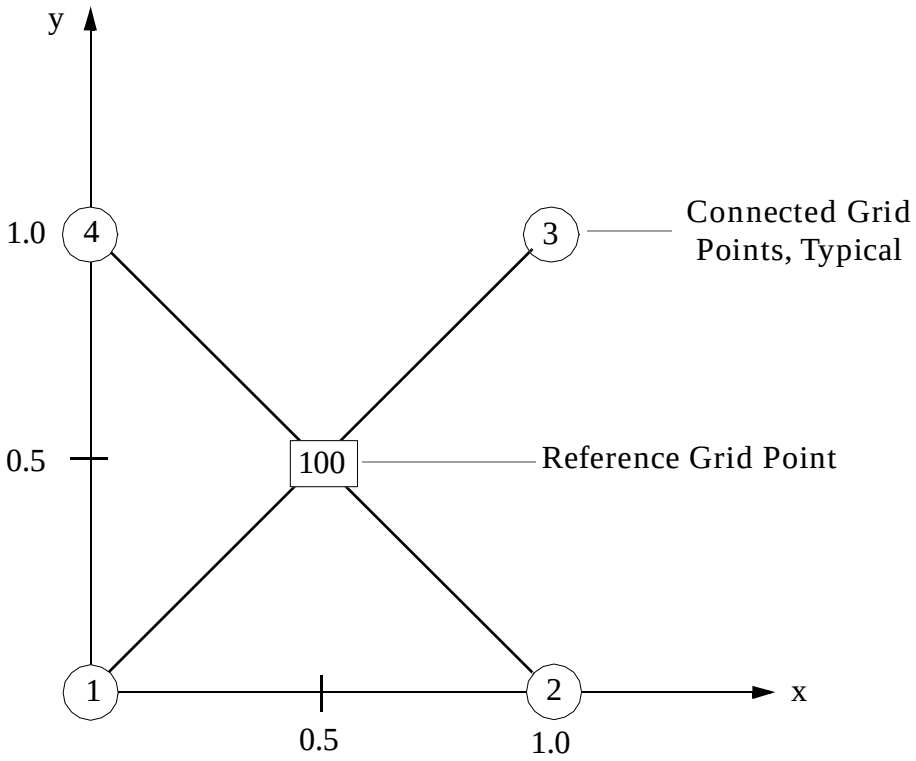


Figure 16 RBE3 Element Checkout Model

If you want the motion of grid point 100 in the x-direction to be the average displacement of grid points 1 through 4, use the following:

$$\text{Average displacement } u_{100_1} X_1 (u_{1_1} + u_{2_1} + u_{3_1} + u_{4_1})$$

A similar equation is used in the T2 direction. X_1 and X_2 are unknown quantities to be determined. By inspection, if the desired motion for point 100 is the average of the motions of points 1 through 4, the X_i are the inverse of the sum of the (implied) unit coefficients on points 1 through 4, or

$$X_1 = X_2 = \frac{1}{4}$$

The following MPC entry provides this action in the T1 direction:

1	2	3	4	5	6	7	8	9	10
MPC	10	100	1	–1.0	1	1	0.25		
		2	1	0.25	3	1	0.25		
		4	1	0.25					

The minus sign occurs because the MPC equation requires that all terms of the equation be on the left-hand side and equates them to zero. The first DOF listed on an MPC entry is always the dependent DOF.

Since a constraint equation may be multiplied by any arbitrary nonzero term without changing its meaning, let us multiply the entry for the T2 direction by 4.0 to reduce the labor of input preparation. The action in the T2 direction is exactly the same regardless of the term used to multiply all coefficients. This entry is then written as follows:

1	2	3	4	5	6	7	8	9	10
MPC	10	100	2	–4.0	1	2	1.0		
		2	2	1.0	3	2	1.0		
		4	2	1.0					

It can be shown that the MPC equations form the matrix equation

$$\{u_m\} = [G_{mn}]\{u_n\}$$

From principles of energy conservation, these constraint elements enforce a relation among the constraint forces q_i of the form:

$$\{q_n\} + [G_{mn}]^T\{q_m\} = 0$$

In the example above, the equation on u_i above implies the following relationships among the constraint forces. If a force is applied on the m-set degree of freedom such that $q_{100_1} = 1.0$, the resulting n-set forces are

$$q_{1_1} = q_{2_1} = q_{3_1} = q_{4_1} = -0.25$$

The RBE3 element shown below provides the same results:

$RBE3	EID		REFGRID	REFC	WT1	C1	G1,1	G1,2	
RBE3	100		100	12	1.0	123	1	2	
$	G1,3	G1,4							
	3	4							

The grid point that is to have the average motion point is called the REFGRID. This point is connected only on the DOFs listed on the REFC field. The T1 and T2 DOFs are connected in this example. The default action of this element is to place the REFC degrees of freedom in the m-set. (The element has provisions to place other DOFs in the m-set instead. However, a discussion of this capability is beyond the scope of this discussion.) Groups of connected grid points follow next. The start of a group is indicated by a real number WTi, which is used as a weighting factor for the grid points in the group. In this example we want a simple average. A uniform weight is applied to all points. The element provides automatic normalization of the weighting coefficients in a manner similar to the discussion on MPC coefficients above. The weighting factors are not required to total any specific value. An example where non-constant weighting factors are used is given at the end of this discussion.

The connected DOFs for the group are listed in the Ci field. Note that all three translation DOFs are listed even though the REFC field does not include the T3 direction. This is done because the DOFs listed for all points must be adequate to define the rigid-body motion of the RBE3 element even when the element is not to carry loads in certain directions. If the T3 DOF is not included in C1 in this example, a fatal message of the form “SINGULAR ELEMENT” occurs. This requirement was established to provide a simpler, more reliable implementation for the element. It is a software, not an engineering requirement. All of the R-type elements have a similar requirement for similar reasons.

It will be shown later that the element described by this RBE3 entry does not transmit forces in the T3 direction. The two reasons for this are that the reference grid point is not connected in this direction and all of the connected points are in the same plane. A list of the grid points in the group follows. They have a common weighting function and a common set of connected DOFs. If other points were to be connected with differing weights or connected DOFs they would be listed in a following group. There is an example of such an element below.

The theory used to generate the internal MPC equations caused by RBE3 elements is difficult to understand. The actual load redistribution is often surprising even to those who understand the theory, particularly for complicated geometry and weighting function specifications. It is good practice to check out a typical RBE3 element specification with a small MSC.Nastran test model to allow you to feel comfortable with its action. Expert users of the element do so every time they use a pattern of geometry and weighting functions that is new to them. They sometimes find it necessary to adjust the weighting factors to obtain the behavior needed to match the device that the element models. A checkout input file for the RBE3 element described above follows. This input file includes extensive comments and is described further below.

```
$ FILE RBE3DEMO.DAT
INIT MASTER(S) $ DELETE DATABASE AT END OF RUN
SOL 101 $ STATICS
CEND
TITLE = RBE3 ELEMENT CHECKOUT MODEL
ECHO = BOTH
SPC = 1
DISP = ALL
OLOAD = ALL
SPCFORCE = ALL $ PRINT FORCE REQUIRED TO REACT FORCES IN RBE3 ELEMENT
SUBCASE 1
LABEL = LOAD IN X DIRETION
LOAD = 1
SUBCASE 2
LABEL = LOAD IN Y DIRECTION
LOAD = 2
SUBCASE 3
LABEL = LOAD IN Z DIRECTION
LOAD = 3
BEGIN BULK
GRID,   1,      ,       0.      0. $ SQUARE PATTERN
GRID    2               1.0     0.
GRID    3               1.0     1.0
GRID    4               0.      1.0
GRID,   100,    ,       0.5     0.5 $ REF. POINT IN CENTER OF SQUARE
$RBE3,  EID,    ,       REFGRID REFC    WT1     C1      G1,1    G1,2
RBE3    100             100     12      1.0     123     1       2
$,      G1,3    G1,4
,       3       4
SPC1,   1,      123456, 1,      2,      3,      4 $ ALL G1,I POINTS
SPC,    1,      4,      1,      1.0 $ MOVE POINT FOUR UNIT DISP. IN T1
FORCE,  1,      100,    ,       1.0,    1.0                 $ UNIT X FORCE
FORCE,  2,      100,    ,       1.0,    0.0     1.0         $ UNIT Y FORCE
FORCE,  3,      100,    ,       1.0,    0.0     0.0     1.0 $ UNIT Z FORCE
GRID,   111 $ GRID POINT FOR TOKEN A-SET DOF
CELAS2, 1110,   1.0,    111,    1 $ TOKEN STIFFNESS TO CAUSE A-SET.
$ OTHER DOF OF GRID POINT 111 ARE AUTOSPCD.
PARAM,  MPCF,   YES $ PRINT THE MPC FORCES
ENDDATA
```

Figure 17 Example Checkout Input File for the RBE3 Element

The example in **Figure 17** demonstrates the load transfer ability of the RBE3. The forces in the 1 and 2 DOFs are equally distributed to the four attachment grid points. Since no connection is made in the 3 direction, AUTOSPC constrains that DOF.

The solution sequences require at least one flexible element and one DOF in the analysis set. The disjoint GRID 111 and its element CELAS2 1110 perform this function. These DOFs need not be constrained or loaded explicitly. In the Structured Solution Sequences, AUTOSPC automatically constrains obvious singularities of the disjoint and other independent grid points.

A review of the output produced by this input file shows the following:

The displacement of all points is zero except grid points 4 and 100. Grid point 4 moves the unit displacement defined on the SPC entry. The reference grid point 100 moves 0.25 units in the same direction. This is the average motion of the four connected points.

The SPC force in the T1 and T2 directions of all connected points is -0.25, which is the sum of the loads needed to balance the applied load in the same direction. (These forces are shown in Figure , for a similar model discussed below.) The only SPC force in the T3 direction is for the reference point, which means that none of this force is transmitted by the element. The OLOAD resultants are equal but opposite to the SPCFORCE resultants. The MPCFORCE output is equal but opposite to the SPCFORCE output on the connected points, and is equal and opposite to the applied load on the reference point.

The most common usage of the RBE3 element is to transfer motion in such a way that all six DOFs of the reference point are connected. The RBE3 element for the model described above, which provides this action, is as follows:

1	2	3	4	5	6	7	8	9	10
$RBE3	EID		REFGRID	refc	WT1	C1	G1,1	G1,2	
RBE3	100		100	123456	1.0	123	1	2	
$	G1,3	G1,4							
	3	4							

Note that the rotations are not used for the independent DOFs. The only field that is changed is the REFC field, as indicated by the lowercase label. Note that all six DOFs are connected. If you wish to inspect the load-transmitting action of this specification, add three more subcases to the checkout input file above, and select a unit moment on the reference point in a different direction in each new subcase.

Notice that the SPC forces for the first two subcases are unchanged (see **Figure 18**). The effect of a missing DOF in a REFC list is that the constraint equation associated with it is discarded. Its absence does not change the equations that are retained by placing other DOFs in the REFC list. Moments at the reference grid are reacted by couples produced by forces at the connected points. In general, the load distribution is quite regular, reflecting the uniform weighting of the element specification and the regular geometry of all connected points.

The translation of the reference point is unchanged from the prior case. There is also a rotation of the reference point because these rotational DOFs are now connected by the REFC entry. The rotation of the reference point is a function of only the translation of the connected points because the rotations of the connected points are not attached to the element. That is, they are not listed in their C1 field.

Now suppose that you have reviewed what is connected to these grid points in your model. You have determined that grid points 1 through 3 should all have about the same load-carrying capacity and stiffness, while point 4 is on much weaker structure. In your judgement, point 4 can carry only one-fourth of the load of the other points in the plane of

the surface, while it has no load-carrying ability at all normal to the surface. One method to model this with an RBE3 element is to change the element in the second example above as follows:

1	2	3	4	5	6	7	8	9	10
$RBE3	EID		refgrid	REFC	WT1	C1	G1,1	G1,2	
RBE3	100		100	123456	1.0	123	1	2	
$	G1,3	wt2	c2	g2,1					
	3	0.25	12	4					

Again, changed fields are indicated by lowercase labels. Grid point 4 is given a smaller weighting factor and its load path in the T3 direction is eliminated by removing its Ci entry. The absolute value of a weighting factor is not important, merely its ratio to the other weighting factors.

A review of the computed results from using this specification shows the following:

The SPC reaction forces are no longer uniform but have a more complicated distribution as shown in **Figure 18**. (Results are rounded off to three significant digits.) An applied load in the T1 direction causes loads in the T1 and T2 directions on the connected points. The loads that are reacted at grid point 4 are smaller than the loads on the other points, but they are not one-fourth of the other loads. An applied load in the T3 direction loads only two of the connected grid points. An applied moment about the T3 axis results in an irregular distribution. The grid point rotation of the reference point is reduced. The T1 motion of the connected point causes both T1 and T2 components of motion in the reference point.

This behavior is difficult to explain in terms of equations that interpolate motion. The behavior is more readily understood in terms of the load distribution implied by the dual nature of the constraint equations.

The reason for the redistribution caused by changing the weighting factors on grid point 4 is that the element enforces equilibrium for the forces and moments which it transmits. This is a requirement for avoiding internal constraints in the element. An internal constraint is a load path between the element and ground that does not appear in SPC force output. The existence of internal constraints results in applied loads and SPC forces that are not in balance. MSC.Nastran elements in general and the R-type elements in particular are not allowed to have the undesirable behavior of internal constraints. By contrast, MPC equations can have internal constraints that are difficult to diagnose and remove. This was one of the prime motivations for developing the R-type elements.

Enforcing equilibrium is the second step in computing MPC coefficients for the element. The first step is to compute loading functions due to the element geometry and the weighting factors of the connected DOFs. “Raw” loading functions are determined from these parameters. These functions depend only on weighting factors and the distances between the reference point and the connected points. In the second step, linear combinations of these functions are used to produce a set that provides zero load resultants for all loads transmitted by the element.

In this example, point 4 has loads smaller than the other points in the plane of the element, but they are not exactly one-fourth of the other loads due to the requirements for equilibrium (see Case 1 in **Figure 18**). The zero load capability in the T3 direction of point 4 results in zero loads in the T3 direction on both it and its opposite grid point (point 2) due to a T3 load on the reference point (see Case 2). Similarly, when a moment is applied about the T3 direction, the loads in point 4 are not the average of the loads in the other connected points (see Case 3 in **Figure 18**). It appears that the element has redistributed the loads in a manner affected only weakly by the weighting factors in some directions and not at all in other directions. Another point of view is that this element defines a poor structure for transmitting loads on some of its load paths. MSC.Nastran has reset the weighting factors to allow some weighting effects while avoiding internal constraints. Internal constraints are difficult to detect and remove in complex models and are usually not wanted by the modeler. You cannot arbitrarily redistribute loads applied on one point to other points without considering some physical principles. The principles used by the RBE3 element are equilibrium foremost with adjustment of weighting functions so that equilibrium is not violated.

The most common user error in RBE3 element specification results from placing 4, 5, or 6 in the Ci (independent DOF) field when also including some translation components. The action of the element is somewhat irrational for this specification. The resulting MPC equations are dimension-dependent. That is, a model described in inches has different ratios of distributed force to moments if it is converted to centimeters. The ability to input 4, 5, or 6 in the Ci field is only for rather unusual applications, such as when all of the connected points are colinear and a device is needed to restrain the rotation of the RBE3 element about a line parallel to the line of the colinear connected points. Your modeling intent here should be never to use this DOF, you merely intend to stabilize this rotation mode of the element. As implied in Remark 1 of the description of this element in the *MSC.Nastran Quick Reference Guide*, do not use 4, 5, or 6 for Ci values unless you have a good reason to do so. Check out any elements so defined with a checkout run, as described above, before inserting them in a model.

Again, checkout models are recommended whenever you are specifying elements with nonuniform weight factors, asymmetric geometry or connected degrees of freedom, or irregular geometry. This is especially true when the reference point is not near the center of the connected points. In general, the more the element deviates from uniformity and the fewer connected DOFs are present, the more the enforcement of equilibrium influences the load distribution. In the limit, when the connected DOFs are reduced to a set that defines a statically determinate load path, the RBE3 element degrades to a rigid RBAR-like constraint. The weighting factors have no influence in this limiting case.

In summary, the intended use for the RBE3 element is to transmit forces and moments from a reference point to several non-colinear points. The rotation components 4, 5, and 6 should be placed in the Ci field only for special cases. If the points are colinear, special attention must be given to stabilizing all rotation modes of the element. When less than three points are connected to the element, it may act like one of the rigid elements. One of the other elements may be better suited to such applications. The load distribution of the

element may not be obvious for complex specifications. Check cases are therefore recommended the first time you use a new type of RBE3 specification. These checks will minimize surprises when reviewing the action of these elements in complex models.

Case 1.

Applied Force in T1 Direction		
	Grid Point 4	Grid Point 3
Fx, Fy	–0.25, 0.0 –.0885, –.011	–0.25, 0.0 –0.354, .029
	Grid Point 1	Grid Point 2
Fx, Fy	–0.25, 0.0 –0.279, –.0461	–0.25, 0.0 –0.279, .029

Case 2.

Applied Force in T3 Direction		
	Grid Point 4	Grid Point 3
Fz, Fy	–0.25 0.0	–0.25 –0.5
	Grid Point 1	Grid Point 2
Fz, Fy	–0.25 –0.5	–0.25 0.0*

Case 3.

Applied Moment in T3 Direction		
	Grid Point 4	Grid Point 3
Fx, Fy	0.25, 0.25 0.1 0.1	0.25, –0.25 0.4, –0.25
	Grid Point 1	Grid Point 2
Fx, Fy	–0.25, 0.25 –0.25, 0.4	–0.25, –025 –0.25, –0.25

Key:
Results for Uniform Weighting
These are the results when grid point 4 has reduced weighting.
*A reaction in this DOF produces a moment.

Figure 18 Load Distributions for Uniform and Nonuniform Weightings

Although space limitations do not allow more uses to be shown, the RBE3 is a powerful tool for distributing applied loads. The default form with scaling factors of 1.0 for all independent DOFs distributes an applied loading from the dependent grid point to the independent grid points in a manner similar to a bolt pattern. The forces are distributed based on the local weight factors (similar to area in a bolt pattern) and any moments are distributed by the ratio of $WT_i \cdot M \cdot r_i / \sum (r_i)^2$ where M is the moment and r_i is the radius from the centroid to the "bolt" (or grid point) of interest. In this expression, WT_i (the weighting factor) again works like the area in the bolt pattern analogy.

This loading distribution makes the RBE3 an ideal element to use to apply loads from a coarse model (or hand calculation) onto a detailed model of a component. For example, the shear distribution on a cross section is a function of the properties of that section. Multiple shear loadings may be distributed on a cross section by performing a calculation of the shear distribution based on unit loading and using an RBE3 with the shear values for the weighting factors at each grid point. In this manner, only one shear distribution need be calculated by hand, and all other loadings (normally there are multiple loading conditions to be considered) may be applied automatically by applying the load to the dependent point on the RBE3.

Another advantage of RBE3 is that no stiffness is added to the structure for a properly specified RBE3; that is, a set of interpolation equations is used to determine the motion of the dependent grid point, and these equations include the effects of the geometry.

The RSPLINE is often used to model a transition from a coarse mesh to a fine mesh. Since an RSPLINE uses the equations of a beam to generate MPC equations, it does an excellent job of approximating the transition for a plate in bending. Note that interpolation elements should always be used away from areas of interest or areas of rapid stress changes. They tend to work well in areas of uniform stress or minor variations in stress, but the approximations may not work well in an area of rapid stress change. The equations used by an RSPLINE approximate a cubic displacement field. However, in a highly stressed area, the variation may need more than a cubic deformation to match the actual deformed shape. Likewise, the RBE3 uses linear interpolation, which may not be able to describe the actual deformed shape.

What Are the Element Codes?

The tables in **Item Codes** (p. 1435) in the *MSC.Nastran Quick Reference Guide* identify the codes for the elements in MSC.Nastran. What do these codes mean, and how are they used in MSC.Nastran?

Each element type in MSC.Nastran is identified internally by a number. Several different element types may be defined by the same connectivity entries; however, because of different material properties, stiffness formulations and output, the element types are identified with different element codes. The element codes are stored in the MSC.Nastran data blocks, and the appropriate calculations are performed based on the codes. For instance, a nonlinear elastic CQUAD4 element has a different element code than a

hyperelastic (finite deformation) CQUAD4 element. The nonlinear element codes for the nonlinear element stress output are different than the nonlinear element codes for other output.

The following table indicates all of the element connectivity entries in MSC.Nastran Version 70. Each connectivity entry may have many different types of elements associated with it, and they are indicated by the different element codes. A description of each element is supplied for each element code. The elements may be identified differently in the printed output based on the element code, and the names are specified in the table. The associated property and material entries for each element are specified as well.

The elements in element punch output and element data blocks are identified by number, not by name. Therefore, the table can be used to identify the elements in the punch output and in printed data blocks.

Table 3 Element Connectivity Entries in MSC.Nastran Version 70

Connectivity Entry	Printed Output	Code	Description	Property Entry	Material Entry
CAABSF		101	Frequency-dependent acoustic absorber element	PAABSF	N / A
CAERO1		--	Aerodynamic panel, double lattice and ZONA	PAERO1	N / A
CAERO2		--	Aerodynamic body, double lattice	PAERO2	N / A
CAERO3		--	Aerodynamic panel, Mach box lifting surface	PAERO3	N / A
CAERO4		--	Aerodynamic macro-strip	PAERO4	N / A
CAERO5		--	Aerodynamic panel, piston theory	PAERO5	N / A
CAXIF2	AXIF2	47	Axisymmetric fluid, 2 fluid points	N / A	N / A
CAXIF3	AXIF3	48	Axisymmetric fluid, 3 fluid points	N / A	N / A
CAXIF4	AXIF4	49	Axisymmetric fluid, 4 fluid points	N / A	N / A
CBAR	BAR	34	Simple beam	PBAR PBARL	MAT1 MAT4 MAT5
		100	Simple beam with intermediate stations	PBAR PBARL	MAT MAT4 MAT5

Table 3 Element Connectivity Entries in MSC.Nastran Version 70 (continued)

Connectivity Entry	Printed Output	Code	Description	Property Entry	Material Entry
CBEAM	BEAM	2	Beam, composite element stress output	PBEAM PBEAML PBCOMP	MAT1 MAT4 MAT5
		94	Nonlinear element stress output	PBEAM PBEAML PBCOMP	MATS1
		105	Beam p-element	PBEAM PBEAML	MAT1
CBEND	BEND	69	Curved beam, curved pipe, or elbow	PBEND	MAT1 MAT4 MAT5
CBUSH	BUSH	102	Frequency-dependent or nonlinear spring and damper	PBUSH	N/A
CCONEAX	CONEAX	35	Conical shell	PCONEAX	MAT1 MAT2
CDAMP1	DAMP1	20	Scalar damper	PDAMP	N/A
CDAMP2	DAMP2	21	Scalar damper	N/A	N/A
CDAMP3	DAMP3	22	Scalar damper	PDAMP	N/A
CDAMP4	DAMP4	23	Scalar damper	N/A	N/A
CDAMP5		106	Heat capacitance	PDAMP5	MAT4 MAT5
CDUMi		54-59	Dummy	PDUMi	MATi
CELAS1	ELAS1	11	Scalar spring	PELAS	N/A
CELAS2	ELAS2	12	Scalar spring	N/A	N/A
CELAS3	ELAS3	13	Scalar spring	PELAS	N/A
CELAS4	ELAS4	14	Scalar spring	N/A	N/A
CFLUID2	FLUID2	43	Axisymmetric fluid, 2 circles	N/A	N/A
CFLUID3	FLUID3	44	Axisymmetric fluid, 3 circles	N/A	N/A
CFLUID4	FLUID4	45	Axisymmetric fluid, 4 circles	N/A	N/A
CFTUBE	FTUBE	73	Fluid tube, heat transfer	PFTUBE	N/A
CGAP	GAP	86	Gap or frictional, nonlinear	PGAP	N/A

Table 3 Element Connectivity Entries in MSC.Nastran Version 70 (continued)

Connectivity Entry	Printed Output	Code	Description	Property Entry	Material Entry
CHACAB	HACAB	83	Acoustic absorber, coupled fluid-structure	PACABS	N/A
CHACBR	HACBR	84	Acoustic barrier	PACBAR	N/A
CHBDY	HBDY	52	Heat boundary	PHBDY	MAT4
CHEXA	HEXA	67	Six-sided solid, 8-20 nodes	PSOLID	MAT1 MAT4 MAT5 MAT9
		93	Nonlinear element stress output	PSOLID	MATS1
		141	Six-sided solid, 8-20 nodes (p-elements)	PSOLID	MAT1
		145	Six-sided solid, p-element results processing for internally generated nodes	PSOLID	MAT1
	HEXPR	76	Fluid, pressure formulation	PSOLID	MAT10
	HEXAFD	140	Six-sided solid, 8 nodes, finite deformation (hyperelastic), nonlinear	PLSOLID	MATHP
		202	Nonlinear element stress output	PLSOLID	MATHP
	HEXAFD	163	Six-sided solid, 9-20 nodes, finite deformation (hyperelastic), nonlinear	PSOLID	MATHP
		207	Nonlinear element stress output	PSOLID	MATHP
CHEXA1	HEXA1	41	Six-sided solid, 8 nodes, 5 tetrahedra, heat transfer	N/A	MAT4 MAT5
CHEXA2	HEXA2	42	Six-sided solid, 8 nodes, 10 tetrahedra, heat transfer	N/A	MAT4 MAT5
CMASS1	MASS1	25	Scalar mass	PMASS	N/A
CMASS2	MASS2	26	Scalar mass	N/A	N/A
CMASS3	MASS3	27	Scalar mass, scalar points only	PMASS	N/A

Table 3 Element Connectivity Entries in MSC.Nastran Version 70 (continued)

Connectivity Entry	Printed Output	Code	Description	Property Entry	Material Entry
CMASS4	MASS4	28	Scalar mass, scalar points only	N/A	N/A
CONM1	CONM1	29	Concentrated mass	N/A	N/A
CONM2	CONM2	30	Concentrated mass, rigid body	N/A	N/A
CONROD	CONROD	10	Rod element	N/A	MAT1 MAT4 MAT5
		92	Nonlinear element stress output	N/A	MATS1
CPENTA	PENTA	68	Five-sided solid, 6-15 nodes	PSOLID	MAT1 MAT4 MAT5 MAT9
		91	Nonlinear element stress output	PSOLID	MATS1
		142	Five-sided solid, 6-15 nodes (p-elements)	PSOLID	MAT1
		146	Five-sided solid, p-element results processing for internally generated nodes	PSOLID	MAT1
	PENPR	77	Fluid, pressure formulation	PSOLID	MAT10
	PENTAFD	160	Five-sided solid, 6 nodes, finite deformation (hyperelastic), nonlinear	PLSOLID	MATHP
		204	Nonlinear element stress output	PLSOLID	MATHP
	PENTAFD	165	Five-sided solid, 7-15 nodes, finite deformation (hyperelastic), nonlinear	PLSOLID	MATHP
		209	Nonlinear element stress output	PLSOLID	MATHP

Table 3 Element Connectivity Entries in MSC.Nastran Version 70 (continued)

Connectivity Entry	Printed Output	Code	Description	Property Entry	Material Entry
CQUAD	QUADFD	164	Five to 9-noded plane strain, finite deformation (hyperelastic), nonlinear	PLPLANE	MATHP
		208	Nonlinear element stress output	PLPLANE	MATHP
CQUAD4	QUAD4	33	Quadrilateral plate element, isoparametric membrane-bending	PSHELL	MAT1 MAT2 MAT4 MAT5 MAT8
		91	Nonlinear element stress output	PSHELL	MATS1
		95	Composite element stress output	PCOMP	MAT1 MAT2 MAT8
		144	Corner stress output	PSHELL PCOMP	MAT1 MAT2 MAT8
		103	Shell p-element	PSHELL	MAT1
CQUAD4	QUADFD	139	Four-noded plane strain, finite deformation (hyperelastic), nonlinear	PLPLANE	MATHP
		201	Nonlinear element stress output	PLPLANE	MATHP
CQUAD8	QUAD8	64	Curved quadrilateral shell element, 4-8 nodes	PSHELL	MAT1 MAT2 MAT4 MAT5 MAT8
		96	Composite element stress output	PCOMP	MAT1 MAT2 MAT8
CQUADR	QUADR	82	Quadrilateral plate, isoparametric membrane and bending, no membrane-bending coupling	PSHELL PCOMP	MAT1 MAT2 MAT8

Table 3 Element Connectivity Entries in MSC.Nastran Version 70 (continued)

Connectivity Entry	Printed Output	Code	Description	Property Entry	Material Entry
CQUADX	QUADXFD	170	Four-noded finite deformation axisymmetric quadrilateral element (hyperelastic), nonlinear	N/A	MATHP
		214	Nonlinear element stress output	N/A	MATHP
	QUADXFD	171	Five to 9-noded finite deformation axisymmetric quadrilateral element (hyperelastic), nonlinear	N/A	MATHP
		215	Nonlinear element stress output	N/A	MATHP
CRAC2D	CRAC2D	60	Two-dimensional crack tip	PRAC2D	MAT1 MAT2 MAT8
CRAC3D	CRAC3D	61	Three-dimensional crack tip	PRAC3D	MAT1 MATT9
CROD	ROD	1	Tension-compression-torsion rod	PROD	MAT1 MAT4 MAT5
		89	Nonlinear element stress output	PROD	MATS1
CSHEAR	SHEAR	4	Shear panel	PSHEAR	MAT1
CSLOT3	SLOT3	50	Radial slot, 3 nodes, 2-D wave equation, acoustic cavity	N/A	N/A
CSLOT4	SLOT4	51	Radial slot, 4 nodes, 2-D wave equation, acoustic cavity	N/A	N/A

Table 3 Element Connectivity Entries in MSC.Nastran Version 70 (continued)

Connectivity Entry	Printed Output	Code	Description	Property Entry	Material Entry
CTETRA	TETRA	39	Four-sided solid, 4-10 nodes	PSOLID	MAT1 MAT4 MAT5 MAT9
		85	Nonlinear element stress output	PSOLID	MATS1
		143	Four-sided solid, 4-10 nodes (p-elements)	PSOLID	MAT1
		147	Four-sided solid, p-elements results processing for internally generated nodes	PSOLID	MAT1
	TETPR	78	Fluid, pressure formulation	PSOLID	MAT10
	TETRAFD	161	Four-sided solid, 4 nodes, finite deformation (hyperelastic), nonlinear	PLSOLID	MATHP
		205	Nonlinear element stress output	PLSOLID	MATHP
	TETRAFD	166	Four-sided solid, 5-10 nodes, finite deformation (hyperlastic), nonlinear	PLSOLID	MATHP
		210	Nonlinear element stress output	PLSOLID	MATHP
CTRIA3	TRIA3	74	Triangular plate element, isoparametric membrane-bending	PSHELL	MAT1 MAT2 MAT4 MAT5 MAT8
		88	Nonlinear element stress output	PSHELL	MATS1
		97	Composite element stress output	PCOMP	MAT1 MAT2 MAT8
		104	Shell p-element	PSHELL	MAT1

Table 3 Element Connectivity Entries in MSC.Nastran Version 70 (continued)

Connectivity Entry	Printed Output	Code	Description	Property Entry	Material Entry
CTRIA3	TRIAFD	162	Three-noded plane strain, finite deformation (hyperlastic), nonlinear	PLPLANE	MATHP
		206	Nonlinear element stress output	PLPLANE	MATHP
CTRIA6	TRIA6	75	Curved triangular plate, 3-6 nodes	PSHELL	MAT1 MAT2 MAT4 MAT5 MAT8
		98	Composite element stress output	PCOMP	MAT1 MAT2 MAT8
	TRIAFD	167	Four- to 6-noded plane strain, finite deformation (hyperelastic), nonlinear	PLPLANE	MATHP
		211	Nonlinear element stress output	PLPLANE	MATHP
CTRIAR	TRIAR	70	Triangular plate element, isoparametric membrane and bending, no membrane-bending coupling	PSHELL PCOMP	MAT1 MAT2 MAT8
CTRIAX	TRIAXFD	168	Three-noded finite deformation axisymmetric triangle element (hyperelastic), nonlinear	N/A	MATHP
		212	Nonlinear element stress output	N/A	MATHP
	TRIAXFD	169	Four- to 6-noded finite deformation axisymmetric triangle element (hyperelastic) nonlinear	N/A	MATHP
		213	Nonlinear element stress output	N/A	MATHP
CTRIAX6	TRIAX6	53	Triangular cross-sectional ring, isoparametric and axisymmetric, 3-6 nodes	N/A	MAT1 MAT3

Table 3 Element Connectivity Entries in MSC.Nastran Version 70 (continued)

Connectivity Entry	Printed Output	Code	Description	Property Entry	Material Entry
CTUBE	TUBE	3	Tension-compression-torsion tube	PTUBE	MAT1 MAT4 MAT5
		87	Nonlinear element stress output	PTUBE	MATS1
CVISC	VISC	24	Viscous damper	PVISC	N / A
PLOTEL	PLOTEL	31	Dummy one-dimensional, plotting	N / A	N / A

p-version Elements

The p-version element is first introduced in Version 68 of MSC.Nastran. The names p-version element and p-element are used interchangeably in this book.

