MSC Nastran 2020

Service Pack 1

Reference Guide

HEXAGON | MSC Software

Corporate
MSC Software Corporation
4675 MacArthur Court, Suite 900
Newport Beach, CA 92660
Telephone: (714) 540-8900
Toll Free Number: 1 855 672 7638
Email: americas.contact@mscsoftware.com

Europe, Middle East, Africa
MSC Software GmbH
Am Moosfeld 13
81829 Munich, Germany
Telephone: (49) 89 431 98 70
Email: europe@mscsoftware.com

Japan
MSC Software Japan Ltd.
Shinjuku First West 8F
23-7 Nishi Shinjuku
1-Chome, Shinjuku-Ku
Tokyo 160-0023, JAPAN
Telephone: (81) (3)-6911-1200
Email: MSCJ.Market@mscsoftware.com

Asia-Pacific
MSC Software (S) Pte. Ltd.
100 Beach Road
#16-05 Shaw Tower
Singapore 189702
Telephone: 65-6272-0082
Email: APAC.Contact@mscsoftware.com

Worldwide Web
www.mscsoftware.com

Support
http://www.mscsoftware.com/Contents/Services/Technical-Support/Contact-Technical-Support.aspx

Disclaimer
MSC Software Corporation reserves the right to make changes in specifications and other information contained in this document without prior notice.

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

User Documentation: Copyright © 2020 MSC Software Corporation. Printed in U.S.A. All Rights Reserved.

This notice shall be marked on any reproduction of this documentation, in whole or in part. Any reproduction or distribution of this document, in whole or in part, without the prior written consent of MSC Software Corporation is prohibited.

This software may contain certain third-party software that is protected by copyright and licensed from MSC Software suppliers. Additional terms and conditions and/or notices may apply for certain third party software. Such additional third party software terms and conditions and/or notices may be set forth in documentation and/or at http://www.mscsoftware.com/thirdpartysoftware (or successor website designated by MSC from time to time).

PCGLSS 8.0, Copyright © 1992-2016, Computational Applications and System Integration Inc. All rights reserved. PCGLSS 8.0 is licensed from Computational Applications and System Integration Inc.

MSC, Dytran, Marc, MSC Nastran, Patran, the MSC Software corporate logo, e-Xstream, Digimat, and Simulating Reality are trademarks or registered trademarks of the MSC Software Corporation and/or its subsidiaries in the United States and/or other countries. Hexagon and the Hexagon logo are trademarks or registered trademarks of Hexagon AB and/or its subsidiaries.

NASTRAN is a registered trademark of NASA. FLEXlm and FlexNet Publisher are trademarks or registered trademarks of Flexera Software. All other trademarks are the property of their respective owners.

Use, duplicate, or disclosure by the U.S. Government is subjected to restrictions as set forth in FAR 12.212 (Commercial Computer Software) and DFARS 227.7202 (Commercial Computer Software and Commercial Computer Software Documentation), as applicable.

U.S. Patent 9,361,413

July 24, 2020

NA:V2020SP1:Z:Z:Z:DC-REF-PDF

Documentation Feedback

At MSC Software, we strive to produce the highest quality documentation and welcome your feedback. If you have comments or suggestions about our documentation, write to us at: documentation-feedback@mscsoftware.com.

Please include the following information with your feedback:

- Document name
- Release/Version number
- Chapter/Section name
- Topic title (for Online Help)
- Brief description of the content (for example, incomplete/incorrect information, grammatical errors, information that requires clarification or more details and so on).
- Your suggestions for correcting/improving documentation

You may also provide your feedback about MSC Software documentation by taking a short 5-minute survey at: http://msc-documentation.questionpro.com.

Note:	The above mentioned e-mail address is only for providing documentation specific feedback. If you have any technical problems, issues, or queries, please contact Technical Support.

Contents

MSC Nastran Reference Guide

Preface

How to Use this Guide **2**
Prerequisites 2
Goals of this Guide 2
List of MSC Nastran Books **3**
Using other Manuals **4**
Typographical Conventions **5**
Accessing MSC Nastran Manuals **5**
. 5
Downloading the PDF Documentation Files 6
Navigating the PDF Files 6
Printing the PDF Files 6
Training and Internet Resources **7**
Technical Support **8**
Visit SimCompanion 9
Help Us Help You 9

1 nast20200 Command and Input Data

Introduction **11**
nast20200 Command **14**
The NASTRAN Statement (Optional) **14**
File Management Section (FMS) **15**
Executive Control Section **15**
Case Control Section **15**
Case Control Section Requirements for Solutions 15
Bulk Data Section **32**
Use of Parameters **32**
Parameters in the Superelement Solution Sequences 33

Main Index

2 Grid Points and Coordinate Systems

Introduction **36**
Grid Point and Coordinate System Definition **36**
Grid Point Properties **40**

3 Structural Elements

Element and Property Definition **42**
Elements with Offsets 42
Axisymmetric Elements 46
Material Properties 46
Line Elements **55**
Beam Element (CBEAM) 55
Three-Node Beam Element (CBEAM3) 63
Bar Element (CBAR) 65
Bend Element (CBEND) 67
Rod Elements (CROD, CONROD, CTUBE) 71
Supplied Beam and Bar Libraries 72
Adding Your Own Beam Cross-Section Library 84
Support for Offsets in Element Coordinates for CBAR & CBEAM Elements 117
Surface Elements **121**
Shear Panel Element (CSHEAR) 121
Shell Elements (CTRIA3, CTRIA6, CTRIAR, CQUAD4, CQUAD8, CQUADR) 123
QUADR and TRIAR Elements 133
Conical Shell Element (RINGAX) 145
Shell Elements Corner Thickness Specification 148
SHELL Structural Damping 151
Solid Elements (CTETRA, CPENTA, CHEXA) **151**
Rigid Elements and Multipoint Constraints (R-type, MPC) **155**
Rigid Element Enhancements 159
Scalar Elements (CELASi, CMASSi, CDAMPi) **174**
Axisymmetric Solid Elements (CTRIAX6, CTRIAX, CQUADX) **175**
Crack Tip Elements (CRAC2D, CRAC3D) **177**
Gap and Line Contact Elements (CGAP, BCONP, BLSEG) **184**
Concentrated Mass Elements (CONM1, CONM2) **190**
General Element Capability (GENEL) **190**
Direct Matrix Input on Grid Points (DMIG) **193**
p-Elements **193**
Hyperelastic Elements **197**

Bushing Elements **201**
Bush Element (CBUSH) 201
Bush1D Element (CBUSH1D) 205
Interface Elements **206**
Curve Interface Elements 206
Surface Interface Elements 208
Connector Elements **211**
Introduction 211
CWELD Element 211
Additional Information 212
Formulation Changes 212
Connector Stiffness 222
Connector Control: 224
CFAST Element 238
CSEAM Element 238
Limitations and Guidelines 239
CINTC Element 241
Element Geometry Checks **245**
ELSUM Case Control Command **256**
Introduction 256
Method and Theory 256
Inputs 256
Outputs 257
Guidelines and Limitations 258
Example 259

4 Material Properties

Isotropic Material **262**
Orthotropic Material **264**
Anisotropic Material **265**
Nonuniform Material and Composite Properties for Shells **267**
Classical Lamination Theory 269
Transverse Shear Theory 275
Data Recovery 279
Failure Indices 280
Strength Ratios 284
“LAM” Options 289
Temperature-Dependent Properties 292
Hyperelastic Material **296**
Reference 296

5 Applied Loads

Static Loads 298
Point Loads 298
Distributed Loads 298
Inertia Loads 299
Thermal and Enforced Element Deformation Loads 299
Loads Due to Enforced Motion 303
Frequency-Dependent Loads 304
Time-Dependent Loads 305
Simplified Static Loading Data in Dynamic Analysis 306
Combining of Loads 308
p-Element Loads and Constraints 309
Summary of Nonlinear Static Loads 312
Summary of Nonlinear Transient Loads 317

6 Constraint and Set Notation

Degree of Freedom Sets 326
Degree-of-Freedom Set Definitions 326
Degree-of-Freedom Set Bulk Data Entries 328
Multipoint Constraints 329
Single Point Constraints 331
Rigid Body Supports 332
Sets for Dynamic Reduction 333
Sets for Aerodynamics 333
Rules of Sets for Undefined Degrees of Freedom 334
Output Selection Via Set Specification 335

7 Solution Sequences

Introduction 337
Output Description 340
MSC Nastran Output 341
Restart Procedures 361
General Operations 362
Geometry Processing in SubDMAP PHASE0 363

Matrix Assembly Operations in SubDMAP SEMG . . . 365
Multipoint Constraint Operations in SubDMAPs SEKR and SEMR2 and SEMRB . . . 366
Single Point Constraint Operations in SubDMAPs SEKR, SEMR2 and SEMRB . . . 369
Constraint and Mechanism Problem Identification in SubDMAP SEKR . . . 370
Static Condensation in SubDMAPs SEKR and SEMR2 . . . 373
Static Solutions in SubDMAP SEKRRS . . . 376
Static and Dynamic Load Generation . . . 379
Dynamic Reduction and Component Mode Synthesis in SubDMAP SEMR3 . . . 382
Formulation of Dynamic Equations in SubDMAP GMA . . . 385
Data Recovery Operations in SubDMAP SEDISP . . . 388
Data Recovery Operations in SubDMAP SEDRCVR . . . 393
Summary of Solution Sequence Operations . . . 394
Static Solutions . . . 394
Real Eigenvalue Analysis in SubDMAPs SEMR3 and MODERS . . . 403
Complex Eigenvalue Analysis in SubDMAP CEIGRS . . . 416
Frequency Response and Random Analysis in SubDMAP FREQRS . . . 417
Transient Response in SubDMAPs SEDTRAN and SEMTRAN . . . 417
Buckling Analysis in SubDMAP MODERS . . . 418
Superelement Analysis . . . 419
Nonlinear Heat Transfer Analysis . . . 432

8 Element Data Recovery Resolved at Grid Points

Introduction . . . 436
Stress Recovery at Grid Points . . . 436
Description of Method . . . 437
Topological Method: Plate Elements and Solid Elements . . . 438
Geometric Method: Surface Elements . . . 441
Treatment of Exception Points: BRANCHing . . . 443
User Information . . . 444
Grid Point Stress Output Description . . . 445
General Remarks . . . 447
Mesh Stress Discontinuities at Grid Points . . . 447

9 DMAP

Introduction . . . 453

10 Database Concepts

Introduction . . . 455
Delivery Databases . . . 457

Dbset Deletion **457**
Database Autoassignment **457**
Database Project and Version **458**
Database Archival, Compression, and Transfer **458**
Database Compression **458**
Database Transfer Across Different Computer Types **459**

11 Plotting

General Capability **461**
Superelement Plotting **461**
Structure Plotter Coordinate System and Orthographic Projection 465
Perspective Projection 467
. 468
Stereoscopic Projection 469
Projection Plane to Plotter Transformations 469
Deformation Scaling 470
Postprocessors **470**

12 Interface With Other Programs

Introduction **472**
POST Case Control Command **472**
PUNCH Output **478**
Patran Support of Heat Transfer Analysis **509**
Loads and Boundary Conditions 510
Results Processing 510
Limitations 510
Documentation 511
The Nastran HDF5 Result Database (NH5RDB) **511**
Introduction 511
Data Schema 512
Nastran Input and Output in NH5RDB 514
Nastran Solution Support 515
NH5RDB Optional Output 516
NH5RDB DMAP modules 517
NH5RDB Data Access **522**
Schema document 572

A Numerical Accuracy Considerations

Introduction **575**
Linear Equation Solution **575**
Eigenvalue Analysis **575**
Matrix Conditioning **576**
Definiteness of Matrices 577
Numerical Accuracy Issues **577**
Sources of Mechanisms **578**
Sources of Nonpositive Definite Matrices **579**
Detection and Avoidance of Numerical Problems **579**
Static Analysis 579
Eigensolutions Using the Inverse Iteration and Lanczos Methods 581
Eigensolutions Using Transformation Methods 581
Frequency Response Analysis 582
Transient Response Analysis 582
The Large Mass Method 582
The Lagrange Multiplier Method 583

B Theory of Eigenvalue Extraction

Sturm Sequence Theory **585**
Eigensolution Methods **585**

C Nonlinear Analysis

General Description of Nonlinear Capabilities **599**
Rudiments of User Interface 602
Material Nonlinearity 613
Nonlinear Elastic Material 614
Hyperelastic Material 615
Elastic-Plastic Material 622
Viscoelastic Material in Nonlinear Analysis 626
Temperature-Dependent Material 627
Geometric Nonlinearity 627
Nonlinear Elements 627
Fully Nonlinear Hyperelastic Elements 644
Bushing Elements 656
Adaptive Gap Element 660
3-D Slideline Contact Capability 664
Nonlinear Static Analysis 673
Examples of Convergence Tolerance Changes 679

Nonlinear Modal Analysis **687**
Differential Stiffness and Follower Force Stiffness in Linear and Nonlinear Solution Sequences. 690
Buckling Analysis Enhancement 691
Nonlinear Transient Response Analysis. 701
Performance in SOL 129. 713
Nonlinear Analysis With Superelements 719
Grid Point Force Balance and Element Strain Energy in Nonlinear Static Analysis 719

D Acoustic Cavity Modeling

Introduction **725**
Assumptions and Limitations **726**
Acoustic Cavity Example Problem **727**

E Cyclic Symmetry

An Overview **733**
Symmetry **733**
A-Frame (Dihedral Symmetry). 734
Pinwheel (Rotational Symmetry) 734
N-Gons. 735
Three-Dimensional Bodies 735
Cyclic Symmetry versus Traditional Reflective Symmetry Techniques. 736
Summary 737
Theory **738**
Symmetrical Components. 739
Transformation from Physical Components to Cyclic Components (ROT) 740
Transformation from Cyclic Components to Physical Components (ROT) 741
Boundary Conditions and Problem Solution (ROT) 742
Summary of Transformations, Boundary Conditions, and Problem Solution (DIH). 747
Transformation from Physical Components to Cyclic Components (DIH) 748
Transformation from Cyclic Components to Physical Components (DIH) 748
Boundary Conditions and Problem Solution (DIH) 749
Using Cyclic Symmetry 750
Solution of Axisymmetric Problems. 752

F p-Elements

Introduction **756**
p-Elements in MSC Nastran **756**
Combining p- and h-Elements 757

p-Version Capabilities 757
p-Element Example Problems 761
Square Plate with a Circular Hole 762
Spherical Structure Under Internal Pressure 766
Motor Blade – Normal Modes 770
Torsion of Elliptical Bar 772
Circular Solid Model 782
Clamped Rhombic Plate 794
Helix Modeled with CBEAM p-Elements 796
Plate with Stiffener 797
Solid p-Elements in Linear Dynamic Solution Sequences 801
p-Element Thermal Analysis 803

G Local Adaptive Mesh Refinement

Introduction 806
Adaptive Mesh Refinement Loop 806
Refinement by Regular Subdivision 808
Location of New Grid Points 811
Hanging Nodes and Multipoint Constraints on Hanging Nodes 814
Selection of Elements to Refine 818
Refinement Criteria 819
Propagation of Refinement 827
Transference of Analysis Data Between Unrefined and Refined Meshes 832
Transference of Element Properties 832
Transference of Distributed Loads and Concentrated Forces 835
Transference of Displacement Coordinate System, Displacement Boundary Conditions, and Constraints . 835
Detection of Geometric Features 838
Detection of Material, and Superelement Interfaces 845

Preface

- How to Use this Guide 2
- List of MSC Nastran Books 3
- Using other Manuals 4
- Typographical Conventions 5
- Accessing MSC Nastran Manuals 5
- Training and Internet Resources 7
- Technical Support 8

How to Use this Guide

This guide is written for a any MSC Nastran user. No prior experience with commercial finite element software is assumed, and no finite element-specific university coursework is required.

Prerequisites

It is assumed that you have a bachelor's degree in any of the fields relevant to structural analysis: mechanical engineering, civil engineering, engineering mechanics, or the equivalent.

Goals of this Guide

This guide provide you with the reference information regarding MSC Nastran input data, grind points and coordinate system, structure elements, material properties, loads, constraints, solution sequence, DMA, data base concepts, and plotting.

It also covers some basic theory regarding nonlinear analysis, p-elements, local adaptive mesh refinement, accuracy considerations, theory of Eigenvalue extraction, and acoustic cavity modeling.

List of MSC Nastran Books

A list of some of the MSC Nastran documents is as follows:

Installation and Release Guides
■ Installation and Operations Guide
■ Release Guide
Reference Guides
■ Quick Reference Guide
■ DMAP Programmer's Guide
■ Reference Guide
■ Utilities Guide
Demonstration Guides
■ Linear Analysis
■ Implicit Nonlinear (SOL 400)
■ Explicit Nonlinear (SOL 700)
User's Guides
■ Getting Started
■ Linear Static Analysis
■ Dynamic Analysis
■ Embedded Fatigue
■ Embedded Vibration Fatigue
■ Thermal Analysis
■ Superelements and Modules
■ Design Sensitivity and Optimization
■ Rotordynamics
■ Implicit Nonlinear (SOL 400)
■ Explicit Nonlinear (SOL 700)
■ Aeroelastic Analysis
■ User Defined Services
■ Non Linear (SOL 600)
■ High Performance Computing
■ DEMATD

You may find any of these documents from MSC Software at:

http://simcompanion.mscsoftware.com/infocenter/index?page=home

Using other Manuals

After reading the *Getting Started Guide*, we recommend that you go through either the *Linear Static Analysis User's Guide*, the *Dynamic Analysis User's Guide*, or the *Nonlinear User's Guide* depending on the type of simulation you are interested in performing. Details of some of the manuals are listed here for your reference.

This guide contains many excerpts from the *Quick Reference Guide (QRG)*. Most of the excerpts have been edited-some extensively-to eliminate material that is not relevant to the topics covered in this book.