What Are h-elements and p-elements? What Are h-adaptivity and p-adaptivity?

In traditional finite element analysis as the number of elements increases, the accuracy of the solution improves. The accuracy of the problem can be measured quantitatively with various entities, such as strain energies, displacements, and stresses, as well as in various error estimation methods, such as simple mathematical norm or root-mean-square methods. The goal is to perform an accurate prediction on the behavior of your actual model by using these error analysis methods. You can modify a series of finite element analyses either manually or automatically by reducing the size and increasing the number of elements. This is the usual h-adaptivity method. Each element is formulated mathematically with a certain predetermined order of shape functions. This polynomial order does not change in the h-adaptivity method. The elements associated with this type of capability are called the h-elements.

A different method to modify the subsequent finite element analyses on the same problem is to increase the polynomial order in each element while maintaining the original finite element size and mesh. The increase of the interpolation order is internal, and the solution stops automatically once a specified error tolerance is satisfied. This is known as the p-adaptivity method. The elements associated with this capability are called the p-elements.

These two methods can be combined to modify the subsequent analyses on the same model by simultaneously reducing the element size and increasing the interpolation order in each element. This combination is called mixed hp-adaptivity.

In addition to the existing h-elements and the manual h-adaptivity method, MSC.Nastran supports the automatic p-adaptivity and semi-automatic mixed hp-adaptivity.

What Types of p-elements Are Supported? What Types of Analyses Are Supported by p-elements? Are p-elements Compatible with Existing h-elements?

The types of p-elements and analyses supported are version-dependent and are summarized in the following table.

Version	p-element Types	Analysis Types
Version 68	CHEXA(8), CPENTA(6), CTETRA(4)	SOLs 101, 103
Version 69	CHEXA(8), CPENTA(6), CTETRA(4)	SOLs 101, 103, 107-112
	CBEAM, CQUAD4, CTRIA3	SOLs 101, 103

These p-elements support point, line, pressure, gravity, and thermal loads with p-level starting from one with no upper limits. These limits may be restricted by practical considerations, such as the availability of computer resources, etc. The stress discontinuity error estimation and strain energy density sensitivity method are employed for the adaptivity control. For SOLs 107-112, the p-elements do not adapt; the analysis is done for a fixed p-level.

When Should I Use p-adaptivity Versus h-adaptivity?

Each method has its advantages over other methods in terms of performance, accuracy, and ease of use. The choice depends, among other things, on the complexity of the geometry, the loading pattern, and the nature of the problem. Remember that all methods have to start with a finite element mesh. A coarser p-element mesh yields similar results as compared to a finer h-element mesh.

In general, if the localized detailed stress with the actual geometry is of interest, then you should use p-elements and p-adaptivity. If the load path and performing global analysis with approximate geometry is sufficient, then h-adaptivity is a better choice. If high stress gradients exist within localized areas, then mixed hp-adaptivity is the best method. Use p-elements for the area of interest and h-elements for the overall model.

How Does the p-element Compare with the Corresponding h-element?

In simple terms, solid elements with a p-level of one are similar to CHEXA(8), CPENTA(6), and CTETRA(4); while elements with a p-level of two are similar to the solid elements with mid-side nodes. However, p-elements have different formulations and integration schemes as compared to the h-elements. Consequently, the results for p-elements may be slightly different when compared to the results for h-elements.

How Can I Use the p-element Adaptivity Feature?

All the p-element features are self contained within MSC.Nastran. These features are supported with the use of Case Control and Bulk Data input. Using MSC.Patran as a graphical processing tool certainly facilitates using p-elements with MSC.Nastran. For further details, refer to *MSC.Nastran Release Guide for Version 69*.

Modeling

How Should I Model a Mesh Transition?

Mesh transition is a complicated subject and a full discussion of the topic goes far beyond the scope of this document. Mesh transitions may simply be used to refine the mesh in an area of interest, connect different element types (for example, a CBAR element to a solid element), and make transitions required to model the geometry of the structure.

A set of guiding rules for mesh transitions can be summarized as follows. Never place a mesh transition in an area of interest or in an area where there is a large variation in stress. The best mesh transitions are those modeled far away from the areas of interest in a region of uniform stress.

Due to incompatibilities between finite element types, any transition between different element types (even a transition from CQUAD4 to CTRIA3 elements) can result in local stress anomalies. Normally these are localized and dissipate quickly as you get away from the transition. A problem arises when the transition occurs in an area of interest. In this case, the local stress rises (or decreases) due to the effect of the transition. This means that the results may be conservative (or unconservative) in an area near a transition. However, if this localized stress variation occurs away from areas of interest, the increase (or decrease) in stress caused by the transition should cause no concern.

How Do I Attach Plate and Bar Elements to Solid Elements?

This question is more involved than it appears at first glance. Solid elements have stiffness only in the translational DOFs at the attachment grid points. They have no stiffness for rotational DOFs. A simple visualization of this is to think of the attachment of a solid element to a grid point as a “ball-and-socket” joint, that is, translational forces may be transmitted, but no moment may be transmitted through the connection.

This causes a modeling problem whenever plate or bar elements are attached to solid elements. Both plate and bar elements have stiffness for rotational DOFs (although the plate may not have a stiffness for the in-plane rotation). Therefore, special modeling must be performed whenever a plate or bar is connected to a solid element. Otherwise, the connection becomes a hinge (for plate elements) or a pinned connection (for bar elements).

A number of methods are available to handle the transition between these elements. These methods range from adding extra elements (for example, simply adding an additional plate or bar which continues into the solid element), to using special elements for the transition.

One method of handling this transition is to use RBE3 elements (see the question “Describe the Rigid Elements”). The RBE3 is an interpolation element, which is ideally suited for this application. By using RBE3s, we can state that the rotation of the attachment grid points is found simply by “slaving” it to translations at the adjacent grid points.

For example in the figure below, the following RBE3 elements attach the rotational DOFs to the translational DOFs on the solid element.

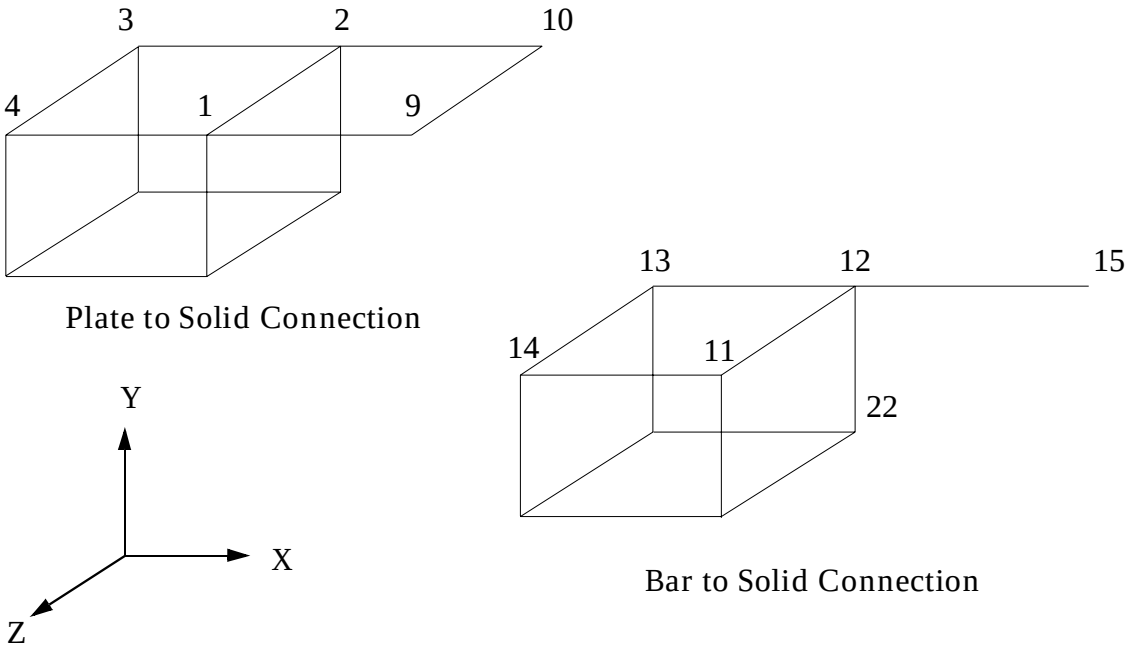


Figure 19 Transitions Between Element Types

For the plate to solid connection, two RBE3 elements suffice:

1	2	3	4	5	6	7	8	9	10
$RBE3	EID		REFGRID	REFC	WT1	C1	G1,1	G1,2	
RBE3	901		1	456	1.0	123	2	3	
	4								

RBE3	902		2	456	1.0	123	1	3	
	4								

For the bar to solid connection, one RBE3 element can make the connection:

RBE3	903		12	456	1.0	123	11	13	
	22								

These RBE3 elements do not add stiffness to the model; rather they transmit the loading to the independent DOFs. If RBE2 elements are used, then the connection is “rigid”.

The important thing to remember when handling these connections is that the solid elements have no stiffness for rotational DOFs, whereas the real structure does. This means that special modeling effort is needed when any element with bending stiffness is connected to a solid element.

When using the RBE3 element, care must be taken to ensure that the independent DOFs are sufficient to transfer any applied loadings. For example, in the sample above for the bar to solid connection, if only two independent grid points are used, the element is “unstable”;

that is, since only the translational DOFs are used as independent in the sample, the element is unstable for rotation about the axis connecting the two points. Therefore, three non-colinear grid points are used. A simple way to remember this is to think “If I were to constrain the DOFs I have listed as independent on the RBE3, could I prevent any possible rigid-body motion?” In this way, you can avoid possible problems in processing the RBE3 elements.

RSSCON – Alternate Shell-to-Solid Transition Element

A new method using the RSSCON for modeling shell-to-solid transition is introduced in Version 69 (see **Figure 20**). This capability conveniently eliminates the need to define RBE3s or MPCs (multipoint constraints) to constrain a shell element’s translational and rotational degrees of freedom to a solid element’s translational degrees of freedom. When using the RSSCON capability, the shell element mesh must line up with the solid element mesh so that there is an exact element-to-element correspondence.

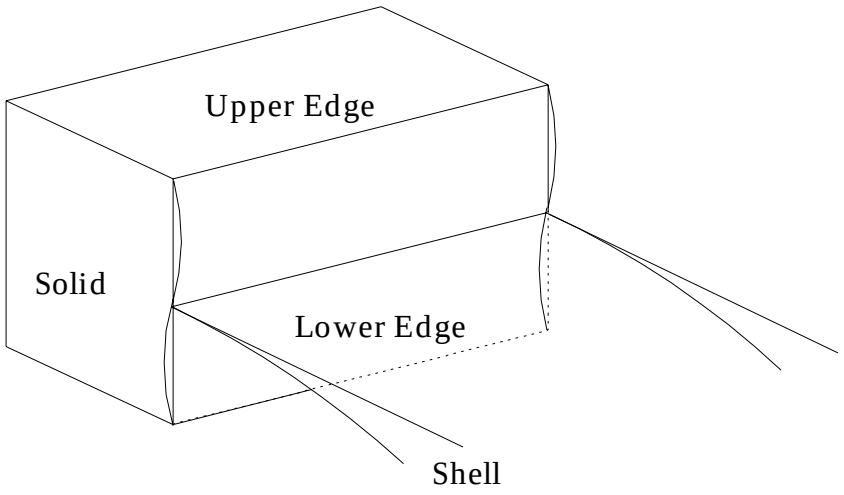
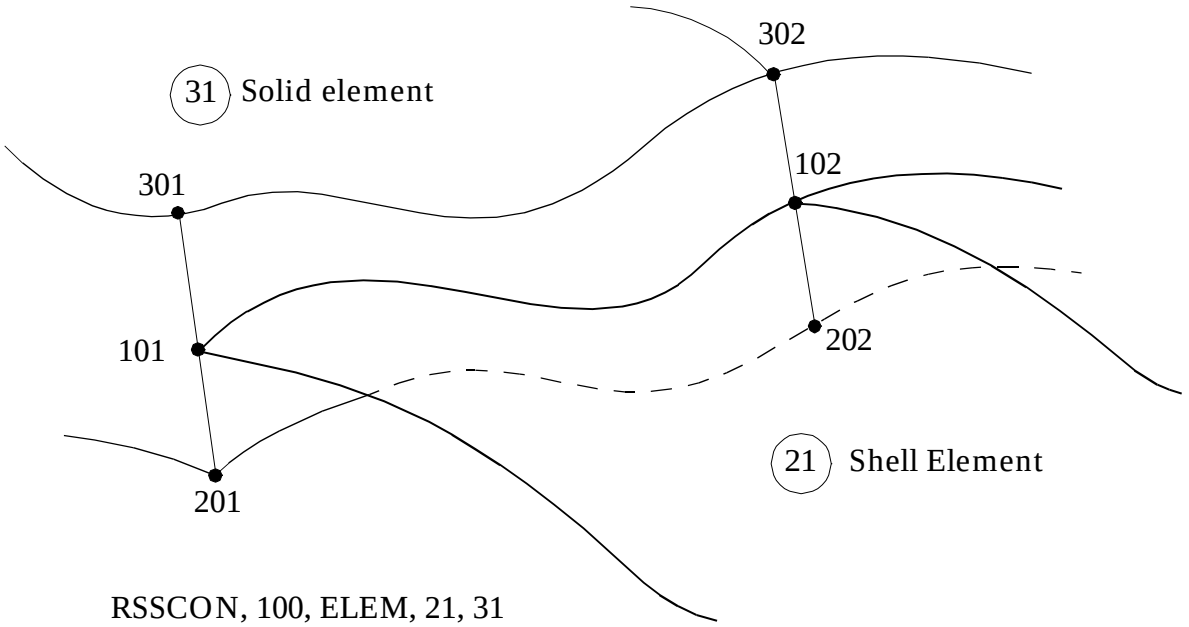


Figure 20 Clamped Connection of a Shell to a Solid

The RSSCON Bulk Data entry supports both p-adaptive elements and h-elements. Through the RSSCON Bulk Data entry, the shell elements CQUAD4, CQUAD8, CTRIA3, CTRIA6, CQUADR, and CTRIAR can be connected to the solid elements CHEXA, CPENTA, and CTETRA. Elements with midside nodes are also supported.

RSSCON generates a multipoint constraint, which puts the shell degrees of freedom in the dependent set (m-set). The three translational degrees of freedom and the two rotational degrees of freedom of the shell edge are connected to the three translational degrees of freedom of the upper and lower solid edge. Poisson’s ratio effects and temperature loads are modeled correctly. The generated multipoint constraints produce six zero-energy modes for rigid-body motion.

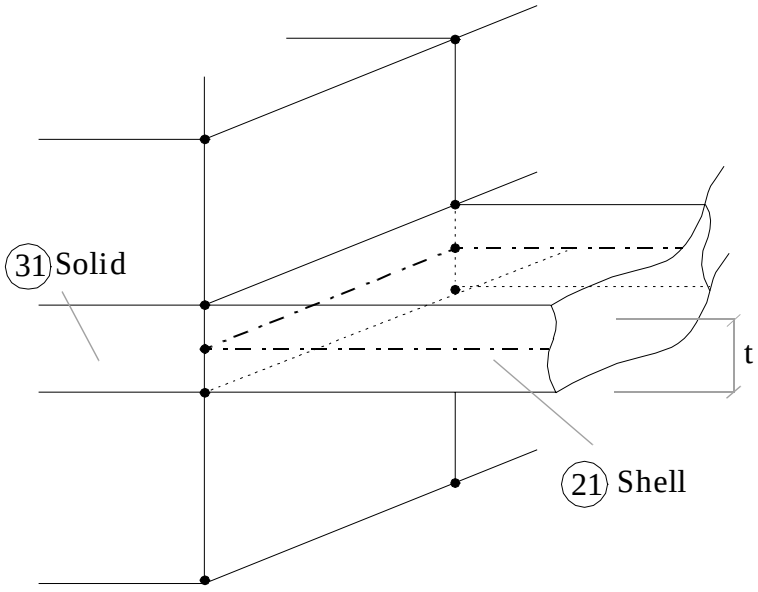
The RSSCON Bulk Data entry defines the connection of a shell element to a solid element. Details of the RSSCON Bulk Data entry are described in the *MSC.Nastran Quick Reference Guide*. Within MSC.Nastran, however, there are two options for making this connection using the RSSCON Bulk Data entry, as shown in **Figure 21**:



RSSCON, 100, ELEM, 21, 31
or
RSSCON, 200, GRID, 101, 201, 301, 102, 202, 302

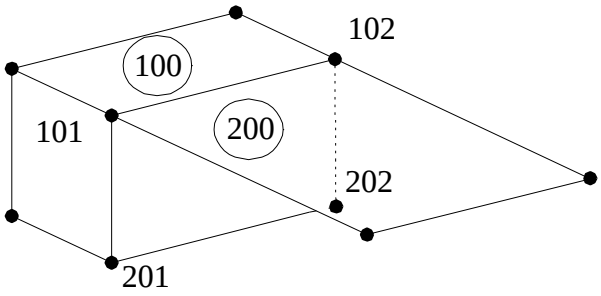
Figure 21 ELEM and GRID Option on the RSSCON Bulk Data Entry

The best modeling practice is illustrated in **Figure 22**. The height of the connected solid element should be chosen equal to the thickness of the shell. All other acceptable mesh topologies are shown in **Figure 23**. If the height of the connected solid element is much larger than the thickness of the shell element, then the connection modeled with RSSCON will be stiffer than the continuum model.



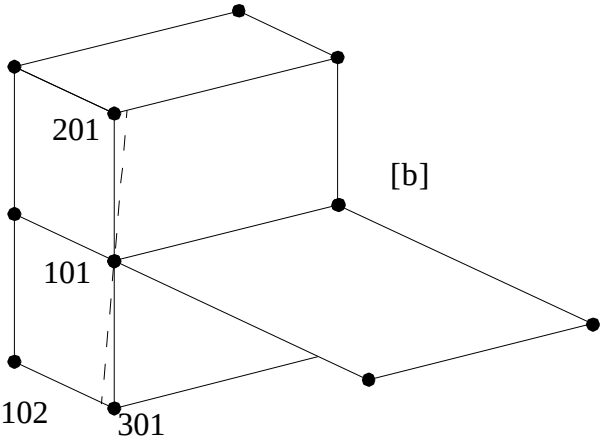
RSSCON, 100, ELEM, 21, 31

Figure 22 Best Modeling Practice for RSSCON

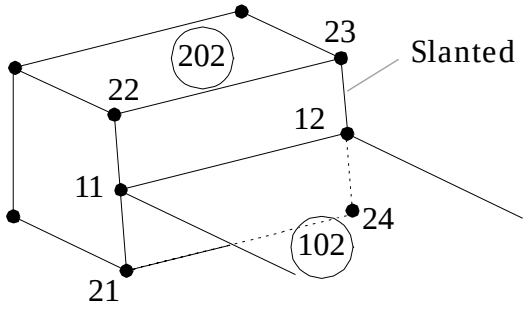


[a]

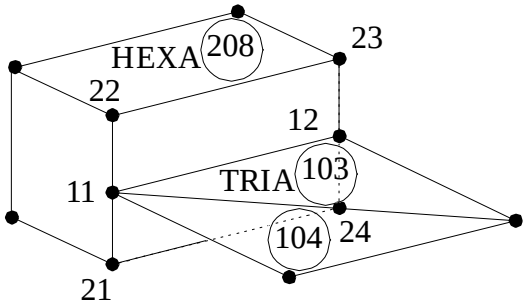
RSSCON, 200, GRID, 101, 201, 101, 102, 202, 102
or
RSSCON, 200, ELEM 200, 100



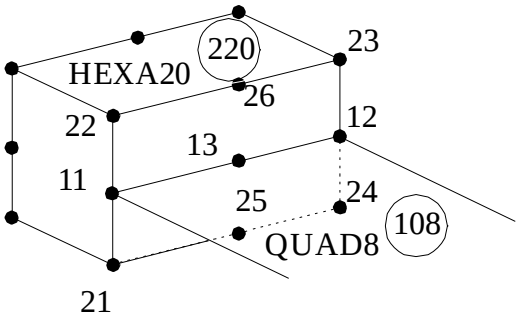
RSSCON, 100, GRID, 101, 201, 301



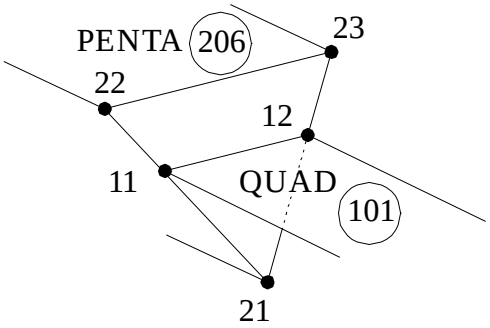
RSSCON, 100, ELEM, 102, 202
or
RSSCON, 100, GRID, 11, 21, 22, 12, 24, 23



RSSCON, 200, ELEM, 103, 208
RSSCON, 201, ELEM, 104, 208 (not necessary)
or
RSSCON, 200, GRID, 11, 21, 22, 12, 24, 23



RSSCON, 200, ELEM, 108, 220
or
RSSCON, 200, GRID, 11, 21, 22, 12, 24, 23
RSSCON, 201, GRID, 13, 25, 26



RSSCON, 200, ELEM, 101, 206
or
RSSCON, 200, GRID, 11, 21, 22, 12, 21, 23

Figure 23 Modeling Options with RSSCON

As mentioned previously, the RSSCON internally generates MPC equations with the grid points on the shell elements as being dependent. For this reason, a shell grid point must not be connected to more than one RSSCON, since a dependent DOF defined on one r-type

element may not be defined as dependent on another r-type element. For further details regarding the RSSCON, see the *MSC.Nastran Linear Analysis Static User's Guide* and the *MSC.Nastran Release Guides* for V69, V69.1, and V70.

If I Apply a Transverse Load on a Cantilever CBEAM, Why Is Axial Shortening Not Present?

When a cantilever beam is loaded in the transverse direction, as shown in **Figure 24** there is an axial deformation when the beam deflects. This deformation is a higher order effect and is not included in the small displacement theory used to formulate finite elements for linear analysis.

The matrices for finite elements in linear analysis do not include secondary effects, such as the shortening of the cantilever beam shown. If you look at the stiffness matrix for the CBAR or CBEAM element, you can see that there is no coupling between bending and axial deformation. Therefore, no matter how large the load becomes, linear theory does not allow the "shortening" effect. If the shortening effect is desired, a nonlinear analysis must be used. In this case, large displacement theory is available, and the element matrices are updated as the structure deforms. The "shortening" terms will show in the solution.

An important point should be made at this point. In a linear analysis, the orientation of a loading never changes with deformation. This is based on the small displacement theory used. If the orientation of a loading changes with deformation (such as a pressure load), these effects are not included in linear analysis and once again, nonlinear analysis must be used.

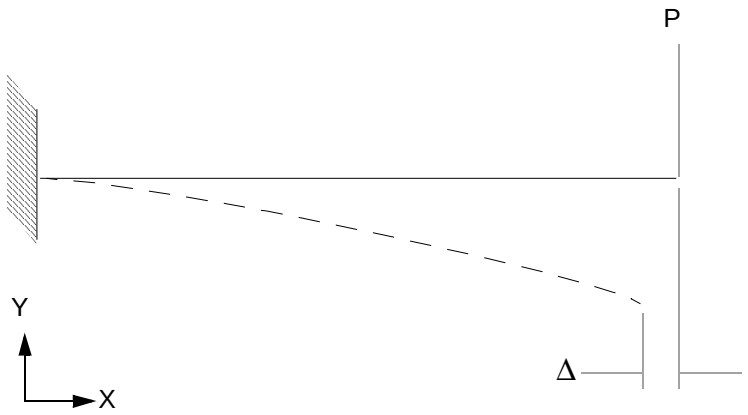


Figure 24 Cantilever Beam Loaded in the Transverse Direction

Linear finite element theory assumes small deflections; therefore the axial shortening term Δ is 0.0. Nonlinear finite element theory allows large deflections and includes the axial motion.

How Do I Use ZOFFS?

Many engineering applications require an offset of a plate from some reference frame. There are three commonly used techniques to define offsets for plate elements in MSC.Nastran: ZOFFS, MID4, and RBAR methods. It is generally recommended that ZOFFS be used to model an offset if you have a sufficiently fine mesh in the region in which offsets are to be defined.

ZOFFS is a field on the connectivity entries for the CQUAD4, CQUAD8, CTRIA3, and CTRIA6 plate elements. A positive value of ZOFFS implies that the element reference plane is offset a distance of ZOFFS from the grid points along the positive z-axis of the element coordinate system. The element coordinate system is attached to the element in its offset position. Refer to the description of the CQUAD4 and PSHELL entries in the *MSC.Nastran Quick Reference Guide*.

The net effect of offsetting a plate is to modify the bending stiffness seen at the grid points relative to the bending stiffness of the plate if there were no offset. Therefore, if ZOFFS is used, the membrane (MID1 field) and bending (MID2 field) properties must be specified on the PSHELL entry, and the membrane-bending coupling (MID4 field) property should be left blank. If the MID2 field is not specified or the MID4 field is used incorrectly, singularities occur for the rotations in the reference plane and ill-conditioned matrices are produced.

If ZOFFS is used, new material properties are calculated to include the effect from the offset as illustrated in the following equations:

$$T \cdot G_1 = t_g \cdot [G_e]_1$$

$$T^2 \cdot G_4 = -z_0 \cdot t_g \cdot [G_e]_1 + t_g^2 \cdot [G_e]_4$$

$$I \cdot G_2 = z_0^2 \cdot t_g \cdot [G_e]_1 - 2 \cdot z_0 \cdot t_g^2 \cdot [G_e]_4 + t_g^3 \cdot \mathrm{F12} \cdot [G_e]_2$$

where:

G_j = material property (MIDj field of the PSHELL entry) after calculation for the ZOFFS.

$[G_e]_j$ = element material property (MATi) referred to by the MIDj field of the PSHELL entry.

z_0 = offset value (ZOFFS field of the element connectivity entries).

F12 = bending stiffness parameter (12I/T3 field of the PSHELL entry).

T = membrane thickness (T field of the PSHELL entry).

t_g = average thickness at the grid points (Ti fields of the element connectivity entries).

I = moment of inertia.

How Should I Model a Stiffened Plate?

A stiffened plate is a special concern in finite element modeling. If you model it as it appears (simply using plate elements), the resulting model is probably too stiff and does not transfer loads properly. The stiffener portion of the component resists a load by bending action, which requires a cubic displacement function to model, while the plate elements in membrane action are capable of representing only a linear displacement.

Therefore, using a single plate as a stiffener almost always results in a model that is too stiff. As more elements are used to model the stiffener, the model approaches the correct stiffness.

Consider the idea of a stiffened plate. The most common use of a stiffener is to remove bending loads from a plate that would fail without the stiffener. This case is shown below. The following illustrations show four different approaches to the same problem. The first three samples use plate elements to model the stiffener. As mentioned, this approach is too stiff for most applications; however, ignoring this, look at the results for these samples. The fourth sample uses offset CBAR elements to model the stiffener. This is normally the preferred modeling practice.

The model is a cantilever beam with a moment applied at the free end. **Figure 25** shows the deformed shape when the stiffener is modeled using plate elements and no special modeling is used. The peak stress in this case is almost 50 times the theoretical value. This is caused by the stress concentration at the point at which the moment is applied. At that point, the stiffener plate has no rotational stiffness, so the moment must be taken by the flat plate (flange) portion of the model. If we look several elements further down the model, the local effects are gone, and the load is beginning to be distributed correctly.

The second model is identical to the first, except that PARAM,K6ROT,10000.0 is added. This adds a fictitious stiffness to the plate elements for the in-plane rotational terms and distributes the loading better (see **Figure 26**), but still gives peak stresses much higher than theory.

The third model is identical to the first, only an RBE3 element is used to distribute the load at the free end. The results of this model are much closer to the theoretical answers (see **Figure 27**).

The fourth model is modeled using offset bar elements to model the stiffener. This is the recommended method of modeling a stiffened plate. Note that the CBAR elements are offset, not the plates. **Figure 28** shows the deformed shape.

Looking at the stress results at the root of the beam, there is no noticeable difference between the different models, but the stress at the tip can differ by more than an order of magnitude. This indicates that the local effects caused by the bad modeling at the tip dissipate as we look further away from the affected region; however, in the affected region, the answers may be incorrect.

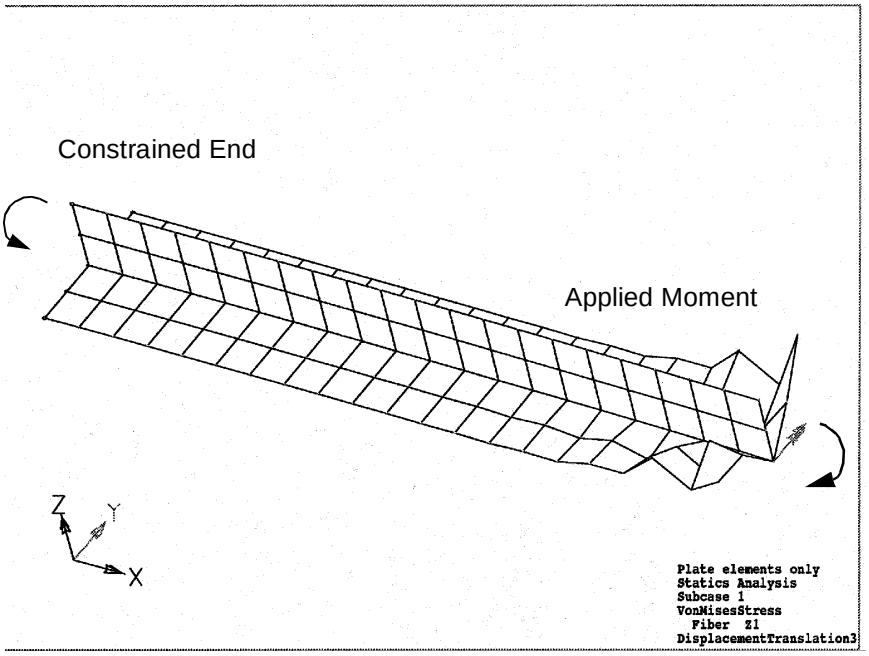


Figure 25 Deformed Shape for Stiffener Modeled with Plate Elements Only

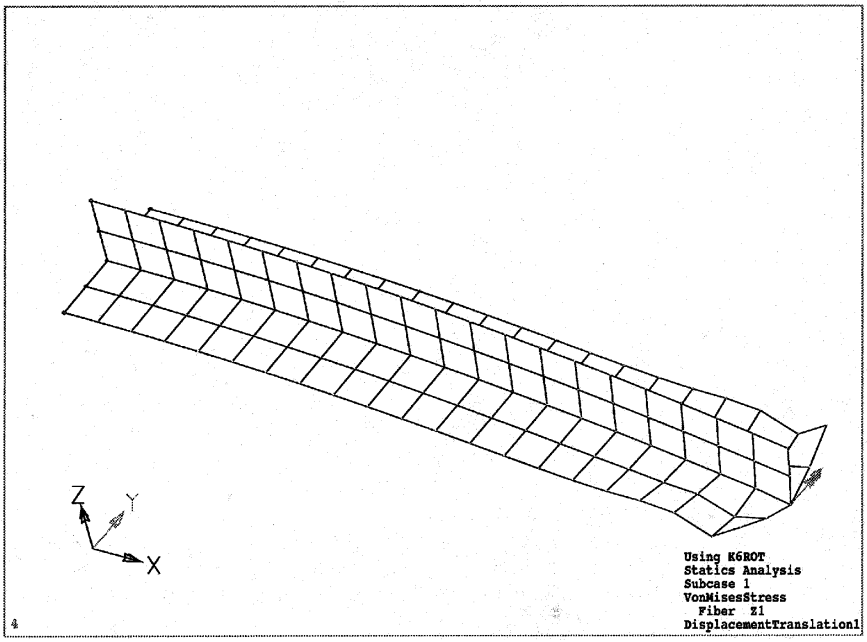


Figure 26 Deformed Shape for Stiffener Modeled with Plate Elements and PARAM,K6ROT

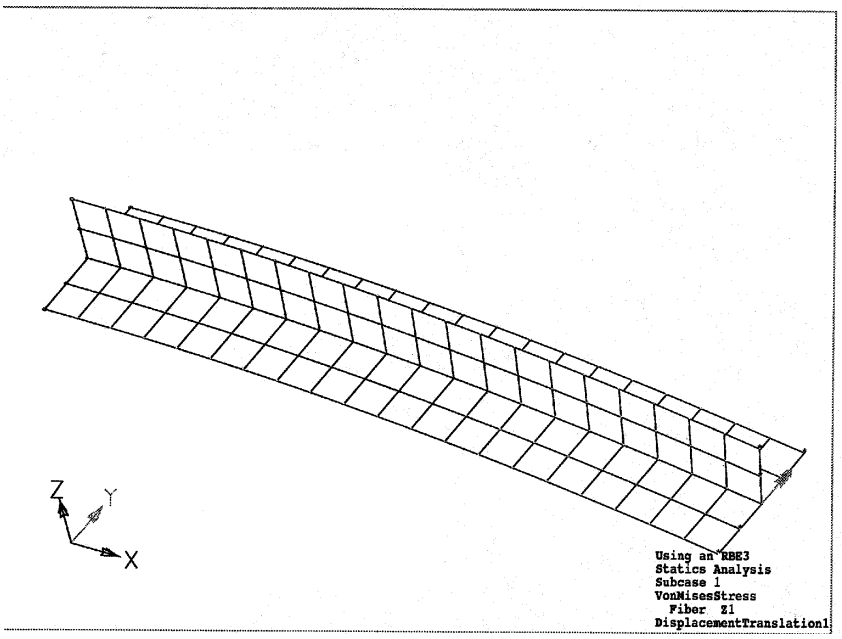


Figure 27 Deformed Shape for Stiffener Modeled with an RBE3 Element at the Tip

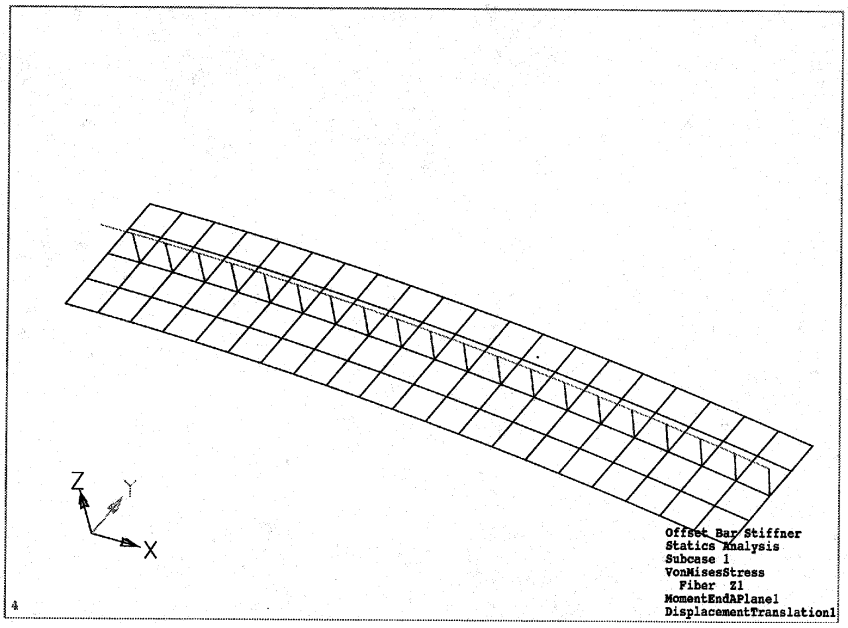


Figure 28 Deformed Shape of Stiffener Modeled with Offset Bar Elements

How Do I Model a Non-Coplanar Fluid-Structure Interface?

The current acoustic design does not allow the modeling of a non-coplanar fluid-structure interface. If you attempt to model a non-coplanar interface, your job will probably fail with one or more of the following error messages:

```
*** SYSTEM FATAL MESSAGE 6160 (GP5D), ELEMENT ID = **** HAS A FACE ON THE FLUID/STRUCTURE BOUNDARY
WHICH IS NOT WELL DEFINED.  THE GEOMETRY SUGGESTS THAT GRID9 LIES ON THE SAME PLANE AS THE FACE.

*** SYSTEM FATAL MESSAGE 6210 (ACMG), AN INTERNAL TABLE HAS BEEN INCORRECTLY CREATED.

*** USER WARNING MESSAGE 6226 (RECTNT), NATURAL COORDINATE CONVERGENCE FAILED AFTER FIVE ITERATIONS.
```

The FSTOL field on the ACMODL entry should not be modified in an attempt to avoid this non-coplanar fluid-structure interface problem because the FSTOL value is used for multiple purposes inside MSC.Nastran. Therefore, the default FSTOL value should be used.

The following modeling technique allows you to model a non-coplanar fluid-structure interface. The top portion of **Figure 29** represents some arbitrary structure. The bottom portion of **Figure 29** represents the fluid. Note that the interface between the structure and the fluid are both non-matching and non-coplanar.

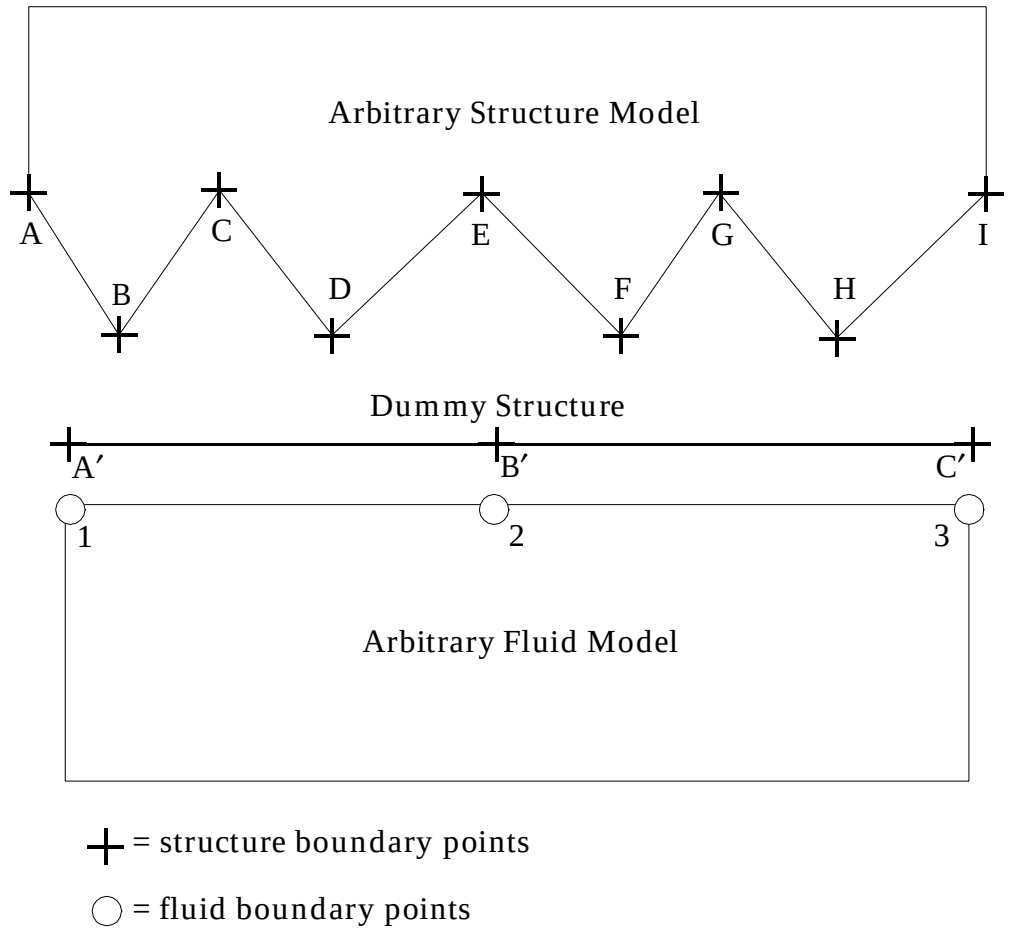


Figure 29 Non-Coplanar Fluid/Structure Interface

The solution is to create an additional dummy structure with a coplanar interface that matches the fluid interface as shown in **Figure 30**.

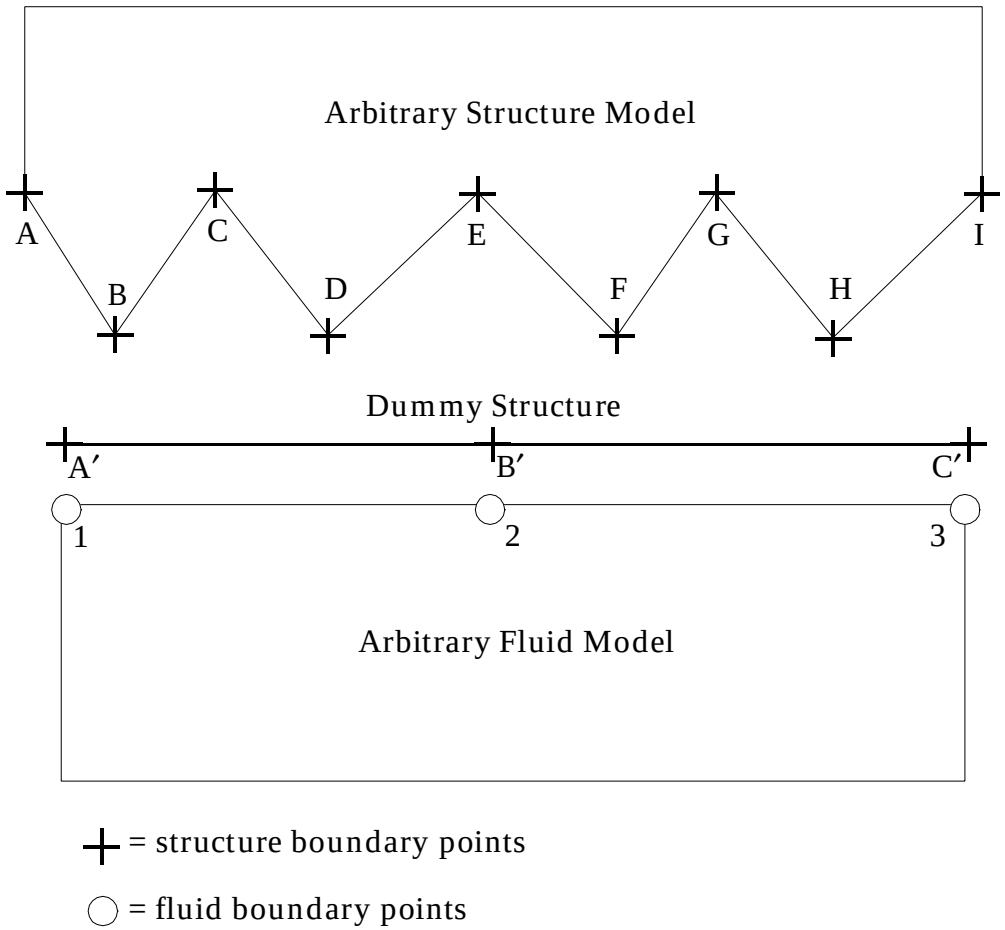


Figure 30 Modeling of Non-Coplanar Fluid/Structure Interface

To further explain the above example, grid points 1 and A', grid points 2 and B', and grid points 3 and C' are separate pairs of grid points that occupy identical locations. The dummy structure should be assigned very small properties as compared to the actual structure. This dummy structure is then connected to the real structure with rigid elements (RBE3s, MPCs, etc.).

Static Analysis

How Do I Get a Printout of the OLOAD, SPCFORCE, and MPCFORCE Resultants?

The OLOAD Resultants can be obtained by setting the parameter PARAM,PRTRESLT to YES, which is the default. To get the SPCFORCE and MPCFORCE Resultants, the corresponding SPCFORCE and MPCFORCE output requests must be included in the Case Control callout, in addition to the above parameter.

How Do I Get a Printout of the Maximum DISPLACEMENTS, APPLIED LOADS, SPCFORCES, and MPCFORCES?

The maximum OLOADS and DISPLACEMENTS can be obtained by setting the parameter PARAM,PRTMAXIM to YES. The default for PRTMAXIM is NO. To get the maximum SPCFORCES and MPCFORCES, the corresponding SPCFORCE and MPCFORCE output requests must be included in the Case Control callout, in addition to the above parameter.

How Do I Use SUBCOM with Thermal Loads?

The SUBCOM/SUBSEQ combination is a way of combining results from previously calculated subcases without recalculations. This combination is quite straightforward for mechanical loads and is documented in the *MSC.Nastran Quick Reference Guide*. The mechanical load case need not be called out in the SUBCOM subcase. However, for thermal loads, the procedure is different. Element stresses and forces are calculated on an element-by-element basis from a knowledge of the displacement vector and the temperature field. Therefore, in a SUBCOM or a SYMCOM, you must supply a definition of the temperature field whenever element stresses and/or forces are requested. The TEMP(LOAD) for the SUBCOM subcase is calculated as follows:

$$T_{SUBCOM} = T_0 + \sum_{i=1}^{n} a_i(T_i - T_0)$$

The new TEMP(LOAD) must be supplied through the standard temperature Bulk Data entries, such as TEMPD, TEMP, etc., and called out in the SUBCOM Case Control Section. This procedure is perhaps best illustrated with an example as follows:

```
$
.
FMS SECTION
.
$
.
EXECUTIVE SECTION
.
$
CEND
$
TEMP(INIT) = 1
$
SUBCASE 1
TEMP(LOAD) = 10
$
SUBCASE 2
TEMP(LOAD) = 15
$
$  THE FOLLOWING SUBCOM WILL COMBINE
$  100% OF THE RESULTS FROM SUBCASE 1
$  WITH 50% OF THE RESULTS FROM SUBCASE 2
$
SUBCOM 3
SUBSEQ = 1.0,0.5
TEMP(LOAD) = 20
$
BEGIN BULK
$
TEMPD,1, 100.
TEMPD,10,150.
TEMPD,15,250.
$
$  “TEMPD,20"  IS CALCULATED AS FOLLOW:
$   100 + 1.0*(150-100) + 0.5*( 250 - 100 )  = 225.
$
TEMPD,20,225.
$
$   REST OF THE BULK DATA ENTRIES
$
ENDDATA
```

Figure 31 Combining Thermal Loads Using SUBCOMs

Note that the appropriate TEMP(LOAD) command must be specified below the SUBCOM command.

What Is Inertia Relief and Why Would I Use It?

Inertia relief is a method used to allow you to model mechanisms and unconstrained structures in the static solutions of MSC.Nastran. Static analysis using the finite element method makes the assumption that a model contains no mechanisms and may not move as a rigid body (strain free). If either of these conditions exist in a conventional finite element analysis, the stiffness matrix for the model is singular, and when MSC.Nastran attempts to decompose a singular matrix (to solve the problem), a singularity occurs at the final degree of freedom (DOF) in the model that describes the unconstrained motion. Once a "divide-by-zero" condition occurs (often shown by UWM 4698), numeric roundoff becomes the dominant part of the solution and worthless answers (if any) result.

Consequently, conventional finite element static analysis cannot be performed on unconstrained structures or mechanisms. However, a method for analyzing these conditions is provided in MSC.Nastran. This method is called inertia relief. In inertia relief, you must provide a list of the DOFs that describe the unconstrained motion and mechanisms using a SUPORT entry. In the case of an unconstrained structure, this means defining six non-redundant DOFs on the SUPORT entry.

A simple description of inertia relief is that the inertia (mass) of the structure is used to resist the applied loadings, that is, an assumption is made that the structure is in a state of static equilibrium even though it is not constrained. Simple samples of this include a spacecraft in orbit or an aircraft in flight. In either case, the structure is in a steady state, although it is capable of unconstrained motion.

In static analysis (SOLs 1 and 101) a SUPORT is necessary whenever unconstrained motion is possible. The easiest way to describe how to use a SUPORT in statics is if you hold the SUPORT DOFs constrained, there will be no possible rigid body motion. This includes mechanisms (SOL 1 only). If all possible rigid body motion is not described on the SUPORT, then the stiffness matrix is singular, and the problem either fails in decomposition or gives unreasonable answers (often in the form of displacements on the order of 10^{11}). The DOFs on the SUPORT entry should describe a non-redundant set. Whenever a SUPORT entry is used in static analysis, the epsilon and strain energy printed in the table from UIM 3035 should be checked to verify this.

In a static run the values printed for epsilon and strain energy in the UIM 3035 table should all be numeric zero. The strain energy printed in this table for matrix KLR represents the strain energy in the model when the associated SUPORT DOF is moved 1.0 unit, while all other SUPORT DOFs are constrained. If the SUPORT is properly specified, the model should be capable of rigid-body motion (strain-free) with no strain energy. The values printed for the strain energy indicate the ability of the model to move as a rigid body. If a SUPORT entry is used, these values should always be checked. If the structure is not constrained, these values should be numeric zero, but roundoff almost always results in a small nonzero value. Acceptable values are a function of the units and the size of the structure. Therefore, a recommended value is not provided in this document.

When inertia relief is specified, MSC.Nastran calculates the forces that result from a rigid body acceleration of the SUPORT DOFs in the specified directions. It then calculates the summation of all applied loadings in the same directions. Then accelerations are applied

to the structure in the SUPORT directions to “balance” the applied loadings. The problem is now in a state of static equilibrium, that is, the summation of all applied loads is 0.0. Since the problem is not constrained, rigid-body displacement is still possible. The next step performed by MSC.Nastran is to constrain the SUPORT DOFs to have a displacement of 0.0 and provide the relative motion of all other grid points to that reference. Hence, the term “reference” DOF is used to describe the SUPORT DOFs in MSC.Nastran. The set of DOFs described on the SUPORT entry belong to the r-set or reference set for the solution. The computed solution is the correct one, and it is relative to any rigid body motion that might be occurring. A simple way to think of this is that the solution coming from MSC.Nastran represents the deformation of the structure you would see if you were standing at the SUPORT DOF.

There are two different approaches to inertia relief in MSC.Nastran. One method is available in SOL 1, and the other is available in SOL 101. The difference is actually internal in MSC.Nastran, and the following is a simple description of the two. The method in SOL 1 is a general approach and is capable of handling mechanisms and unconstrained problems. It is also capable of handling problems that have possible rigid body motion in selected directions.

The implementation in SOL 101 is a more efficient implementation but is limited because it can only handle problems that are fully unconstrained (that is, those with no constraints preventing rigid body motion of any kind and whose model does not contain mechanisms).

The way that inertia relief is activated also varies between the solutions. In both methods, a SUPORT entry is used to list the DOFs that describe the possible mechanisms and rigid body motion. In SOL 101, an additional parameter (PARAM,INREL,-1) must also be specified. If superelements are used, the SUPORT grid points(s) must belong to the residual structure. If the parameter (PARAM,GRDPNT,xx) is used, it must be exterior to all superelements (this can be done using the CSUPEXT entry).

Both methods require that a realistic mass distribution exists, and the DOFs listed on the SUPORT entry must be connected to the model in all specified DOFs. (For example, DOFs 4, 5, and 6 on a grid point with only solid elements attached cannot be used since solid elements have no stiffness in these DOFs.)

An additional special case allowed in SOL 101 is the solution of a problem under uniform acceleration. This problem is posed using the DMIG,UACCEL entry in addition to the previously mentioned requirements. In this case, uniform accelerations are applied to the model, and the solution is found. This is useful for situations such as spacecraft liftoff and landing loadings, which are often specified as static accelerations.

Stress Analysis

What Is the Grid Point Stress Capability?

The grid point stress request causes stresses at grid points to be calculated from the adjoining plate and solid elements. Grid point stress uses the following Case Control commands:

GPSTRESS	Requests grid point stresses for printing
STRFIELD	Requests grid point stresses for graphical postprocessing and stress discontinuities
STRESS or ELSTRESS	Requests element stress output
OUTPUT(POST)	Delimiter
SET	Defines a set of elements that define a surface or volume
SURFACE	Defines a surface of plate elements
VOLUME	Defines a volume of solid elements

The OUTPUT(POST) section applies to all subcases. In other words, the SURFACE(s) and VOLUME(s) apply to all subcases.

You need to request enough element stresses to cover the SET definition in the OUTPUT(POST) section; in other words, the stresses must be calculated for all elements that attach to the grid points for which grid point stresses are requested. Consider **Figure 32** below.

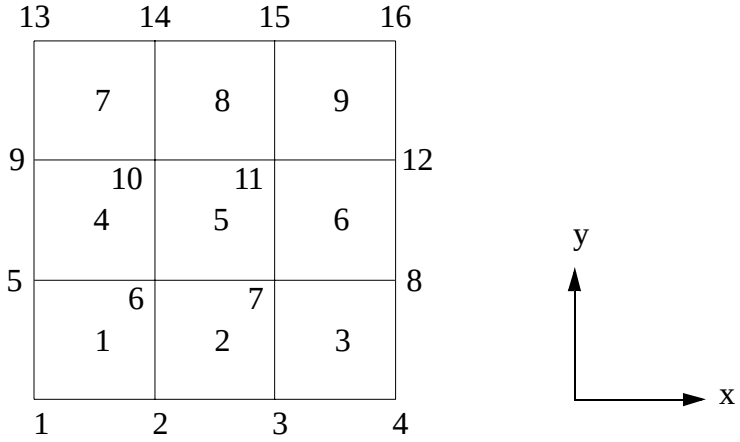


Figure 32 3 by 3 Mesh

If grid point stresses are requested for grid point 11, then element stresses must be requested for a set that includes at least elements 5, 6, 8, and 9. In addition, the set of elements in the SURFACE definition must also include elements 5, 6, 8, and 9. The Case Control Section for this example is shown below:

```
CEND
SET 99 = ALL
ELSTRESS = 99
SET 98 = 1
GPSTRESS = 98
SUBCASE 1
  LOAD = 100
SUBCASE 2
  LOAD = 200
OUTPUT(POST)
  SET 97 = 5,6,8,9
  SURFACE 1  SET 97  NORMAL Z
BEGIN BULK
```

Figure 33 Example – Case Control Section

The SET defined in the OUTPUT(POST) section must be equal to (or a subset of) the SET for which element stresses (ELSTRESS) are computed. If the SET defined in the OUTPUT(POST) section exceeds the SET for which element stresses are computed, UWM 4621, "NO STRESS DATA IS AVAILABLE FOR ELEMENT ***" is written. Further information about grid point stress can be found in the *MSC.Nastran Reference Manual* and the *MSC.Nastran Linear Analysis Static User's Guide*.

How Can I Compute Stress Output for CQUAD4 Elements in a Specified Coordinate System for Linear Static Analysis?

Stresses in QUAD4 elements can be rotated to align with a specified coordinate system by the following procedure:

1. Specify the desired coordinate system via a CORDi Bulk Data entry. (This is required even if the desired coordinate system is the basic coordinate system.)
2. In field 5 of the MAT1 continuation entry, specify the desired coordinate system ID. (This is field 2 of the second MAT2 continuation entry.)
3. Insert PARAM,CURV,1 and PARAM,S1M,1 in the Bulk Data input.

This stress transformation is not available for the corner option. For more information, see parameters CURV and S1i in the *MSC.Nastran Quick Reference Guide.*

How Do I Perform Stress Filtering in Transient Analysis?

MSC.Nastran Version 69 introduces a new stress filtering method based on element groups that can be used in all the solution sequences that already support pre-Version 69 output filtering methods. In addition, the element group option can be used in SOLs 109 and 112, the solution sequences for transient response analysis. Note, however, SOLs 109 and 112 do not support any other filtering options besides stress filtering based on element groups.

Stress filtering can be activated in a transient analysis by adding the following entries to the Bulk Data Section:

PARAM,S1,1
PARAM,NUMOUT,0
PARAM,BIGER,x

where x is the stress threshold to be filtered.

The stress filtering enhancement extends the logic so that if any selected stress is greater than BIGER, its entire element group of records is retained; otherwise, the group is deleted. For SORT1 output an element group is defined as all the element records at a given time step. For SORT2 output an element group is defined as all the element records for a given element at all time steps. This option enables you to greatly reduce the size of the output file.

An example problem—consisting of four elements (elements 12, 23, 34, and 45) and ten time steps—is used to illustrate the filtering options. A value of 300.0 is used for BIGER (param,biger,300.). For SORT2 output, only elements 12 and 23 are printed in the output (**Figure 34**). Elements 34 and 45 do not have any stress values that exceed 300.0. For SORT1 output, only two time steps (t = 0.6 and t = 0.8) are printed (**Figure 35**). The other eight time steps do not any stress values that exceed 300.0.

```
VERIFICATION PROBLEM.  TRAN. RESP. OF RODS WITH SPC    FILTER2
     ELEMENT-ID =      12
S T R E S S E S   I N   R O D   E L E M E N T S      ( C O N R O D )
                     AXIAL       SAFETY      TORSIONAL     SAFETY
       TIME         STRESS       MARGIN        STRESS      MARGIN
    .0             .0                        .0
   1.000000E-01   1.216969E+01                .0
   2.000000E-01   1.622681E+02                .0
   3.000000E-01  -1.574618E+01                .0
   4.000000E-01  -2.931780E+02                .0
   5.000000E-01   1.642058E+01                .0
   6.000000E-01   4.003575E+02                .0
   7.000000E-01  -1.253635E+01                .0
   8.000001E-01  -4.897087E+02                .0
   9.000001E-01   4.807502E+00                .0

     ELEMENT-ID =      23
                     AXIAL       SAFETY      TORSIONAL     SAFETY
       TIME         STRESS       MARGIN        STRESS      MARGIN
   1.000000E-01   3.149272E+01                .0
   2.000000E-01   1.234668E+02                .0
   3.000000E-01  -2.478340E+01                .0
   4.000000E-01  -2.306717E+02                .0
   5.000000E-01   1.594586E+01                .0
   6.000000E-01   3.227227E+02                .0
   7.000000E-01  -6.780667E+00                .0
   8.000001E-01  -4.015388E+02                .0
   9.000001E-01  -2.173207E+00                .0
```

Figure 34 SORT2 Output Example

Note that only elements 12 and 23 are listed. The other two elements do not have any stress values that exceed BIGER.

```
VERIFICATION PROBLEM.  TRAN. RESP. OF RODS WITH SPC    FILTER1
      TIME =  6.000000E-01
S T R E S S E S   I N   R O D   E L E M E N T S      ( C O N R O D )
       ELEMENT       AXIAL       SAFETY      TORSIONAL     SAFETY
         ID.        STRESS       MARGIN        STRESS      MARGIN
            12    4.003575E+02                .0
            23    3.227227E+02                .0
            34    1.868571E+02                .0
            45    7.054543E+01                .0

      TIME =  8.000001E-01
S T R E S S E S   I N   R O D   E L E M E N T S      ( C O N R O D )
       ELEMENT       AXIAL       SAFETY      TORSIONAL     SAFETY
         ID.        STRESS       MARGIN        STRESS      MARGIN
            12   -4.897087E+02                .0
            23   -4.015388E+02                .0
            34   -2.417573E+02                .0
            45   -9.110515E+01                .0
```

Figure 35 SORT1 Output Example

Note that only two time steps are output, although ten are calculated. The other time steps do not have any stresses that exceed BIGER.