- MSC Nastran Quick Reference Guide: The *QRG* contains a complete description of all the input entries for MSC Nastran. It contains complete descriptions of all the finite element input data and options available in MSC Nastran. Each entry provides a description, formats, examples, details on options, and general remarks. You will find the full descriptions for all SOL 400 input entries in the *QRG*.
- MSC Nastran Reference Guide: It provides supporting information that relates to the theory of MSC Nastran inputs, element libraries, and loads and boundary conditions.
- MSC Nastran Linear Static Analysis User's Guide: It provides support information on the basic use of MSC Nastran which can also be applied to SOL 400.
- Nonlinear User's Guide SOL 400: This guide provides a complete background to MSC Nastran Implicit Nonlinear (SOL 400) and describes how to use SOL 400 within the MSC Nastran environment. The theoretical aspects of nonlinear analysis methods, types, and techniques are included along with thorough descriptions for nonlinear material models, properties, and loads and constraints.
- Dynamic Analysis User's Guide: This guide describes the proper use of MSC Nastran for solving various dynamic analysis problems. This guide serves as both an introduction to dynamic analysis for the new user and a reference for the experienced user.
- MSC Nastran High Performance Computing User's Guide: It describes the strategy for getting the best performance out of MSC Nastran for SOL 101, 103, 108, 111, and 400. The guide also gives details on selecting the hardware system that provides the optimal performance for your models.
- DMAP Programmer's Guide: MSC Nastran provides a wide variety of solution sequences. Each solution sequence consist of a series of DMAP (Direct Matrix Abstraction Program) statements. DMAP is a high level language with its own compiler and grammatical rules. MSC Nastran allows you to modify the solution sequences or to write your own solution sequences using DMAP.

This guide contains many highlighted links (in blue) to other MSC Nastran documents and all the documents were delivered together as a collection. If you keep the collection together the links between documents will work.

Two ways of working with links are as follows:

- Use alt ← to return back to the window your curser is in
- Open the other "linked to" documents in a new window from an Adobe Reader
  a. Choose Edit → Preferences → Documents → ☐
  b. Open cross-document links in the same window.
  c. Un-check the ☐ checkbox.

d. Select OK.

Typographical Conventions

The section provides a brief overview of the typographical conventions used in the document to help the user better follow the MSC Nastran documentation.

This section describes some syntax that will help you in understanding text in the various chapters and thus in facilitating your learning process. It contains stylistic conventions to denote user action, to emphasize particular aspects of a MSC Nastran run or to signal other differences within the text.

Courier New	Represents command-line options of MSC Nastran and results from f04/f06 files. `Example: nast20170 memorymax=16gb myjob.dat`
Quoted Text	Represents command-line options of MSC Nastran for in-line text. Example: memorymax=16gb
Arial font	To represent elements. Example: RBE3 and RSPLINE are interpolation elements and are not rigid.
Red Text	Represents items in the examples that we to emphasize. Example: smp=16
Bold Text	Represents items in the text that we want to emphasize. Example: **dmp=4**
Italic Text	Represents references to manuals/documents.

Accessing MSC Nastran Manuals

This section describes how to access the MSC Nastran documentation outside of MSC Software. MSC Nastran documentation is available through PDF files. The PDF files can be obtained from the following sources:

- MSC Nastran documentation installer
- SimCompanion
- Combined documentation

The PDF documentation files are appropriate for viewing and printing with Adobe Acrobat Reader (version 5.0 or higher), which is available for most Windows and Linux systems. These files are identified by a .pdf suffix in their file names.

Downloading the PDF Documentation Files

You can download the PDF documentation from SimCompanion (http://simcompanion.mscsoftware.com).

Navigating the PDF Files

For the purpose of easier online document navigation, the PDF files contain hyperlinks in the table of contents and index. In addition, links to other guides, hyperlinks to all cross-references to chapters, sections, figures, tables, bibliography, and index entries have been applied.

Printing the PDF Files

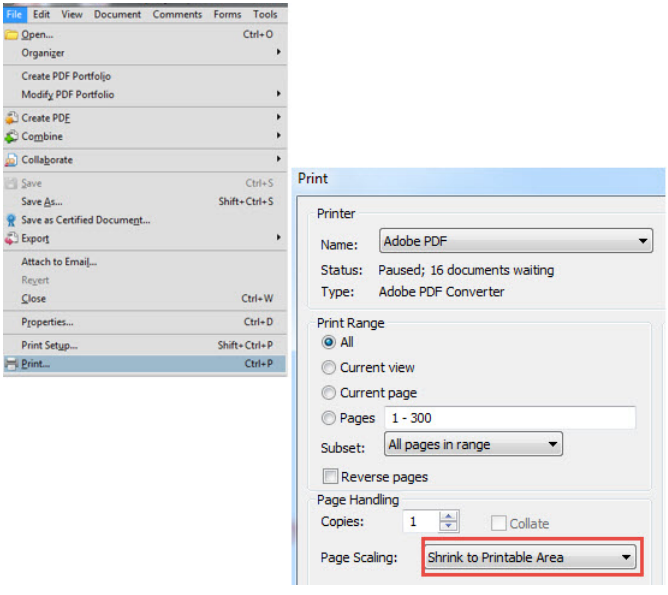
Adobe Acrobat PDF files are provided for printing all or part of the manuals. You can select the paper size to which you are printing in Adobe Acrobat Reader by doing the following:

1. Click **File**.
2. Select the **Print**.... option. The **Print** dialog box is displayed.
3. Select **Page Setup**....
4. Choose the required paper size in the **Page Setup** menu.

The PDF files are recommended when printing long sections since the printout will have a higher quality.

If the page is too large to fit on your paper size, you can reduce it by doing the following:

1. Select the **File -> Print**.
2. Under **Page Scaling**, choose the **Shrink to Printable Area** option.



Training and Internet Resources

Information about MSC's products, services and latest events is available on our website www.mscsoftware.com.

The information about MSC Seminars is available on the training link http://www.mscsoftware.com/msc-training. You can also use this link to schedule the seminars.

If you are a new MSC Nastran user, we recommend the following courses:

NAS101A: Linear Static and Normal Modes Analysis using MSC Nastran

This course serves as an introduction to fundamental use of MSC Nastran. Through this course you will learn the following:

- Finite Element Analysis (FEA).
- Features available in MSC Nastran for solving structural engineering problems.
- To create all finite element models and edit using a text editor.
- Proper data structure of the MSC Nastran input file.

NAS101B: Advanced Linear Analysis using MSC Nastran

This course is a continuation of NAS101A. Through this course you will learn the following:

- Theory of buckling analysis and how to perform a buckling analysis.
- About rigid elements—MPC, RBAR, RBE2, and RBE3.
- Modeling with interface element CINTC and connectors.
- Lamination theory and composite materials.
- MSC Nastran composite theory.
- Failure theories.
- Linear contact and permanent glued contact.
- Different model checks.
- Modeling tips and tricks.

NAS 102A: Dynamic Analysis using MSC Nastran

Through this course you will learn the following:

- Fundamentals of dynamics
- Normal modes
- Direct and modal methods
- Frequency response
- Transient response
- Damping
- Enforced motion
- Response spectrum
- Random analysis

- Residual vectors

NAS 102B: Advanced Dynamic Analysis using MSC Nastran

This course will help you learn the following:

- Review and expand upon the basic dynamic analysis topics.
- Introduce model checkout tools for dynamics.
- Introduce advanced dynamic analysis capabilities.
- Show where to get technical support.
- Show the available training for MSC Nastran.

NAS 124: Thermal Analysis using MSC Nastran and Patran

This course will help you learn the following:

- Heat transfer
- Modes of heat transfer
- Types of analysis
- Thermal model elements and material properties
- Thermal model boundary conditions
- Thermal model loads
- Steady and transient state equation

NAS120: Linear Static Analysis using MSC Nastran and Patran

This seminar provides the foundation required for intermediate and advanced MSC Nastran applications:

- Linear static
- Normal modes
- Buckling analysis of structures with the use of MSC Nastran and Patran

In this course MSC Nastran data structure, the element library, modeling practices, model validation, and guidelines for efficient solutions are discussed and illustrated with examples and workshops. Patran is an integral part of the examples and workshops. It will be used to generate and verify illustrative MSC Nastran models, manage analysis submission requests, and to visualize results.

Technical Support

If you encounter difficulties while using MSC Nastran, first please refer to the section(s) of the manual containing information on the commands you are trying to use or the type of problem you are trying to solve.

Visit SimCompanion

The product documentation is available in SimCompanion (http://simcompanion.mscsoftware.com). The SimCompanion gives you access to a wealth of resources for MSC Software products. You will find various information such as:

- Product documentations
- Knowledge base articles
- Product error lists (fixed and known issues for each release)
- SimAcademy webinars
- Product and support contact information

SimCompanion is a searchable database which allows you to find articles relevant to your inquiry. Valid MSC customer entitlement and login is required to access the database and documents. It is a single sign-on that gives you access to product documentation for complete list of products from MSC Software, allows you to manage your support cases, and participate in our discussion forums.

Help Us Help You

Clients frequently call up the support engineers at MSC Software with enquiry regarding models that do not run correctly. Our technical support staff can help you much more efficiently and effectively if you are working with a small model, since debugging a small model is much easier, and the turnaround time to rerun a (hopefully) corrected test model is minutes rather than hours.

- For information on the latest events, products and services for all products, refer to the MSC Software corporate site (www.mscsoftware.com).
- For technical support phone numbers and contact information, please visit: http://www.mscsoftware.com/Contents/Services/Technical-Support/Contact-Technical-Support.aspx.

1 nast20200 Command and Input Data

- Introduction 11
- nast20200 Command 14
- The NASTRAN Statement (Optional) 14
- File Management Section (FMS) 15
- Executive Control Section 15
- Case Control Section 15
- Bulk Data Section 32
- Use of Parameters 32

Introduction

nastran Command

The recommended procedure for invoking MSC Nastran is to use the nastran command. The nastran command is described in Executing MSC Nastran (p. 1) in the *MSC Nastran Quick Reference Guide*.

Input Data

MSC Nastran input requires records that are 80 characters (or columns) in length. The input file is comprised of five sections, which must be assembled in the sequence shown in Figure 1-1.

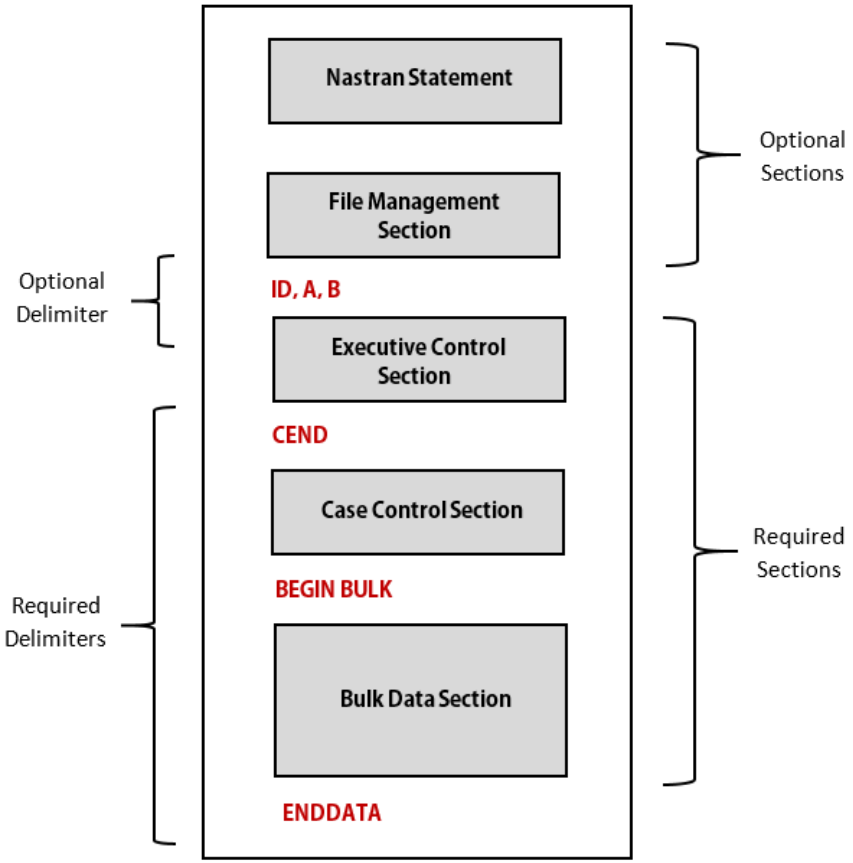


Figure 1-1 Structure of the MSC Nastran Input File

The records of the first four sections are input in free-field format and only columns 1 through 72 are used for data. Any information in columns 73 through 80 may appear in the printed echo, but will not be used by the program. If the last character in a record is a comma, then the record is continued to the next record.

The NASTRAN statement is optional and is used in special circumstances (see the Executing MSC Nastran (p. 1) in the *MSC Nastran Quick Reference Guide*).

The File Management Section (FMS) is optional and typically follows the NASTRAN statement(s). It ends with the specification of an Executive Control statement. This section provides for database initialization and management, along with job identification and restart conditions. The File Management statements are described in the File Management Statements (p. 41) in the *MSC Nastran Quick Reference Guide*.

The Executive Control Section begins with the first Executive Control statement and ends with the CEND delimiter. It identifies the job and the type of solution to be performed. It also declares the general conditions under which the job is to be executed, such as maximum time allowed and type of system diagnostics desired. If the job is to be executed with a solution sequence, the actual solution sequence is declared, along with any alterations to the solution sequence that may be desired. If DMAP (Direct Matrix Abstraction Program) is used, the complete DMAP sequence must appear in the Executive Control Section. The Executive Control statements and examples of their use are described in the Executive Control Statements (p. 105) in the *MSC Nastran Quick Reference Guide*.

The Case Control Section follows CEND and ends with the BEGIN BULK delimiter. It defines the subcase structure for the problem, defines sets of Bulk Data, and makes output requests for printing, punching and plotting. A general discussion of the functions of the Case Control Section and a detailed description of the commands used in this section are given in the Case Control Command Descriptions (p. 193) in the *MSC Nastran Quick Reference Guide*. The special requirements of the Case Control for each solution sequence are discussed in Restart Procedures, 361.

The Bulk Data Section follows BEGIN BULK and ends with the ENDDATA delimiter. It contains all of the details of the model and the conditions for the solution. BEGIN BULK and ENDDATA must be present even though no new Bulk Data is being introduced into the problem or if all of the Bulk Data is coming from an alternate source, such as user-generated input. The BEGIN BULK entry is in free field format. The ENDDATA delimiter must begin in Column 1 or Column 2.

The input file may be formed by the insertion of other files, with the INCLUDE statement. This may be specified in any of the five parts of the input file. Note, however, the data referenced by the INCLUDE statement must be appropriate for that section of the input file.

Comments may be inserted in any of the parts of the input file. They are identified by a dollar sign ($) in any column followed by any characters out to Column 80. Columns 2 through 72 may contain any desired text.

Table 1-1 gives valid characters for MSC Nastran.

Table 1-1 ASCII Subset for MSC Nastran

Character	ASCII (HEX)	Character	ASCII (HEX)
HT (Horizontal Tabulation)	09	@	40
		A	41
SP (Space “blank”)	20	B	42
		C	43
!	21	D	44
"	22	E	45
#	23	F	46
$	24	G	47
%	25	H	48
&	26	I	49
‘	27	J	4A
(	28	K	4B
)	29	L	4C
*	2A	M	4D
+	2B	N	4E
’	2C	O	4F
-	2D	P	50
.	2E	Q	51
/	2F	R	52
0	30	S	53
1	31	T	54
2	32	U	55
3	33	V	56
4	34	W	57
5	35	X	58
6	36	Y	59
7	37	Z	5A
8	38	[	5B
9	39	\	5C

Table 1-1 ASCII Subset for MSC Nastran (continued)

Character	ASCII (HEX)	Character	ASCII (HEX)
:	3A	]	5D
;	3B	^	5E
<	3C	-	5F
=	3D		
>	3E		
?	3F		

nast20200 Command

MSC Nastran is executed from the command line using the command `nast20200`. This command sets environment variables in order for MSC Nastran to correctly execute on your system and it also has capabilities to predict memory and solvers (as of MSC Nastran 2020 with `solve=auto`). The basic format of the `nast20200` command is:

`nast20200` *input_data_file* `keywords`

`nast20200` *input_data_file* [keyword1=value1 keyword2=value2 …]

where input_data_file is the name of the file containing the input data and keyword=valuei is one or more optional keyword assignment arguments. For example, to run an MSC Nastran job using the data file example1.dat, enter the following command:

nast20200 example1.dat

Various options to the `nast20200` command are available using keywords. Keyword assignments consist of a keyword, following by an equal sign, followed by a keyword value. For example,

```
nast20200 example1.dat memorymax=16gb
```

Note: In Windows you can use a hash mark # instead of the equal sign. This is useful if `nast20200 example1.dat memorymax=16gb` command is placed in `.bat file`.

`nast20200` example1.dat memorymax=16gb

The NASTRAN Statement (Optional)

The NASTRAN statement is used to change the default values for certain operational parameters (also called system cells). See the Executing MSC Nastran (p. 1) in the *MSC Nastran Quick Reference Guide* for a description of the format and most often used system cells. The NASTRAN statement is used for special circumstances and is therefore not needed in most runs. The NASTRAN statement may also be specified in the Runtime Configuration (RC) files at the system, user, and job level. See the MSC Nastran Installation and Operation Guide.

File Management Section (FMS)

The File Management Section (FMS) is primarily intended for the attachment and initialization of Database sets (DBsets) and FORTRAN files. The initialization of DBsets includes specification of their maximum size, member names, and physical filenames. The initialization of FORTRAN files includes the specification of their filenames, FORTRAN unit numbers, and FORTRAN attributes.

In most classes of problems that use MSC Nastran solution sequences, no File Management statements are required because a default File Management Section is executed at the beginning of every run. The default File Management Section is described in Database Concepts, 454. For details, see File Management Statements (p. 41) in the *MSC Nastran Quick Reference Guide*.

Executive Control Section

The Executive Control statements, in general, select the solution sequences to be run and various diagnostics. The input specifications are given in Executive Control Statements (p. 105) in the *MSC Nastran Quick Reference Guide*.