How Can I Perform Stress Sorting Using MSC.Nastran?

Stress sorting is controlled by the parameter S1. A value of PARAM,S1,1 enables stress sorting. Additional parameters controlling stress sorting are NUMOUT, BIGER, SRTOPT, and SRTELTYP. A description of these parameters is provided in the *MSC.Nastran Reference Manual*.

The following is an abbreviated description of the function of these parameters:

NUMOUT A positive value of NUMOUT controls the number of elements that is printed after sorting. For example, if PARAM,NUMOUT,20 is specified, MSC.Nastran responds by printing only the 20 elements with the highest stresses.

NUMOUT = 0 outputs all elements in a group when one or more exceeds BIGER. NUMOUT = 0 does not sort but filters according to BIGER by element group.

BIGER Cut-off value. For example, if PARAM,BIGER,30000.0 is specified, then only elements whose sorted stress exceeds 30000 are printed. This is a handy filter to obtain printout of highly stressed elements only. (Be careful-if NUMOUT and / or BIGER are used, and you are using a postprocessor, only the stresses printed in the output are available for the postprocessor.)

SRTOPT	Option controlling the sorting method. The default (0) sorts the output based on the largest magnitude. Options are available to sort based on minimum magnitude and maximum or minimum algebraic value.
SRTELTYP	Element type for stress sorting. If sorting is desired for only one element type, then the element type (from the *MSC.Nastran Quick Reference Guide*) is specified here. The default is for all element types.

For a typical problem, simply specifying PARAM,S1,1 and PARAM,NUMOUT,-1 enables stress sorting. The other parameters are optional.

It should be pointed out that these parameters can be placed in the Case Control Section of the input, thereby allowing different options for different subcases.

As implemented in MSC.Nastran, stress sorting is performed within each subcase and by element type. That is, the results for each SUBCASE are sorted individually and the results for each element type within the SUBCASE are sorted individually.

Only one stress value is sorted in any run. The default stress quantity for each element type is shown in the table with the DTI,INDTA Bulk Data entry included in the *MSC.Nastran Quick Reference Guide*. For example, for a CQUAD4 element, the default stress quantity is von Mises stress on the Z2 surface.

The selected value may be changed by using the DTI,INDTA entry in the Bulk Data. This entry allows you to select which stress quantity is used for sorting by element type. The format for this entry and the possible quantities that can be selected for sorting are described in the *MSC.Nastran Reference Manual*.

How Can I Compute “Running Loads”?

Running loads are defined as loads along the edge of elements. There is an option to obtain the grid point force balance output as running loads.

The default format is in grid-point-oriented format. Forces and moments (both applied and element related) are calculated and printed using the “global” coordinate system of the grid points. This is performed by requesting GPFORCE for a selected set of grid points in the Case Control Section.

A second set of output is available and is controlled by parameters NOELOF and NOELOP. These two parameters have a default value of -1, which disables the second set of output. If these parameters are given positive integer values, then a second set of output occurs that contains grid point forces aligned with the edges of the elements (PARAM,NOELOF,1) and additional output containing the sum of the forces parallel to the edges of the elements (PARAM,NOELOP,1).

A sample input file to generate this output is as follows:

```
SOL 101
TIME 10
CEND
TITLE = TEST OF ELEMENT-ORIENTED OUTPUT
SUBCASE 1
LOAD = 1
SPC = 1
GPFORCE = ALL
BEGIN BULK
PARAM,NOELOF,1
PARAM,NOELOP,1
.
.
.
ENDDATA
```

Figure 36 Sample Input File

The following is a simple cantilever plate problem with 10-pound forces in the y-direction at grid points 11 and 33 (see **Figure 37**).

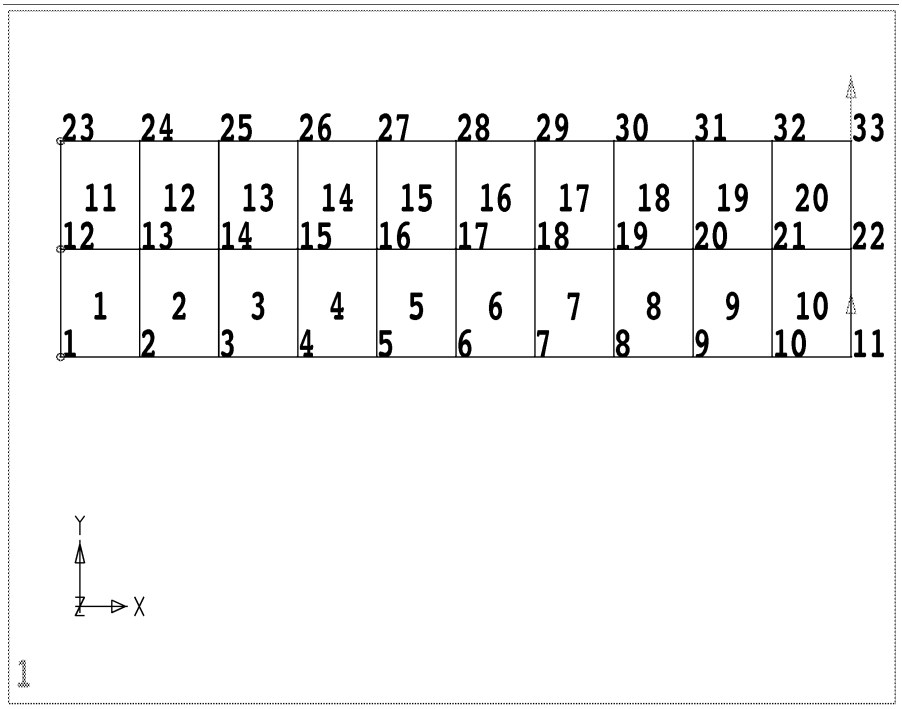


Figure 37 Simple Cantilever Plate Problem

The output for grid point 1 appears in **Figure 38** as follows:

1

G R I D P O I N T F O R C E B A L A N C E

POINT-ID	ELEMENT-ID	SOURCE	T1	T2	T3	R1	R2	R3
1		F-OF-SPC	-1.000000E+02	-1.805020E+01	.0	.0	.0	.0
1	1	QUAD4	1.000000E+02	1.805020E+01	.0	.0	.0	.0
1		*TOTALS*	.0	.0	.0	.0	.0	.01

E L E M E N T I N T E R N A L F O R C E S A N D M O M E N T S
(INTERNAL ACTIONS FROM REFERENCE-POINTS TO LOAD-POINT)

LOAD POINT-ID	ELEMENT -ID	ELEMENT TYPE	REFERENCE POINT-1	FORCE-1	MOMENT-1	REFERENCE POINT-2	FORCE-2	MOMENT-2	KICK-LOAD
1	1	QUAD4	2	1.000000E+02	.0	12	1.805020E+01	.0	.01

S U M M A T I O N O F E L E M E N T O R I E N T E D F O R C E S O N A D J A C E N T E L E M E N T S
(ONE AND TWO DIMENSIONAL ELEMENTS ONLY)

POINT-ID	ORIENT-ID	TENSION =(+)
1	2	1.000000E+02
1	12	1.805020E+01

Figure 38 Output for Grid Point 1

The first set is the standard output for the grid point force balance. The second set is the element-oriented forces (NOELOF), and the final set is the sum of the forces oriented along the elements.

Normal Modes

What Real Eigensolvers Are Available in MSC.Nastran?

Tracking Methods. MSC.Nastran has several real eigenvalue extraction methods including inverse power (INV), inverse power with enhancements (SINV), Givens, modified Givens, automatic Givens, Householder, modified Householder, automatic Householder, and the Lanczos method. The INV method is an older method and should rarely be used, if ever. It can miss modes yet inform you that all of the roots in the range were found. It is maintained for compatibility reasons only. The SINV method is the modern version of the INV method. It is much more reliable and is effective when the model is large and only a few modes are needed. However, it is costly when more than a few modes are needed.

Transformation Methods. The Givens method, the Householder method and their modified forms are suited for small models that fit into memory but these methods are prohibitively expensive for moderate to large size problems that do not fit into memory. The Givens (GIV) and the Householder (HOU) methods cannot be used on singular or near singular mass matrices. To overcome this limitation, modified forms of these solvers, the modified Givens (MGIV) and modified Householder (MHOU) methods, were developed. Commonly these modified methods are used in all situations just in case the mass matrix is poorly conditioned. However, the modified methods require additional processing and the lower order modes may be less accurate than those obtained using the unmodified versions. It is for this reason that the automatic methods AGIV and AHOU were developed. These methods inspect the mass matrix and determine if the modified method is needed without you having to decide.

Many use these methods in conjunction with a reduction technique, such as Guyan reduction or generalized dynamic reduction (GDR). These reduction methods reduce the complete model to a smaller number of degrees of freedom that approximate the dynamic characteristics of the full model. The eigenvalues of the reduced models are usually obtained with the modified Givens or modified Householder method. While the reduction provides a cost-effective method for performing eigenvalue extraction for large models, it can produce poor results and can miss modes. Always scrutinize the results when using a reduction procedure to ensure that no modes were missed. When GDR is used, it is easy to determine whether any modes were missed by using the Sturm sequence User Information Message 5010. However, when Guyan reduction is used in conjunction with GDR, the Sturm sequence check is not performed on the full model.

Sturm Sequence Check. To determine the actual number of modes in the frequency range of interest, it is common to perform an eigenvector extraction using SINV with a very narrow frequency range specified. The upper bound on the EIGR entry should correspond to the highest frequency of interest. MSC.Nastran should be stopped after the first Sturm sequence check or just give MSC.Nastran enough time to perform only one check. The information obtained will help greatly during model verification.

Lanczos Method. The Lanczos method was implemented in MSC.Nastran because of the limitations of the other methods. The Lanczos method combines the best features of all of the other solvers. It is robust, it does not miss roots, and the eigenvalues and eigenvectors are accurate. If Lanczos cannot find all roots in the range you specify, user diagnostics are printed. The Lanczos method is the preferred method for all medium- or large-sized problems.

Describe the Lanczos Method

The Lanczos method is implemented as a block shifted method. In this method a trial eigenvalue is assumed, and an attempt is made to extract all of the eigenvalues close to this value. The trial eigenvalues are referred to as shift points. It is called a block method because it extracts several eigenvectors within a frequency block close to the trial eigenvalue. A Sturm sequence check is made at each of the shift points to determine the number of eigenvalues below that shift point. This information is used to determine when all of the eigenvalues of interest have been found.

Several error messages that may result from the Lanczos eigenvalue solver are enhanced to help you determine the cause of the problem should a fatal error occur. One fatal error that can occur using the Lanczos method is User Fatal Message 5299. This is often caused by a massless mechanism, such as a grid point floating in space or a BAR not connected to the rest of the structure in torsion. Another potential error, which may or may not indicate a modeling problem, may occur when a shift point is chosen too close to an actual eigenvalue.

If UFM 5299 occurs and massless mechanisms are not present in the model, try adjusting the frequency range on the EIGRL entry. This adjustment forces the Lanczos method to use different shift points that may result in a better numerically conditioned solution. If rigid body modes are not present or are not needed, it is sometimes helpful to increase V1 to a small positive number. If the Lanczos method cannot find all of the roots after modifying V1 and V2 several times, this generally indicates a modeling problem.

Which Real Eigensolution Method Should I Use?

MSC.Nastran provides a variety of real eigensolution methods and you must decide which is the best for your application. The best method for a particular model depends on three factors: the size of the model, the number of eigenvalues desired, and the available real memory of your computer. In general, the Lanczos method is the most robust and is the recommended choice if only one method is to be used for moderate to large models.

The SINV method is sometimes the fastest when the model is too large to fit into memory, only a few modes are needed, and a reasonable search range can be specified on the EIGR entry. For other large problems, Lanczos is the preferred method.

How Do I Use the Sturm Sequence Check?

The Sturm sequenc is a mathematical tool that is used to determine the number of eigenvalues that exist within a certain frequency range when performing real eigenvalue analysis. The procedure involves picking a trial eigenvalue λ_i and forming the matrix

$[K - \lambda_t M]$. When this matrix is decomposed into the triangular factors, the number of negative terms on the factor diagonal is equal to the number of eigenvalues below λ_t. The Sturm sequence check is extremely reliable. The information message should be checked by you, especially when using generalized dynamic reduction (GDR). The check is also performed when using the SINV and the Lanczos methods. The Sturm sequence check can be requested for the INV method using DIAG 16.

The Sturm output is generated as a User Information Message 5010. A typical output is shown in **Figure 39**.

```
*** USER INFORMATION MESSAGE 5010, STURM SEQUENCE DATA FOR EIGENVALUE EXTRACTION
     TRIAL EIGENVALUE =  1.579137D+08, CYCLES -  2.000000D+03 NUMBER OF EIGENVALUES BELOW THIS VALUE  =     3
```

Figure 39 Sturm Sequence Check Ouput Message

For the message shown in **Figure 39**, a trial eigenvalue of 1.579×10^8, which corresponds to a frequency of 2000 Hz, was picked by MSC.Nastran to perform the Sturm sequence check. The number of eigenvalues below this is three. Note that this check is performed before any eigenvalues have been extracted. If the number of eigenvalues found below 2000 Hz is less than the number indicated by the Sturm check (which is three for this case), some modes were not found, and you will need to investigate the problem.

How Do I Select an Eigenvalue Range?

The eigenvalue range you select depends on the analysis to be performed and the loads applied on the structure. If the normal modes analysis is the only analysis to be performed, the task of selecting an appropriate range is fairly straightforward: choose a range that computes the characteristics of the dynamics of the structure. If the analysis modes are to be compared to test modes, select an upper range somewhat higher than the highest test frequency. Do not set the upper bound at the highest test frequency. Your model may be too stiff, causing an important mode to be missed.

If the normal mode analysis is used in conjunction with designing a structure to a dynamic specification, it is often necessary to perform the analysis with an upper bound higher than the highest frequency of the specification. For instance, suppose you are designing a component that is subjected to a sine dwell at various frequencies between 20 and 2000 Hz. It is important to compute all of the modes that will be excited by the sine dwell input. A mode at 2001 Hz will be excited by a 2000 Hz input load. However, if you use an upper bound of 2000 Hz, this 2001 Hz mode will not even be included in the analysis.

When additional dynamic analysis (such as transient response analysis or frequency response analysis) is to be performed, the task of choosing a frequency range is more difficult. Your frequency range should be specified to ensure that the structure responds dynamically to the input loads. This requires knowledge of the structure and the frequency content of the input loading. As a general rule, the upper frequency of the modal analysis should be two to ten times that of the highest dominant frequency of the input depending on the structure's damping. If there is any doubt as to whether enough modes were included (which is often the case), include more modes and rerun the analysis. If the results do not change, you have gained confidence in the frequency range chosen.

Likewise, if the analysis times are excessive, try reducing the number of modes; if the results do not change, consider using the reduced number of modes. It is also possible to print or plot the response of the modes themselves using the SDISPLACEMENT command. If the response in the higher models is small, the higher modes may be safely ignored. For related topics, see the sections on "*Describe the Mode Acceleration Method*" and "*Describe the Residual Vector Method*."

Why Do I Sometimes Miss Roots When Using GDR?

It is very important for you to examine the printed output for User Warning and User Information Messages when the GDR method is used. A Sturm Sequence Message (User Information Message 5010) is issued in the output identifying the number of modes that exist below the frequency selected on the DYNRED entry. If the structure has clusters of closely spaced modes and the NIRV parameter on the DYNRED entry is improperly selected, the GDR procedure may miss some of these modes. Unless you are aware of this condition prior to the analysis, the default value for NIRV may lead to missing modes. However, examining the printed output messages will identify this situation although no direct messages are issued.

The GDR method is an approximate method. It is more accurate than the Guyan reduction it was intended to replace, but it is still an approximate method. If exact answers are desired, the Lanczos method should be used.

What Are Rigid Body Modes?

Rigid body modes are similar to mechanisms since they may be described as motion without any strain occurring. For each rigid body mode, there is no relative displacement between the grid points.

A simple example of rigid body motion occurs when you move a pencil from one location on your desk to another. In this case, there is no strain occurring in the pencil, only translation and rotation as a rigid body.

In statics, the presence (or possibility) of rigid body modes results in a singularity in the stiffness matrix and causes a failure in the solution of the problem (during matrix decomposition). This can be prevented either by applying constraints (if no rigid body motion should occur) or using inertia relief (if the structure is truly unconstrained).

In dynamics, rigid body modes are a common occurrence. Examples of this are aircraft in flight and spacecraft in orbit. In these cases, the possible rigid body motion is a part of the solution and may even be important. In fact, constraining a model to remove the rigid body modes changes the dynamic properties and response of the model.

When rigid body modes are mentioned, many people expect to see "clean" modes. That is, they expect to see the first one being motion of the structure in the x-direction, followed by the second one being in the y-direction, and so on. This situation is rarely the case (unless a SUPORT entry is used to ensure this). Rigid body modes are an example of an interesting feature of dynamics known as "repeated roots," that is, they all occur at the same frequency (0.0 Hz).

In dynamics, when repeated roots occur, the eigenvectors calculated are a linear combination of all eigenvectors that occur at that frequency. As engineers, we like to think of "clean" rigid body modes, that is the first rigid body mode is x-translation of the structure, etc. Mathematically, this is possible, but not probable. Normally the rigid body modes that are calculated are a combination of the "clean" rigid body modes ($\alpha_1 \cdot$ unit translation x + $\alpha_2 \cdot$ unit translation y + …).

Since the rigid body modes calculated by the computer are rarely "clean" modes, it is often a good idea to verify the modes. This procedure can be as simple as looking at the eigenvalues (which should be numeric zero) or may involve detailed checking, such as plots and checking element strain energy. In a true rigid body mode, the strain is zero; therefore the element strain energy should also be zero. Due to round-off in the computer, a true zero is rarely obtained. Instead, we often obtain a "numeric" zero that requires engineering judgement to determine if the value is really zero or not. You may accept 10^{-3} as zero in one situation, while insisting on 10^{-11} in another. The value you are willing to accept is a function of the units you are using, the physical size of the structure you are modeling, and the complexity of the model itself.

Dynamic Response Analysis

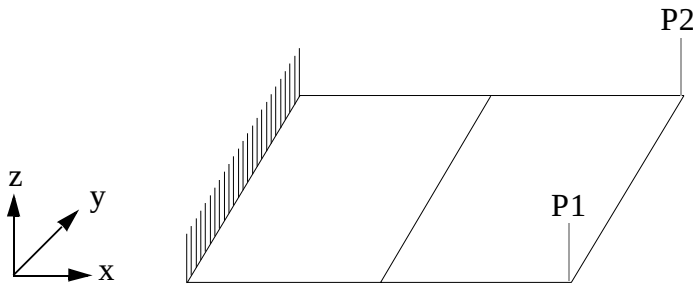
How Do I Apply Dynamic Loads?

The dynamic solutions in MSC.Nastran belong to one of two categories, either transient response analysis or frequency response analysis. In transient response analysis, the goal is to determine the response of the structure as a function of time due to a time-dependent load. The analysis is performed in the time domain, that is, the loads are defined as a function of time and all of the output is a function of time. The transient response analysis Structured Solutions Sequences are SOLs 109 and 112.

The purpose of a frequency response analysis is to determine the steady-state response of the structure to a sinusoidal input at a set of discrete frequencies. Since the structure is considered linear in the frequency response analysis, the response to the sinusoidal input is also sinusoidal and has the same frequency (though a different phase) as the input. The frequency response Structured Solutions Sequences are SOLs 108 and 111. In both transient response and frequency response, the input load may be entered on individual grid points or it may entered as a dynamically applied static load, or as a combination of both.

Applying Dynamic Loads in Transient Response Analysis. The transient loads are entered with the TLOAD1 and TLOAD2 entries in the Bulk Data Section. The TLOAD1 entry is a brute force method where the input time history is entered at discrete time steps with a TABLED1 entry. The TLOAD2 entry can be used when the input load is in the form of a simple function, such as a cosine function. For further information regarding these entries, refer to the *MSC.Nastran Quick Reference Guide.*

For a TLOADi entry to be used in the analysis, it must be referenced by a DLOAD command in the Case Control Section. The TLOADi entry specifies the DAREA ID which defines the location and amplitude of the load. This DAREA ID refer to one or more DAREA Bulk Data entries or it may be a link to a static load. As an example, consider the application of a decaying sine wave forcing function applied to the end of a cantilever plate as shown in **Figure 40**.



$$P1 = 0.1e^{-3t}\sin(2\pi \cdot f \cdot t)$$

$$P2 = 0.5e^{-5t}\sin(2\pi \cdot f \cdot t)$$

where $f = 100$ Hz

Figure 40 Cantilever Plate with End Loads

The required Case Control and Bulk Data input is shown in **Figure 41**.

```
ID PLATE, COM_QUES
SOL 112
CEND
METHOD =1
DISP = ALL
DLOAD = 99
TSTEP = 5
BEGIN BULK
EIGRL   1               500.
DLOAD   99      1.0     1.0     1       1.0     2
TLOAD2  1       11              0       0.      1000.     100.      -90.
        -3.
TLOAD2  2       21              0       0.      1000.     100.      -90.
        -5.
DAREA   11      1       3       .1
DAREA   21      2       3       .5
TSTEP    5      100     .001    5
.
ENDDATA
```

Figure 41 Input File for the Cantilever Plate with Point Loads

In the above example, the DLOAD entry in the Bulk Data was used because two TLOAD2s are required for this analysis. If your input load can be described with a single TLOADi, then the DLOAD in the Bulk Data is optional. Since the solution is a modal transient response, an eigenvalue method is required. The displacement request in the Case Control refers to the transient displacement response not to the eigenvectors as in the normal modes analysis. The TSTEP entry is used to control the analysis time step, the duration of the analysis, and how often the results are printed. The T1 and T2 fields on your TLOAD2 entry do not control the analysis duration; they are used only to define the input load. As a final note on this example, the goal was to input a sine function whereas the form of the TLOAD2 is a cosine function. Placing a phase angle of -90 degrees in field 9 of TLOAD2 entry converts the input load to a sine function.

As a second example, suppose you wanted to apply a 5 psi pressure load to the cantilever plate with the same time dependency as P1. The input file that satisfies this is shown in **Figure 42**.

```
ID PLATE, COM_QUES
SOL 112
CEND
METHOD =1
DISP = ALL
DLOAD = 99
LOADSET = 98
TSTEP = 5
BEGIN BULK
EIGRL   1               500.
TLOAD2  1       11              0       0.      1000.     100.      -90.
        -3.
LSEQ    98      11      2
PLOAD4  2       1       2
TSTEP   5       100     .001    5
.
.
.
ENDDATA
```

Figure 42 Input File for the Cantilever Plate with Pressure Loads

For this input file, the static pressure described by the PLOAD4 entry is linked to the dynamic load by the DAREA ID defined in field 3 on the TLOAD2 entry. A DAREA entry does not have to exist; the ID in the DAREA field is only a pointer. The ID of the DAREA field corresponds to the DAREA field on the LSEQ entry. The LSEQ entry specifies the static loads to be used, which in this case is the PLOAD4. Both the DLOAD and LOADSET (which call out the appropriate LSEQ in the Bulk Data) must be included in the Case Control Section.

Applying Dynamic Loads in Frequency Response Analysis. The input file for frequency response is similar to that for transient response analysis. The RLOAD1 and RLOAD2 entries are used to define the frequency-dependent loads. The RLOAD1 entry is presented in real and imaginary format, whereas the RLOAD2 is presented in magnitude and phase format.

A common misconception of the RLOADi entries is that these entries are used to define the actual input sine function; however, that is not the case. The input load for a frequency response analysis is a sinusoidal function by definition. The RLOADi entries are used to define the amplitude of the sinusoidal input and its phase and delay with respect to other RLOADi inputs as a function of frequency. For example, consider the form of the RLOAD1 equation as follows:

$$P(f) = \{A[C(f)] + iD(f)e^{i(\theta - 2\pi f\tau)}\}$$

where:

A = DAREA scale factor

C = frequency-dependent real coefficient

D = frequency-dependent imaginary coefficient

θ = phase term

τ = delay term

Suppose you enter

A = 1

C = 1 for all frequencies (entered on the TABLED1 entry)

D = 0 for all frequencies (field 7 is left blank)

θ = 0 (field 5 is left blank)

τ = 0 (field 4 is left blank)

It may appear that you have entered a load of the form: $P(f) = 1$, a static load, for all frequencies. Actually, what you requested is the steady state response of the structure due to an input load of the form: $P(f) = 1\sin(2\pi ft)$. The sinusoidal part of this equation is inferred because you are performing a frequency response analysis. It is important to note that MSC.Nastran does not convert the input load into a time-varying load, rather the solution is performed in the frequency domain. For more information on the implementation within MSC.Nastran, please see the *MSC.Nastran Basic Dynamic Analysis User's Guide*.

The frequency at which the analysis is performed is not controlled by the RLOADi entries but rather by the FREQi entries. The FREQ entry is used to specify discrete frequencies for the analysis. The FREQ1 and FREQ2 entries are used to specify a range of frequencies with linear and logarithmic distribution, respectively. It is common to use a combination of the FREQ and FREQ1 or FREQ2 entries to obtain a smooth frequency response curve without an excessive number of frequencies analyzed in a direct frequency response analysis. For a modal frequency response analysis, three more FREQi entries—FREQ3, FREQ4, and FREQ5—are available. See the section on "*Describe the Guidelines for Frequency Response Analysis of Structures.*" The FREQi Bulk Data entries must be called out in the Case Control Section with a FREQ command.

A common question is how to apply an enforced displacement input in frequency response analysis. In transient response analysis, the TLOADi entries allow you to enter the "type" of input desired (the large mass approach is assumed). However, the type is not available on the RLOADi entries. Therefore, the most convenient method is to use the TABLED4 entry to generate a frequency-dependent coefficient. The approach is as follows:

- The desired input displacement is defined by $x(t) = A \cdot \sin(2\pi \cdot f \cdot t)$.
- This corresponds to an input acceleration defined by the following equation: $a(t) = -A \cdot 4\pi^2 \cdot f^2 \cdot \sin(2\pi ft)$.
- When the large mass approach is used, the acceleration of the input point is proportional to the applied load. Therefore, the applied load should be $P(f) = -A \cdot M \cdot 4\pi^2 \cdot f^2 \cdot \sin(2\pi ft)$, where M is the large mass value.
- The scale factor A times M is entered on the DAREA entry.

- The factor $-4\pi^2 \cdot f^2 = -39.4784 \cdot f^2$ is a frequency-dependent factor that is entered on a TABLED4 entry.

As an example, consider the input file shown in **Figure 43** where the desired excitation is defined by a constant amplitude displacement equal to 1 inch at grid point 1 in DOF 3 over the frequency range of 20 to 500 Hz, with 50 analysis frequencies.

```
ID PLATE, COM_QUES
SOL 111
CEND
METHOD =1
DISP = ALL
DLOAD = 1
FREQ = 5
BEGIN BULK
EIGRL   1               1000.
RLOAD2  1       11                      3
TABLED4 3       0.      1.      0.      10000.
        0.      0.      -39.4784
DAREA   11      1       3       10.E5
FREQ2   5       20.     500.    50
CONM1   10      1       10.E5
.
.
.
ENDDATA
```

Figure 43 Input File for the Cantilever Plate with Constant Displacement Frequency Response

It should be noted that none of the examples shown in this section have damping included. This was done only to save space. It is advisable to include damping in all your dynamic models, especially for frequency response. The peak response of an undamped structure at its natural frequencies is infinite. This is not only unrealistic, it will cause numeric problems.

Additional information on dynamic loads can be found in the *MSC.Nastran Basic Dynamic Analysis User's Guide*.

What Does a SUPORT Entry Do in a Dynamic Analysis?

In an eigenvalue solution, the main result of using a SUPORT entry is that the corresponding modes will be set to a frequency of 0.0 Hz. Since there is almost always some numeric roundoff in a solution, it is difficult otherwise to obtain "perfect" 0.0 Hz modes. In the Givens, Householder, and inverse power methods, a SUPORT entry guarantees "perfect" 0.0 Hz modes. In these methods, the SUPORT entry specifies that the frequency for the first N_r modes (N_r = number of SUPORT DOFs) is 0.0 Hz. This is accomplished by replacing the first N_r eigenvalues with a value of 0.0. It should be noted that the program has no built-in checks to terminate if any of the replaced eigenvalues are not numeric zero. This replacement can hide potential modeling problems if the results of UIM 3035 are not checked. The method to perform the checks using the information in

UIM 3035 is described below. The Lanczos method will replace the first N_r frequencies with 0.0 Hz only if the first N_r frequencies are less than $\sqrt{\text{SHFSCL}}$ (an estimate of the first elastic frequency, which you (optionally) supply on the EIGRL entry).

If a non-redundant set of DOFs are specified on the SUPORT entry, the values printed for epsilon and strain energy in the UIM 3035 table should all be numeric zero. The strain energy printed in this table for matrix KLR represents the strain energy in the model when the associated SUPORT DOF is moved 1.0 unit while all other SUPORT DOFs are constrained. If the SUPORT is properly specified, the model should be capable of rigid body motion (strain-free) with no strain energy. The values printed for the strain energy indicate the ability of the model to move as a rigid body. If a SUPORT entry is used, these values should always be checked. If the structure is not constrained, these values should be numeric zero, but roundoff almost always results in a small nonzero value. Acceptable values are a function of the units used and the size of the structure modeled. Therefore, a recommended value is not provided in this writeup.

Note also that using a SUPORT entry in a dynamic analysis causes an additional matrix decomposition. This decomposition is used to calculate the deformations resulting when the SUPORT DOFs are moved.

In the Lanczos method, the SUPORT entry attempts to give 0.0 Hz modes. The Lanczos routines compare the values of the calculated modes and determine if the calculated frequencies for the N_r modes are near 0.0. If they are, then the computed value is replaced with 0.0; but if a computed value is not near 0.0, then it is retained. (This may cause problems in response spectrum analysis where 0.0 Hz is required for the SUPORT modes.)

In the Lanczos method, MSC.Nastran uses the calculated eigenvectors. In the other eigensolvers, the routines replace the first N_r modes with the vectors calculated by moving the SUPORT DOFs each a unit distance (they are orthogonalized to the other modes calculated). This is the reason that the SUPORT entry may improve the accuracy in modal methods (unless Lanczos is used). The static deformation of the structure resulting from the enforced motion of each SUPORT DOF is always contained within the eigenvectors when the Givens, Householder, or Inverse Power method is used. The motion should be contained in the eigenvectors when Lanczos is used.

When Do I Need to Use a SUPORT Entry?

In dynamic analysis, there is only one solution type that requires the use of a SUPORT entry and that is response spectrum analysis (available in SOL 103). When performing this solution, the DOFs on the SUPORT are used to define the input DOFs for the spectra. The mode acceleration data recovery method also requires the use of the SUPORT entry.

A common misbelief is that a SUPORT entry is necessary to use with the large mass method for enforced motion. As long as the large mass method is used, the SUPORT entry is not necessary. In a direct solution, all dynamic properties are available (therefore, the SUPORT is not needed); and in modal solutions, as long as the low frequency (large mass) modes are retained, the input motion is included.

Describe the Guidelines for Linear Transient Response Analysis of Structures

Several factors are important in computing accurate linear transient response for structural analysis including: the number of retained modes (if modal transient response is used, as opposed to direct); the integration time step Δt; the time duration of the computed response; and damping.

The guidelines described are only approximate and vary depending on the kind of model, complexity of the spatial distribution of the loading, and skill and experience of the analyst. In all cases, however, running a normal modes analysis first facilitates the determination of the factors for computing an accurate transient response.

For a more complete discussion of these guidelines, see the *MSC.Nastran Basic Dynamic Analysis User's Guide*.

Number of Retained Modes. In modal transient response analysis, the larger the number of modes used, the more accurate the response (although at the expense of increased runtimes). The number of modes must be large enough to cover the frequency range of interest. (The term "range of interest" means the range of frequencies whose response has to be computed as well as the range of frequencies for which the loading is applied.) As a general rule, there should be enough modes to cover a range up to about two times the highest frequency of interest. For example, if response needs to be computed to 100 Hz, then enough modes should be used in the transient response to cover up to at least 200 Hz. As another general rule, you should use enough modes to cover a range up to two to ten times the dominant frequency of response.

The number of modes used in modal transient response is governed by two choices of Bulk Data parameters:

1. PARAM,LMODES,n. LMODES is the number of lowest modes to use in a modal formulation; the default 0 includes all modes.
2. PARAM,HFREQ,fh and PARAM,LFREQ,fl. The real values fh and fl give the frequency range of the modes to be used in the modal formulation. The default for fh is 1.0E30, which should cover all of the modes computed.

A sufficient number of modes must be retained to cover the time and spatial distribution of loading. For example, if the applied transient load has a spatial distribution resembling the fifth mode shape, then that mode should be included in the transient response.

There are similar guidelines for design sensitivity analysis of transient response. See the *MSC.Nastran Design Sensitivity and Optimization User's Guide* for a discussion of this subject.

Size of the Integration Time Step. The value of the integration time step, denoted by DT on the TSTEP Bulk Data entry, is important for accurate integration of the equations of motion. Rough guidelines for its selection are as follows:

1. DT must be small enough to accurately capture the magnitude of the peak response. This means that about ten time steps per cycle of the highest mode be used. For example, if the highest frequency of interest is 100 Hz, then DT should be 0.001 second or smaller.
2. DT must be small enough to accurately represent the frequency content of the applied loading. If the applied loading has a frequency content of 1000 Hz, then DT must be 0.001 second or less.
3. DT should be 1/5 to 1/10 the size described above if there are nonlinearities present that are described by NOLINi entries. This is because the NOLINi puts the structure out of equilibrium by one time step, so a very small integration time step is needed to minimize this limitation.

Duration of Computed Response. The length of the time duration is important so that the lowest flexible (e.g., non rigid-body) mode oscillates through at least one cycle. For example, if the lowest flexible mode has a frequency of 0.2 Hz, then the time duration of the computed response should be at least 5.0 sec. A better guideline is to make the duration the longer of: (1) twice the period of the lowest flexible mode, or (2) one period of the lowest flexible mode after the load has stopped or reached a steady-state. The time duration is set on the TSTEP entry by multiplying DT (the integration time step) by the number of time steps (N).

Value of Damping. It is very difficult to generalize about the proper value of damping other than to recommend using the value that your structure actually has. This value (or values, since damping can vary by mode) can be determined from structural tests or by published literature that gives values for structures similar to yours.

The proper selection of the damping value is relatively unimportant for analyses that are of very short duration, such as a crash impulse or a shock blast. The specification of the damping value is most important for long duration loadings (such as earthquakes) and is critical for loadings (such as sine dwells) that continually “pump” energy into the system.

Can I Change the Integration Time Step in a Linear Transient Response Analysis?

The integration time step can be changed in a linear transient response analysis, but it is not recommended. To begin with, much of the cost of linear transient response occurs with the decomposition of the dynamic matrix, which occurs only once if there is a constant DT. A new decomposition is performed every time DT changes, thereby increasing the cost of the analysis if the time increment is changed.

Another reason not to change the time increment is that an artificial "spike" occurs when the time increment is changed. This spike is larger when there are nonlinearities present via NOLINi entries.

Therefore, the recommendation is to determine the smallest DT required based on the loading and frequency range and use that DT throughout the entire linear transient response analysis.

Describe the Guidelines for Frequency Response Analysis of Structures

The guidelines for accurate frequency response analysis are similar to those described above for transient response. These include:

Number of Retained Modes. Use enough modes to cover the range of excitation. For example, if the model is to be excited from 1 to 100 Hz, use all modes with frequencies up to at least 100 Hz. A better guideline is to use enough modes to cover two to three times the range of excitation in order to provide accurate answers at the high end of the range. Therefore, in the example where the excitation is applied to 100 Hz, modes with frequencies up to 200 to 300 Hz should all be used.

PARAM,LMODES, PARAM,HFREQ, and PARAM,LFREQ apply to modal frequency response as well as modal transient response.

How Do I Apply Excitation Frequencies As a Function of Natural Frequencies? In order to capture the dynamic response properly in a frequency response analysis, enough excitation frequencies must be specified in the vicinities of the resonant frequencies. For modal frequency response solution sequences (SOLs 111, 146, and 200), you can specify excitation frequencies as a function of the natural frequencies using the FREQ3, FREQ4, and / or FREQ5 entries. This feature does not require you to know the natural frequency beforehand or to use a very small frequency increment. The excitation frequencies are computed automatically after the natural frequencies are extracted. Thus, results quality is maximized while minimizing the number of excitation frequencies.

FREQ3 specifies excitation frequencies in the range between two modal frequencies. The increments between excitation frequencies are calculated linearly or logarithmically. These frequencies may be clustered towards the end points of the range or towards center by specifying a cluster parameter. FREQ4 calculates excitation frequencies as a spread around each modal frequency. FREQ5 calculates excitation frequencies as a factor of natural frequencies. All FREQ, FREQ1, FREQ2, FREQ3, FREQ4 and FREQ5 entries with the same SID will be combined to derive the list of excitation frequencies. See the *MSC.Nastran Quick Reference Guide* for further details regarding the format of the FREQi entries.

Adaptive excitation frequency Bulk Data entries (FREQ3, FREQ4 and FREQ5) are applicable only for the modal frequency response solutions (SOLs 111, 146, and 200) since the knowledge of the natural frequencies are required during the run. A warning message is issued if these Bulk Data entries are specified for direct frequency response solutions (SOLs 108, 118, or 200).

In design optimization, (SOL 200) excitation frequencies are not changed as the natural frequencies change during the optimization procedure. Excitation frequencies are based on the initial analysis. The DRESP1 frequency values are unaffected by these new Bulk Data entries. There is no tie to the tracking of modes in design optimization.

It is paramount to specify some damping if using the new Bulk Data entries. This is because the system will be excited exactly at the modal frequencies, and the solution may become unbounded without some damping.

Size of the Frequency Increment. The size of the frequency increment Δf must be small enough to ensure that the magnitude of the peak response is accurately computed. To ensure this, you need to choose a frequency increment small enough so that there are at least five to ten increments within the "half-power bandwidth" frequencies as shown below.

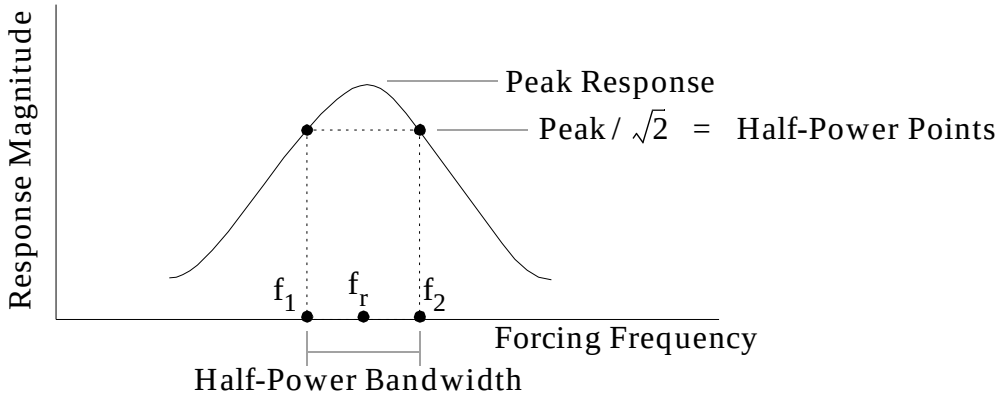


Figure 44 Half-Power Band Curve

This is especially important in random response analysis where most of the energy lies between the half-power points. An insufficient number of frequency points can lead to unconservative answers (i.e., computed response that is too low).

The frequency increment is defined by DF on the FREQ1 Bulk Data entry. Note that FREQ and FREQ2 entries can be used in conjunction with FREQ1 in order to define more solutions in the areas of resonance (the frequencies of which should have been determined by a prior normal modes analysis) in a direct frequency response analysis. In a modal frequency response analysis, the combination of the FREQ4 entry—which can be used to specify the number of frequencies between half-power points—and the FREQ1 entry provides a way of picking frequencies around the resonant points and even distribution throughout the rest of the frequency range. A nonuniform DF imposes no cost increase, unlike a nonuniform DT for transient response analysis.

Relationship of Damping to Frequency Increment. The less damping in the structure, the greater the response at resonance and the narrower in frequency the half-power bandwidth. For lightly-damped structures, an approximate relationship between the half-power bandwidth, resonant frequency, and modal damping ratio is:

$$B = \text{modal damping ratio} = (f2 - f1) / (2 \cdot fr)$$

where f1 and f2 define the beginning and ending frequencies of the half-power bandwidth and fr is the frequency at resonance. Assuming 1 percent modal damping and six points to define the half-power bandwidth, the frequency increment is $0.004 \cdot fr$. The frequency increment is less for lighter damped structures.

Numeric Truncation. Due to numeric truncation, two different computers may compute different responses. This is especially true at resonance, and is also exaggerated with little or no damping.

How Do I Model Frequency-Dependent Springs and Dampers?

Vibration control devices, such as engine and component vibration isolation mounts, can have impedance (stiffness and damping) values that vary with excitation frequency. Starting in Version 69, you can define frequency-dependent mounts in a frequency response analysis using the CBUSH element. The frequency-dependent impedance values are input on table entries—frequencies versus impedance values. The frequency-dependent features may be used in SOLs 108, 111, and 200. These elements may also be used with other solution sequences with the frequency-dependent characteristics ignored. The CBUSH element has up to six uncoupled stiffness and damping values to define its nominal properties. These impedances act at a point in space (the elastomer center of elasticity) defined on the CBUSH entry. The stiffness values at the elastomer center of elasticity are transferred to the connected grid points by rigid element equations. The orientation of these stiffness values at the elastomer center of elasticity is defined also on this connection entry, in a manner similar to the CGAP element.

The PBUSH property entry is used to specify the nominal properties of the element. The nominal stiffness is used for real-mode analysis and for cases where no tabular input is used to define frequency-dependent impedance. If frequency-dependent stiffness and/or damping values are needed, they are defined on the PBUSHT entry with the same ID as the PBUSH entry.

A typical CBUSH element is shown below.

```
CBUSH,   1,      12,     1,      2,                              0
PBUSH,   12,     K       1.      1.      1.      2.      4.      2.
$
PBUSHT,  12,     K,      1112
TABLED1  1112.......
```

In the above example, the CBUSH element 1 is connected between grid points 1 and 2 with nominal spring values defined on the PBUSH 12 entry. The nominal stiffnesses—as denoted by K in field three of the PBUSH entry—are 1., 1., 1., 2.,4., and 2., in the 1,2,3,4,5, and 6 directions, respectively, of the basic coordinate system. Only the stiffness in the 1-direction is frequency-dependent (non blank field 4 on the PBUSHT entry) with the stiffness values in the other five directions as frequency-independent. TABLED1 1112 contains the frequencies versus stiffness values for the CBUSH element 1 in the 1-direction.

Element force and stress output is available for the CBUSH element. If present, the frequency-dependent effects are accounted for exactly in all output quantities, including element force and stress, and single- and multipoint constraint forces.

For versions prior to Version 69, you can obtain similar functions using the zfreq* alter delivered with the sssalter library.

When Should Modal Formulations Be Used As Opposed to Direct Formulations?

MSC.Nastran offers both modal and direct formulations for transient and frequency response. For moderate to large models, modal formulations (i.e., modal transient response or modal frequency response) are almost always preferred over direct formulations because of their lower cost. In order to perform accurate transient or frequency response analyses, you should always run a normal modes analysis first in order to determine the resonant frequencies; this information will help guide you in selecting the proper parameters (see above). After the normal modes analysis has been verified, it is straightforward to restart into modal transient response or modal frequency response.

Modal formulations can also be less expensive than direct formulations even when they have not been restarted from a normal modes analysis. The primary cost in a modal formulation is the eigenanalysis; after that, the dynamic response is computed with a much smaller number of equations. For example, while there may be 100,000 DOFs in the model, there may be as few as only 20 modes that are needed for dynamic response.

Modal formulations also allow output of the modal variable output (with the SACCELERATION, SDISPLACEMENT, and SVELOCITY Case Control commands). This output provides the contribution of each individual mode to the overall response, which can be useful when making redesign decisions.

Using a direct formulation can be expensive, particularly if there are many time steps or frequency points in the solution. The cost for a direct transient solution is approximately the cost of a similar static analysis plus the price of data recovery for all of the time steps, while the cost of a direct frequency response solution is approximately the cost of a similar static analysis times the number of frequency points. It is not uncommon for there to be several thousand time steps in an earthquake analysis, for example, or several hundred frequency points in simulating a stepped-sine shake table test.

Direct formulations make sense when there are very few time or frequency points in the solution and the solution cost for them is outweighed by the cost to perform the eigenanalysis. Direct formulations are more efficient than modal formulations when there is very high modal density and there is high frequency excitation.

There are instances where modal formulations cannot be used, and the direct formulation is the only choice. Initial conditions for transient response are available as a standard feature only for direct formulation. Nonlinear forces (NOLINi) are easier to implement in direct, as opposed to modal, transient response. Material and geometric nonlinearities can be analyzed only with direct transient response (SOL 129, for example). Direct formulations can provide more accurate local stresses than modal formulations (although modal formulation stresses can be improved by using the mode acceleration method).

Describe the Mode Acceleration Method

Mode acceleration is a method of data recovery provided in the dynamic response solutions that improves the accuracy of the stress and strain results for many cases. Although the standard modal formulations provide accurate overall response near the

natural frequencies, the missing effects of the truncated modes cause errors in the local stresses and forces. In order to improve these results, you have several options, which include increasing the number of modes, using DMAP to append static shape vectors, or selecting the mode acceleration option (PARAM,MODACC,0) for data recovery.

The matrix operations for the mode acceleration method are described in the *MSC.Nastran Basic Dynamic Analysis User's Guide.* In summary, the method generates "pseudo static" solutions using the calculated accelerations and velocities from the standard solution as a starting point. The steps are:

1. The modal solutions are expanded to analysis set vectors and multiplied by the mass, damping, and direct input matrices to generate equivalent loading vectors. These vectors are added to the applied load function to generate a "pseudo load matrix" for all selected output time steps or frequencies.
2. The analysis-sized structural stiffness matrix is decomposed (with special treatment for free bodies).
3. A static solution matrix is obtained for the required output points.
4. Stresses and forces are obtained through the conventional data recovery process. These results are automatically compatible with the inertial loads, damping loads, and other loads.

The overall performance and accuracy of the method falls somewhere between the standard modal outputs and the expensive direct solutions. It works well for large structures where the local forces are important by reducing the required number of modes. It is not cost-effective when the mode shapes provide good results or when only peak modal displacements are important.

Describe the Residual Vector Method

The Residual Vector Method provides an alternate way of improving the results using the modal approach. Starting in Version 70, the residual vector can be activated by adding "PARAM,RESVEC,YES" to the input file. These static vectors can be generated by applying an external load, or by specifying DOFs where unit loads are applied automatically. These static vectors are then orthogonalized and augmented to the calculated mode shapes. The Residual Vector Method is more efficient than the Mode Acceleration Mode and it can be applied to both the superelements and residual structure. For earlier versions of MSC.Nastran, similar capabilities are provided by the resflexa.vxx alter delivered with the sssalter library.

How Do I Input Cross Spectra in Random Analysis?

Random analysis in MSC.Nastran is part of postprocessing for the frequency response solutions. When a random analysis is performed, the program can take one of two paths. If no cross-spectral information is input (that is, if J is equal to K on all of the RANDPS

entries), the input loads defined by the subcases are taken to be uncorrelated. If this is the case, MSC.Nastran checks to ensure that each J, K pair is unique. If they are not unique, the following fatal error occurs:

```
*** USER FATAL MESSAGE 5418, ILLEGAL DUPLICATION ON RANDPS CARDS.
```

If a RANDPS entry exists in the input in which J is not equal to K, then MSC.Nastran takes a different path that allows for repeated J, K pairs. The PSDs for each RANDPS entry are summed to generate the output PSD for the run. This operation provides a convenient method to input complex cross spectra. To input a complex cross-spectral table, use two RANDPS entries—each with the same J, K pair. One RANDPS entry only has a real coefficient (Y = 0.0), and it points to the TABRND1 table for the real part. The second RANDPS entry has only the imaginary coefficient (X = 0.0), and it points to the TABRND1 table for the imaginary part. The two parts are combined internally to form a complex cross spectra. An example using multiple RANDPS entries to model partially correlated random input is shown below.

```
RANDPS,100,1,1,1.,0.,100
RANDPS,100,2,2,1.,0.,200
RANDPS,100,1,2,1.,0.,300
RANDPS,100,1,2,0.,1.,400
$
TABRND1,100,
 ,20.,0.1,30.,1.,100.,1.,500.,.1
 ,1000.,.1,ENDT
$
TABRND1,200,
 ,20.,0.5,30.,2.5,500.,2.5,1000.,0.
 ,ENDT
$
TABRND1,300,
 ,20.,-.099619,100.,-.498097,500.,.070711,1000.,0.
 ,ENDT
$
TABRND1,400,
 ,20.,.0078158,100.,.0435791,500.,-.70711,1000.,0.
 ,ENDT
$
```

Figure 45 Using Multiple RANDPS Entries to Model Partially Correlated Random Input

Miscellaneous

Describe the Various Grid Point Coordinate Systems in MSC.Nastran

Basic. MSC.Nastran has one implicitly defined coordinate system that is called the basic coordinate system. It is a rectangular (Cartesian) coordinate system. The long-established choice of name may be unfortunate for users who are familiar with other finite element codes.

Global. The term “global coordinate system” in some other codes has a meaning that is identical to that of MSC.Nastran’s basic coordinate system. The term “global coordinate system” has a unique meaning in MSC.Nastran and will is explained later.

Local. In addition to the basic coordinate system, MSC.Nastran supports three types of local coordinate systems—rectangular, cylindrical, and spherical. A local coordinate system can reference either the basic coordinate system or another local coordinate system. The only rule in MSC.Nastran is that the local coordinate system must ultimately be able to be transformed back into the basic coordinate system. For a more detailed description of these coordinate systems, refer to the CORDij entries in the *MSC.Nastran Quick Reference Guide*.

There are two fields on GRID entries that can reference these coordinate systems. Field 3, also known as the CP field, is used solely for identifying the coordinate system used to define the location of a grid point in physical space. This coordinate system can be any of the valid MSC.Nastran coordinate systems including the basic coordinate system. Field 7, also known as the CD field, serves a much bigger role. This field identifies the coordinate system in which the displacements, degrees of freedom, constraints, some applied loads, and solution vectors are defined at the particular grid point. The connectivity relationships, e.g., SPCs, MPCs, RBAR, etc., are written in the coordinate system defined on the CD field, not the CP field.

The union of all the coordinate systems defined on all the CD fields on all the GRID entries is called the global coordinate system in MSC.Nastran. The global coordinate system in MSC.Nastran is, therefore, not a single unique system. Note that the basic coordinate system may also, but not necessarily, be a part of the global coordinate system.

Describe the Grid Point Weight Generator

The grid point weight generator (GPWG) is activated with the parameter GRDPNT. The data entry format is:

```
PARAM,GRDPNT,i
```

where i is an integer value defining a reference point. The value of “i” can be any grid point in the model, or if it is set to zero, the reference point is the origin of the basic coordinate system.

The output from the GPWG includes a rigid body mass matrix, various coordinate transformations, and the location of the center of mass as shown in **Figure 46**.

```
    O U T P U T   F R O M   G R I D   P O I N T   W E I G H T   G E N E R A T O R
                                  REFERENCE POINT =        0
                                        M O
*  1.480000E-03  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00 -1.480000E-03 *     Rigid Body
*  0.000000E+00  1.480000E-03  0.000000E+00  0.000000E+00  0.000000E+00  7.400000E-03 *     Mass
*  0.000000E+00  0.000000E+00  1.480000E-03  1.480000E-03 -7.400000E-03  0.000000E+00 *     Properties
*  0.000000E+00  0.000000E+00  1.480000E-03  2.220000E-03 -7.400000E-03  0.000000E+00 *     Matrix
*  0.000000E+00  0.000000E+00 -7.400000E-03 -7.400000E-03  5.032000E-02  0.000000E+00 *
* -1.480000E-03  7.400000E-03  0.000000E+00  0.000000E+00  0.000000E+00  5.254000E-02 *
                                        S                                              Transformation
                      *  1.000000E+00  0.000000E+00  0.000000E+00 *                     From the
                      *  0.000000E+00  1.000000E+00  0.000000E+00 *                     Basic to the
                      *  0.000000E+00  0.000000E+00  1.000000E+00 *                     Principal
          DIRECTION                                                                    Directions
      MASS AXIS SYSTEM (S)     MASS              X-C.G.        Y-C.G.        Z-C.G.
              X             1.480000E-03      0.000000E+00  1.000000E+00  0.000000E+00  CG Relative
              Y             1.480000E-03      5.000000E+00  0.000000E+00  0.000000E+00  to the
              Z             1.480000E-03      5.000000E+00  1.000000E+00  0.000000E+00  Reference Pt
                                     I(S)
                      *  7.399999E-04  0.000000E+00  0.000000E+00 *                     Moments of
                      *  0.000000E+00  1.332000E-02  0.000000E+00 *                     Inertia w.r.t.
                      *  0.000000E+00  0.000000E+00  1.406000E-02 *                     Prin. Mass Axes
                                        I(Q)
                      *  7.399999E-04                              *                  Principal
                      *                1.332000E-02                *                  Moments of
                      *                                  1.406000E-02 *               Inertia
                                           Q
                      *  1.000000E+00  0.000000E+00  0.000000E+00 *                     Transformation
                      *  0.000000E+00  1.000000E+00  0.000000E+00 *                     from the Prin.
                      *  0.000000E+00  0.000000E+00  1.000000E+00 *                     Mass Axes to
                                                                                       the Prin. Dir.
                                                                                       of the Momental
                                                                                       Ellipsoid
```

Figure 46 Output from the Grid Point Weight Generator

The output from the GPWG is generally more than most users need. The mass and center of gravity (CG) location is typically all that is used. Note the following items:

- For most models, the mass in all three directions should be the same. If they are not, the cause may be due to a CONM1, CMASSi, or DMIG input.
- Because the mass may be different in the three translational directions, the mass may have directional properties. It is for this reason that CG location is given for each of the three translational masses. If the mass is the same in all directions, a unique CG exists and is located at the X component of the Y (or Z) mass, the Y component of the X (or Z) mass and Z component of the X (or Y) mass. In the above example, the CG of the structure is located at (5.0,1.0,0.0) in the principal mass coordinates.
- The scale factor entered in the parameter WTMASS is applied to the assembled element mass before GPWG. The GPWG module, however, converts the mass back to the original input units that existed prior to the scaling effect of the parameter WTMASS. (Note that the parameter WTMASS does not get applied to

M2GG or M2PP input, but the M2GG mass does get assembled into the mass matrix prior to GPWG. Therefore, for GPWG output only, the mass input using M2GG is scaled to the same units as the element mass input. M2GG input may be scaled independently using the CM2 parameter.)

- GPWG is performed on the G-size mass matrix which is the mass matrix prior to the processing of the rigid elements, SPCs, and MPCs.
- Mass at scalar points is not included in the GPWG calculation.
- GPWG for a superelement does not include the mass from upstream superelements. Therefore, the GPWG for the residual structure includes only the mass on the residual points. The CG location is also based on the residual mass only.
- If a large mass is used for enforced motion, the large mass dwarfs the structural mass. For model checkout, remove the large mass and constrain the driving point. A static run is a convenient way to generate a mass matrix and obtain output from GPWG.
- The output from GPWG is for informational purposes only and is not used in the analysis.
- Additional information can be found in Section 4 of the *MSC.Nastran Programmer's Manual* and the *MSC.Nastran Linear Analysis Static User's Guide.*

What Is Included in the Grid Point Force Balance?

The grid point force balance is computed only from the linear stiffness elements. Therefore, forces due to rigid elements, GENEL elements, DMI, TFs, MPCs, NOLINs, and those resulting from the inertia and damping terms in the dynamic solutions are not included in the GPFORCE output. If any of the above exists in your model, then the grid point force balance shows an unbalanced force at some or all of the grid points.

How Do I Interpret Scalar Point Output?

Scalar point output uses the same format as the grid point output. However, grid points have six degrees of freedom per grid, whereas the scalar point has only one. To reduce the output, MSC.Nastran prints up to six scalar point outputs per line. T1 corresponds to the output scalar point ID listed at the start of the line. T2 corresponds to the next scalar point output, etc. The T1, T2, ..., R3 are not the directions for the scalar point. To denote that the output is for a scalar point, the letter S is printed on the left side under TYPE. If the output is for a grid point, the letter G is printed. For the example shown in **Figure 47**, the displacement for scalar point 2 is shown in the T2 column of the output line for scalar point 1.

```
BEGIN BULK
SPOINT  1      THRU     9
CELAS3  1       99      1       2
.
.
.
CELAS3  8       99      8       9
PELAS   99      100.            1.
SPC     1       1       0
SLOAD   2       9       100.
GRID    999
SPC     1       999     123456
ENDDATA



                                    D I S P L A C E M E N T   V E C T O R
POINT ID.   TYPE          T1            T2            T3            R1            R2            R3
     1      S      0.0            1.000000E+00  2.000000E+00  3.000000E+00  4.000000E+00  5.000000E+00
     7      S      6.000000E+00   7.000000E+00  8.000000E+00
     999    G      0.0            0.0           0.0           0.0           0.0           0.0
```

Figure 47 Displacements for Scalar Points

Note that the grid point is needed because MSC.Nastran requires at least one grid point in the model. In this case, it is not connected to the structure and is fully constrained, so it does not affect the results.

What Is Meant by the Term “Mechanism”?

A mechanism occurs when a part of a structure is capable of strain-free or rigid-body motion. When performing finite element analysis, a mechanism occurs when it is necessary to decompose a singular stiffness matrix. This mechanism is indicated either by UWM 4698 or a fatal message (often SFM 4276).

Mechanisms may be intentional. For example, when modeling a turbine, an ideal set of bearings is assumed. This allows the unconstrained rotation of the fan blades. This is a simple mechanism that requires special user effort, such as inertia relief, to analyze.

Most mechanisms are unintentional, that is, they arise from modeling mistakes. Some common modeling mistakes that can cause mechanisms are as follows:

- Pinned-pinned CBAR or CBEAM elements with all three rotations released at both ends (the element may spin about its axis).
- Plate elements without bending properties (MID2 on the PSHELL entry).
- CBAR or CBEAM elements connected to plate elements without special modeling to transfer the in-plane moment. (The plates have no stiffness for in-plane bending.)
- Plate, CBAR, or CBEAM elements connected to solid elements. (Solid elements have no stiffness for any bending.)
- CROD elements used to resist bending loads. (The CROD element has stiffness only along its axis.)

- CROD elements connected in a colinear fashion. In this case there is no stiffness perpendicular to the line of the elements.
- Elements or constraints omitted unintentionally.

The problem arises since the finite element method is an approximate method and works with matrices, not a physical structure. It is up to you to use the elements properly when modeling the physical structure. Errors made by the user almost always result in either incorrect results or mechanisms.

What Is the Meaning of AUTOSPC, MAXRATIO, and BAILOUT?

When performing the solution to linear equations, singularities lead to conditions where a unique solution is not possible. The above parameters in MSC.Nastran can help remove singularities in the model and enable the program to obtain a solution. MSC.Nastran considers two types of singularities:

- Grid point singularity that is identified by considering the stiffness terms of only one grid point.
- Mechanism type of singularity that requires the consideration of stiffness terms of more than one grid point.

For more information, refer to the *MSC.Nastran Reference Manual.*

Singularities cause ill-conditioned matrices that can be detected in four different phases of the MSC.Nastran execution as follows:

1. After matrix assembly, the grid point singularities are detected. At each grid point a 3x3 partition of the stiffness matrix corresponding to the three translational and three rotational DOFs is solved as an eigenvalue problem to determine the principal stiffnesses. Each stiffness term is compared to the principal stiffness as follows:

$$\varepsilon = \frac{K_{jj}}{K_{max}}$$

where K_{jj} is the term in the *j*-th row and *j*-th column of the matrix and K_{max} is the principal stiffness. For more information, refer to the *MSC.Nastran Reference Manual.*

If ε is less than the value of PARAM,EPZERO, the global direction nearest j is considered singular. The default value for EPZERO is 10^{-8}. A list of potential singularities is printed in the grid point singularity table printed by the GPSP module. If PARAM,AUTOSPC,YES is specified (this is the default in the Structured Solution Sequences), the potential singularities are automatically constrained if possible. AUTOSPC does not have any effect in SOL 106. PARAM,AUTOSPCR is used in SOL 106 for the o-set in the residual structure. For the default values of AUTOSPC in each solution sequence, refer to the *MSC.Nastran Reference Manual.*

Below is an example of the Grid Point Singularity Table. If PARAM,AUTOSPC,YES is specified, the singular degrees of freedom are shifted to other sets by the logic documented in the *MSC.Nastran Quick Reference Guide.* In this example, they are moved from the f-set into the sb-set, which is constrained as indicated by the asterisks.

```
            G R I D   P O I N T   S I N G U L A R I T Y   T A B L E
POINT  TYPE    FAILED    STIFFNESS      OLD USET           NEW USET
  ID         DIRECTION     RATIO    EXCLUSIVE  UNION   EXCLUSIVE  UNION
6711     G        4       0.00E+00       B        F        SB        SB     *
6711     G        5       0.00E+00       B        F        SB        SB     *
6711     G        6       0.00E+00       B        F        SB        SB     *
6712     G        4       0.00E+00       B        F        SB        SB     *
6712     G        5       0.00E+00       B        F        SB        SB     *
6712     G        6       0.00E+00       B        F        SB        SB     *
6713     G        4       0.00E+00       B        F        SB        SB     *
6713     G        5       0.00E+00       B        F        SB        SB     *
6713     G        6       0.00E+00       B        F        SB        SB     *
```

Figure 48 Example – Grid Point Singularity Table

In both linear and geometric nonlinear analyses, PARAM,K6ROT can be used to add stiffness to the normal rotation for QUAD4 and TRIA3 elements. For more information on PARAM,K6ROT, refer to the question “What Value Should I Use for K6ROT?”