Most statements are order independent. Exceptions are the `COMPILE`, `COMPILER`, `ALTER`, `ENDALTER`, and `LINK` statements. The LINK statement must appear after all `COMPILE` statements. The `COMPILER` statement (or equivalent DIAGs) must appear before all `COMPILE` statements. This statement sets the default print control for subsequent `COMPILE` statements.

Case Control Section

The Case Control Section defines the subcase structure for the problem, makes selections from the Bulk Data Section, and makes output requests. Grid point stress and stress discontinuities output request are treated separately in Stress Recovery at Grid Points, 436 and Mesh Stress Discontinuities at Grid Points, 447, respectively. Structure plotter output requests and curve plot requests are treated separately in Section 11.2 and Section 11.3, respectively.

The Case Control commands are described in the Case Control Command Descriptions (p. 193) in the *MSC Nastran Quick Reference Guide.*

The Case Control Section has several basic functions:

- Selects loads and constraints.
- Requests printing, plotting, and/or punching of input and output data (plotter commands are described in Plotting, 460).
- Defines the subcase structure for the analysis.

Case Control Section Requirements for Solutions

This section describes the options and requirements in the Case Control Section for the various solution sequences.

In general, a separate subcase is defined for each loading condition and/or each set of constraints. Subcases may also be used in connection with output requests, such as requesting different output for each mode in a real eigenvalue problem. Only

one level of subcase definition is provided. All items placed above the subcase level (ahead of the first subcase) will be used for all following subcases unless overridden within the individual subcase.

Case Control Commands for Solutions

This section describes the Case Control structure for the various solution sequences.

Static Analysis (SOLutions 101, 144, 200, and 400)

- A separate subcase must be defined for each unique combination of constraints and static loads. Multiple constraint sets are permitted in SOLutions 101, and 200 (for `ANALYSIS = STATICS` only) and for residual structures only.
- A static loading condition must be defined for each subcase with a `LOAD`, `TEMPERATURE(LOAD)` or `DEFORM` selection unless all loading is specified with grid point displacements on SPC entries. In SOLution 144, a loading condition is optional, but a trim variable constraint must be selected with the `TRIM` command.
- An `SPC` set must be selected for each subcase, unless the model is a properly supported free body, or all constraints are specified on `GRID` entries, scalar connection entries, or with general elements.
- `REPCASE` may be used to repeat subcases in order to allow multiple sets of the same output item.
- Direct input matrices may be selected with K2GG and/or M2GG above the subcase level. P2G is allowed in all the above SOLutions.
- In SOLutions 101 and 200, the `SUPORT1` command may be used to select `SUPORT1` Bulk Data entries for inertia relief analysis. See Superelement Subcase Structure, 25 for additional Case Control requirements for superelements.
- For an adaptive p-version element solution (multiple p-levels) the `ADAPT` command must select an ADAPT Bulk Data entry. The `ADAPT` command may be specified to limit the subcases participating in the adaptive analysis. For a single p-level analysis the `ADAPT` command must select a PSET entry. The `DATAREC` and `OUTRCV` commands control the output of p-element stresses, strains, and displacements. The `VUGRID` command requests output at view-elements and view-grids.
- The `SMETHOD` command may be used to select parameters for an iterative solution technique.

In statics problems, provision has been made for the combination of the results of several subcases. This is convenient for studying various combinations of individual loading conditions and for the superposition of solutions for symmetrical and antisymmetrical boundaries.

The following examples of Case Control indicate typical ways of defining subcases:

- Static analysis with multiple loads and boundary conditions. Multiple boundary conditions are supported only in SOLutions 101, 103, 105, 145, and 200. See the BC Case Control command.

```
   DISPLACEMENT = ALL
   MPC = 3
SUBCASE 1
   SPC = 2
   TEMPERATURE(LOAD) = 101
   LOAD = 11
SUBCASE 2
   SPC = 2
   DEFORM = 52
   LOAD = 12
SUBCASE 3
   SPC = 4
   LOAD = 12
SUBCASE 4
   MPC = 4
   SPC = 4
```

Four subcases are defined in this example. The displacements at all grid points will be printed for all four subcases. `MPC = 3` will be used for the first three subcases and will be overridden by MPC = 4 in the last subcase. Since the constraints are the same for subcases 1 and 2 and the subcases are contiguous, the static solutions will be performed simultaneously. In `SUBCASE 1`, thermal load 101 and external load 11 are internally superimposed, as are the external and deformation loads in `SUBCASE 2`. In `SUBCASE 4`, the static loading will result entirely from enforced displacements of grid points.

- Linear combination of subcases

```
   SPC = 2
   SET 1 = 1 THRU 10,20,30
   DISPLACEMENT = ALL
   STRESS = 1
SUBCASE 1
   LOAD = 101
   OLOAD = ALL
SUBCASE 2
   LOAD = 201
   OLOAD = ALL
SUBCOM 51
   SUBSEQ = 1.0,1.0
SUBCOM 52
   SUBSEQ = 2.5,1.5
```

Two static loading conditions are defined in subcases 1 and 2. `SUBCOM` 51 defines the sum of subcases 1 and 2. `SUBCOM` 52 defines a linear combination consisting of 2.5 times `SUBCASE 1` plus 1.5 times `SUBCASE 2`. The displacements at all grid points and the stresses for the element numbers in SET 1 will be printed for all four subcases. In addition, the nonzero components of the static load vectors will be printed for subcases 1 and 2.

- Statics with one plane of symmetry. (Note that this is not supported in the superelement solution sequences.)

```
   SET 1 = 1,11,21,31,51
   SET 2 = 1 THRU 10, 101 THRU
110
   DISPLACEMENT = 1
   ELFORCE = 2
SYM 1
   SPC = 11
   LOAD = 21
   OLOAD = ALL
SYM 2
   SPC = 12
   LOAD = 22
SYMCOM 3
SYMCOM 4
   SYMSEQ 1.0,-1.0
```

Two SYM subcases are defined in subcases 1 and 2. SYMCOM 3 defines the sum and SYMCOM 4 the difference of the two SYM subcases. The nonzero components of the static load will be printed for SUBCASE 1 and no output is requested for SUBCASE 2. The displacements for the grid point numbers in SET 1 and the forces for elements in SET 2 will be printed for subcases 3 and 4.

- Use of REPCASE in statics problems

```
   SET 1 = 1 THRU 10, 101 THRU 110,201 THRU 210
   SET 2 = 21 THRU 30,121 THRU 130,221 THRU 230
   SET 3 = 31 THRU 40,131 THRU 140,231 THRU 240
SUBCASE 1
   LOAD = 10
   SPC = 11
   DISP = ALL
   SPCFORCE = 1
   FORCE = 1
REPCASE 2
   FORCE = 2
REPCASE3
   FORCE=3
```

This example defines one subcase for solution and two subcases for output control. The displacements at all grid points and the nonzero components of the single-point forces of constraint along with forces for the elements in SET 1 will be printed for SUBCASE 1. The forces for elements in SET 2 will be printed for REPCASE 2 and the forces for elements in SET 3 will be printed for REPCASE 3.

Normal Modes (SOLutions 103, 111, 200, and 400)

- METHOD must be used to select an EIGR or EIGRL entry that exists in the Bulk Data Section. In SOLutions 103 and 200 multiple subcases may be specified with different sets of METHOD, MPC, SPC, and SUPORT1 commands. In all other SOLs only one set may be specified. (See the BC Case Control command.)
- An SPC set must be selected unless the model is a free body or all constraints are specified on GRID entries, scalar connection entries or with general elements.

- Multiple subcases may be used to control output requests. A single subcase is sufficient if the same output is desired for all modes. If multiple subcases are present, the output requests will be honored in succession for increasing mode numbers. MODES may be used to repeat subcases in order to make the same output request for several consecutive modes.
- Direct input matrices may be selected with K2GG and/or M2GG above the subcase level.
- In SOLutions 103 and 200, the SUPORT1 command may be used to select SUPORT1 Bulk Data entries for computing rigid body modes.
- See Superelement Subcase Structure, 25 for additional Case Control requirements for superelements.
- Use of MODES in Eigenvalue Problems

```
   METHOD = 2
   SPC = 10
SUBCASE 1
   DISP = ALL
   STRESS = ALL
   MODES = 2
SUBCASE 3
   DISP = ALL
```

  In this example, the displacements at all grid points will be printed for all modes. The stresses in all elements will be printed for the first two modes. If subcases are not present, output requests act on all computed eigenvectors.

- Use of REPCASE in Normal Modes Analysis

  In normal modes analysis the REPCASE command is applied only to the last mode in the previous subcase.

```
SET 200 = 1,7,10,15
DISP(PLOT) = ALL
SUBCASE 1
   LABEL = MODE 1
REPCASE 2
   LABEL = REPCASE FOR MODE 1
   DISP = 200
SUBCASE 3
   LABEL = MODE 2
REPCASE 4
   LABEL = REPCASE FOR MODE 2
   DISP = 200
```

  This example requests the output of the first two eigenvectors. In the SUBCASEs, output is computed at all points for plotting purposes. In the REPCASEs, output is printed only at selected points.

- Prestressed structures may be analyzed by specifying static subcases to define the pre-loads. Then the STATSUB(PRELOAD) command that references the desired static subcases must be specified in the dynamic response subcases to request the inclusion of differential stiffness in the dynamic formulation. By default, stiffness due to follower forces is also included in the differential stiffness.

- For an adaptive p-version element solution (multiple p-levels) the ADAPT command must select an ADAPT Bulk Data entry. The ADAPT command may be specified to limit the subcases participating in the adaptive analysis. For a single p-level analysis the ADAPT command must select a PSET entry. The DATAREC and OUTRCV commands control the output of p-element stresses, strains, and displacements. The VUGRID command requests output at view-elements and view-grids.
- For fluid-structure analysis, the METHOD(FLUID) and SDAMPING(FLUID) may be used to request eigensolution and modal damping parameters for the fluid which are different from the structure.

Buckling Analysis (SOLutions 105 and 200)

- The Case Control Section must contain at least two subcases. In SOL 105, multiple subcases may be specified with different sets of METHOD, SPC, and MPC commands. See also the BC Case Control command. The buckling subcase(s) may have a different boundary condition than the statics subcase(s).
- In SOL 105 and 200, the METHOD and STATSUB commands must be specified in one or more subcases. In SOLution 105 and 200, two different STATSUB conditions may be selected:
  - STATSUB(PRELOAD) for prestressing
  - STATSUB(BUCKLE) for determining the buckling loading condition. Both may appear in the same subcase with a METHOD command.
- A static loading condition must be defined in the first subcase with a LOAD, TEMPERATURE(LOAD), or DEFORM selection, unless all loading is specified by grid point displacements on SPC entries.
- Output requests that apply to both the static solution and the buckling modes may be placed above the subcase level.

Modal Formulation (SOLutions 110, 111, 112, 145, 146, 200, and 400)

- An SPC set must be selected above the subcase level unless the model is a free body or all constraints are specified on GRID entries, scalar connection entries, or with general elements.
- SDAMPING must be used to select a TABDMP1 entry if modal damping is desired. Only one selection is allowed in superelement solution sequences.
- For SOLutions 110, 111, and 112, STATSUB selects a static solution used to form differential stiffness.
- For SOLution 145, see also the BC Case Control command.

Complex Eigenvalue and Flutter Analysis (SOLutions 107, 110, 145, 200, and 400)

- At least one subcase must be defined for each unique set of direct input matrices (K2PP, M2PP, B2PP). Only one set is allowed in SOLutions 107 and 110. More than one set may be selected in SOLution 145.
- Multiple subcases for each set of direct input matrices are used only to control output requests. A single subcase for each set of direct input matrices is sufficient if the same output is desired for all modes. If consecutive multiple subcases are present for a single set of direct input matrices, the output requests will be honored in succession for increasing mode numbers. MODES may be used to repeat subcases in order to make the same output request for several consecutive modes.

- CMETHOD must be used to select an EIGC entry in the Bulk Data Section for each set of direct input matrices. Only one selection is allowed in SOLutions 107 and 110. In SOLution 145, CMETHOD is required only if k-method flutter analysis is also selected.
- Constraints must be defined above the subcase level for SOLutions 107 and 110. Multiple boundary conditions are allowed for SOLution 145 with a BC Case Control command.
- In SOLution 145, FMETHOD must be used to select a FLUTTER entry in the Bulk Data Section.
- Prestressed structures may be analyzed by specifying static subcases to define the pre-loads. Then the STATSUB(PRELOAD) command that references the desired static subcases must be specified in the dynamic response subcases to request the inclusion of differential stiffness in the dynamic formulation. By default, stiffness due to follower forces is also included in the differential stiffness.
- For fluid-structure analysis, the METHOD(FLUID) and SDAMPING(FLUID) may be used to request eigensolution and modal damping parameters for the fluid which are different from the structure.
- The SMETHOD command may be used to select parameters for an iterative solution technique.

Frequency, Random, and Aeroelastic Response (SOLutions 108, 111, 146, 200, and 400)

- At least one subcase must be defined for each unique set of direct input matrices (K2PP, M2PP, B2PP) or frequencies. Only one set is allowed in SOLutions 108, 111, 118, 146, and 200.
- Consecutive subcases for each set of direct input matrices or frequencies are used to define the loading conditions - one subcase for each dynamic loading condition.
- Constraints must be defined above the subcase level.
- DLOAD must be used to select a frequency-dependent loading condition DLOAD and RLOADi Bulk Data entries, for each subcase. In SOLutions 108, 111, and 146, DLOAD may also select a time-dependent loading condition, i.e., TLOADi entries.
- FREQUENCY must be used to select a set of FREQ, FREQ1, FREQ2, FREQ3, FREQ4, or FREQ5 entries from the Bulk Data Section for each unique set of direct input matrices. Only one selection is allowed in SOLutions 118 and 146.
- OFREQUENCY may be used above the subcase level or within each subcase to select a subset of the solution frequencies for output requests. The default is to use all solution frequencies.
- If Random Response calculations are desired, RANDOM must be used to select RANDPS and RANDTi entries from the Bulk Data Section. Only one OFREQUENCY and FREQUENCY command can be used for each set of direct input matrices.
- LOADSET may be used in the first subcase to select a set of LSEQ entries which define a set of static load vectors to be used as DAREA entries in dynamic analysis.
- Prestressed structures may be analyzed by specifying static subcases to define the pre-loads. Then the STATSUB(PRELOAD) command that references the desired static subcases must be specified in the dynamic response subcases to request the inclusion of differential stiffness in the dynamic formulation. By default, stiffness due to follower forces is also included in the differential stiffness.
- Multiple subcases with different SPC, MPC and SUPORT1 requests may be specified in SOLs 108 and 111.

Transient Response (SOLutions 109, 112, 200, and 400)

- One subcase must be defined for each dynamic loading condition. DLOAD or NONLINEAR must be used to define a time-dependent loading condition for each subcase. Only one DLOAD and NONLINEAR command is allowed in SOLution 200.
- Constraints must be defined above the subcase level.
- TSTEP must be used to select the time-step intervals to be used for integration and output in each subcase. Only one TSTEP selection is allowed in SOLution 200.
- If nonzero initial conditions are desired, lC must be used to select a TIC entry in the Bulk Data Section.
- `LOADSET` may be used in the first subcase to select a set of `LSEQ` entries which define a set of static load vectors to be used as DAREA entries in dynamic analysis.
- Prestressed structures may be analyzed by specifying static subcases to define the pre-loads. Then the STATSUB(PRELOAD) command that references the desired static subcases must be specified in the dynamic response subcases to request the inclusion of differential stiffness in the dynamic formulation. By default, stiffness due to follower forces is also included in the differential stiffness.
- At least one subcase must be defined for each unique set of direct input matrices (K2PP, M2PP, B2PP), dynamic loading (DLOAD) and time step parameters (TSTEP). Only one set of direct input matrices is allowed.

Linear Steady State Heat Transfer (SOLution 101 and 400)

- A separate subcase must be defined for each unique combination of constraints and static loads.
- A static loading condition must be defined for (not necessarily within) each subcase with a LOAD selection, unless all loading is specified with grid point temperatures on SPC entries.
- An SPC set must be selected for (not necessarily within) each subcase, unless all constraints are specified on `GRID` entries or scalar connection entries.
- REPCASE may be used to repeat subcases in order to allow multiple sets of the same output item.

Nonlinear Steady State Heat Transfer (SOLution 153 and 400)

- A separate subcase must be defined for each unique combination of temperature loads (LOAD Case Control command), temperature constraints (SPC and MPC command), and nonlinear iteration strategy (NLPARM command). ANALYSIS = HEAT must be specified above the subcase level.
- The LOAD Case Control command references the static temperature load entries: QVECT, QVOL, QHBDY, and QBDYi.
- An estimated temperature distribution must be defined on TEMP or TEMPD entries and selected by the TEMPERATURE(INITIAL) Case Control command. Temperature constraints are specified on SPCi entries.
- Restarts are controlled by the PARAMeters SUBCASID, SUBID and LOOPID.

  See the MSC Nastran Thermal Analysis User's Guide for further details.

Nonlinear Transient Heat Transfer (SOLution 159 and 400)

- Only one set of temperature constraints (MPC and SPC Case Control command) may be requested and must be specified above the subcase level. ANALYSIS = HEAT must be specified above the subcase level.

- A subcase must be defined for each unique combination of dynamic temperature load conditions (DLOAD command) and nonlinear iteration strategy (TSTEPNL command).
- The DLOAD and/or NONLINEAR command must be used to specify a time-dependent loading condition. The static temperature load entries: QVECT, QVOL, QHBDY, and QBDYi may be used in defining a dynamic load specified by the TLOADi entry. The set identification number on the static load entries, usually field 2, is specified in the DAREA field of the TLOADi entry.
- The initial temperature condition is requested by the IC command which selects TEMP or TEMPD entries. The IC command must be specified above the subcase level.
- Restarts are controlled by the PARAMeters STIME, LOOPID, and SLOOPID.

See the MSC Nastran Thermal Analysis User's Guide for further details.

Cyclic Symmetry Statics (SOLution 114)

- Each loading condition must have a separate subcase.
- All constraints must be specified above the subcase level.
- A single set of direct input matrices may be selected with K2GG and/or M2GG above the subcase level. P2G is allowed in SOLution 114.

For details, see Cyclic Symmetry, 703.