The automatic constraint may mask a modeling error, for example, plate elements with no bending stiffness. It is recommended that all singularity messages be carefully inspected to ensure that modeling errors are not being masked.

2. Before decomposition, if null columns are identified, an information message is issued as shown below:

```
***SDCOMP INFORMATION MESSAGE--COLUMN *** OF DATA BLOCK *** IS NULL.
```

By default, MSC.Nastran prints up to 50 messages for null columns. In many cases, the program obtains a good solution. However, it can also produce incorrect results or divergence in the nonlinear solution sequences.

You can control the operation of the module for exceptional conditions with the DECOMP parameter. The valid values and actions for the parameter are listed in the *MSC.Nastran Quick Reference Guide* under the DECOMP module description. If DECOMP=1, the program execution is terminated when the first null column is encountered, and the following message is issued. The parameter can also be set with the NASTRAN statement keyword SYSTEM(69).

```
***SDCOMP FATAL MESSAGE---DECOMPOSITION OF DATA BLOCK *** ABORTED DUE TO ABOVE MESSAGES AND USER OPTION.
```

This option can be used during model checkout so that the null columns can be resolved before the decomposition is performed. This option is particularly of benefit in large models.

3. During decomposition, mechanisms can be detected based on the maximum ratio of matrix diagonal to factor diagonal,

$$\text{MAXRATIO} = \frac{K_{ii}}{D_{ii}}$$

where K_{ii} is the *i*-th diagonal term of the original stiffness matrix and D_{ii} is the *i*-th diagonal term of the factor diagonal matrix.

All terms whose ratio exceed the value of PARAM,MAXRATIO are printed. The default for MAXRATIO is 10^5. UIM 4158 prints the statistics for the decomposition that include the number of negative terms on the factor diagonal and the maximum ratio of matrix diagonal to factor diagonal at a specified row number. UWM 4698 prints the degrees of freedom that have a factor diagonal ratio greater than the MAXRATIO value or have negative terms on the factor diagonal. Both of these messages are issued by the DECOMP module and are shown below:

```
*** USER INFORMATION MESSAGE 4158---STATISTICS FOR SYMMETRIC DECOMPOSITION OF DATA BLOCK KLL      FOLLOW
           NUMBER OF NEGATIVE TERMS ON FACTOR DIAGONAL =     1
           MAXIMUM RATIO OF MATRIX DIAGONAL TO FACTOR DIAGONAL = 7.2E+15 AT ROW NUMBER    16
 *** USER WARNING MESSAGE 4698.  STATISTICS FOR DECOMPOSITION OF MATRIX KLL     .
 THE FOLLOWING DEGREES OF FREEDOM HAVE FACTOR DIAGONAL RATIOS GREATER THAN 1.00000E+05 OR HAVE NEGATIVE TERMS ON
THE FACTOR DIAGONAL.

      GRID POINT ID       DEGREE OF FREEDOM   MATRIX/FACTOR DIAGONAL RATIO     MATRIX DIAGONAL

           6714                  T1                -7.19297E+15                  6.02908E+07
 ^^^ DMAP FATAL MESSAGE 9050 (SEKRRS) -  RUN TERMINATED DUE TO EXCESSIVE PIVOT RATIOS
     IN MATRIX KLL. USER PARAMETER BAILOUT MAY BE USED   TO CONTINUE THE RUN.
```

Figure 49 Error Messages Issued by the DECOMP Module

At this point, MSC.Nastran takes the appropriate action as directed by PARAM, BAILOUT. A value of 0 for BAILOUT terminates the processing of the superelement or the run (this is the default action in the Structured Solution Sequences). A negative value for BAILOUT causes MSC.Nastran to continue processing the superelement or run with near singularities. This continuation may lead to poor solutions or fatal messages later in the run. It is recommended that the default values be used. Refer to the *MSC.Nastran Quick Reference Guide* for the default values of BAILOUT in each solution sequence. Refer to Chapter 4 of the *MSC.Nastran Numerical Methods User's Guide* for more information on decomposition.

Taking the log of MAXRATIO indicates how many significant digits may have been lost during the decomposition.

4. After decomposition, a singularity may lead to an incorrect solution. As a check in static analysis, MSC.Nastran solves

$$Ku = P$$

to obtain u. MSC.Nastran then calculates a "residual" loading vector as follows:

$$Ku - P = \delta P$$

This vector should be null but may not be due to numeric roundoff.

An error measure ε is calculated by

$$\varepsilon = \frac{u^T \cdot \delta P}{u^T \cdot P}$$

δP can be printed by PARAM,IRES,1. ε is the value of epsilon that is printed in UIM 5293 as shown below. Epsilon values that are greater than 0.001 are flagged for a possible loss of accuracy due to numeric conditioning.

```
0*** USER INFORMATION MESSAGE 5293 FOR DATA BLOCK KLL
    LOAD SEQ. NO.            EPSILON             EXTERNAL WORK      EPSILONS LARGER THAN 0.001 ARE FLAGGED WITH ASTERISKS
               1         4.8341316E-18           1.5595239E-03
```

Figure 50 Example User Information Message 5293

In summary, the following types of output are indicators of singularities in the model:

1. Grid point singularity table.

   Some general causes for singularity are

   - Degree of freedom without stiffness because of missing elements.
   - 2-D plate problem with the normal rotation unconstrained.
   - Solid model with rotational DOFs at the corners.
   - Planar joint in space structure.

   AUTOSPC does not detect the singularities associated with

   - Offset beams.
   - Mechanisms and free bodies such as sloped plates, beam to plate connections, beam to solid connections, and plate to solid connections.

2. Warning messages from the DECOMP module.

   Some of the causes for ill-conditioned matrices are:

   - Low stiffness in rotation.
   - Large mass in dynamic solutions.
   - Very stiff beam.
   - Mechanisms.
   - Stiff element adjacent to a very flexible element.

3. Unreasonable displacements with reasonable forces. If a near-singular problem gets through the decomposition, large displacements may result in parts of the structure. A clue that some grid points are in the same mechanism is that all grid points have the same very large (greater than one radian) rotations.

4. A flagged epsilon value

5. Unexpected zero frequency eigenvalues.
6. Nonlinear divergence.

What Diagnostics Are Available for Understanding My Solution Process?

There are numerous diagnostic tools available in MSC.Nastran for debugging and understanding the solution process (see the *MSC.Nastran Quick Reference Guide*). The most widely used diagnostic is DIAG 8, which is invoked by:

```
DIAG 8  $ Executive Control statement
```

DIAG 8 prints matrix information in the .F04 file, such as:

Size of the matrix: COLS
ROWS

Form of the matrix: F

1 = square
2 = rectangular
3 = diagonal
4 = lower triangular
5 = upper triangular
6 = symmetric
8 = identity matrix
10 = Cholesky factor
11 = partial lower triangular factor
13 = sparse symmetric factor
15 = sparse unsymmetric factor

Type of the matrix: T =

1 = real single precision
2 = real double precision
3 = complex single precision
4 = complex double precision

Number of nonzero terms in the densest column (NZWDS)

Density of the matrix (DENS)

Size of the matrix (in MSC.Nastran GINO blocks)

Average string length (STRL)

Number of strings in the matrix (NBRSTR)

Three unused entries

Average bandwidth (BNDAVG)

Maximum bandwidth (BNDMAX)

} This information is used internally for numerical method selection

Number of null columns (NULCOL)

The most important matrix properties COLS, ROWS, T, and DENS are normally used to understand the solution path. For example in the output below, the module EMA has created 270x270 symmetric stiffness and mass matrices KJJZ and MJJX, respectively.

```
 6:57:34     0:26     5.0       .0      4.8       .0    SEMG        53  EMA     BEGN
*8** MODULE  DMAP MATRIX     COLS   ROWS  F  T  NZWDS   DENS BLOCKS STRL  NBRSTR BNDL  NBRBND ROW1 BNDAVG BNDMAX NULCOL
     EMA     53   SCRATCH       3     45  1  1      0  .0000      1    0       1    0       1    0      0      0      0 *8**
     EMA     53   KJJZ        270    270  6  2     36  .0396      5    2    1167    0       1    0     22     33     60 *8**
 6:57:34     0:26     5.0       .0      5.0       .1    SEMG        55  EMA     BEGN
*8** MODULE  DMAP MATRIX     COLS   ROWS  F  T  NZWDS   DENS BLOCKS STRL  NBRSTR BNDL  NBRBND ROW1 BNDAVG BNDMAX NULCOL
     EMA     55   SCRATCH       3     45  1  1      0  .0000      1    0       1    0       1    0      0      0      0 *8**
     EMA     55   MJJX        270    270  6  2      2  .0017      2    0     126    0       1    0      0      1    144 *8**
```

Figure 51 Output from the EMA Module

This type of information can be used to understand where the matrices are generated, reduced, and solved. For more information about the solution process for each solution sequence, refer to the *MSC.Nastran Numerical Methods User's Guide.*

How Do I Use DMAP?

The complete answer to this question could easily fill this book, so the following is a brief explanation on how to use DMAP. DMAP (Direct Matrix Abstraction Programming) is a high-level programming language that is available in MSC.Nastran. All solution sequences in MSC.Nastran are composed of DMAP statements. Any solution sequence may be listed by inserting COMPILER LIST into the Executive Control Section of the input file.

DMAP alters allow you to modify any solution in MSC.Nastran.

If you want to print the sparse factor matrix LLL in SOL 101, the following procedure allows you to modify SOL 101 to provide that output.

First, a little explanation is appropriate. DMAP statements have a standard format:

```
MODULE      IN1,IN2,IN3,..../OUT1,OUT2,OUT3,.../
            PARAM1/PARAM2/PARAM3/... $
```

where:

MODULE is the DMAP module name.

IN1, IN2, IN3, ... are the input data blocks (matrices or tables) for the module.

OUT1, OUT2, OUT3,... are the output data blocks from the module.

PARAM1, PARAM2, PARAM3, ... are parameters that are used either to control the operations of the module or are assigned values by the module.

If you are familiar with FORTRAN, it may be easy to think of a DMAP statement as a subroutine call statement with input and output information.

Most module names are acronyms for the function that is carried out by the module. For example, the ADD module is used to add matrices together. Some of the module names are not as easy to figure out, but still make sense once you know what they mean. For example, REIGL stands for Real EIGenvalue analysis using the Lanczos method. These and a number of other modules are described in the *MSC.Nastran DMAP Programmer's Guide*.

A handy rule to help read DMAP is that the first letter of a matrix name is often used to indicate what type of matrix is being represented (P = loading, Q = constraint force, U = displacement, K = stiffness, M = mass, B = damping, G = transformation, etc.).

In order to modify an existing solution in MSC.Nastran, first it is necessary to list the solution. The following input file lists SOL 101.

```
SOL 101, NOEXE $ get SOL 101, but do not execute it
COMPILER LIST
CEND
TITLE = LISTING OF SOL 101
BEGIN BULK
ENDDATA
```

The following is a selected set of lines from the resulting listing:

```
  1      SUBDMAP SESTATIC $ SOL 101 - SUPERELEMENT STATICS
  2      $  PHASE 0
  .
  .
202      END $ SESTATIC


  .
  .

  1      SUBDMAP SEKRRS  KAA,USET,SILS,EQEXINS/
                         KLL,LLL,DM,ULL,LRSEQ/
                         NOLSET/NORSET/ERROR/FIXEDB/NOQSET/NOTSET $
  2      $
  2      TYPE PARM,,CHAR8,N,SUBDMAP='SEKRRS  ' $
  .
  .
 14      $     MODE ACCELERATION OR SUPORT PROCESSING
 14         DECOMP=-1 $
 15         IF ( ASING>=0 ) DECOMP=16 $ REPLACE NULL ON DIAG WITH
            1.0 AND PROCEED
 16         DCMP   USET,SILS,EQEXINS,KLL,/
                   LLL,ULL,LRSEQ/
                   -1/0/BAILOUT/MAXRATIO/'L'/1.E-20/DECOMP//////
                   S,N,SING/S,N,NBRCHG/S,N,ERR $
 17         IF ( SING<0 ) THEN $
  .
  .
```

Figure 52 Example Listing from SOL 101

Each solution sequence starts with a main subDMAP. In the case of SOL 101 the main subDMAP is SESTATICS. SESTATICS calls a series of subDMAPs, which in turn call other subDMAPs. In this case, the LLL matrix is output by the DCMP module, which resides in the SEKRRS subDMAP.

Since we want to print the LLL matrix, we use the MATPRN module (matrix print). This and other modules are described in the *MSC.Nastran DMAP Programmer's Guide*. The MATPRN module is perhaps one of the simplest modules. It can have up to five input matrices, with no output matrices or parameters.

The form of this DMAP Alter is as follows:

```
.
.
compile sekrrs $
alter 'dcmp' $
$alter 'dcmp'(1,0) $ equivalent statement
$alter 16 $ equivalent statement
matprn lll// $
endalter $
CEND
.
.
```

Figure 53 Form for the Example DMAP Alter

This DMAP Alter should be inserted into the input file just before the CEND statement. In this example, the MATPRN is inserted after the DCMP module, inside the SEKRRS subDMAP. Since the DCMP module is the 16th statement in the SEKRRS subDMAP, this procedure can be accomplished using either ALTER 16 or ALTER 'DCMP.' The use of string-based DMAP can reduce the amount of maintenance between versions. See the *MSC.Nastran Release Guide for Version 68.2* and *Version 69* for further details on string DMAP. When this is included in the input file, the output includes additional output that resembles the following:

```
0                                                                                                    SUBCASE 1
       SPARSE FACTOR MATRIX LLL      ( GINO NAME   101 ) IS A REAL    DOUBLE PRECISION MATRIX OF SIZE     2 AND     1 FRONT(S).
 FRONT MATRIX      1 HAS      2  COLUMN(S)       2 ROW(S)      ---------------------------------------------------
 INDICES
               2           1
 COLUMN      1          ROWS      1 THRU      2
    ROW
      1)    2.5000D-06  1.0000D+00
 COLUMN      2          ROWS      2 THRU      2
    ROW
      2)    2.5000D-06
0*** USER INFORMATION MESSAGE 5293 FOR DATA BLOCK KLL
```

Figure 54 Additional Output

In summary, DMAP is a powerful programming language that can be used to solve almost any problem that can be written in matrix form.

How Can I Use the INCLUDE Statement to Decrease the Complexity of My Input File?

The INCLUDE statement is documented in the *MSC.Nastran Quick Reference Guide*. The INCLUDE statement inserts an external file into the MSC.Nastran input file. It can be placed anywhere within the input file, and therefore, can be used to avoid repetitious input in the FMS, Executive, Case Control, and Bulk Data Sections of the input file. Some examples for its use are indicated below.

In the FMS Section, the INCLUDE statement can be used to assign multiple database files with one statement. In the Executive Section, the INCLUDE statement can be used to include alters to a solution sequence. In the Case Control Section, the INCLUDE file can be useful if the same subcase structure and/or the same output requests are desired for different models.

The INCLUDE statement is probably most useful in the Bulk Data Section. Each superelement of a superelement model can be placed in a different physical file. The files can be shared by more than one user and analyzed separately. Since superelement models are usually very large and complex, the INCLUDE statement can greatly reduce the complexity of the input file.

With the INCLUDE capability, the MSC.Nastran input file can consist of a series of INCLUDE statements. However, they cannot be nested; only one level of include file is supported. Another limitation is that the INCLUDE statement plus the filename must be less than 72 characters. Also, continuations cannot be used.

Below is an example of the use of the INCLUDE statement. Note that the syntax for the filenames is machine dependent. The format of lowercase filenames within single quotes works on all computer operating systems that support MSC.Nastran.

```
INCLUDE 'nastran.dat'
INCLUDE 'database.dat'
SOL 101
INCLUDE 'sol101.alter'
CEND
TITLE = Complete Model
SUBTITLE = Three load cases
INCLUDE 'output.dat'
SUBCASE 1
  LOAD = 10
BEGIN BULK
INCLUDE 'se0.dat'
INCLUDE 'se10.dat'
INCLUDE 'se20.dat'
ENDDATA
```

Each of the INCLUDE files in the example contains different information. For this example, “nastran.dat” contains NASTRAN system cells, “database.dat” contains FMS statements, “sol101.alter” contains a DMAP Alter for SOL 101, “output.dat” contains Case Control commands, and “se*.dat” contains modeling information for superelements 0, 10, and 20.

Can I Migrate Old Databases for Compatibility with Version 69 or Version 70?

Databases created from previous versions of MSC.Nastran are not compatible with Versions 69, 69.1, or 70. There is no method to convert pre-Version 69 databases to Version 69 databases due to changes required to support 64-bit precision of geometry information. The databases for Versions 69 and 69.1 are not compatible with those of Version 70 due to a new matrix storage technology introduced in Version 70.

Input files from previous versions of MSC.Nastran may be used to create Version 69, 69.1, or 70 databases.

How Do I Determine the VERSION Number for a Restart?

The general format for the RESTART command is as follows:

RESTART VERSION = a,b

where “a” is the version from which you restart (default value for “a” is LAST) and “b” indicates whether version “a” will be kept (KEEP) or deleted (NOKEEP) at the end of the run. The default value for “b” is NOKEEP. Due to the default values, the following two restart commands are identical:

RESTART VERSION=LAST,NOKEEP

RESTART

Each time a run that uses a database is submitted, a new version number is automatically created in the database. For each restart, the current version number is incremented by one, regardless of whether the job ran successfully or not. There are two exceptions to this rule and they will be discussed later. A message is printed near the top of the .F04 file indicating the version number that the current job is running on. However, not all versions are restartable. In general, if a job completes without any fatal messages, then that particular version is restartable. It is a good idea to keep a brief log of all the restartable versions since this is the most reliable source of information regarding whether a version is restartable or not. If a restart job failed; e.g., due to a Bulk Data error, then this newly created version is not restartable, and this is indicated by the following error message at the bottom of the .F06 file.

```
*** USER WARNING MESSAGE 784 (XCLNUP)
VERSION =  yyy PROJECT =  "zzz"  OF THIS DATA BASE IS NOT     VALID FOR RESTART PURPOSES.

USER ACTION:
SUBSEQUENT RESTARTS SHOULD REFERENCE VERSION =  xxx OR A
PRIOR VALID VERSION
```

Figure 55 Sample Restart Messages in the .F06 File

In this case, you must restart from a previous valid version.

If for some reason the records for the old runs are no longer available, then the DBDIR FMS statement can be used to query the database contents to find out which versions are being stored in the database. The following simple setup is all that is required for this purpose.

```
ASSIGN MASTER='ddddd.MASTER'
DBDIR VERSION=*,PROJECT=*
ENDJOB
```

Neither the Executive Control, Case Control, nor the Bulk Data Section is required in this case. For this run, a new version is not created. Near the top of the .F06 output, a Project Version Directory Table is printed listing all the versions in the database. An asterisk next

to a version number indicates that this particular version was deleted from the database. This may be due to the "NOKEEP" option or the use of the DBCLEAN command when performing one of the restart runs. A version number with an "asterisk" is not restartable. A version number without an "asterisk" is restartable if the run that created the version did not terminate with UFM 784. For more advanced users, the DBDIR FMS statement can also be used to check the database directory for the existence of data blocks; e.g., UG. If the job fails very early in the run, e.g., an error in the FMS section, then a new version may not be created.

It is always a good practice to back up the database on a regular basis. If the system aborts the run, e.g., the disk space is exhausted or the time quota is exceeded on a system level, then there is a chance that the database is corrupted. To ensure that only good models are retained in the database, use the following practices:

- Use RESTART VERSION=a,KEEP
- If a version contains errors, or is no longer of interest, then use the FMS statement DBCLEAN to remove obsolete or incorrect versions from the database. This allows the Executive System to reuse some of this space for new versions.

What Should I do if I Have Trouble Punching Out an Output Quantity?

The default punch format is based on an older method which does not perform as much checking as the newer method, which is based on the NDDL description. The older method (default) should work for most cases, and is the recommended method since it normally runs faster than the newer method. Should the default method fails, you can turn on the newer method by adding the following nastran statement to your input file:

```
nastran system(210)=1
```

What is the Advantage of Using OUTPUT2 Versus Punch File?

If you are requesting a large amount of punch output, the time spent in it can be a substantial portion of your runtime. This is due to the fact that formatted output (such as punch output request) is very time consuming. A better alternative is to use the OUTPUT2 option. If you are using this OUTPUT2 file for subsequent post processing on the same computer, then use the binary (unformatted option). If you are transferring this OUTPUT2 file to a different machine, then use the "ascii" (formatted option). A 10,000 grids model-small in today's standard-is used to illustrate the time comparison using the three different options.

PUNCH	OUTPUT2 (Unformatted)	OUTPUT2 (Formatted
200.6	3.9	82.6

Can I Use an OUTPUT2 Binary File Generated by MSC.Nastran on a Different Machine?

An OUTPUT2 file created on one machine can be used directly on a different machine provided that they belong to the same class of machine. The machines supported by MSC.Nastran can be divided into six groups based on the way each machine addresses data.

Table 4 Machines Supported by Version 69 MSC.Nastran

Machine Categories	Machine	Operating System
32 bit, big endian, IEEE	IBm RISC System/6000	AIX
	SUN	SOLARIS
	SGI	IRIX
	HP	HP-UX, HP Exemplar, SPP-UX
32 bit, little endian, IEEE	INTEL	Windows NT
	Digital Alpha	UNIX, Windows, NT
32 bit, little endian	Digital Alpha	VMS
64 bit, big endian, IEEE	Cray T90	Unicos
	NEC SX-4	SUPER-UX
64 bit, big endian	Cray C90/T90	Unicos
	Cray J90	Y-MP, Unicos
32 bit, big endian	IBM S390	MVS

Prior to Version 67.5, the OUTPUT2 file generated by MSC.Nastran was available only in binary format. This makes the OUTPUT2 file machine-dependent. However, if you are running MSC.Nastran on one of the above machines, then the OUTPUT2 file generated by one of these machines can be used directly by MSC.Nastran on another machine provided that they both belong to the same group. This OUTPUT2 file can be transferred directly across machines using a file transfer program (e.g., FTP), or you can read this file directly if your machine is on a network and you are remote mounted to the disk that the OUTPUT2 file resides on. For example, if the OUTPUT2 file is generated on a SUN, you can use this OUTPUT2 file directly on an IBM RISC System/6000 since they are both 32 bit, big endian, IEEE machines. On the other hand, an OUTPUT2 binary file generated on the HP cannot be used directly by an INTEL-based machine since they belong to different machine categories.

Starting in Version 67.5, an option is added such that the OUTPUT2 file can be written in a neutral format (i.e., compressed ASCII). This compressed ASCII format makes the OUTPUT2 file machine-independent, and this file can be transferred to any computer desired. This compressed ASCII file, however, cannot be edited. The INPUTT2 module of

MSC.Nastran is also modified to allow the direct use of the neutral files. The program RCOUT2 that is furnished on the delivery tape can read the neutral files and convert them into binary. This program is machine-dependent, and the source code is provided for the program (MSC.Nastran Version 67.5 and later). Consequently, this code may be modified to suit your needs and allows you to transfer the OUTPUT2 file to computers other than those supported by MSC.

You only need this stand-alone program if you do not have MSC.Nastran on that platform; otherwise, as mentioned previously, a copy of RCOUT2 is delivered with the MSC.Nastran delivery tape. A detailed description of the OUTPUT2 neutral format is available in the *MSC.Nastran Release Notes for Version 67.5*.

How is the qsub Memory Used on the Cray?

The qsub memory is NOT a hard limit, it is an upper limit on what the process may use. So, unlike the NASTRAN memory request, which will hold that memory for the duration of the run, the -lm and -lM process limits do not 'lock' up the memory from other processes. Therefore, if a user requests 7 mw and it is not required, the job will use only the value required.

MSC.Nastran Version 69 has a new non-segmented executable approach. Prior to Version 69, MSC.Nastran would load that portion of the executable required into mem, and swap out code from disk as necessary. This would result in increased elapsed and system cpu time due to I/O. Every job would load its own executable independently, and required a que mem (memory) of approximately 5 mw above the MSC.Nastran mem.

The new non-segmented approach in Version 69, similar to most UNIX applications, loads the entire executable into memory, reducing the swap (I/O) penalty for linking various portions of the code. This effect is most pronounced on transient, nonlinear, and large DMAP looping problems, and may save up to 5% on cpu time, and have a larger effect on real time.

The other advantage of this method is that the shared portion of the executable, approximately 2.5 mw, is accessed simultaneously by all users. The first user will load the executable into memory, by requesting and using 7 mw of memory, but all subsequent users will share that portion of the code, and use approximately 4.5 mw for que mem. Using the available scripts, the users must request 7 mw for que mem, but the large majority of jobs will not use all of it, making that space available for other processes.

The Version 69 executable has grown in addition to the new non-segmented approach with the addition of 300,000 lines of code. The Version 69 MSC.Nastran executable is larger due to many additional features (e.g., enhanced superelements, beam library, parallel Lanczos, p-elements, nonlinear, etc.).

Nonlinear Analysis

What Elements Are Supported for Nonlinear Analysis?

The following elements are supported in both nonlinear static analysis and nonlinear transient analysis to the extent described in the following table. Most of the other structural elements can be included in a nonlinear model with the provision that they remain linear throughout the analysis. The most common misapplication of elements in a nonlinear analysis is related to use of the CBAR, RBE, and CGAP elements. The CBAR is a linear element only, the RBE element is a small displacement element (its coordinates are not updated as it deforms), and the CGAP element is a small displacement element (its coordinates are not updated as it deforms). A summary of properties of the nonlinear elements is shown in the following table.

Table 5 Summary of Properties of the Nonlinear Elements

Connectivity	Properties	Geometric Nonlinearity	Material Property			Static Loads			DATA Recovery		
			Nonlinear Isotropic	Anisotropic	Orthotropic	Thermal	Pressure	Gravity	Stress	Grid Point Stress	Force
BCONP BLSEG	BWIDTH BFRIC	X							X		X
CBEAM*	PBEAM	X	MAT1			X		X	X		X
	PBCOMP	X									
CBUSH	PBUSH								X		X
CGAP	PGAP								X		X
CHEXA†	PSOLID	X	MAT1	MAT9		X	X	X	X	X	
	PLSOLID	X	MATHP			X	X	X	X		
CONROD	---	X	MAT1			X		X	X	X	
CPENTA†	PSOLID	X	MAT1	MAT9		X	X	X	X	X	
	PLSOLID	X	MATHP			X	X	X	X		
CQUAD4	PSHELL	X	MAT1	MAT2	MAT8	X	X	X	X	X	X
	PCOMP	X									
	PLPLANE	X	MATHP			X		X	X		
CQUAD8	PLPLANE	X	MATHP			X		X	X		
CQUAD	PLPLANE	X	MATHP			X		X	X		
CROD	PROD	X	MAT1			X		X	X		X

Table 5 Summary of Properties of the Nonlinear Elements (continued)

Connectivity	Properties	Geometric Nonlinearity	Material Property			Static Loads			DATA Recovery		
			Nonlinear Isotropic	Anisotropic	Orthotropic	Thermal	Pressure	Gravity	Stress	Grid Point Stress	Force
CTETRA†	PSOLID	X	MAT1	MAT9		X	X	X	X	X	
	PLSOLID	X	MATHP			X	X	X	X		
CTRIA3	PSHELL	X	MAT1	MAT2	MAT8	X	X	X	X	X	X
	PCOMP	X									
	PLPLANE	X	MATHP			X		X	X		
CTRIA6	PLPLANE	X	MATHP			X		X	X		
CTUBE	PTUBE	X	MAT1			X		X	X		X
CQUADX	PLPLANE	X	MATHP			X	X	X	X		
CTRIAX	PLPLANE	X	MATHP			X	X	X	X		

* For the beam element, only elastic-perfectly plastic material is available.

†Edge nodes are not applicable to nonlinear solid elements, except for hyperelastic element.

MATS1 and CREEP may be attached to MAT1.

For more information on nonlinear elements, refer to the *MSC.Nastran Reference Manual.*

Which Nonlinear Iteration Method Should Be Used?

Since each nonlinear problem is different, it is difficult to generalize the selection of an iteration solution method. In nonlinear statics (SOL 106), the AUTO, SEMI, and ITER methods are available. In a nonlinear transient analysis (SOL 129), the AUTO, TSTEP, and ADAPT methods are available. In addition, a number of convergence acceleration techniques (e.g., quasi-Newton, line search, bisection) exist to enhance the nonlinear solution convergence. Each of the methods examines the appropriate parameters, convergence criteria, and error indicators to proceed through the solution.

AUTO Method. The AUTO method works very well as a starting point and is the default method in the nonlinear static solution. This method essentially examines the solution convergence rate and uses the quasi-Newton, line search and / or bisection methods to perform the solution as efficiently as possible sometimes without a stiffness matrix update. The AUTO method can also be used in a nonlinear transient analysis. Highly nonlinear behavior in some cases may not be handled effectively using the AUTO method.

SEMI Method. The SEMI method is also an efficient technique for nonlinear static iteration. It is similar to the AUTO method since it uses the quasi-Newton, line search and / or bisection methods, but it differs from the AUTO method in one respect. The SEMI method forces a stiffness matrix update after the first iteration of a load increment. This update is performed irrespective of the convergence status of the solution. It is effective in many highly nonlinear problems where regular stiffness matrix updates help the solution to converge. The SEMI method does not have a counterpart in nonlinear transient analysis.

ITER Method. The ITER method is known as the "brute force" method in nonlinear static analysis because a stiffness matrix update is forced every KSTEP iterations with a resultant increase in CPU time. The TSTEP method is its counterpart in nonlinear transient analysis. Its best applications tend to be those involving highly nonlinear behavior for which using only multiple iterations may not be efficient.