Cyclic Symmetry Modes (SOLution 115)

- METHOD must be included above the subcase level to select an EIGR or EIGRL entry that exists in the Bulk Data Section.
- All constraints must be specified above the subcase level.
- A single set of direct input matrices may be selected with K2GG and/or M2GG above the subcase level.

For details, see Cyclic Symmetry, 703.

Cyclic Symmetry Buckling (SOLution 116)

- Case Control requirements for cyclic symmetry are the same as in SOLution 105. However, only one static and buckling subcase may be specified.
- For the static subcase requirements see the description under Cyclic Symmetry Statics (SOLution 114), 23.

For details, see Cyclic Symmetry, 703.

Design Sensitivity Analysis and Optimization (SOLution 200)

- SOLution 200 has the same case control requirements as the solution sequences identified on the ANALYSIS Case Control command for SOL 200. The ANALYSIS command is specified in each subcase to identify the analysis to which the subcase is to be applied. If superelements are present, then the ANALYSIS command need only be specified in the residual structure subcases.
- Table 1-2 indicates the analysis types that are available in SOL 200 and whether the types support multiple subcases and multiple boundary conditions.

Table 1-2 Analysis Disciplines Supported in Multidisciplinary Analysis and Design of SOL 200

Analysis	SOL* Numbers	Multiple Subcases	Multiple Boundary Conditions
Statics	101	Y	Y
Normal Modes	103	Y	Y
Buckling	105	Y	Y
Direct Complex Eigenanalysis	107	N	N
Direct Frequency	108	Y	N
Modal Complex Eigenanalysis	110	N	N
Modal Frequency	111	Y	N
Modal Transient	112	N	N
Static Aeroelasticity	144	Y	Y
Flutter	145	Y	Y
*The column of Solution Numbers refers to the number MSC Nastran would use to perform the separate analyses.			

- The DESGLB and DESSUB commands select design constraints specified on DCONSTR and DCONADD Bulk Data entries. The DESGLB selection is independent of subcase and the DESSUB applies only to the current subcase.
- The DESOBJ command selects a scalar design objective specified on a DRESP1, DRESP2, or DRESP3 Bulk Data entry.
- The optional DESVAR command selects the set of design variables to be used. If this command is not specified, all DESVAR Bulk Data entries are used.
- The MODTRAK command selects mode tracking parameters.
- The AUXCASE and AUXMODEL indicate subcases to be applied to auxiliary models.
- The DSAPRT command requests the output of design sensitivity coefficients.

See the Design Sensitivity and Optimization User's Guide for further information.

Nonlinear Static Analysis (SOLution 106)

- A separate subcase must be defined for each unique combination of static loads (LOAD and TEMPERATURE(LOAD)), constraints (SPC and MPC), and nonlinear iteration strategy (NLPARM and NLPCI).
- A static loading condition may be defined for each subcase with a LOAD and/or TEMPERATURE(LOAD) selection unless all loading is specified through enforced displacements on SPC entries. If unloading or creep under residual load is desired, then no static load is specified in the subcase.

- If thermal loads are present, then TEMPERATURE(INIT) must be specified above the subcase level.
- An SPC command must be specified unless the model is constrained on the GRID entries.
- A single set of direct input matrices may be selected with K2GG, M2GG, and P2G.
- If upstream superelements have static loads, then LOADSET must be used to select LSEQ entries and, in the residual structure subcases only, the CLOAD command must be used to assemble the upstream loads.
- Restarts are controlled by PARAMeters LOOPID and SUBID or SUBCASID. See Parameters (p. 785) in the *MSC Nastran Quick Reference Guide*.
- The normal, complex, or unsymmetric modes due to the total nonlinear stiffness may be computed at the end of each subcase as long as PARAM,NMLOOP,<>0 and a METHOD, CMETHOD, or UMETHOD command is specified in the desired subcase and selects EIGB, EIGR, EIGRL, EIGC, or Bulk Data UEIG entry.
- The buckling modes due to the total nonlinear stiffness may be computed at the end of each subcase as long as PARAM,BUCKLE,>0 and a METHOD command is specified in the desired subcase and selects an EIGB, EIGR, or EIGRL Bulk Data entry.

See Nonlinear Analysis, 535 for further information.

Nonlinear Transient Response Analysis (SOLution 129)

- A subcase must be defined for each unique combination of dynamic loads (DLOAD) and nonlinear iteration strategy (TSTEPNL).
- Only one set of constraints (SPC and MPC) may be selected and must be specified above the subcase level.
- LOADSET may be used to select LSEQ entries which reference static load sets to be used in the generation of dynamic loads.
- Restarts are controlled by PARAMeters STIME, LOOPID, and SLOOPID. See Parameters (p. 785) in the *MSC Nastran Quick Reference Guide*.
- Upstream superelements may be reduced dynamically using the METHOD commands.

See Nonlinear Analysis, 535 for further information.

Superelement Subcase Structure

Subcases must be defined for each superelement to specify load, constraint, and output selection. The syntax rules described above also apply for superelements, i.e., requests above the subcase level are overridden by requests inside the subcase level. See Superelement Case Control in this section for exceptions to the above examples. Static analysis with multiple loads, two superelements.

```
   DISP = ALL
   ESE = ALL
   SPC = 1000
SUBCASE 11
   LOAD = 1000
SUBCASE 12
   LOAD = 1002
```

For most applications, a conventional subcase structure may be used in superelement analysis. The subcase command SEALL = ALL and SUPER = ALL are the default in superelement solution sequences. SEALL = ALL requests the program to run to completion. SUPER = ALL instructs the program to use the same subcases for all superelements. See Restart Procedures, 361, for use of commands alternative to the SEALL command for restarts and for applications where it is necessary to use more than one SUPER command.

The specification of the METHOD command in a superelement subcase requests the calculation of component modes. In this example, the same output, constraint, and load requests are used for all superelements and the residual structure. Two loading conditions are analyzed.

Case Control Commands for Output

Printed output may be requested in either a SORT1 or SORT2 format. The SORT1 format is a tabular listing of the requested output, sorted by grid point number or element number, for each requested loading condition, forcing frequency or time step. The SORT2 format is a tabular listing of the requested output, sorted by loading condition, forcing frequency or time step, for each requested grid point or element.

In any one run, the output requests must be all SORT1 or all SORT2. If both types of output format are desired, a restart must be made to satisfy the second format request. Any request for deformed structure plots results in a request for the SORT1 format. Any X-Y output request results in a request for the SORT2 format.

The default for all analyses except transient response is the SORT1 format. The default for transient response is the SORT2 format.

In the case of the SORT1 format for static or transient analysis, there is an additional option to sort the stresses on the basis of magnitude and to print only the largest stresses. See the description of DTI,INDTA (p. 1802) in the , for details on preparing the user request.

This section describes the output options for the various solution sequences. MSC Nastran is designed to allow the solution of large problems, which can generate very large amounts of output. For this reason, most types of output (e.g., stresses in static analysis, or eigenvectors in modal analysis) are not automatically printed, but must be specifically requested by the user. Options are provided which allow the user to restrict the printed output to only significant results. See Parameter Descriptions for additional output options.

Static Analysis (SOLutions 101, 106, 114, 144, and 200)

- Displacements at selected grid points (DISPLACEMENT).
- Nonzero components of the applied static loads at selected grid points (OLOAD).
- Nonzero components of single point forces of constraint, including reactions on free-body supports due to applied loads, at selected grid points or scalar points (SPCF).
- Forces and stresses in selected elements (FORCE and STRESS). In SOLution 144, forces and pressures on aerodynamic elements (APRESSURE and AEROF).
- In all SOLutions except 4 and 106, grid point force balance at selected grid points (GPFORCE).
- In all SOLutions except 4 and 106, strain energy in selected elements (ESE).

- Undeformed and deformed plots of the structural model (geometry for fundamental region in cyclic symmetry) (OUTPUT(PLOT)).
- Multipoint forces of constraint for MPCs and rigid elements (MPCF).
- Contour plots for stresses in selected elements and displacements at selected grid points (geometry for fundamental region in cyclic symmetry) (OUTPUT(PLOT)).
- Grid point stresses and strains in selected surfaces or volumes are available (GPSTRESS and GPSTRAIN).
- Nonlinear stress output is controlled by NLSTRESS. Its default is ALL.

Normal Modes (SOLutions 103, 115, and 200)

- The eigenvector for a list of selected GRID points (g-set) or SOLUTION points (a-set) along with the associated eigenvalue for each mode (VECTOR or SVECTOR).
- Nonzero components of the single-point forces of constraint for selected modes at selected grid points (SPCF).
- Forces, stresses, and strains in selected elements for selected modes (FORCE, STRESS, and STRAIN).
- Grid point force balance at selected grid points (GPFORCE).
- Strain energy and kinetic energy in selected elements (ESE, EKE).
- Undeformed plot of the structural model and mode shapes for selected modes (geometry for fundamental region in cyclic symmetry) (OUTPUT(PLOT)).
- Grid point stresses and strains in selected surfaces or volumes (GPSTRESS and GPSTRAIN).
- The application of response spectrum curves may be performed in SOLutions 103 and 115. See Local Adaptive Mesh Refinement.
- Multipoint forces of constraint for MPCs and rigid elements (MPCF).

Buckling Analysis (SOLutions 105 and 200)

- Displacements and nonzero components of the static loads and single-point forces of constraint at selected grid points for the static analysis (DISPLACEMENT, OLOAD and SPCF).
- Forces and stresses in selected elements for the static loading condition and selected buckling modes (FORCE and STRESS).
- Mode shapes and nonzero components of the single-point forces of constraint at selected grid points for selected modes (VECTOR and SPCF).
- Undeformed plot of the structural model and mode shapes for selected buckling modes (OUTPUT(PLOT)).
- Grid point force balance at selected grid points for the static loading condition (GPFORCE).
- Strain energy in selected elements for the static loading condition (ESE).
- Grid point stresses and strains in selected surfaces or volumes for the static loading condition (GPSTRESS and GPSTRAIN).
- Multipoint forces of constraint for MPCs and rigid elements (MPCF).

Complex Eigenvalue Analysis (SOLutions 107, 110, 145, and 200)

The following printed output, sorted by complex eigenvalue root number (SORT1), may be requested for any complex eigenvalue extracted, as either real and imaginary parts or magnitude and phase angle (0° – 360° lead):

- The eigenvector for a list of PHYSICAL points (grid points and scalar points) or SOLUTION points (points used in formulation of the general K system) (VECTOR or SVECTOR).
- Nonzero components of the single-point forces of constraint for a list of PHYSICAL points (SPCF).
- Forces, stresses, or strains in selected elements (FORCE and STRESS STRAIN).
- In addition, an undeformed plot of the structural model may be requested. (OUTPUT(PLOT))
- Multipoint forces or constraint for MPCs and rigid elements (MPCF).
- Strain energy, kinetic energy, and energy loss (ESE, EKE, EDE).

Frequency, Random, and Aeroelastic Response (SOLutions 108, 111, 118, 146, and 200)

The following printed output, sorted by frequency (SORT1) or by point number or element number (SORT2), is available, either as real and imaginary parts or magnitude and phase angle (0° - 360°), for the list of frequencies specified by OFREQUENCY:

- Displacements, velocities, and accelerations for a list of PHYSICAL points (grid points and scalar points) or SOLUTION points (points used in formulation of the generalized coordinate system) (DISPLACEMENT or SDISPLACEMENT, VELOCITY or SVELOCITY, ACCELERATION or SACCELERATION).
- Nonzero components of the applied load vector and single-point forces of constraint for a list of PHYSICAL points (OLOAD and SPCF).
- Stresses and forces in selected elements (ALL available only for SORT1) (STRESS and FORCE). In SOLution 146, forces on aerodynamic elements. (AEROF).
- The following plotter output is available for frequency response calculations:
  - Undeformed and deformed structure plots of the structural model (OUTPUT(PLOT)).
  - X-Y plot of any component of displacement, velocity, or acceleration of a PHYSICAL point or SOLUTION point (OUTPUT(XYPLOT)).
  - X-Y plot of any component of the applied load vector of single-point force of constraint (OUTPUT(XYPLOT)).
  - X-Y plot of any stress or force component for an element (OUTPUT(XYPLOT)).
- The following plotter output is available for random response calculations: The data used for preparing the X-Y plots may be punched or printed in tabular form. Also, a printed summary is prepared for each X-Y plot which includes the maximum and minimum values of the plotted function. The printing of the X-Y pairs is the only form of printed output for Random Response. SORT2 output is required for X-Y plots. If both SORT1 and SORT2 requests are made, only the SORT2 output will be printed.
  - X-Y plot of the power spectral density versus frequency for the response of selected components for points or elements (OUTPUT(XYPLOT)).
  - X-Y plot of the autocorrelation versus time lag for the response of selected components for points or elements (OUTPUT(XYPLOT)).

- Multipoint forces of constraint for MPCs and rigid elements (MPCF).
- Strain energy, kinetic energy, and energy loss (ESE, EKE, EDE).

Transient Response (SOLutions 109, 112, 129, and 200)

The following printed output, sorted by point number or element number is available at selected multiples of the integration time step:

- Displacements, velocities, and accelerations for a list of PHYSICAL points (grid points and scalar points) or SOLUTION points (points used in formulation of the general K system) (DISPLACEMENT or SDISPLACEMENT, VELOCITY or SVELOCITY, ACCELERATION or SACCELERATION).
- Nonzero components of the applied load vector and single-point forces of constraint for a list of PHYSICAL points (OLOAD and SPCF).
- Nonlinear force vector for a list of SOLUTION points (NONLINEAR).
- Stresses and forces in selected elements (STRESS and FORCE). In SOLutions 109 and 129, forces are not computed for nonlinear elements.
- Grid point stresses in selected surfaces or volumes (GPSTRESS).
- The following plotter output is available. The data used for preparing the X-Y plots may be punched or printed in tabular form. Also, a printed summary is prepared for each X-Y plot which includes the maximum and minimum values of the plotted function.
  - Undeformed plot of the structural model (OUTPUT(PLOT)).
  - Deformed shapes of the structural model for selected time intervals (OUTPUT(PLOT)).
  - Contour plots for stresses in selected elements, and displacements, velocities or accelerations at selected grid points for selected time intervals (OUTPUT(PLOT)).
  - X-Y plot of any component of displacement, velocity, or acceleration of a PHYSICAL point or SOLUTION point (OUTPUT(XYPLOT)).
  - X-Y plot of any component of the applied load vector, nonlinear force vector, or single-point force of constraint (OUTPUT(XYPLOT)).
  - X-Y plot of any stress or force component for an element (OUTPUT(XYPLOT)).
- Multipoint forces of constraint for MPCs and rigid elements except SOLution 129.

Linear Steady State Heat Transfer (SOLutions 101 and 153)

- Temperatures (THERMAL) and nonzero components of static loads (OLOAD) and constrained heat flow (SPCFORCE) at selected grid points or scalar points.
- The punch option of a THERMAL request to produce TEMP Bulk Data entries.
- Flux density (FLUX) in selected elements.
- Grid point heat balance (GPFORCE) at a selected set of grid points or scalar points (SOLution 101).
- Undeformed plots of the structural model and temperature profiles.
- Contour plots for temperatures at selected grid points.

Steady State Heat Transfer (SOLution 153)

The following output may be requested for the last iteration in Heat Transfer Analysis:

- Temperature (THERMAL) and nonzero components of static loads (OLOAD) and constrained heat flow (SPCFORCE) at selected grid points or scalar points.
- The punch option of a THERMAL request to produce TEMP Bulk Data entries.
- Flux density (FLUX) in selected elements. In the case of CHBDYi elements, a flux density summary is produced that includes applied flux, radiation flux, and convective flux.
- Undeformed plots of the structural model and temperature profiles.
- Contour plots of temperatures at selected grid points.

See the MSC Nastran Thermal Analysis User's Guide for further details.

Nonlinear Transient Heat Transfer (SOLution 159)

The following printed output, is available at selected multiples of the integration time step:

- Temperatures (THERMAL or SDISPLACEMENT) and derivatives of temperatures (VELOCITY or SVELOCITY) for a list of PHYSICAL points (grid points and scalar points) or SOLUTION points (points used in formulation of dynamic equation).
- Nonzero components of the applied load vector (OLOAD) and constrained heat flow (SPCFORCE) for a list of PHYSICAL points.
- Nonlinear load vector (NLLOAD) for a list of SOLUTION points.
- Flux density (FLUX) in selected elements (ALL not allowed).
- Rate of change of enthalpy (HDOT).

The following plotter output is available for Transient Heat Transfer Analysis:

- Undeformed plot of the structural model.
- Temperature profiles for selected time intervals.
- X-Y plot of temperature (TEMP or SDISP) or derivative of temperature (VELO or SVELO) for PHYSICAL or SOLUTION points.
- X-Y plot of the applied load vector (OLOAD), nonlinear load vector (NOLI), or constrained heat flow (SPCF).
- X-Y plot of flux density (FLUX) for an element.
- Contour plots of temperatures at selected grid points.

The data used for preparing the X-Y plots may be punched or printed in tabular form. Also, a printed summary is prepared for each X-Y plot which includes the maximum and minimum values of the plotted function.

Case Control Looping

Boundary Condition Changes

More than one set of MPC, SPC, and SUPORT1 selections is allowed in all of the subcases for the residual structure during one run in SOLutions 101, 103, 105, 106, 108, 111 and 200. Only one boundary condition is allowed for superelements.

Frequency and Dynamic Load Set Changes in SOLutions 108, 111, 118, 146 and 200

More than one FREQUENCY Case Control command may be used per run except in SOLutions 146 and 200. Residual structure subcases may be used to select different dynamic loads using the DLOAD command. Output requests may differ between subcases.

Transient and Dynamic Load Set Changes in SOLutions 109, 112 and 200

More than one TSTEP and DLOAD Case Control command may be used per run except in SOLution 200. Only one residual structure subcase is allowed.

Formulation of Dynamic Equations in All Dynamic Response SOLutions

Multiple subcases with different requests of K2PP, M2PP, B2PP, TFL, and SDAMPING Case Control commands may be specified in SOLs 107 through 112, 129 and 145.

DMIG Bulk Data Entry Processing

Two forms of direct input matrix processing are provided. The Case Control commands `B2GG`, `M2GG`, `K2GG`, and `P2G` select matrices added to the global matrices before constraints are applied and only one set may be selected for superelement or residual structures. These terms may be added to any superelement or the residual structure. `K2GG`, `B2GG` and `M2GG` must be real and symmetric (IF = 6). Their precision should be set equal to the precision of [`K2GG`] (TOUT = 0 will provide this).

The Case Control commands `B2PP`, `M2PP` and `K2PP` select matrices processed only in the residual structure. In the modal formulations they are added after real eigenvalue extraction. These matrices need not be symmetric.

REPCASE, `SUBCOM` and SUBSEQ Case Control Commands

These commands may be used for superelement processing, with the provision that the `SUPER` Case Control command must be used with `REPCASE` and `SUBCOM`. An example of a statics subcase structure follows:

```
SUBCASE 10
SUPER = 10,1
LOAD = 1
SET 1 = 1 THRU 10
DISP = 1
REPCASE 11
SUPER = 10,1
SET 2 = 11 THRU 100
DISP = 2
REPCASE 12
SUPER = 10,1
SET 3 = 101 THRU 200
DISP = 3
SUBCASE 20
SUPER = 10,2
...
```

The REPCASEs must immediately follow the subcase they reference. They should contain the same SUPER = i,j command. In this example, the two REPCASEs give output for the loading condition described in SUBCASE 10.

Bulk Data Section

The primary MSC Nastran input medium is the Bulk Data entry. These entries are used to define the model and data which may be selected in the Case Control Section at execution time. For large problems the Bulk Data may consist of several thousand entries.

The Bulk Data may be submitted with the entries in any order since a sort (see XSORT (p. 2601) in the *MSC Nastran DMAP Programmer's Guide*) is performed prior to the execution of the Input File Processor (see IFP (p. 1908) in the *MSC Nastran DMAP Programmer's Guide*). It should be noted that the machine time to perform this is minimized for input that is already sorted. The sort time for badly sorted data will become significant for large numbers of entries. The user may obtain a printed copy of either the unsorted or sorted Bulk Data by selection in the Case Control Section. A sorted echo is automatically provided unless you specifically suppress it.

In order to minimize the handling of a large number of entries, provision has been made in the Structured Solution Sequences (see Solution Sequences, 336) to store the Bulk Data in the database, and it may be modified on subsequent runs. For any cold start, the entire bulk data must be submitted. Thereafter, if the database from any previous run is saved, the Bulk Data exists in the database in sorted form where it may be modified and reused on restart. Upon restart the Bulk Data entries contained in the Bulk Data are added to the Bulk Data stored in the database. Entries are removed from the database by using a "/" entry.

See the Bulk Data Entries (p. 1119) in the *MSC Nastran Quick Reference Guide*, for a detailed description of the Bulk Data.

Use of Parameters

Parameters are used extensively in the solution sequences for input of scalar values and for requesting special features. Parameters' values are specified on PARAM Bulk Data entries or PARAM Case Control commands.

- The PARAM Bulk Data entry is described in the Bulk Data Entries (p. 1119) in the *MSC Nastran Quick Reference Guide*.
- The PARAM Case Control command is described in the Case Control Command Descriptions (p. 193) in the *MSC Nastran Quick Reference Guide*.
- The manner in which MSC Nastran processes PARAMeters is described in general terms in the MSC Nastran DMAP Programmer's Guide.

The specification of PARAMeters differs between the rigid formats and the superelement solutions sequences and SOL 4. These differences require different user action in placing the PARAM entries in the MSC Nastran section, and in changing PARAMeter values on restarts.

Parameters in the Superelement Solution Sequences

The program module reads the Case Control Section for PARAM entries. If PARAM entries exist, the VPS table is updated with the values found in the Case Control Section. The PARAM values are updated at the beginning of the generation, assembly, reduction, and the data recovery loops of the superelement solution sequences. This allows the values of some user-supplied parameters to change for each superelement during the course of one run. In the nonlinear solution sequences the PARAM values are updated at the beginning of the iteration loop. This allows the values of some user-set parameters to vary between subcases. These parameters may appear in either the Case Control and/or Bulk Data Section. Other PARAMeters must appear in the Bulk Data Section. If they appear in the Case Control Section, they will not cause a preface error, but will be ignored. Other parameters may be reset to default values during the course of the run. This type should be set only in the subcase of the Case Control Section. See Table 5-1 and Table 5-2 in the , for parameter applicability.

The PARAM Case Control command is treated in the same manner as constraint and output commands. Parameters set above the subcase level are effective in every subcase. If there is also a parameter value set in the subcase, it takes precedence. Only one value of a parameter may be current at any time during the execution of the run. These rules imply an order of precedence if several forms are used in the same run. The rules are illustrated in the following example.

A partial subcase structure is listed on the following page. The subcases are listed in ascending superelement number order. PARAM,GRDPNT is varied between them.

Subcase Structure, Sorted by SEID

```
SUBCASE 1
SUPER = 1
...
SUBCASE 2
SUPER = 2
PARAM,GRDPNT,100
...
SUBCASE 3
SUPER = 3
PARAM,GRDPNT,200
...
SUBCASE 4
SUPER = 4
...
SUBCASE 100 $ RESIDUAL STRUCTURE
PARAM,GRDPNT,-1
...
BEGIN BULK
PARAM,GRDPNT,0
```

Processing Order	Value of GRDPNT
SEID	
1	0
2	100
3	200
4	0
0	–1

The Bulk Data entries are processed at the beginning of the run. Any PARAM entries in the Bulk Data Section set the value of that parameter until it is changed by a Case Control command. Superelement 1 is processed first. Since it has no PARAM,GRDPNT entry the value of 0, as set by the Bulk Data entry PARAM,GRDPNT,0 is used. The value for superelement 2 is set at 100 by a Case Control command. Superelement 4 is processed next. Its subcase contains no PARAM entry, so that the value of the parameter that remains is set by the bulk data. The values for superelement 3 and the residual structure are set by Case Control commands. An alternate method of parameter specification is to set the value most used above the subcase level, and exceptional values within the subcase. For example, if PARAM,GRDPNT,0 were placed above the subcase level and the values of 100, 200, and -1 placed in subcases 2, 3 and 100, respectively, the value used in each superelement will be the same as described above, regardless of processing order.

2 Grid Points and Coordinate Systems

- Introduction 36
- Grid Point and Coordinate System Definition 36
- Grid Point Properties 40

Introduction

Analysis model geometry in MSC Nastran is represented by grid point locations. The physical behavior of the analysis model is represented by the constraints and their associated coordinate system, and by the elements connecting the grid points (elements are described in Structural Elements, 41).

Grid Point and Coordinate System Definition

Geometric grid points are defined on GRID Bulk Data entries by specifying coordinates in either the basic coordinate system or in a local coordinate system. The implicitly defined basic coordinate system is rectangular, except when using axisymmetric elements. Local coordinate systems may be rectangular, cylindrical, or spherical and are related directly or indirectly to the basic coordinate system. Figure 2-1 shows the coordinate system definition.

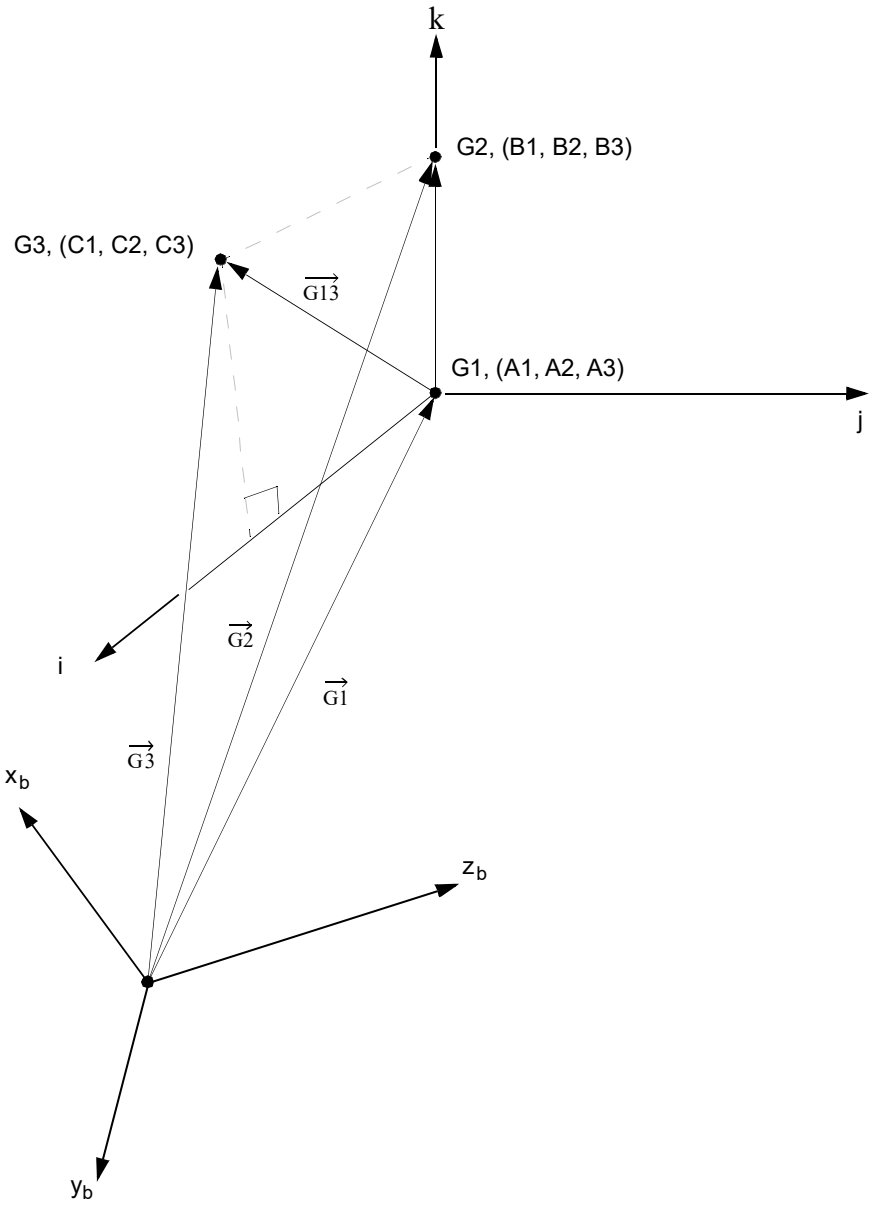


Figure 2-1 Coordinate System Definition

where:

(i, j, k) = user defined coordinate system (x, y, z) or (r, θ, z) or (r, θ, ϕ)

(x_b, y_b, z_b) = basic coordinate system

$\vec{k}$ = $(\vec{G2} - \vec{G1})$ normalized

$\vec{G13}$ = $(\vec{G3} - \vec{G1})$

$\vec{j}$ = $(\vec{k} \times \vec{G13})$ normalized

$\vec{i}$ = $(\vec{j} \times \vec{k})$

The CORD1C, CORD1R and CORD1S entries are used to define cylindrical, rectangular, and spherical local coordinate systems, respectively, in terms of three previously defined geometric grid points. The CORD2C, CORD2R and CORD2S entries are used to define cylindrical, rectangular, and spherical local coordinate systems, respectively, in terms of the coordinates of three points in a previously defined coordinate system.

Six rectangular displacement components (three translations and three rotations) are defined at each grid point. The displacement coordinate system, which is used to define the directions of motion, may be different from the "location coordinate system," which is used to locate the grid point. Both the location coordinate system and the displacement coordinate system are specified on the GRID entry for each geometric grid point. The orientation of displacement components depends on the type of local coordinate system used to define the displacement components. If the defining local system is rectangular, the displacement components are parallel to the local system and are independent of the grid point location as indicated in Figure 2-2(a). If the local system is cylindrical, the displacement components are in the radial, tangential, and axial directions as indicated in Figure 2-2(b). If the local system is spherical, the displacement components are in the radial, meridional, and azimuthal directions as indicated in Figure 2-2(c).

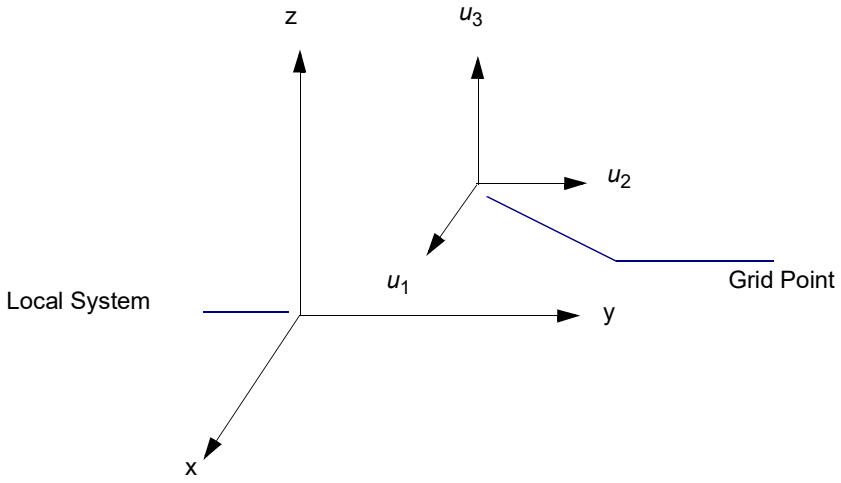
If the displacements of a grid point are defined to be in a spherical or cylindrical coordinate system and the grid point is located on, or relatively close to, the polar axis, then it will be replaced by the "defining" rectangular coordinate system. The z-axis of the "defining" rectangular coordinate system is coincident with the polar axis and the x-axis is coincident with $\theta = 0$ in cylindrical system, or $\theta = 0$ and $\phi = 0$ in the spherical system. It is recommended that a rectangular system is explicitly specified by the user.

Each geometric grid point may have a unique displacement coordinate system associated with it. The collection of all displacement coordinate systems is known as the *global coordinate system*. All matrices are formed and all displacements are output in the global coordinate system. The symbols T1, T2 and T3 on the printed output indicate translations in the 1, 2, and 3-directions, respectively, for each grid point. The symbols R1, R2, and R3 indicate rotations about the three axes.

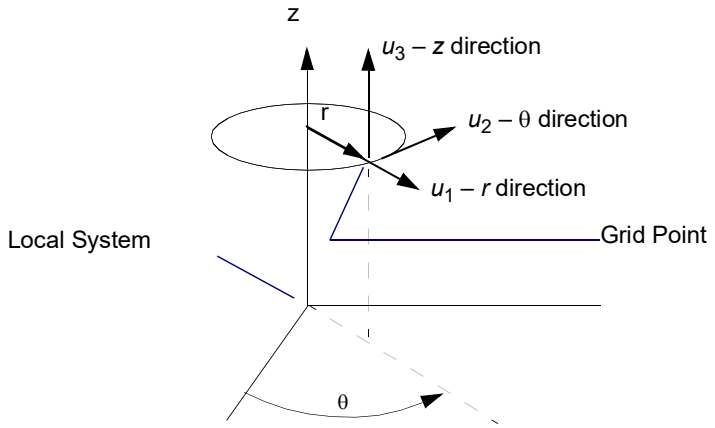
Provision is also made on the GRID entry to apply single-point constraints to any of the displacement components. Any constraints specified on the GRID entry will automatically be used for all solutions. Constraints specified on the GRID entry are usually restricted to those degrees of freedom that will not be elastically constrained and hence must be removed from the model in order to avoid singularities in the stiffness matrix.

Element offsets are defined by a Cartesian coordinate system, which is also parallel to the displacement coordinate system at the connecting grid point. However, the components of the offsets are always defined in units of translation, even if the displacement coordinate system is cylindrical or spherical.

(a) Rectangular



(b) Cylindrical



(c) Spherical

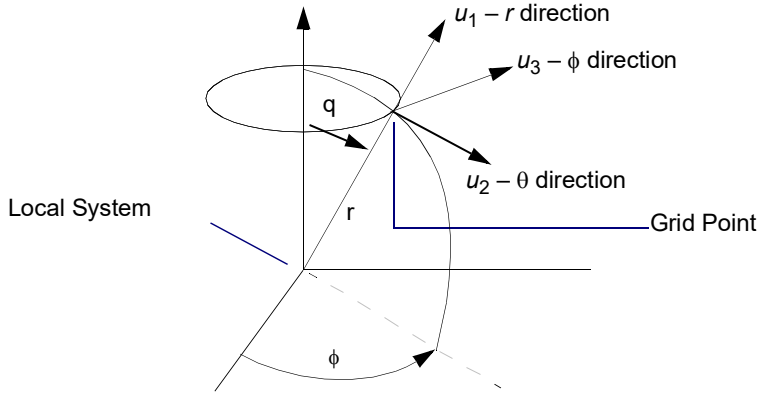


Figure 2-2 Displacement Components

The GRDSET Bulk Data entry is provided to avoid repeating the specification of location coordinate systems, displacement coordinate systems, and single point constraints when many (or all) of the GRID entries have the same entries for these items. When any of the three items are specified on the GRDSET entry, the entries are used to replace blank fields on the GRID entry for these items. This feature is useful in the case of such problems as space trusses where one wishes to remove all of the rotational degrees of freedom or in the case of plane structures where one wishes to remove all out-of-plane or all in-plane motions.

Scalar points are defined either on an SPOINT entry or by reference on a connection entry for a scalar element. SPOINT entries are used primarily to define scalar points appearing in constraint equations but to which no structural elements are connected. A scalar point is implicitly defined if it is used as a connection point for any scalar element. Special scalar points, called “extra points,” may be introduced for dynamic analyses. Extra points are used in connection with transfer functions and other forms of direct matrix input used in dynamic analyses and are defined on EPOINT entries.

GMCORD defines a convective/follower coordinate system on a FEEDGE, GMCURV, FEFACE, or GMSURF Bulk Data entry for use in adaptive (p-element) analysis.

Grid Point Properties

Some of the characteristics of the model may be specified as properties of grid points rather than as properties of elements.