ADAPT Method. The ADAPT method is used in nonlinear transient analysis to modify the integration time step during the analysis. This is done by monitoring the dominant response frequency of the system and changing the integration time step automatically (within bounds) to respond to changes in the frequency content of the loading and response. However, it may force the solution to solve for unnecessary high frequency modes. The ADAPT method is the default in nonlinear transient analysis and can be used for the linear structures in nonlinear solution sequences.

For more information on the nonlinear iteration methods, refer to the NLPARM and TSTEPNL entry descriptions in the *MSC.Nastran Quick Reference Guide* and the *MSC.Nastran Handbook for Nonlinear Analysis*.

What Can Be Changed In a Nonlinear Restart?

The following can be changed in a restart of a nonlinear analysis:

1. Linear elements may be added or modified. This modification should not be drastic in order to avoid large initial unbalanced loads.
2. Elastic material properties (e.g., MAT1, etc.) may be changed.
3. Additional applied loads for new solutions.
4. Boundary conditions (statics only).

In Nonlinear Analysis, What Is the Difference Between Load Increment, Subcase, and Iteration?

In a nonlinear analysis, loading is typically applied in stages both to allow for the nonlinear behavior to occur in the numerical processing and to give you control over restarts if problems (divergence, excessive iterations, etc.) occur during the solution. The choice of the solution strategy is based on the structure itself, the nature of the loading, and the anticipated nonlinear behavior. Part of the overall solution strategy includes dividing the loading into logical steps, controlling the numerical processing, and planning for the possibility of changing the solution strategy during the analysis using restarts.

The subcase structure in a nonlinear analysis differs from a typical linear analysis. In a nonlinear analysis, subcases are cumulative; that is, the loads and boundary conditions at the end of a subcase are the initial conditions for the next subcase. Only one independent loading history can be applied during an analysis. In general, a different loading sequence requires a complete new analysis.

Function of the Subcase. In a nonlinear static analysis, you first determine the total value of loading to be applied at a particular stage of the analysis. This loading value is selected with the LOAD Case Control command specifying a load set ID that exists in the Bulk Data. In this case, the subcase functions as a type of landmark in the loading history. It may be an expected point or a point at which the nature of the loading changes (for example, first applying an internal pressure loading and then an axial loading on a cylinder). The subcase is a major partition of the loading history. The loading history should be divided into subcases since this provides the user with more control over the solution and restart strategy.

Restarts. When planning restarts, you should be aware of the INTOUT field on the NLPARM Bulk Data entry. If INTOUT is NO (the default), the only solution information saved in the database is the data for a converged subcase. If a particular subcase has 10 load increments and INTOUT = NO, only information related to the last load increment is saved. If INTOUT = YES, data is saved after each converged load increment. This operation gives the user more control over restarts especially for rapid changes in the behavior but uses more space in the database.

Load Increment. In the loading history, the total change of loading applied during a subcase can be subdivided into smaller parts to allow the solution to converge. These subdivisions within a subcase are termed load increments and are defined by the NINC field on the NLPARM Bulk Data entry. Selecting a number of increments divides the total load change applied during the subcase into NINC equal parts. Selecting YES in the INTOUT field on the NLPARM entry allows for that load increment to be saved to the database for possible restart. This is important because many times the solution does not converge during a subcase. If the loading is divided into increments and these values are saved to the database, the restart strategy can continue from a loading value closer to the problem value than having to go back to the previous subcase.

Iteration. In the incremental solution process, the unbalanced forces that occur during a load increment are re-introduced internally into the solution until the solution has converged. The process of redistributing the unbalanced force within a load increment is known as an iteration. The iteration is the lowest level of the solution process. Iterations continue within a load increment until the solution converges or any of the solution parameters on the NLPARM Bulk Data entry are exceeded.

How Can the Printed Output Produced by the Nonlinear Solution Sequences (Solutions 106 and 129) Be Reduced?

Much of the printed output for the nonlinear solution sequences consists of the following:

- DIAG 51 Output. DIAG 51 is intended for debugging purposes only; therefore, it is not recommended for general usage.
- Nonlinear Stress Output. By default, the nonlinear stress output is always printed. To avoid the printing of the nonlinear stress output use the following Case Control command.

  NLSTRESS = NONE

Are There Changes in the Use of LOOPID, SUBID, and LOADINC for Nonlinear Restarts?

The major change in the nonlinear restart parameters is that PARAM,LOADINC is no longer used starting in Version 67. Parameters SUBID and LOOPID are used as they were in Versions 65 and 66.

What Is the Orientation of the CGAP Element?

The CGAP element coordinate system is defined by one of the two following methods:

1. Coincident Grid Points

   If the grid points defining the element are coincident, the orientation of the element is defined using the coordinate system CID. In this case, the gap element x-axis is the T1 direction of the CID coordinate system, the y-axis is the T2 direction, etc. If unspecified, the value of the coordinate system is the basic system.

2. Non-Coincident Grid Points

   If the grid points defining the element are not coincident, the orientation of the element is defined using the element x-axis defined by the GA-GB grid orientation, and the v vector is defined using the X1, X2, X3, or G0 entries.

How Are the Subcases in Nonlinear Analysis Different from Linear Analysis?

In a linear analysis, subcases represent an independent loading condition. Each subcase is distinct from all others. In a nonlinear analysis, the end of a subcase is the initial condition for the next subcase. Nonlinear static analysis permits only one independent loading condition per run. However, this loading does not need to be constant in magnitude or in position. Loadings can vary from subcase to subcase. Since a nonlinear static analysis is path dependent, the geometric and material changes in subcases are cumulative. For further details, refer to the *MSC.Nastran Handbook for Nonlinear Analysis.*

How Do Restarts Differ in Nonlinear Analysis Versus Linear Analysis?

Restarts in a static analysis are typically performed for such things as additional data recovery, new loadings, new boundary conditions, etc. When the Structured Solution Sequences are used, the MSC.Nastran NDDL performs the restart using its automatic restart logic. In the simplest cases, special input data and / or parameters are not required.

Restarts in a nonlinear analysis are usually performed to continue a solution into a new series of loading steps using the previous analysis as the starting conditions. In addition, a nonlinear static analysis can be used as initial conditions for a nonlinear transient analysis, or a nonlinear transient analysis can be continued for a longer duration.

PARAM,LOOPID. The key in any nonlinear restart is the output from the prior run that identifies the LOOPID which forms the initial conditions for the restart run. A LOOPID is created after each converged load increment in static analysis and at the end of each converged subcase in nonlinear transient analysis. Only LOOPIDs written to the database and identified as such in the printed output can be used for a restart, i.e., if INTOUT = NO on the NLPARM entry, only the LOOPID at the end of the converged subcase is saved.

PARAM,SUBID. In the nonlinear static restart, the user enters PARAM,LOOPID,n (where n = LOOPID from previous run to be used as initial conditions) in the Bulk Data or Case Control Section. An additional parameter SUBID identifies the subcase ID which is being used for the current analysis. The subcase ID is not the subcase number. It is the sequence number of the subcase in the current subcase structure (first subcase = 1, second subcase = 2, etc.). Usually the new SUBID = last SUBID + 1. The SUBID and LOOPID values are printed in the nonlinear iteration module output after each converged load increment.

Data Recovery Restarts. Data recovery restarts in nonlinear static analysis are performed as follows:

1. Include the desired output requests above the subcase level.
2. Set SUBID = last SUBID+1
3. Set PARAM,LOOPID to the desired load increment. Data recovery is performed for all saved load increments through the LOOPID load increment.

PARAM,STIME. When restarting a nonlinear transient analysis, PARAM,LOOPID,n is used to provide the initial conditions for the restart. The value of n must refer to the last LOOPID in a converged subcase that is printed in the nonlinear iteration module output. PARAM,STIME,t (where t = starting time for the analysis) is also set. For data recovery restarts, PARAM,STIME is not required, and PARAM,LOOPID should be set equal to the last converged LOOPID.

PARAM,SLOOPID. If a nonlinear static analysis is used as the initial condition for a nonlinear transient analysis, PARAM,SLOOPID,n (where n = LOOPID from static analysis) is used to select the nonlinear static loading condition that is to be used in the nonlinear transient analysis. The initial transient loading must be identical to the static loading (specified in the LOOPID) selected as the initial conditions.

In the Structured Solution Sequences for nonlinear analysis, the automatic restart logic only applies to upstream superelements.

Solution of Large Models

What Is the MSC.Nastran Database?

The MSC.Nastran database is a place to store data for future restarts. The default MSC.Nastran database consists of two DBsets: .MASTER and .DBALL. The .MASTER and .DBALL DBsets are commonly used to store data for restarts. Additional DBsets can be assigned using the INIT FMS statement. Refer to the *MSC.Nastran Quick Reference Guide*.

Describe the Structured Solution Sequence (SSS) Restart

Structured Solution Sequences (SSS) restarts in MSC.Nastran are performed by using the RESTART FMS statement, assigning an old SSS database, and supplying the Bulk Data changes from the previous runs. The most efficient restarts are additional data recovery requests, load changes, solution sequence changes, and model changes for superelements.

How Do I Know What an Automatic Restart Run Has Done?

The automatic restart logic is designed to detect changes in the Executive Control Section (SOL changes), the Case Control Section (LOAD, SPC, METHOD, etc. changes) as well as the Bulk Data Section. Most major changes are detected by the RESTART module. This module echoes Case Control and Bulk Data changes in the .F04 file. For example, if an EIGR Bulk Data entry is changed, the RESTART module issues the following message:

```
6:58:09     0:20     6.6        .0      4.3        .0    IFPL         86  RESTART BEGN
6:58:09     0:20     6.6        .0      4.3        .0    IFPL         87  RESTART BEGN
      DYNAMICS EIGR               SID      HAS CHANGED
      DYNAMICS EIGR               METHOD   HAS CHANGED
      DYNAMICS EIGR               F1       HAS CHANGED
      DYNAMICS EIGR               F2       HAS CHANGED
      DYNAMICS EIGR               NE       HAS CHANGED
      DYNAMICS EIGR               ND       HAS CHANGED
      DYNAMICS EIGR               NORM     HAS CHANGED
      DYNAMICS EIGR               G        HAS CHANGED
      DYNAMICS EIGR               C        HAS CHANGED
6:58:09     0:20     6.6        .0      4.4        .0    IFPL         88  RESTART BEGN
6:58:09     0:20     6.6        .0      4.4        .0    IFPL         89  RESTART BEGN
```

How Do I perform Read-Only Restarts?

Read-only restarts allows MSC.Nastran to attach a database for restart purposes that is different from the database used to store new information. The original database is accessed in a read-only mode. No new data is added to its files; consequently, the size of the read-only database does not increase with each restart. Instead, all new data is written to the files of the new database. For example, RUN1 contains the results from a real eigenvalue analysis. If the real eigenvalue information is used in a modal frequency response analysis, the RUN2.dat file would read as follows:

```
RESTART  LOGICAL= RUN1
ASSIGN RUN1='run1.MASTER'
$
$ REST OF THE FILE MANAGEMENT SECTION
$
.
$
$ EXECUTIVE CONTROL SECTION
$
SOL 111
.
.
$
$ CASE CONTROL SECTION
$
$    CASE CONTROL STRUCTURE SHOULD BE IDENTICAL TO ORIGINAL
$    CASE CONTROL SECTION EXCEPT FOR ANY REQUIRED COMMANDS
$    FOR NEW ANALYSIS PROCEDURE AND DATA RECOVERY
$
.
$
BEGIN BULK
$
$ ADD ANY NEW ENTRIES REQUIRED FOR NEW ANALYSIS
$
ENDDATA
```

The restart procedure will access the RUN1 database for use in the new analysis. Any new information will be added to the new database, leaving the original database intact. The new database can be deleted, or it can be saved for restart purposes. The database created during the restart run will retain an internal link to the original database. Note that if the following FMS statements

```
RESTART
ASSIGN MASTER = 'run1.MASTER'
```

are used instead of the previous statements, new data will be written into the RUN1 database.

When a restart is performed, the program will automatically open all the files required to access any previously stored information. For example, the run is a restart of the modal frequency-response analysis from a real mode analysis. If the second database is saved, a second restart for additional data recovery can be performed using the database created from the second run.

The RUN3.dat file would appear as follows:

```
RESTART  LOGICAL= RUN2
ASSIGN RUN2='run2.MASTER'
$
$ FILE MANAGEMENT SECTION
$
.
```

```
$
$ EXECUTIVE CONTROL SECTION
SOL 111
.
$
$ CASE CONTROL SECTION
$
$  CASE CONTROL STRUCTURE SHOULD BE IDENTICAL TO ORIGINAL
$  RESTART CASE CONTROL EXCEPT FOR ANY REQUIRED COMMANDS
$   FOR ADDITIONAL DATA RECOVERY
.
.
BEGIN BULK
$
ENDDATA
```

The second restart will use the database from the second run (first restart run) for restart checking and data recovery. MSC.Nastran will automatically access the RUN1 and RUN2 databases for any required data.

The RUN2 database contains a record of information stored in the RUN1 database and will automatically access the RUN1 database for any required data. This capability makes restart chaining possible for each run that has its own database containing the additional data calculated during the run.

How Can I Assign a New DBset?

Use the INIT FMS statement to initialize a new DBset and the ASSIGN statement to assign the DBset to a physical file. See the following example. For more information, refer to the *MSC.Nastran Quick Reference Guide*.

```
INIT DBDOWN, LOGICAL=(DBDOWN(1500MB))
ASSIGN DBDOWN='BODY.DBDOWN'
```

How Can I Store Data into My Newly Created DBset?

The location parameters DBDN or DBUP can be assigned in either the Bulk Data Section or in the Case Control Section. Refer to the *MSC.Nastran Quick Reference Guide* for more information. For example:

```
$ FMS Section of the MSC.Nastran input file
INIT DBDOWN, LOGICAL=(DBDOWN(1500MB))
ASSIGN DBDOWN='BODY.DBDOWN'
$ Case Control or Bulk Data Section of the
$ MSC.Nastran input file

PARAM,DBDN,DBDOWN
```

How Can I Expand the DBALL DBset?

DBsets can be expanded in a restart run or by rerunning the job from scratch. The expansion is accomplished by adding a new physical file to the DBALL DBset.

- For a restart run, assign the database and submit the restart job with the following FMS statements:

```
RESTART
EXPAND DBALL, LOGICAL=(DBALL1(yyyyy))
ASSIGN DBALL1='file_name'
```

  where yyyyy is the size of DBALL1. Prior to Version 68, the yyyyy value is in units of GINO blocks. Beginning in Version 68, this value can be in terms of GINO blocks, bytes, words, megabytes, megawords, kilobytes, or kilowords. See the *MSC.Nastran Release Notes for Version 68* for further details.

- For a scratch run, delete the database and resubmit the original job with the following FMS statements:

```
INIT DBALL,LOGICAL=(DBALL(yyyyy))
ASSIGN DBALL='file_name'
```

  where yyyyy is the size of DBALL.

- For a scratch run using two disks, delete the database and resubmit the original job with the following FMS statements:

```
INIT DBALL,LOGICAL=(DBALL(xxxxx),DBALL1(yyyyy))
ASSIGN DBALL='/disk1/file_name'     $ residing on disk pack '/disk1'
ASSIGN DBALL1='/disk2/file_name1'   $ residing on disk pack '/disk2'
```

  where xxxxx is size of 'file_name' and yyyyy is size of 'file_name1'. The default value for xxxxx and yyyyy in is 250000 MSC.Nastran blocks. Refer to the *MSC.Nastran Quick Reference Guide* for more information.

How Can I Reduce the Size of the SCRATCH DBset?

The size of SCRATCH DBset can be reduced by:

1. Assigning the SCRATCH DBset to multiple disks. This process does not reduce the size of SCRATCH DBset; however, it does distribute the size across different disks.
2. Using more efficient numerical methods in the following order:
   a. Increase the available memory, which reduces the number of spill groups and passes.
   b. Use the Lanczos method with the sparse solver for normal modes analyses.
   c. Use the resequencers in MSC.Nastran.
   d. Increase the available memory to the functional modules by setting the "smem" to zero. These parameters are documented in the *MSC.Nastran Configuration and Operations Guide*.

Does the Size of My Database Increase If I Specify a Very Large BUFFSIZE?

Using a very large BUFFSIZE for small- to medium-sized models may increase the size of the database (especially the DBALL and SCRATCH DBsets). The minimum amount of data written is one BUFFSIZE; therefore, the space used by small data blocks is increased by "padding" it up to a full block. However, large-size models will show relatively little increase in database size for increased BUFFSIZE.

How Can I Use the Lanczos Method Effectively?

The Lanczos method performs decomposition at the lower and upper bound frequencies specified on the EIGRL entry. It also saves the lower triangular factors of the lower and upper bound frequency shifts. The following steps are recommended to reduce the disk and CPU requirements:

1. Use the sparse solver if possible. It uses more memory but tends to be faster and uses less disk space. The sparse solver is now the default solver.
2. Specify only the upper frequency boundary (V2); only one boundary shift factor is saved.
3. Use the space saver option invoked by

   ```
   NASTRAN SYSTEM(146)=1
   ```

For more information, refer to this guide's section on normal modes and the EIGRL Bulk Data entry description in the *MSC.Nastran Quick Reference Guide*.

How Can I Run My Job Most Efficiently?

Use the following MSC.Nastran parameters to tune your job:

1. Increase BUFFSIZE if your model is large. Do this using

   ```
   NASTRAN BUFFSIZE=n $
   ```

   This is machine dependent; for more information, see the *MSC.Nastran Configuration and Operations Guide.*
2. Reduce the number of spill groups or passes by increasing the available memory by using the "mem" keyword (or REGION for IBM computers) on the MSC.Nastran submittal procedure.
3. If there is no spill or passes, increase smem by using the "smem" keyword on the MSC.Nastran submittal procedure or use

   ```
   INIT SCRATCH(MEM=n)
   ```

   where n is in MSC.Nastran blocks.

How Can I Reduce the Size of the DBALL DBset?

Reduce the size of the DBALL DBset by

- Assigning it to multiple files. This process does not reduce the size of DBALL DBset. However, it distributes the size across different disks.
- Using split database techniques.
- Use scr = mini when submitting the job. With this option, only data recovery type restarts are allowed.

How Do I Allocate Larger Disk Space for the SCRATCH and SCR300 Files?

The INIT statement is used for creating / initializing a permanent and / or temporary DBset. The INIT statement has two basic formats: one for all the DBsets and one specifically for the SCRATCH DBsets. This section is dedicated to the description of the disk space allocation for the SCRATCH DBsets. The allocation of disk space for all the other DBsets is quite straightforward and warrants no further discussion since it is covered extensively in the *Introduction to MSC.Nastran Version 67* guide.

The general format for allocating the SCRATCH DBset is as follows:

```
INIT SCRATCH [LOGICAL=(log-name1(max-size1)   ,
  log-name2(max-size2),...log-namei(max-sizei))  ,
  SCR300=(log-namei+1(max-sizei+1),...log-namen(max-sizen))]
```

Log-name1 through log-namei are allocated for regular scratch files as temporary workspace. This space is not released until the end of the job. SCR300 is a special keyword that indicates that the log-names are members reserved for DMAP module internal scratch files. The space occupied by these files is for the duration of the execution of the module. This space is released at the end of the module execution. You can have up to a combined total of 20 logical names for the SCRATCH DBset (1 v n v 20). The usage is perhaps best illustrated with the following example.

```
INIT SCRATCH LOGICAL=(SCR1(150MB),SCR2(100MB)) ,
     SCR300=(SCRA(250MB),SCRB(300MB))
ASSIGN SCR1='cqa.scr1'
ASSIGN SCR2='cqa.scr2'
ASSIGN SCRA='cqa.scra'
ASSIGN SCRB='/mydisk1/cqa.scrb'
```

This example creates the SCRATCH DBset with logical names of SCR1, SCR2, SCRA, and SCRB. Two physical files, cqa1.scr1 and cqa.scr2, are created with 150 and 100 megabytes, respectively. These two files are regular scratch files and reside in the current directory. Two additional physical files, cqa.scra and cqa.scrb, are created with 250 and 300 megabytes, respectively. These two files are SCR300-type files. The SCR300 file cqa.scra resides in the current directory. The SCR300 file cqa.scrb is allocated to the file system (directory) / mydisk1.

To increase only the size of the SCR300 part of the SCRATCH DBset, use the following:

- Increase the size of the SCR300 allocation:

```
INIT SCRATCH, SCR300=(SCR1(zzzzzz))
ASSIGN SCR1='file_name'
```

- Increase the size of the SCR300 using two disks:

```
INIT SCRATCH, SCR300=(SCR1(zzzzzz), SCR2(wwwww))
ASSIGN SCR1='/disk1/file_name3'  $ residing on the scratch pack '/disk1'
ASSIGN SCR2='/disk2/file_name4'  $ residing on the scratch pack '/disk2'
```

  where zzzzzz is size of 'file_name3' and wwwww is size of 'file_name4' in MSC.Nastran blocks. The default value for zzzzzz and wwwww is 250000 MSC.Nastran blocks. Refer to the *MSC.Nastran Quick Reference Guide* for more information.

How Can I Estimate the Size of My Database?

Starting with Version 69, a utility program called ESTIMATE is included with the delivery media which can be used to estimate memory and disk space requirements for MSC.Nastran runs. The *MSC.Nastran Resource Usage Guide* also provides empirical data that will serve as a guideline to estimate the resources requirement for your run.

How Can I Estimate the CPU Time Required for My Run?

It is difficult to obtain an accurate estimate of the CPU time required for any finite element job because the CPU time is controlled by many factors. Among these factors are the model size, type of elements used in the model, element connectivity, type of analysis solution method, and types and amount of output desired.

The *MSC.Nastran Resource Usage Guide* can be used as a guideline for your resource estimation. These empirical data are tuned for Solutions 101 and 103. They provide reasonable estimates for other solutions except for the nonlinear solutions sequences.

What Are the Considerations When Specifying "mem" in the MSC.Nastran Submittal Procedure?

The "mem" keyword in the submittal procedure enables you to request a specific amount of computer memory for use by MSC.Nastran. For example, to submit an MSC.Nastran input file named "test.dat" with 5,000,000 words for memory, you enter:

```
nastran  test  mem=5000000
```

Beginning with Version 68, you can specify this memory value in units other than words. For details, refer to the *MSC.Nastran Release Notes for Version 68*. When using the mem keyword, it is important for efficiency reasons to specify a value that is significantly less than the real memory available. (Real memory is the true physical memory available.) If MSC.Nastran is given more memory (using the mem option) than there is real memory available, most operating systems will use virtual memory. With virtual memory, the machine swaps between memory and disk. This arrangement defeats the purpose of the MSC.Nastran spill logic, which is designed to handle large matrices and tables. The spill logic within MSC.Nastran is much more efficient than using virtual memory. Because the

operating system (including shells and windowing systems) requires a substantial amount of memory, a useful recommendation in an X-Windows environment is to set mem to no more than half of the available real memory.

Note that nastran is the command procedure name, and this procedure may differ depending on how MSC.Nastran was installed on your machine.

When Should I Use the Iterative Method?

For a large problem that consists primarily of solid elements, the iterative solver may be a better choice than the direct solver. For these types of models, substantial disk space savings may be realized. For well-conditioned solid models, if sufficient memory is available, performance improvement can also be obtained on most machines. The Iterative Solver requires substantially more memory than the direct solver and is most efficient if there is no spill. To turn on the iterative solver, insert the following nastran statement at the top of your input file.

```
nastran iter=yes
```

Optimization and Design Sensitivity

How Are the Design Optimization and Design Sensitivity Different in Various Versions?

The tables below show the optimization and sensitivity capabilities in Version 67, Version 67.5, Version 68, and Version 68+.

Table 6 Design Optimization in Versions 67 and 67.5 of MSC.Nastran

Solution		Responses	Design Quantities
No.	Discipline(s)		
200	Statics, Normal Modes, Buckling	Weight, Volume, Eigenvalue, Buckling Load Factor, Displacement, Force, Strain, Stress, Lamina Strain, Stress, and Composite Failure Criterion	Element Properties (Size)

Table 7 Design Sensitivity Analysis in Versions 67 and 67.5 of MSC.Nastran

Solution		Responses	Design Quantities
No.	Dscipline(s)		
108	Direct Frequency	Displacements, Velocities, Accelerations, Forces, and Stresses at specified times or frequencies	Element Properties (Size) and Grid Coordinates (Shape)
111	Modal Frequency		
112	Modal Transient		
200*	Statics, Normal Modes, Buckling	Weight, Volume, Eigenvalue, Buckling Load Factor, Displacement, Force, Strain, Stress, Lamina Strain, Stress, and Composite Failure Criterion	Element Properties (Size) and Grid Coordinates (Shape)

* Design model may span superelements in Solution 200 for design sensitivity analysis.

Table 8 Design Sensitivity/Optimization Analysis in Version 68+ of MSC.Nastran

Solution		Responses	Design Quantities
No.	Discipline(s)		
200*	Direct Frequency, Modal Frequency, Modal Transient	Weight, Volume, Displacements, Velocities, Accelerations, Forces, and Stresses at Specified Times or Frequencies.	Element Properties (Size) and Grid Coordinates (Shape)
	Statics, Normal Modes, Buckling	Weight, Volume, Eigenvalue, Buckling Load Factor, Displacement, Force, Strain, Stress, Lamina Strain, Stress, and Composite Failure Criterion	
	Static Aeroelasticity	Weight, Volume, Trim Variable, Divergence Dynamic Pressure, Stability Derivative, and all Static Responses	
	Flutter	Weight, Volume, Damping	

* Design model may span superelements. In Version 68, Solution 200 integrates all the different disciplines of design sensitivity / optimization into a single solution sequence. Some of the design sensitivity / optimization entries from previous versions may not be upward compatible. An external program to convert these existing entries to Version 68 format is provided with the Version 68 delivery tape. For further information, refer to the *MSC.Nastran Release Notes for Version 68.*

What Types of Restarts Are Available in Design Optimization?

There are two types of restarts available in design optimization (SOL 200). The first method utilizes the automatic restart capability in a limited fashion; the second method is a “pseudo-restart,” which is a cold start run using the results from a previous design cycle.

The only automatic restart supported in SOL 200 is a restart from one of the six predefined exit points selected with the parameter OPTEXIT. (The Case Control Command DSAPRT overwrites the PARAM,OPTEXIT,4, –4, or 7.) The extent of any reanalysis is dependent upon the point of exit. The mechanics of this type of SOL 200 restart (job submittal, database definition, etc.) is identical to other Structured Solution Sequence restarts.

The “pseudo-restart” in SOL 200 uses a punch file from the previous design cycle for the current set of design variables and solution parameters. If shape optimization is performed, the punch file also contains a set of updated GRID entries for the previous design cycle. The parameter “DESPCH” controls how the design data is written in the punch file. By default, a set of DESVAR and GRID entries are written for the final design cycle. While this restart method does not directly take full advantage of previous design iteration calculations, it does present you with the ability to effectively monitor the various stages of the computation and make the appropriate changes to the design model

Design optimization restarts are described in the *MSC.Nastran Design Sensitivity and Optimization User's Guide.*

What Should I Look for When Checking the Results of Hard Convergence?

If hard convergence is achieved, it will be noted in the Summary of Iteration History table. This test compares the results of two successive finite element analyses and is a conclusive test of convergence at the design cycle level. All constants used in the convergence test can be changed from their defaults using the DOPTPRM Bulk Data entry. (See the *MSC.Nastran Design Sensitivity and Optimization User's Guide.*)

There are essentially two types of converged designs:

1. Convergence to a feasible design
2. Convergence to a best compromise infeasible design

Feasible Design. If all constraints are satisfied, a feasible design has been found, and can be confirmed by a maximum constraint value that is less than GMAX (default = 0.005).

Best Compromise for an Infeasible Design. If one or more constraints are violated at the converged design, the conclusion is that the optimizer has done the best it can, and returned with a design representing the best compromise between least objective function, and violated constraints. This is actually useful design information, since it indicates that one or more constraints need to be either relaxed or restated in order to achieve a feasible design.

What Does the "Maximum Value of Constraints" Column Mean in the Summary of Iteration History?

This refers to the maximum normalized constraint over all constraints in the design model. A positive value indicates constraint violation, a negative value constraint satisfaction, and a near zero value a critical constraint.

Normalized constraints are used in MSC.Nastran to remove the dependence of the constraint value on the magnitude of the underlying responses. To illustrate this, the combination of a DRESP and a DCONSTR entry results in a response with specified bounds r_j^L and r_j^U:

$$r_j^L \le r_j\left(\vec{X}\right) \le f_j^U$$

These bounds are used to generate a pair of normalized constraints as follows:

$$g_{2j-1} = \frac{r_j^L - r_j\left(X\right)}{\left|r_j^L\right|} \le 0$$

$$g_{2j} = \frac{r_j(X) - r_j^U}{\left|r_j^U\right|} \leq 0$$

where the response bounds are used as normalizing factors. The Summary of Iteration History table reports on the maximum value of constraints of the form given in the above equations. However, this table does not provide the information necessary to determine where this value comes from, i.e., the response which is nearest its bound. This question can be easily answered by a closer inspection of the combined optimizer and data recovery output.

The following example is taken from input file D200X1 in the TPL (Test Problem Library). (This example is also illustrated in the *MSC.Nastran Design Sensitivity and Optimization User's Guide.*) The only modifications are to include a Bulk Data parameter description PARAM,NASPRT,1 (which provides for analysis output on every iteration) and to set IPRINT to a value of 1 using the DOPTPRM Bulk Data entry.

From the Design Constraints on Responses table (see **Figure 56**), note that the maximum constraint value for the initial design is –0.3235. Note that the Internal Response IDs are 2 and 5.

```
                                  -----   DESIGN CONSTRAINTS ON RESPONSES   -----
                                   (MAXIMUM RESPONSE CONSTRAINTS MARKED WITH **)
 ------------------------------------------------------------------------------------------------------
                          INTERNAL                              INTERNAL
  INTERNAL      DCONSTR   RESPONSE     RESPONSE      L/U        REGION       SUBCASE
     ID           ID        ID           TYPE        FLAG         ID           ID            VALUE
 ------------------------------------------------------------------------------------------------------
         1          21          2       STRESS     UPPER             1           1      -3.2350E-01**
         2          21          3       STRESS     LOWER             3           1      -3.9351E-01
         3          21          4       STRESS     LOWER             1           2      -3.9351E-01
         4          21          5       STRESS     UPPER             3           2      -3.2350E-01**
```

Figure 56 Summary of Iteration History

If you are interested in identifying the response that contributed to this constraint, look back to the optimizer output (provided for values of IPRINT ≥ 1), and observe that constraint numbers 1 and 4 share this value:

```
-- INITIAL FUNCTION VALUES

OBJ =   4.8284

CONSTRAINT VALUES (G-VECTOR)
     1)  -3.23495E-01  -3.93512E-01  -3.93512E-01  -3.23495E-01  -9.00000E+00
     6)  -5.00000E-01  -1.00000E+00  -1.00000E+00  -1.00000E+00  -3.33333E-01
    11)  -3.33333E-01  -3.33333E-01
```

Figure 57 Optimizer Output

To determine which responses contribute to these constraints, the Initial Analysis Subcase Table can be consulted, which shows that design constraints 1 and 4 are the upper bounds on axial stresses in elements 1 and 3 for subcases 1 and 2, respectively.