Mass, damping, and stiffness properties may be provided, in part or entirely, as properties of grid points through the use of direct input matrices. The DMIG entry is used to define direct input matrices. These matrices may be associated with components of geometric grid points, scalar points or extra points introduced for dynamic analysis. The TF entry is used to define transfer functions that are internally converted to direct matrix input. The DMIAX entry is an alternate form of direct matrix input that is used for hydroelastic problems (see Coupled Fluid-Structure Interaction, 663).

Thermal fields are defined by specifying the temperatures at grid points. The TEMP entry is used to specify the temperature at grid points for use in connection with thermal loading and temperature-dependent material properties. The TEMPD entry is used to specify a default temperature in order to avoid a large number of duplicate entries on a TEMP entry when the temperature is uniform over a large portion of the structure. The TEMPAX entry is used for conical shell problems.

Mass properties may be input as properties of grid points by using the concentrated mass element. The CONM1 entry is used to define a 6×6 matrix of mass coefficients at a geometric grid point in any selected coordinate system. The CONM2 entry is used to define a concentrated mass at a geometric grid point in terms of its mass, the three coordinates of its center of gravity, the three moments of inertia about its center of gravity, and its three products of inertia, referred to any selected coordinate system.

The SNORM Bulk Data entry and/or the PARAM,SNORM assigns the singularity normal of a shell element to a degree of freedom. This singularity is then removed via AUTOSPC.

3 Structural Elements

- Element and Property Definition 42
- Line Elements 55
- Surface Elements 121
- Solid Elements (CTETRA, CPENTA, CHEXA) 151
- Rigid Elements and Multipoint Constraints (R-type, MPC) 155
- Scalar Elements (CELASi, CMASSi, CDAMPi) 174
- Axisymmetric Solid Elements (CTRIAX6, CTRIAX, CQUADX) 175
- Crack Tip Elements (CRAC2D, CRAC3D) 177
- Gap and Line Contact Elements (CGAP, BCONP, BLSEG) 184
- Concentrated Mass Elements (CONM1, CONM2) 190
- General Element Capability (GENEL) 190
- Direct Matrix Input on Grid Points (DMIG) 193
- p-Elements 193
- Hyperelastic Elements 197
- Bushing Elements 201
- Interface Elements 206
- Connector Elements 211
- Element Geometry Checks 245
- ELSUM Case Control Command 256

Element and Property Definition

Structural elements are defined on Bulk Data connection entries that identify the grid points to which the element is connected. The mnemonics for all such entries have a prefix of the letter “C”, followed by an indication of the type of element, such as CBAR and CROD. The order of the grid point identification defines the positive direction of the axis of a one-dimensional element and the positive surface of a plate element. The connection entries include additional orientation information when required. Some elements allow for offsets between its connecting grid points and the reference plane of the element. The coordinate systems associated with element offsets are defined in terms of the grid point coordinate systems described in Grid Points and Coordinate Systems, 35. For most elements, each connection entry references a property definition entry. If many elements have the same properties, this system of referencing eliminates a large number of duplicate entries.

The property definition Bulk Data entries define geometric properties such as thicknesses, cross-sectional areas, and moments of inertia. The mnemonics for all such entries have a prefix of the letter “P”, followed by some or all of the characters used on the associated connection entry, such as PBAR and PROD. Other included items are the nonstructural mass and the location of points where stresses will be calculated. For most elements, each property definition entry will reference a material property entry.

In some cases, the same finite element can be defined by using different Bulk Data entries. These alternate entries have been provided for user convenience. In the case of a rod element, the normal definition is accomplished with a connection entry (CROD) which references a property entry (PROD). However, an alternate definition uses a CONROD entry which combines connection and property information on a single entry. This is more convenient if a large number of rod elements all have different properties.

Most of the elements may be used with elements of other types within the limitations of good modeling practice. Exceptions are the axisymmetric elements, which are designed to be used by themselves. There are two types at present, linear and nonlinear. The linear are described in the following section. The nonlinear elements used in SOL 106 and SOL 129 are described under Hyperelastic Elements, 197. The conical shell element (“CCONEAX”) describes a thin shell by sweeping a line defined on a plane by two end points through a circular arc. Loads may be varied with azimuth angle through use of harmonic analysis techniques. This element has a unique set of input entries, which may be used with a limited set of other entries. These unique entries, if mixed with entries for other types of elements, cause a preface error. The advanced elements used in nonlinear solution, SOL 400, are described in the *MSC Nastran Nonlinear User’s Guide*, Chapter 11: Element Library. The elements used in explicit nonlinear solution, SOL 700, are described in the *MSC Nastran Explicit Nonlinear (SOL 700) User’s Guide*, Chapter 6: Elements.

Elements with Offsets

The CBAR, CBEAM, CBEAM3, CQUAD4, CQUAD8, CQUADR, CTRIA3, CTRIA6, and CTRIAR elements allow the user to apply offsets at the grid connections. The beam elements specify offset vectors at the connecting grids and the shell elements use a ZOFF from the surface of grid points to the element reference plane.

MSC Nastran provides the bulk data entry: **MDLPRM, OFFDEF, option**; for control of offset behavior. For legacy reasons, the default is **MDLPRM, OFFDEF, ELMOFF**, which imposes the following limitations:

- The differential stiffness is not supported. Therefore, they are not applicable in solution sequence where differential stiffness is required, such as linear buckling analysis (SOL 105).

- The effect of offsets is not included the mass matrix.
- The effect of offset is not included in computation of thermal loads, pressure loads, or gravity loads.
- For curved shell problems, the offset is defined in the z direction of the element and not the shell normal direction, this results in gaps between elements when offset geometry is applied.
- The transformation is linear and therefore, it cannot be used in nonlinear analysis.

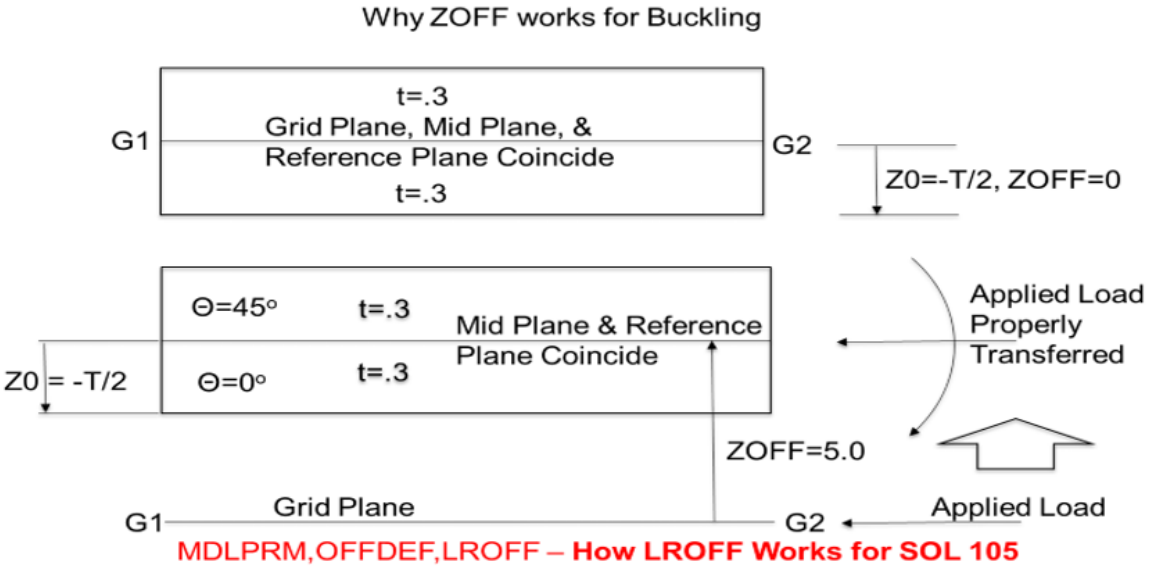
In general, the above limitations are removed if the recommended MDLPRM, OFFDEF, LROFF, is used. MDLPRM, OFFDEF, LROFF and its options are supported in all solution sequences with the exception of SOL106, SOL129, and SOL200.

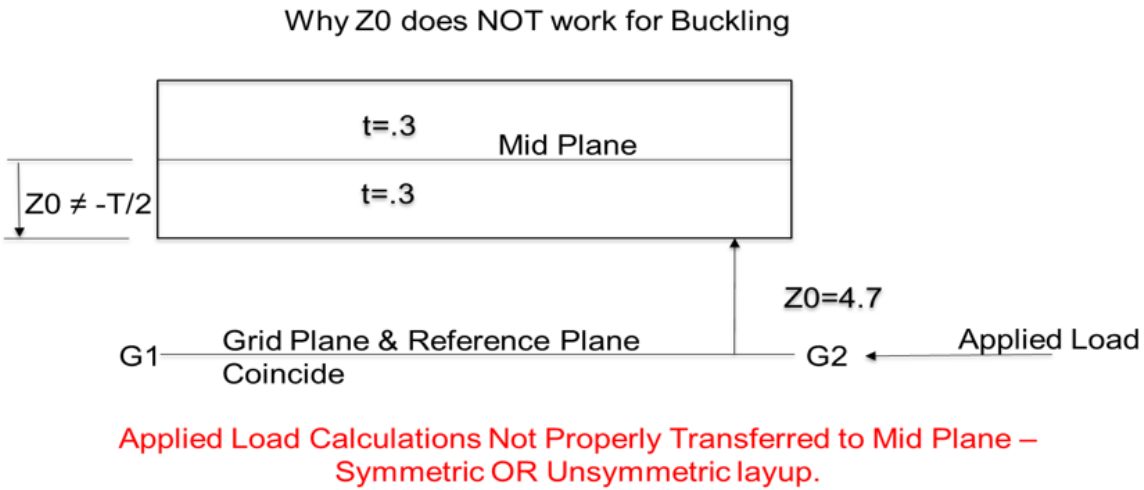
The enhanced offset method has the following benefits:

- DIFFERENTIAL STIFFNESS - The differential stiffness is computed for the offset so that it is applicable to solution sequences that required differential stiffness. These include, for example, SOL 103, SOL 105, and SOL 400.
- MASS - The effect of offset is included in the mass matrix generation.
- LOADS - The effect of offset is included in the load generation.
- GEOMETRY COMPATIBILITY - For QUAD4, TRIA3, QUADR, TRIAR, QUAD8 and TRIA6 elements the shell normal is used as the offset direction. Thus, the new offset will enhance solution of a model in two aspects: there is no gap in the offset geometry, and the mass and stiffness matrices are computed based on the offset geometry.
- NONLINEAR EFFECTS - The transformations are nonlinear so that it can be used in nonlinear analysis such as SOL 400

In SOL103, if the case control ADAMSMNF FLEXBODY=YES or AVLEXB EXBBODY=YES is present, then MDLPRM, OFFDEF, **LROFF must be replaced by MDLPRM, OFFDEF, NOMASS**.

For shell elements, there are two ways to account for an offset from the surface of grid points to the element reference plane. The ZOFF on the shell connection entry and Z0 on the PCOMP or PCOMPG entry. The **MDLPRM, OFFDEF, option** is only valid for the connection ZOFF method. Further, the Z0 composite option will in general lead to incorrect loading results for SOL105.





If there are offsets on shells or beams, MDLPRM, OFFDEF, LROFF should be used except in SOL106, SOL129, and SOL200. Remember, however, that SOL105, a linear buckling analysis, is an attempt to approximate a nonlinear instability, so it can be sensitive to stiffness and large rotation of offset vectors.

In SOL105, K6ROT must be thought of as a model imperfection for CQUAD4 and CTRIA3 elements. For these elements:

1. LROFF uses the shell normal at each grid
2. LROFF scales that normal by the element offset to get an offset vector
3. LROFF moves the mid plane to the offset position.(internal m-set, n-set, $G_{mn}{}^{T}\ K_{mm}\ G_{mn}$)
4. K6ROT applies a numerical drilling stiffness to this moved normal to form K_{gg}
5. For differential stiffness K_{dd}, LROFF does a 1-3 above
6. Then LROFF rotates that offset vector based on preload rotations
7. The differential stiffness matrix K_{dd} is based on this rotated offset vector

In SOL105, spurious modes often occur when rigid elements such as RBE2 or RBE3 are present because they generate Lagrange multipliers. When rigid elements are present **RIGID=LGELIM or LAGRAN** must be used in SOL101, SOL103 or SOL105 to account for rigid element rotation. In SOL105:

1. Spurious modes are not caused by LROFF. LROFF never produces Lagrange multipliers in any solution sequence.
2. Spurious modes may be caused by the rigid elements such as the RBE2 element which generates Lagrange multipliers.
3. In SOL105, RIGID=LAGRAN & LGELIM should get same result if for RIGID=LAGRAN the parameters LMFACT & PENFN are set properly.
4. LMFACT and PENFN are the scale factor and penalty function for the Lagrange rigid elements and the contact analysis.
5. The purpose of LMFACT and PENFN is to make the values of stiffness matrix of the Lagrange rigid elements and/or the contact components about the same relative magnitude as those of the other elements in the model.
6. The default value is 1.0E+5 for all solution sequences except SOL 400 and SOL 101 linear contact. In SOL 400 and SOL 101 linear contact, MSC Nastran will compute the appropriate default values for LMFACT and PENFN.
7. This computed SOL400 default is good in general. But when the material/element stiffness (e.g., Young's Modulus, Spring Stiffness) used in the model vary in a wide range, users may have to adjust the computed LMFACT and PENFN.
8. For some SOL105 models, reducing LMFACT and PENFN to 1.0E+4 allows the RIGID=LAGRAN to remove spurious modes. But "rotation" modes are still obtained.
9. Because of the above discussion, it is recommended in SOL105 if rigid elements are present that RIGID= LGELIM be used to avoid the pain of LMFACT and PENFN adjustment. (Note RIGID=LGELIM is not valid is SOL400).
10. It is recommended in SOL105 to start with a value of K6ROT=100. as a low value such as K6ROT=1. is in the noise of the solution. It may be necessary to increase the value of K6ROT > 100 in order to provide higher drilling stiffness to counter large rotation values and their impact on rigid elements and LROFF.
11. Models containing, RBE2 and RBE3 elements, BCONTACT surfaces and LRFOFF and may be extremely flexible. Thus rotations are large. To get control of the rotations for the CQUAD4/CTRIA3 elements K6ROT values as high as K6ROT=1800 may be required.
12. Always us the lowest possible converged K6ROT value, too high a value may be just as detrimental as to low a value.

13. To alleviate the K6ROT problem, NASTRAN QRMETH=5 may be tried. However, if spurious modes occur with this method, there is no way to add rotational stiffness such as K6ROT to correct them. Also, some of its spurious modes are difficult to detect.
14. In buckling analysis, always plot and examine the mode shapes.

Axisymmetric Elements

The axisymmetric elements CTRIAX6, CTRIAX, and CQUADX define a solid ring by sweeping a surface defined on a plane through a circular arc. Loads are constant with azimuth for these elements; that is, only the zeroth harmonic is considered. There may be innovative modeling techniques that allow coupling this class of axisymmetric element with other elements, but there are no features to provide correct automatic coupling.

Although the axisymmetric elements CCONEAX and CTRIAX6 provide useful functions, the user should be aware that their usage precludes the use of many features available for other elements (see Tables 3-1, 3-2, and 3-3). An alternative modeling technique is to use conventional elements with the cyclic symmetry solution sequences to provide equivalent models. This technique generally provides the features not available with axisymmetric elements. The CTRIAX and CQUADX axisymmetric elements are finite deformation, hyperelastic elements with fully nonlinear characteristics (see Hyperelastic Elements, 197).

Material Properties

The material property definition entries are used to define the properties for each of the materials used in the structural model. The MAT1 entry is used to define the properties for isotropic materials and may be referenced by any of the structural elements. The MATT1 entry specifies table references for isotropic material properties that are temperature dependent. The TABLEM1, TABLEM2, TABLEM3, and TABLEM4 entries define four different types of tabular functions for use in generating temperature dependent material properties. More information is given in Material Properties, 260.

The MAT2 entry is used to define the properties of anisotropic materials for triangular or quadrilateral membrane and bending elements. The MAT2 entry specifies the relationship between the in-plane stresses and strains. It may also be used for anisotropic transverse shear. The angle between the material coordinate system and the element coordinate system is specified on the connection entries. The MATT2 entry specifies table references for anisotropic material properties that are temperature dependent. This entry may reference any of the TABLEM1, TABLEM2, TABLEM3, or TABLEM4 entries.

The MAT3 entry is used to define the properties for orthotropic materials used in the modeling of axisymmetric shells. This entry may only be referenced by CTRlAX6 entries. The MATT3 entry specifies table references for use in generating temperature-dependent properties for this type of material. More information is given in Isotropic Material, 262.

The MAT8 entry is used to define the properties of orthotropic materials used in the modeling of quadrilateral and triangular shell elements for composite structures.

The MAT9 entry is used to define the properties of anisotropic materials for the CHEXA, CPENTA, and CTETRA elements. The MATT9 entry specifies table references for use in generating temperature-dependent properties for this type of material. More information is given in Orthotropic Material, 264.

The MATS1 entry specifies table references for isotropic material properties that are stress and/or temperature dependent. The TABLES1 entry defines a tabular stress-strain function for use in creep analysis (CREEP entry) or material nonlinear analysis. The TABLEST entry specifies TABLES1 references for nonlinear elastic temperature-dependent material properties. The MATS1 entry may be combined with a corresponding MAT1 to define the following types of nonlinear material analysis: (1) plastic (2) nonlinear elastic and (3) creep. More information is given in the *MSC Nastran Handbook for Nonlinear Analysis*.

The MATHP entry specifies hyperelastic material properties and must be referenced by one of the PLPLANE or PLSOLID property entries, specifying a finite deformation nonlinear analysis, including large strain and large rotation. Experimental data may be input on TABLES1 entries for automatic curve fitting of material parameters (see TABLES1 (Ch. A)).

In linear analysis, temperature-dependent material properties are computed once only at the beginning of the analysis. In nonlinear static analysis (SOLution 106), temperature-dependent material properties for linear (MAT1, MAT2, and MAT9 entries) and nonlinear elastic materials (MAT1 and MATS1 entries) may be updated many times during the analysis.

The advanced materials used in nonlinear solution, SOL 400, are described in the *MSC Nastran Nonlinear User's Guide*, Chapter 10: Materials. The elements used in explicit nonlinear solution, SOL 700, are described in the *MSC Nastran Explicit Nonlinear (SOL 700) User's Guide*, Chapter 6: Elements.

The General Element (GENEL)

The GENEL entry is used to define general elements whose properties are defined in terms of deflection influence coefficients or stiffness matrices which can be connected between any number of grid points. One of the important uses of the general element is the representation of part of a structure by means of experimentally measured data. No output data is prepared for the general element. Detailed information on the general element is given in General Element Capability (GENEL), 190.

Dummy, Mass, and Damping Elements

Dummy elements are provided in order to allow the user to investigate new structural elements with a minimum expenditure of time and money. A dummy element is defined with a CDUMi (i = index of element type, $1 \le i \le 9$) entry and its properties are defined with the PDUMi entry. The ADUMi entry is used to define the fields on the connection and property entries.

Mass elements have inertia properties, but do not have stiffness or damping. The CONM2 entry provides convenient input of mass attached to a grid point whereas the CONM1 entry provides a more general description of mass at a grid point.

Damping elements have damping properties only. The CVISC entry provides damping along a line between two grid points.

Constraint Elements

There are two classes of constraint elements. They generate constraint equations that allow elimination of some variables, rather than add coefficients to the stiffness matrix. However, they do have other element-like properties, such as the satisfaction of equilibrium conditions. The RBAR, RBE1, RBE2, RROD, RJOINT, RSSCON, and RTRPLT entries describe rigid elements. The RBE3 and RSPLINE entries describe interpolation elements. (See also Rigid Elements and Multipoint Constraints (R-type, MPC), 155).

Other Element Types

Other types of elements are described in different subsections of Additional Topics, 511. They include elements with fluid, acoustics, and nonlinear analysis capability. Heat transfer elements are described in the MSC Nastran Thermal Analysis User's Guide.

Details of the finite elements in MSC Nastran are given in the following sections. A summary of the finite elements and their characteristics is given in Table 3-1. An X in the table indicates the existence of an item.

Element identification numbers must be unique across all element types.

Table 3-1 Element Summary -- Conventional MSC Nastran Elements

Element Type	Structural Matrices								Materials			Static Load				Heat Transfer		
	Stiffness	Mass	Differential Stiffness	Geometric Nonlinear	Material Nonlinear	Viscous Damping	Axisymmetric	p-Adaptivity	Isotropic	Anisotropic	Orthotropic	Thermal	Pressure	Gravity	Element Deformation	Heat Conduction	Heat Capacity	Thermal Load
CAXIFi							X											
CBAR	X	LC	X						X			EB	X	X	X	X	X	X
CBEAM	X	LC	X	X	X			X	X			EB	X	X	X	X	X	X
CBEAM3	X	LC	X	X	X			X	X			EB	X	X	X			
CBUSH	FD				X	FD												
CBUSH1D	X	X	X	X	X				X									
CBEND	X	C	X						X			EB	X	X		X	X	X
CCONEAX	X	L					X		X	X	X	E	X	X				
CONMi		LC												X				
CONROD	CS	LC	X	X	X				X			E		X	X	X	X	X
CRAC2D	I	LC							X	X		E		X				
CRAC3D	I	LC							X	X		E		X				
CDAMPi						X											X	
CELASi	X								X							X		
CFAST	X	X							X									

Table 3-1 Element Summary -- Conventional MSC Nastran Elements (continued)

Element Type	Structural Matrices								Materials			Static Load				Heat Transfer		
	Stiffness	Mass	Differential Stiffness	Geometric Nonlinear	Material Nonlinear	Viscous Damping	Axisymmetric	p-Adaptivity	Isotropic	Anisotropic	Orthotropic	Thermal	Pressure	Gravity	Element Deformation	Heat Conduction	Heat Capacity	Thermal Load
CFLUIDi							X											
CGAP	X				X													
CHBDYi																X	X	X
CHEXA	I	LC	X	X*	X*			X	X	X		E	X	X		X	X	X
CMASSi		L												X				
CPENTA	I	LC	X	X*	X*			X	X	X		E	X	X		X	X	X
CQUAD4	I	LC	X	X	X			X	X	X	X	EB				X	X	X
CQUAD8	I	LC	X						X	X	X	EB	X	X		X	X	X
CQUADR	I	LC	X						X	X	X	EB	X	X		X	X	X
CROD	CS	LC	X	X	X				X			E		X	X	X	X	X
CSEAM	X	X							X									
CSHEAR	CS	L	X						X			E	X	X				
CSLOTi							X											
CTETRA	I	LC	X	X	X			X	X	X		E	X	X		X	X	X
CTRIA3	I	LC	X	X	X			X	X	X	X	EB	X	X		X	X	X
CTRIA6	I	LC	X						X	X	X	EB	X	X		X	X	X
CTRIAR	I	LC	X						X	X	X	EB	X	X		X	X	X
CTRIAX6	I	LC					X		X		X	E	X	X		X	X	X

Table 3-1 Element Summary -- Conventional MSC Nastran Elements (continued)

Element Type	Structural Matrices								Materials			Static Load				Heat Transfer		
	Stiffness	Mass	Differential Stiffness	Geometric Nonlinear	Material Nonlinear	Viscous Damping	Axisymmetric	p-Adaptivity	Isotropic	Anisotropic	Orthotropic	Thermal	Pressure	Gravity	Element Deformation	Heat Conduction	Heat Capacity	Thermal Load
CTUBE	CS	LC	X	X	X				X			E		X	X	X	X	X
CVISC						X												
CWELD	X	X							X									

†For the fully nonlinear hyperelastic elements, see Table 3-1.

*With the exception of hyperelastic elements, no midside grid points may be defined with the nonlinear stiffness formulation.

Table 3-2 Element Data Recovery Summary - Part 1

Element Type	Data Recovery								
	Stress Real*	Stress Complex*	Force Real*	Force Complex*	Force Sum Global	Force Sum Element Edges	Structure Plot	Contour Plot	Grid Point Stresses
CAXIFi	12	23							
CBAR	16	19	9	17	X	X	X		
CBEAM	89	111	89	177	X		X		
CBEND	23	23	17	31	X		X		
CBUSH	7	13	7	13	X				
CBUSH1D	8				X				

Table 3-2 Element Data Recovery Summary - Part 1 (continued)

	Data Recovery								
Element Type	Stress Real*	Stress Complex*	Force Real*	Force Complex*	Force Sum Global	Force Sum Element Edges	Structure Plot	Contour Plot	Grid Point Stresses
CCONEAX**	18		7				X		
CONMi									
CONROD	5	5	3	5	X	X	X		
CRAC2D	8						X		
CRAC3D	10								
CDAMPi			2	3					
CELASi	2	3	2	3	X				
CFLUIDi							X		
CGAP	8						X		
CHBDYP, CHBDYG							X		
CHEXA	193	121			X	X	X		X
CMASSi									
CPENTA	151	95			X	X	X		X
CQUAD4	17	15	9	17	X	X	X	X	X
CQUAD8	87	77	47	87	X		X		X
CQUADR	87	77	47	87	X		X		X
CROD	5	5	3	5	X	X	X		
CSHEAR	8	5	17	33	X	X	X	X	
CSLOTi	7	13					X		
CTETRA	109	69			X		X		X
CTRIA3	17	15	9	17	X	X	X	X	X
CTRIA6	70	62	38	70	X		X		X

Table 3-2 Element Data Recovery Summary - Part 1 (continued)

	Data Recovery								
Element Type	Stress Real*	Stress Complex*	Force Real*	Force Complex*	Force Sum Global	Force Sum Element Edges	Structure Plot	Contour Plot	Grid Point Stresses
CTRIAR	70	62	38	70	X	X	X		X
CTRIAX6	30	34			X		X	X	
CTUBE	5	5	3	5	X	X	X	X	
CVISC				5			X		
CWELD			9	17					

*The integers represent the number of output words per element, useful for storage requirement calculations.

**This requires the presence of an AXISYM Case Control command.

Table 3-3 Element Data Recovery Summary - Part 2

	Data Recovery									
Element Type	Strain Energy Density	Strain Energy	SORT2	MSGSTRESS	Heat Transfer	MSGMESH	Composite* Failure Indices	Composite* Stresses	Strain Real*	Strain Complex*
CAXlFi			X							
CBAR	X	X	X		X	X				
CBEAM	X	X	X		X	X				
CBEND	X	X	X	X	X					
CCONEAX**										
CBUSH		X	X							
CBUSH1D		X	X						8	
CONMi										
CONROD	X	X	X		X		X			
CRAC2D										
CRAC3D										
CDAMPi			X							
CELASi		X	X		X					
CFLUIDi							X			
CGAP		X	X							
CHBDYi			X		X	X				
CHEXA	X	X	X	X	X	X			193	121
CMASSi										
CPENTA	X	X	X	X	X	X			151	95
CQUAD4	X	X	X	X	X	X	9	11	17	15
CQUAD8	X	X	X		X	X	9	11	87	77

Table 3-3 Element Data Recovery Summary - Part 2

Element Type	Data Recovery									
	Strain Energy Density	Strain Energy	SORT2	MSGSTRESS	Heat Transfer	MSGMESH	Composite* Failure Indices	Composite* Stresses	Strain Real*	Strain Complex*
CQUADR	X	X	X		X	X	9	11	87	77
CROD	X	X	X		X	X				
CSHEAR	X	X	X			X				
CSLOTi			X							
CTETRA	X	X	X		X				109	69
CTRIA3	X	X	X	X	X	X	9	11	17	15
CTRIA6	X	X	X		X	X	9	11	70	62
CTRIAR	X	X	X		X	X	9	11	70	62
CTRIAX6	X	X	X	X	X	X				
CTUBE	X	X	X		X	X				
CVISC										
CWELD	X	X	X							

*The integers represent the number of output words per element, useful for storage requirement calculations.

**This requires the presence of an AXISYM Case Control command.

Stiffness Matrix	Mass Matrix
CS -- Constant strain element	L -- Lumped mass only
LS -- Linear strain element	C -- Coupled mass only
I -- Modified isoparametric element	LC -- Lumped mass or coupled mass
FD -- Frequency dependent	

Thermal Load
E -- Extension load only
EB -- Both extension and bending load

Integer values indicate number of items output.

Line Elements

The line elements consist of the following:

- CBEAM
- CBEAM3
- CBAR
- CBEND
- CROD
- CONROD
- CTUBE

Beam Element (CBEAM)

The beam element is defined with a `CBEAM` entry and its properties are defined with a PBEAM, PBCOMP, or PBEAML entry. The beam element includes extension, torsion, bending in two perpendicular planes, the associated shears, and the features listed below:

1. Distributed mass polar moment of inertia.
2. Separate shear center, neutral axis, and nonstructural mass center of gravity.
3. Arbitrary variation of the section properties (A, I1, 12, I12, J) and of the nonstructural mass (NSM) along the beam (PBEAM only).
4. Shear relief due to taper (PBEAM only).
5. The ability to apply either concentrated or distributed loads along the beam, using the `PLOAD1` entry.
6. The effect of cross-sectional warping on torsional stiffness (PBEAM only).
7. The ability to model a beam made up of offset rods, using the PBCOMP entry.
8. Nonlinear material properties: elastic-perfectly plastic only (see TYPE = PLASTIC on MATS1 entry).

Any five of the six forces at either end of the element may be set equal to zero by using the pin flags on the `CBEAM` entry. The integers 1 to 6 represent the axial force, shearing force in Plane 1, shearing force in Plane 2, axial torque, moment in Plane 2 and moment in Plane 1, respectively. The structural and nonstructural mass of the beam are lumped at the ends of

the elements, unless coupled mass is requested with the PARAM entry, COUPMASS (see Parameters (p. 785) in the *MSC Nastran Quick Reference Guide*).

The beam element coordinate system and nomenclature are shown in Figure 3-1. End A is offset from grid point GA an amount measured by vector $\vec{w}_a$, and end B is offset from grid point GB an amount measured by vector $\vec{w}_b$. The vectors $\vec{w}_a$ and $\vec{w}_b$ are measured in the global coordinates of the connected grid point. The *x*-axis of the element coordinate system is defined by a line connecting the shear center of end A to that at end B of the beam element. The orientation of the beam element is described in terms of two reference planes. The reference planes are defined with the aid of vector $\vec{v}$. This vector may be defined directly with three components in the global system at end A of the beam, or by a line drawn from end A parallel to the line from GA to a third referenced grid point. The first reference plane (Plane 1) is defined by the *x*-axis and the vector $\vec{v}$. The second reference plane (Plane 2) is defined by the vector cross product $\vec{x} \times \vec{v}$ and the *x*-axis. The subscripts 1 and 2 refer to forces and geometric properties associated with bending in Planes 1 and 2, respectively. The reference planes are not necessarily principal planes. The coincidence of the reference planes and the principal planes is indicated by a zero product of inertia (I_{12}) on the PBEAM entry. When pin flags and offsets are used, the effect of the pin is to free the force at the end of the element *x*-axis of the beam, not at the grid point.

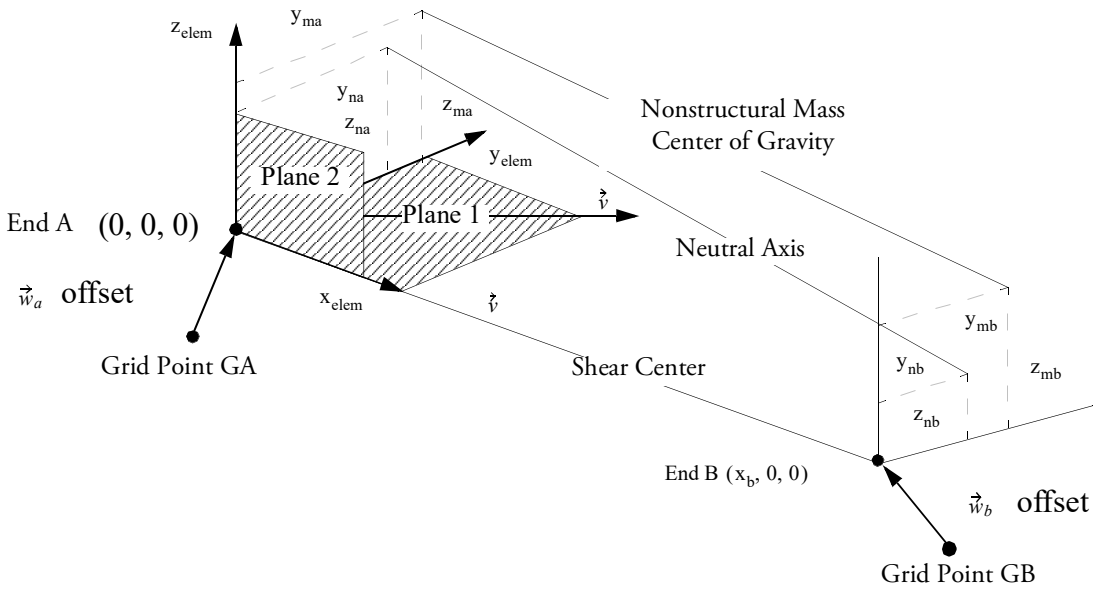


Figure 3-1 CBEAM Element Coordinate System

The positive directions for element forces are shown in Figure 3-2. The following element forces, either real or complex (depending on the solution sequence), are output on request at both ends and at intermediate locations defined on the PBEAM entry:

Beam element internal forces and moments

- Bending moments in the two reference planes at the neutral axis.
- Shear forces in the two reference planes at the shear center.
- Axial force at the neutral axis.
- Total torque about the beam shear center axis.
- Component of torque due to warping.

The following real element stress data are output on request:

- Real longitudinal stress at the four points prescribed for each cross section defined along the length of the beam on the PBEAM entry.
- Maximum and minimum longitudinal stresses.
- Margins of safety in tension and compression for the element if the user enters stress limits on the MAT1 entry.

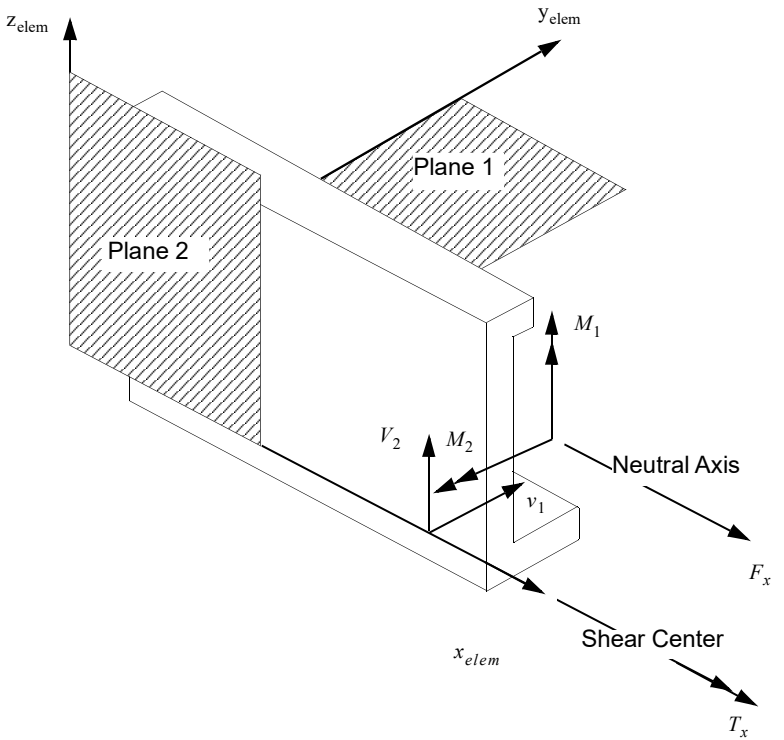
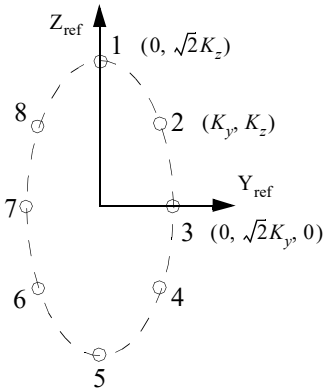


Figure 3-2 CBEAM Element Internal Forces and Moments

Tensile stresses are given a positive sign and compressive stresses a negative sign. Only the longitudinal stresses are available as complex stresses. The stress recovery coefficients on the PBEAM entry are used to locate points on the cross section for stress recovery. The subscript 1 is associated with the distance of a stress recovery point from Plane 2. The subscript 2 is

associated with the distance from Plane 1. Note that if zero value stress recovery coefficients are used, the axial stress is output.