The analysis output shows that these are the maximum axial stresses in the elements under tension:

```
LOAD CONDITION 1                                                                SUBCASE 1

                          S T R E S S E S   I N   R O D   E L E M E N T S     ( C R O D )
 ELEMENT       AXIAL       SAFETY      TORSIONAL     SAFETY      ELEMENT       AXIAL       SAFETY      TORSIONAL     SAFETY
   ID.        STRESS       MARGIN        STRESS      MARGIN        ID.        STRESS       MARGIN        STRESS      MARGIN
     1    1.353010E+04               0.0                          2       4.432777E+03                    0.0
     3   -9.097320E+03               0.0
THREE BAR TRUSS DESIGN OPTIMIZATION - SYMMETRIC        D200X1         FEBRUARY   9, 1993  MSC.Nastran   2/ 5/93   PAGE    17
 BASELINE - 2 CROSS SECTIONAL AREAS AS DESIGN VARIABLES
 LOAD CONDITION 2                                                               SUBCASE 2

                          S T R E S S E S   I N   R O D   E L E M E N T S     ( C R O D )
 ELEMENT       AXIAL       SAFETY      TORSIONAL     SAFETY      ELEMENT       AXIAL       SAFETY      TORSIONAL     SAFETY
   ID.        STRESS       MARGIN        STRESS      MARGIN        ID.        STRESS       MARGIN        STRESS      MARGIN
     1   -9.097320E+03        0.0        2         4.432777E+03     0.0
     3    1.353010E+04        0.0
```

Figure 58 Analysis Output Stresses in ROD Elements

Finally, looking back at the Bulk Data input, we can confirm that these responses are, indeed, near their upper bounds of 20,000. We can also confirm that DRESP1 numbers 23 and 25 are the particular responses of interest.

```
DCONSTR 23      ALL     -15000. 20000.
DCONSTR 24      ALL     -15000. 20000.
DCONSTR 25      ALL     -15000. 20000.

DRESP1  23      S1      STRESS  PROD            2               11
DRESP1  24      S2      STRESS  PROD            2               12
DRESP1  25      S3      STRESS  PROD            2               13
```

Figure 59 Bulk Data Input

What Does the “Fractional Error of Approximation” Column Reference in the Objective and Maximum Constraint History?

This number refers to the differences in values between the objective function after approximate optimization, and the corresponding value after a subsequent analysis update. As discussed in the *MSC.Nastran Design Sensitivity and Optimization User’s Guide*, the optimizer works with structural response approximations. A measure of the approximation’s quality could then be computed as:

$$E = \frac{F_{opt} - F_{true}}{F_{true}}$$

where:

F_{opt} = objective function value returned from the optimizer

F_{true} = true objective function computed via a finite element analysis of the current design

However, be aware that this is based on the objective function only and not on any of the constraints. Often, this error is small since a common design objective is weight, which frequently is linear in the design variables (e.g., linearly proportional to plate thicknesses, bar areas, and so on). The constraint approximations may not be as good since they are often based on responses, such as stresses, which generally are nonlinear functions of the design variables. Thus, it could be a mistake to increase move limits (DELP) in order to increase the rate of convergence based solely on this error measure.

How Are Move Limits Implemented?

In MSC.Nastran, move limits are placed on analysis model properties rather than directly on the design variables. The rationale for this is twofold. First, the analysis model integrity is a direct function of the properties used to define it. Since we do not want the estimates of structural responses to be too far off from the analysis model responses, we limit the maximum allowable variation in analysis model properties for the current design cycle. Second, it is difficult to know beforehand how much a given property will change for a given change in a design variable. This is because properties can be expressed as a linear combination of design variables on DVPREL1 entries

$$p_i\left(\vec{X}\right) = C_o + \sum_i C_j X_j$$

or as a nonlinear function of design variables on DVPREL2 entries

$$p_i\left(\vec{X}\right) = p_i\left(\vec{X}, \vec{C}\right)$$

The optimizer only has direct control over the design variables. The corresponding properties, in turn, are only known once a vector of design variables is at hand. Thus, move limits on properties are expressed in terms of equivalent constraints as follows:

$$p_i^L \le p_i\left(\vec{X}\right) \le p_i^U$$

from which a pair of constraints in standard form is written and used internally:

$$g_1 = \frac{p_i^L - p_i\left(X\right)}{\left|p_i^L\right|} \le 0$$

$$g_2 = \frac{p_i\left(X\right) - p_i^U}{\left|p_i^U\right|} \le 0$$

Since critical constraints may not be satisfied exactly at the optimum, the corresponding move limits might not be either. For example, if the lower bound is given by PMIN (see the DVPREL1 and DVPREL2 entries), the final design could actually contain properties which are slightly less than PMIN. (By default, one-half of one percent maximum constraint violation is tolerated at the optimum—see the GMAX field on the DOPTPRM Bulk Data entry.)

This effect may be even more noticeable if the initial design is infeasible. Under this condition, the optimizer's primary task is to minimize the total constraint violation. To do so may require violating some constraints in favor of reducing others. Thus, move limits, which are seen by the optimizer only as constraints, may be relaxed in favor of reductions in more highly violated constraints.

What Is the Best Way to Express Equality Constraints in Design Optimization?

Equality constraints are constraints of the form

$$h_k\left(\vec{X}\right) = 0 \qquad k = 1, \ldots, n_k$$

where $\vec{X}$ is the vector of design variables and n_k is the number of equality constraints in the design problem. $h\left(\vec{X}\right)$ is usually an implicit function of the design variables.

Equality constraints frequently occur in design tasks where correlation of test and analysis data is sought, frequency matching is desired, and so on. For example, suppose we want to match the *i*-th mode of a structure to some target value Λ_i or

$$\lambda_i - \Lambda_i = 0$$

Using a DRESP1 entry to identify the eigenvalue of interest and a DCONSTR entry to specify the bounds, the equality could be enforced using equal lower and upper bounds as follows:

$$\Lambda_i \le \lambda_i \le \Lambda_i$$

Often, however, we are only interested in satisfying the equality within a certain tolerance ε or

$$(\Lambda_i - \varepsilon) \le \lambda_i \le (\Lambda_i + \varepsilon)$$

Not only does this more accurately reflect our design goals but the addition of ε makes the numerical task of satisfying the constraints easier. Again, a DRESP1 entry is required to identify the *i*-th eigenvalue with a corresponding DCONSTR entry to establish the lower and upper bounds.

An alternate, and recommended, method is to normalize the responses first; then add a tolerance that represents a percentage satisfaction of equality:

$$(1 - \varepsilon) \leq \frac{\lambda_i}{\Lambda_i} \leq (1 + \varepsilon)$$

This approach would require a DRESP2 entry to express the eigenvalue ratio and a DCONSTR entry to establish the lower and upper bounds.

Are There Numerical Scaling Considerations I Should Be Aware of?

There are a number of default parameters that can affect the overall design optimization process. Although these default values are chosen to enable the optimizer to yield satisfactory results for a wide range of expected problem types, they should always be checked. The scaling of the overall design task should always be considered since it may influence the choice of these parameter values.

For example, convergence with respect to absolute change in the objective function is indicated at the optimizer level by

$$\left|F_p - F_{p-1}\right| \leq DABOBJ = \max\left\{0.001 \cdot \left|F\left(\vec{X^o}\right)\right|, 0.0001\right\}$$

and at the design cycle level by

$$\left|F_p - F_{p-1}\right| \leq CONV2 = 1.0\text{E}{-}20$$

A small CONV2 value is chosen to avoid premature convergence when using certain SI units.

Similarly, convergence is indicated in a relative sense at the optimizer level by a change in the objective function of

$$\frac{\left|F_p - F_{p-1}\right|}{\left|F_{p-1}\right|} \leq DELOBJ = 1.0\text{E}{-}20$$

and at the design cycle level by

$$\frac{\left|F_p - F_{p-1}\right|}{\left|F_{p-1}\right|} \leq CONV1 = 0.001$$

That is, relative convergence is achieved for changes in the objective function of 1/10th of one percent or less. This would translate into a one kilogram change for a design having an initial mass of 1000 kg. Whether or not this is a reasonable test depends on the necessary accuracy of the converged solution. In short, the inspection of the absolute and relative convergence test parameters will suggest a usable scale of objective function values.

Constraint scaling may also have an effect on the quality of the optimal solution. The DCONSTR Bulk Data entry is used to place lower and upper bounds on responses

$$r_j^L \le r_j\left(\vec{X}\right) \le r_j^U$$

which are, in turn, converted into a normalized form used internally

$$g_{2j-1} = \frac{r_j^L - r_j\left(X\right)}{\left|r_j^L\right|} \le 0$$

$$g_{2j} = \frac{r_j\left(X\right) - r_j^U}{\left|r_j^U\right|} \le 0$$

If either of the bounds is specified as 0.0, an epsilon of 1.0E-4 is used to avoid a divide by zero error. This is usually adequate, unless some of the $\left|r_j\left(\vec{X}\right)\right|$ are less than this value, in which case the optimizer tries to enforce $\left|r_j\left(\vec{X}\right)\right| \ge \varepsilon$. This, of course, can lead to unrealistic results.

Beyond these simple scaling issues, the accuracy of constraint sensitivities also depends on the magnitude of the underlying responses. If second-level responses (DRESP2) appear in the model, their sensitivities are computed using chain rule differentiation

$$\frac{\partial r^{(2)}}{\partial x} = \frac{\partial r^{(2)}}{\partial r^{(1)}} \cdot \frac{\partial r^{(1)}}{\partial x}$$

where superscripts (1) and (2) indicate first- and second-level responses, respectively.

The first term on the right-hand side ($\partial r^{(2)}/\partial r^{(1)}$), is approximated using finite differences, with a minimum absolute variation in the denominator of 1.0E-6. The minimum Δx used in the second term is also 1.0E-6. Thus, inaccurate sensitivities result if any first-level responses or design variables (and properties) are on the order of 1.0E-6 or less.

Finally, other frequent causes of scaling difficulties result from the absolute values of the analysis model properties affected by the design model. One difficulty relates to PMIN, or the minimum allowable property value listed on the DVPREL1 and DVPREL2 entries. Its default of 1.0E-3 should be changed for properties, which are expected to take on values less than this amount.

Another common source of difficulty is with regard to DPMIN, or the minimum move limit imposed. DPMIN is an absolute quantity (not a relative one) and has a default value of 0.01. This absolute quantity can be modified using the DOPTPRM entry. Thus, for properties that are numerically very small, the move limits are at least ± 0.01. By default, properties on the order of 1.0E-2 can experience up to 100% moves per design cycle, which may be sufficient to invalidate the approximate model used by the optimizer. If small property values do appear in the model, it is recommended that the entire problem be rescaled (by changing units, for example, from meters to centimeters), or DPMIN be changed to allow more reasonable 20% to 50% move limits.

Why are Some of My Optimization Runs More Efficient in Version 70 than Prior Versions?

The introduction of the Adjoint Sensitivity Method in Version 70 provides substantial savings in resources—both cpu and disk space usage—for certain types of optimization problems. The Adjoint Sensitivity method works well if the number of retained responses (NRESP) is small and it is grid-type-only responses (e.g., disp, velocity, etc.). The criteria for selecting the Adjoint Sensitivity Method as opposed to the "direct" method is based on the following relationship:

$$NRESP < NDV \times NLF$$

where:

NRESP = number of retained responses

NDV = number of design variables

NLF = number of loadings in static analysis or number of frequencies (# of frequencies per subcase times the # of subcases).

The selection of the Adjoint Method is transparent to the users. MSC.Nastran will automatically select either the "direct" or Adjoint Method based on the model information. In other words, you can take an existing Version 69 optimization input file and run it in Version 70 without changing your input file.

Why is Mode Tracking Important?

In design optimization, a particular mode number (frequency or eigenvalue) is selected as a design response by identifying the mode number on a DRESP1 Bulk Data entry. The eigenvalue solver arranges the modes in order from the lowest to the highest frequency during each design cycle. The mode number is an artificial designation that does not have any physical interpretation such as "first vertical bending mode" or "first lateral bending mode".

This ordering convention may create difficulties in design optimization. As the design is modified, it is entirely possible that two or more modes will switch with each other. As an example, assume that the first two modes of the current design are 18 hz (first vertical bending mode) and 18.3 hz (first lateral bending mode), respectively, and the loading is primarily in the vertical direction. Since the loading is primarily in the vertical direction, you want to select mode number one as the design response to monitor. During the design process, the first vertical bending mode and first lateral bending mode change to 18.7 hz and 18.5 hz, respectively. Since the eigenvalue solver arranges the modes in order from the lowest to the highest frequency, the lateral bending mode now becomes the first mode instead of the second mode. In this case, if this mode switching is not accounted for, you would be monitoring the lateral bending mode for a vertical applied load, which in turn could lead to incorrect optimization conclusions.

One way of keeping track of the mode is to plot the mode shapes at each design cycle and keep track of them manually. The mode tracking capability introduced in Version 69 attempts to do as its name implies; investigate the mode shapes for the current design

cycle, compare with those of the previous cycle, and identify if any mode switching has occurred. Mode tracking can be activated with the MODTRAK Case Control command. See the *MSC.Nastran Version 69 Release Guide* for further details. Similar features are available in Version 68.2 with the modtraka.v682 sssalter.

Superelements

Can Superelements Be Used in Acoustic Analysis, and If So, How?

Superelements can be used in acoustic analysis, but there are some restrictions. Each superelement may contain either fluid points and elements only, or structural points and elements only. The coupling between the fluid and structure occurs in the residual structure. This coupling is based on the actual contact between the different elements. Therefore, for the coupling to occur, all fluid elements that are in contact with structural elements must be in the residual structure, and all structural elements that are in contact with fluid elements must also belong to the residual structure. Starting in Version 70, the fluid-structure interface points can be placed in a superelement using the rflagb.v70 sssalter.

The residual structure may contain both structural and acoustic points, and elements. At a minimum the residual structure must contain all interface elements and points. If a modal reduction is to be used on fluid superelements, then a METHOD(FLUID) Case Control command must occur in the superelement subcase and an SEQSET Bulk Data entry must be defined for the superelement.

The enhanced superelement capabilities (parts) do not support acoustic analysis. The pre-V69 superelement method (SESET) supports acoustic analysis.

How Do I Apply Loads on Superelements in Solution 106? Thermal Loads?

In an effort to reduce the size of the matrices involved in the nonlinear iteration process, the superelement modeling approach can be used to isolate the nonlinear behavior to the residual structure. The subcase structure can be used in nonlinear statics to control the load application, output requests, etc., as is done in linear superelement analysis with one exception. Static loads on upstream superelements cannot be directly referenced with the LOAD Case Control command. Combinations of interior loads on upstream superelements must be defined with the CLOAD/LOADSET/LSEQ commands. This causes MSC.Nastran to generate load combinations acting on upstream superelements prior to the nonlinear solution.

To apply upstream loads in a nonlinear analysis, the LOADSET, CLOAD, and LOAD Case Control commands are used in combination with the LSEQ, CLOAD, and typical static load entries. The following example should clarify the interrelation of these entries.

```
SOL 106
.
.
CEND
$
SEALL = ALL  $  optional
.
.
SUBCASE 1
   LABEL = SUPERELEMENT 10
   SUPER = 10
   LOADSET=1000
SUBCASE 11
   LABEL = LOAD CASE 1
   NLPARM=10
   CLOAD=1001
   LOAD=401
SUBCASE 12
   LABEL = LOAD CASE 2
   NLPARM=20
   CLOAD=1002
   LOAD=402
BEGIN BULK
$
$LSEQ   LOADSET  DAREA   LOAD      TEMP
$         ID     ID      ID        ID
$
LSEQ     1000    101      1
LSEQ     1000    102      2
$
FORCE    1       .......
GRAV     2       .......
$
$CLOAD  CLOAD    OVERALL  SCALE     FIRST  SCALE     SECOND
$        ID      SCALE    FACTOR    DAREA  FACTOR    DAREA
$                FACTOR   FOR 1ST   ID     FOR 2ND   ID
$                         DAREA ID         DAREA ID
$
CLOAD   1001     1.0      2.0       101      1.0      102
CLOAD   1002     1.0      2.0       101      2.0      102
$
$  RESIDUAL STRUCTURE LOADS
$
FORCE   401      .......
FORCE   402      .......
$
$   DEFINE SUPERELEMENT
$
SESET    10     ......
$
$   REST OF THE BULK DATA SECTION
$
ENDDATA
```

Figure 60 Input Used for Applying Upstream Loads in a Nonlinear Analysis

In the above Case Control Section, LOADSET = 1000 is activating all LSEQ entries in the Bulk Data Section with a set ID of 1000. In addition, the CLOAD entry in the first subcase (subcase 11) is referring to CLOAD set ID 1001 in the Bulk Data. Together they are linked by common reference to DAREA set IDs 101 and 102. In addition, the LSEQ entries refer to the FORCE and GRAV entries with set IDs 1 and 2, respectively. The net effect of these entries is to apply 1.0 (overall scale factor from CLOAD) times the combination of 2.0 times the FORCE entry (from CLOAD scale factor) plus 1.0 times the GRAV entry (from CLOAD scale factor). These loads are applied to all upstream superelements that have interior loadings included by static load sets 1 and 2. These upstream superelement loads are combined with LOAD set ID 401, which applies the FORCE (401) to the residual structure.

In the second subcase (subcase 12) the GRAV loading scale factor is increased to 2.0, the FORCE scale factor remains at 2.0, and these upstream superelement loads are combined with LOAD set ID 402 that applies the FORCE (402) to the residual structure. Thermal loads on upstream superelements are applied in a similar manner except that they refer to the TEMP ID field instead of the LOAD ID field on the LSEQ entry.

How Do I Perform Random Analysis with a Superelement Model?

The approach used for random analysis with a superelement model is similar to that for a residual only (non-superelement) model. In the residual only model, a single RANDOM specification is required that must be above the subcase level or in the first subcase. When multiple subcases are present in the residual only analysis, each is considered to be a unique load case that can be uncorrelated or correlated depending on the RANDPS entries. It is important to note that only one random analysis is performed per run regardless of the number of subcases.

For a superelement model either single-level or multilevel, the analysis approach is the same. Only one random analysis is performed per run. The main difference is that a unique RANDOM specification must be made in the first subcase for each superelement for which you request random output. The RANDPS entries for each of the superelements requesting random output should specify the same TABRND1 entries. Do not specify a RANDOM entry for any superelement for which an XYPLOT (or XYPRINT or XYPEAK) request was not made; otherwise, a fatal error will occur. Remember that, even though multiple RANDOM specifications are requested in the Case Control, only one random analysis is performed per run.

As an example, consider the single-level model shown in **Figure 61**.

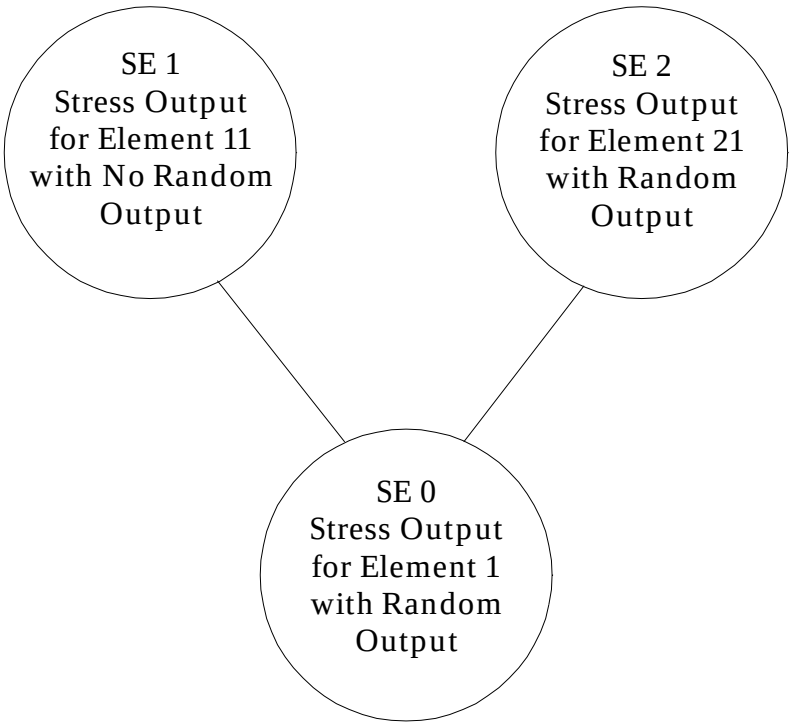


Figure 61 Superelement Model with Random Analysis

An input file showing the correct setup is shown in **Figure 62**. Only the necessary Case Control commands and Bulk Data entries are shown to make it more readable. The goal for this problem was to obtain random output for element 1, which is located in the residual, and element number 21, which is located in superelement 2. Therefore, a RANDOM specification is needed in the first subcase for superelement 2 and in the first residual subcase. If you wish to obtain stress data recovery for the other superelements in the model, additional subcases may be included as long as there is no RANDOM specification for these subcases.

```
ID,PSD,SOL111
SOL 111
TIME 100
CEND
TITLE=  SE MODEL WITH CORRELATED RANDOM INPUT
METHOD=100
SET 111 = 1 $ ELEMENT 1 IS IN THE RESIDUAL
SET 222= 11 $ ELEMENT 11 IS IN SE 1
SET 333= 21 $ ELEMENT 21 IS IN SE 2
FREQ = 100
SDAMP = 100
$
SUBCASE 1
  LABEL = SE 1 LOAD CASE 1
  SUPER = 1,1
  STRESS(SORT2,PLOT,REAL)= 222
SUBCASE 2
  LABEL = SE 1 LOAD CASE 2
  SUPER = 1,2
  STRESS(SORT2,PLOT,REAL)= 222
$
SUBCASE 11
  LABEL = SE 2 LOAD CASE 1
  SUPER = 2,1
  STRESS(SORT2,PLOT,REAL)= 333
  RANDOM= 101
SUBCASE 12
  LABEL = SE 2 LOAD CASE 2
  SUPER = 2,2
  STRESS(SORT2,PLOT,REAL)= 333
$
SUBCASE 21
  LABEL = RESIDUAL LOAD 1
  SUPER = 0,1
  RANDOM= 201
  DLOAD = 9
$
SUBCASE 22
  LABEL = RESIDUAL LOAD 2
  SUPER = 0,2
  DLOAD = 19
$
OUTPUT(XYOUT)
$
SEPLOT 2
XYPRINT STRESS PSDF /21(10)
SEPLOT 0
XYPRINT STRESS PSDF /1(10)
$
$ RANDOM INPUT FOR SE 2
$
BEGIN BULK
randps       101  11  11     1.0     0.0     911
randps       101  12  12     1.0     0.0     922
randps       101  11  12     1.0     0.0     912
randps       101  11  12     0.0    -1.0     913
$
$ RANDOM INPUT FOR RESIDUAL
$
randps       201  21  21     1.0     0.0     911
randps       201  22  22     1.0     0.0     922
randps       201  21  22     1.0     0.0     912
randps       201  21  22     0.0    -1.0     913
ENDDATA
```

Figure 62 Typical Superelement Input File for Random Analysis

What Are Some of the Major Differences Between the Enhanced Superelement and Basic Superelement Capabilities?

The basic superelement is defined as the superelement capabilities that exist prior to Version 69 and these capabilities still exist in Version 69 and subsequent versions. In other words, if a superelement input file exists, and no changes have been made using any of the enhanced superelement features, then the job will run exactly the way it did as in previous versions of MSC.Nastran. The only exception is the default for the super command is SUPER=ALL starting in Version 69.

The enhanced superelement is defined as the new superelement capabilities introduced in Version 69. This superelement capability is also referred to as PART.

Some of the major differences between the Enhanced Superelement and Basic Superelement Capabilities are listed below.

1. The basic superelement is defined by assigning the grid points to specific superelements manually using the SESET entries, and the elements are automatically assigned to the superelements based on their connectivities. The enhanced superelement is defined by putting all the model descriptions of each superelement (part) in a separate Bulk Data Section. By default, if the location of a grid point in one part falls within a specified tolerance of a grid point located in another part, then these parts will be connected together at that location automatically.
2. The basic superelement is contained in a single Bulk Data Section. The enhanced superelement uses multiple Bulk Data Sections, each superelement Bulk Data Section begins with a BEGIN [BULK] SUPER = SEIDx delimiter and terminates with another BEGIN [BULK] SUPER = SEIDy or ENDDATA for the last superelement.
3. All IDs of the same type (e.g., grids) must be unique across different superelements using the basic superelement capabilities. The same IDs can be used for different superelements using the PART method.
4. The PART method does not support p-element, optimization, and acoustics. The basic superelement method supports these features.

There are other differences between the two methods. See the *MSC.Nastran Release Guides* for Versions 69, 69.1, and 70 for further details.

Can I Mix the Basic Superelements with the Enhanced Superelements?

You can mix the basic superelement with the enhanced superelement capabilities, although this practice is not normally recommended. This feature can, however, be quite convenient for certain applications. For instance, if you have an existing superelement model that was created in a previous version using the the basic superelement capabilities and you want to attach a new component to your existing model. This problem can be solved rather easily by putting the original superelement in the main Bulk Data Section,

and attach the new superelement (part) by placing the complete model description for this part in a separate Bulk Data Section that begins with a BEGIN [BULK] SUPER = SEIDx. MSC.Nastran will automatically attach the new part to the existing superelement model based on the model tolerance.

Heat Transfer

What Are the New Heat Transfer Capabilities in Version 68+?

Substantial enhancements have been added to the MSC.Nastran heat transfer capabilities in Version 68. **Table 9** summarizes the major differences in the heat transfer capabilities between Version 68+ and prior versions.

Table 9 Heat Transfer Capabilities of MSC.Nastran

Capabilities	Available in Version 67.5 and Prior Versions?	Available in Version 68+?
Enforced Temperature Boundary	Yes.	Yes. New user interface.
Free Convection	Yes. Convection coefficients in a table format.	Yes. In addition to the table format, convection coefficients can be supplied in an equation form.
Forced Convection	Yes. Convection coefficients in a table format.	Yes. Table format is not supported. Can supply flow parameters.
Radiation Exchange (View Module)	Yes. A more efficient View Module design is implemented in Version 67.	Yes. Multiple cavities are supported. This reduces the amount of required shading calculation.
Radiation to Space	Yes. User must supply radiation matrix.	Yes. Automatic generation of radiation matrix.
Phase Change	Not available.	Yes. New feature.
Spectral radiation, where emissivity is a function of the wavelength.	Not available.	Yes. New feature.
Emissivity as a function of temperature.	Not available.	Yes. New feature.
Volumetric Heating	Not available.	Yes. New feature.
Control node as a function of temperature (thermostat feedback).	Not available.	Yes. New feature.

Prior to Version 68, depending on the features desired, there were eight different solution sequences available for the MSC.Nastran heat transfer analysis. Beginning with Version 68, these eight solution sequences are combined into two solution sequences. **Table 10** summarizes the different features along with their corresponding solution sequences and version numbers.

Table 10 Heat Transfer Solution Sequences of MSC.Nastran

Solution Type	Solution Sequences for Version 67.5 and Prior Versions	Solution Sequence for Version 68+
Linear Steady State	SOLs 101, 24, 61	153
Nonlinear Steady State	SOLs 153, 74	153
Linear Transient	SOLs 159, 89	159
Nonlinear Transient	SOL 17	159

In order to add all these new heat transfer features, the user interface has been changed. A separate utility program, heatconv, is provided in order to convert pre-Version 68 heat transfer input files into Version 68 format. For further information regarding Version 68 heat transfer features, refer to the *MSC.Nastran Release Notes for Version 68.*

MSC.Patran Version 6+ supports pre- and postprocessing of MSC.Nastran heat transfer capabilities.

Pre- and Postprocessing

How Can I Create a PostScript Plot?

A utility called PLOTPS allows you to produce PostScript plots from either a binary or neutral file. This interface is generic across all platforms except for the IBM(MVS/XA). In addition, the "wallpaper" feature is supported. See the *MSC.Nastran Configuration and Operations Guide* for more details regarding PostScript files.

How Are Data Blocks Obtained for Pre- and Postprocessing?

PARAM,POST is used to obtain data blocks from an MSC.Nastran analysis for use in pre- and postprocessing. Currently, this parameter supports several pre- and postprocessors including MSC.Patran, MSC.Nastran for Windows, MSC/ARIES, MSC/XL, SDRC I-DEAS, LMS International/MSC_NF, Dynamic Design Solutions/Fem Tools, and EDS/Unigraphics.

The following values support the indicated programs.

0	MSC.Nastran for Windows, MSC/ARIES and MSC/XL. This option converts the data blocks using the database converter module (DBC) into the suitable database format.
–1	MSC.Patran. PARAM,PATVER selects the appropriate version of PATRAN either 2.0 or 3.0. The default value for PATVER is 3.0. For Version 2.5, only the results data blocks are required as input to the NASPAT program. For Version 3.0, both the geometry and results data blocks are required. The data blocks are obtained using the OUTPUT2 module and written to a user-specified FORTRAN file specified by PARAM,OUNIT2. The default for OUNIT2 is 12.
–2	SDRC I-DEAS. The geometry and results data blocks are obtained for the I-DEAS FEM Data Loader using the OUTPUT2 module. All of the data blocks are written to one user-specified FORTRAN file specified by PARAM,OUNIT2. The default for OUNIT2 is 12.
–4	Results Data blocks for LMS International/MSC_NF.
–5	Results Data blocks for Dynamic Design Solutions/Fem Tools.
–6	Results Data blocks for EDS/Unigraphics.

PARAM,POST is available in all Structured Solution Sequences. Refer to the *MSC.Nastran Quick Reference Guide* for other applicable parameters and default values as well as a detailed description of each including which data blocks are obtained depending on the values of the parameters.

I N D E X

MSC.Nastran Common Questions and Answers

A

B

C

D

I N D E X

MSC.Nastran Common Questions and Answers

E

F

G

I N D E X

MSC.Nastran Common Questions and Answers

I N D E X

MSC.Nastran Common Questions and Answers

I N D E X

MSC.Nastran Common Questions and Answers

S

T

I N D E X

U

V

W

X

Z