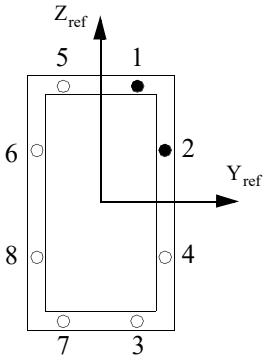
The PBCOMP entry allows the input of offset rods to define the beam's section properties. A program automatically converts the data to an equivalent PBEAM entry. The input options that allow efficient descriptions of various symmetric cross sections are shown in Figure 3-3.



SECTION = 0 (default)
Symmetric about Y_{ref} and Z_{ref}

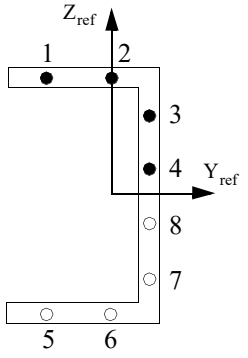
$$K_y = \sqrt{\frac{I_{zz}}{A}},\ K_z = \sqrt{\frac{I_{yy}}{A}},\ C1 = \frac{1}{8}$$

I_{zz} - moment of inertia about z-axis
I_{yy} - moment of inertia about y-axis



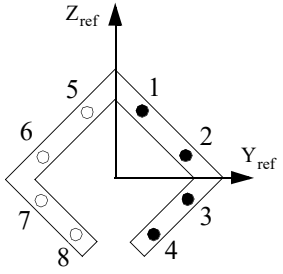
SECTION = 1
(w/continuation entry)
Symmetric about Y_{ref} and Z_{ref}

Y1 = Y3 = -Y5 = -Y7
Z1 = -Z3 = Z5 = -Z7, etc.



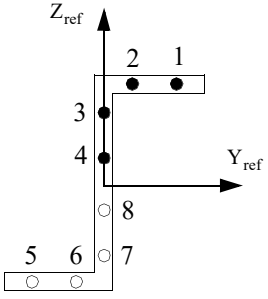
SECTION = 2
Symmetric about Y_{ref}

Y1 = Y5
Z1 = -Z5, etc.



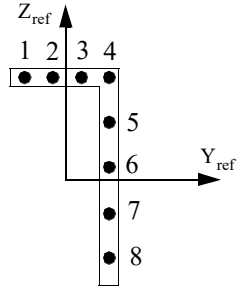
SECTION = 3
Symmetric about Z_{ref}

Y1 = -Y5, Z1 = Z5, etc.



SECTION = 4
Mirror Symmetry about Y_{ref} and Z_{ref}

Y1 = -Y5, Z1 = -Z5, etc.



SECTION = 5
No symmetry

Figure Notes:

1. Integration points (lumped area) are numbered 1 through 8.
2. User-specified points are denoted by ● and the program default point denoted by ○.

Figure 3-3 PBCOMP Entry SECTION Types

Element Properties on the PBEAM Entry

The cross-sectional properties of the element which may be defined by the user and the manner in which these properties are interpolated along the x-axis of the element are listed in the following table:

Quantity	Definition	Locations at Which User Specifies Properties	Method of Interpolation
A	Cross-sectional Area	Ends, Interior points	Linear between points
I1,I2,I12	Moments and product of inertia about neutral axis for planes 1 and 2	Ends, Interior points	Linear between points
J	Torsional stiffness parameter	Ends, Interior points	Linear between points
NSM	Nonstructural mass per unit length	Ends, Interior points	Linear between points
K1,K2	Shear factors in (K)AG for planes 1 and 2	One value	Constant
S1,S2	Shear relief factors for planes 1 and 2	One Value	Constant
CW	Warping stiffness parameter	Ends	Linear between ends
NSI	Nonstructural mass moment of inertia per unit length	Ends	Linear between ends

Mass Matrix

The inertia properties of the CBEAM element include the following terms:

- Structural mass per unit length, $\text{RHO} \cdot \text{A}$, on the neutral axis.
- Nonstructural mass per unit length, NSM.
- Moment of inertia of structural mass per unit length $\text{RHO} \cdot (\text{I1} + \text{I2})$, about neutral axis.
- Moment of inertia of nonstructural mass, NSI.

where RHO is the density defined on the MAT1 entry.

Cross Sectional Warping

Open section members such as channels will undergo torsion as well as bending when transverse loads act anywhere except at the shear center of a cross-section. This torsion produces warping of the cross-section so that plane sections do not remain plane and, as a result, axial stresses are produced. This situation can be represented in the following differential equation for the torsion of a beam about the axis of shear centers:

$$G\frac{d}{dx}\left(J\frac{d\theta}{dx}\right) - E\frac{d^2}{dx^2}\left(C_w\frac{d^2\theta}{dx^2}\right) = m \tag{3-1}$$

E	=	Young's modulus of elasticity
C_w	=	Warping constant
G	=	Shear modulus
J	=	Torsion constant
θ	=	Angle of rotation at any cross-section
m	=	Applied torsional moment per unit length

Note that C_w, the warping constant, has units of (length)^6. The development of the above differential equation and methods for the numerical evaluations of the warping constant are available in the literature. (See, for example, Timoshenko and Gere, *Theory of Elastic Stability,* McGraw Hill Book Company, 1961.) An example that demonstrates the use of the warping constant is the section titled "Example Problem of Channel Section".

Shear Relief

The shear relief factor accounts for the fact that in a tapered flanged beam the flanges sustain a portion of the transverse shear load. This situation is illustrated in Figure 3-4:

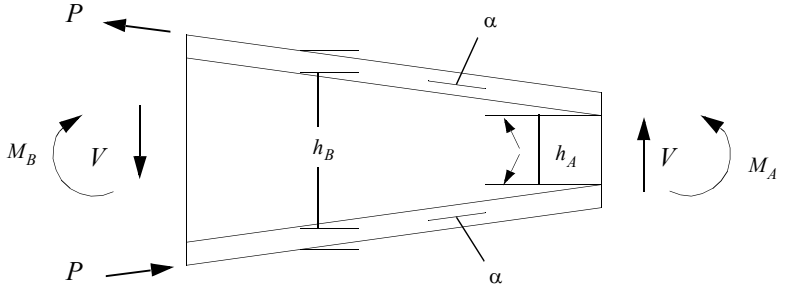


Figure 3-4

Here, the net transverse shear, Q, at the cross-section of depth h_B is

$$Q = V - 2P\sin\alpha \tag{3-2}$$

and, if the entire bending moment is carried by the flanges,

$$Q = V\frac{2\tan\alpha}{h_r}M_B \tag{3-3}$$

Equation (3-3) is represented in MSC Nastran as:

$$Q_1 = V_1 - \frac{S_1}{l}M_1 \tag{3-4}$$

$$Q_2 = V_2 - \frac{S_2}{l}M_2$$

where l is the length of the CBEAM element in question and the subscripts refer to plane 1 and plane 2, respectively. The terms S_1 and S_2 are denoted as the shear relief coefficients. The value of the shear coefficient for a tapered beam with heavy flanges that sustain the entire moment load may then be written as:

$$S_1 = \frac{2(h_A - h_B)}{(h_A + h_B)}$$

where:

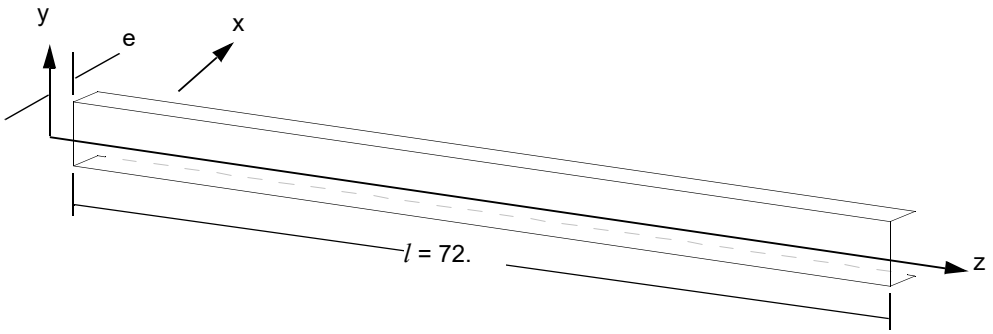
h_A = depth at end A of CBEAM element (GA on CBEAM entry)
h_B = depth at end B of CBEAM element (GB on CBEAM entry)

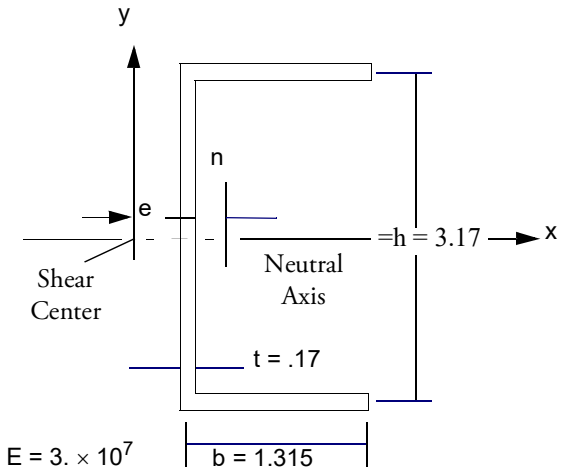
Example Problem of Channel Section

The simply supported beam illustrated in Figure 3-5 is modeled with five CBEAM elements. Since the beam is an open section channel with a single plane of symmetry, buckling failure can occur either through a combination of torsion and bending about the element x-axis or the lateral bending about the y-axis. The effect of cross-sectional warping coefficients, CW(A) and CW(B), on the PBEAM entry are necessary to capture these effects. Since the column is of uniform cross-section only the PBEAM entry illustrated below is required.

1	2	3	4	5	6	7	8	9	10
PBEAM	1	1	.986	1.578465	.1720965		.0094985		
	NO	1.							
							.3010320	.3010320	
						.7659450		.769450	

Warping requires seven degrees of freedom for the beam element. Thus, for warping, each grid must have an associated scalar point.





Section Properties

$$e = \frac{3b^2}{6b+h} = .4690484$$

$$J = \frac{t^3}{3}(2b+h) = .0094985$$

$$C_w = \frac{tb^3h^2}{12}\frac{3b+2h}{6b+h} = .3010320$$

$$A = .986$$

$$n = .2968966$$

$$I_x = 1.579465$$

$$I_y = .1720965$$

$$X_o = (e+n) = .7659450$$

Figure 3-5 Simply-Supported Channel

Example of Tapered Beam

The cantilevered tapered flanged beam illustrated in Figure 3-6 is utilized to demonstrate the use of shear relief factors, S1 and S2, with the CBEAM element. As noted earlier in the section, shear relief provides for the fact that in a tapered flanged beam some of the transverse shear load is carried by the flanges. All of the bending load is carried by the flanges. The shear relief coefficient for the tip cell is computed by the following formula:

$$S_1 = \frac{2(h_A - h_B)}{(h_A + h_B)} = \frac{2(10. - 20.)}{(10. + 20.)} = -.66667$$

The PBEAM entry for this tip cell is illustrated below:

1	2	3	4	5	6	7	8	9	10
PBEAM	1	1	1.	60.	1.				
	5.		-5.						
	YES	1.	2.	240.					
	10.		-10.						
			-.666667						

Note: A unique PBEAM entry is required for each CBEAM element as the shear relief factor among other properties vary from element to element. Also note that the default values of 1. are accepted for the shear stiffness factors.

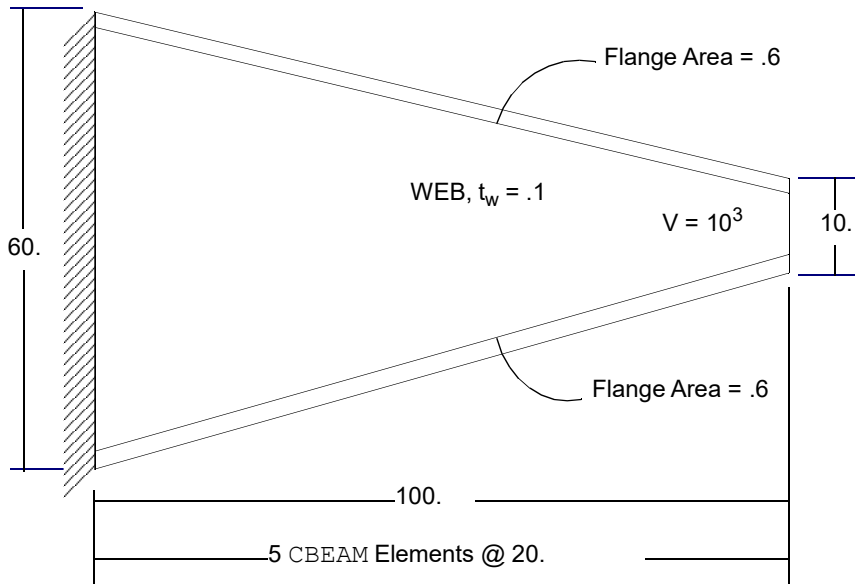


Figure 3-6 Tapered Cantilevered Beam With Shear Relief

Three-Node Beam Element (CBEAM3)

The Three-Node Beam Element is defined with a CBEAM3 entry and its properties are defined with a PBEAM3 entry. The CBEAM3 element is described in the MSC Nastran Implicit Nonlinear User's Guide. See the *MSC Nastran Linear Static Analysis User's Guide*, CBEAM3 Three-Node Beam Element. The material presented here will only be formulations and equations not found in the other guides.

Formulation

Strain-Displacement Relations

The formulation of a three-node beam element is based on Timoshenko beam theory. The kinematical assumption hereafter is that the displacement vector at an arbitrary point of the beam cross section is uniquely defined by the displacement vector at the beam reference axis and the rotations of the cross section, superposed by a longitudinal warping displacement. For small deformation theory, the rotation of the cross section is a vector. The displacement vector at point (s, y, z) of a beam cross section is

$$\boldsymbol{u}(s, y, z) = U_x \boldsymbol{e}_x + U_y \boldsymbol{e}_y + U_z \boldsymbol{e}_z$$

where:

$$U_x(s, y, z) = u(s) + z\theta_y(s) - y\theta_z(s) + \omega(s)\psi(y, z)$$
$$U_y(s, y, z) = v(s) - z\theta_x(s)$$
$$U_z(s, y, z) = w(s) + y\theta_x(s)$$

in which $u(s)$, $v(s)$ and $w(s)$ are three translative displacement components at the beam axis; $\theta_x(s)$, $\theta_y(s)$ and $\theta_z(s)$ are three rotational degrees of freedom of the beam cross section; and $\psi(y, z)$ is a given warping function and $\omega(s)$ the warping degree of freedom. The assumption of the warping displacement is based on the Saint-Venant torsion theory of straight bars.

The Green-Lagrangian normal and shear strains, $\{\varepsilon, \gamma_y, \gamma_z\}$, are

$$\varepsilon = \varepsilon_0 + z\chi_y - y\chi_z + \psi\omega_s$$
$$\gamma_y = \gamma_{y0} - z\chi_x + \bar{\psi}_y\omega$$
$$\gamma_z = \gamma_{z0} + y\chi_x + \bar{\psi}_z\omega$$

where:

$$\varepsilon_0 = u_{,s} + \kappa_y w - \kappa_z v$$
$$\gamma_{y0} = v_{,s} - \theta_z + \kappa_z u - \kappa_x w$$
$$\gamma_{z0} = w_{,s} + \theta_y - \kappa_y u + \kappa_x v$$
$$\chi_x = \theta_{x,s} - \kappa_z\theta_y + \kappa_y\theta_z$$
$$\chi_y = \theta_{y,s} + \kappa_z\theta_x$$
$$\chi_z = \theta_{z,s} - \kappa_y\theta_x$$
$$\bar{\psi}_y = \frac{\partial\psi}{\partial y} + \kappa_z\psi$$
$$\bar{\psi}_z = \frac{\partial\psi}{\partial z} - \kappa_y\psi$$

in which $(\,)_{,s} = \partial(\,)/(\partial s)$, the derivative with regard to the arc-length of beam axis.

Constitutive Relation

The general stress-strain relations are given, including thermal loads, at an arbitrary point of a beam cross-section, as

$$\{\sigma\} = \begin{Bmatrix} \sigma \\ \tau_y \\ \tau_z \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ & C_{22} & C_{23} \\ Sym & & C_{33} \end{bmatrix} \left(\begin{Bmatrix} \varepsilon \\ \gamma_y \\ \gamma_z \end{Bmatrix} - \begin{Bmatrix} \alpha \Delta T \\ 0 \\ 0 \end{Bmatrix} \right)$$

where C_{ij} are material elastic constants, α the thermal expansion coefficient and ΔT the variation of temperature.

Element Forces and Moments

The axial and shear forces as well as torsion and bending moments of the cross-section in the local coordinate system are

$$N_x = \int_A \sigma dA$$

$$V_y = \int_A \tau_y dA$$

$$V_z = \int_A \tau_z dA$$

$$M_x = \int_A (y\tau_z - z\tau_y) dA$$

$$M_y = \int_A \sigma z dA$$

$$M_z = \int_A (-y\sigma) dA$$

and the bi-shear force and bi-moment, which are associated with the warping degree of freedom and its derivative, are

$$V_b = \int_A (\tau_y \bar{\psi}_y + \tau_z \bar{\psi}_z) dA$$

$$M_b = \int_A \sigma \psi dA$$

Bar Element (CBAR)

The bar element is defined with a CBAR entry and its properties are defined with a PBAR entry. The bar element is a one-dimensional bending element which is prismatic, and for which the elastic axis, gravity axis, and shear center all coincide. The use of pin flags and the orientation of the bar element are similar to the beam element as shown in Figure 3-7. If shearing deformations are included in the bar element, the reference axes (Planes 1 and 2) and the principal axes must coincide.

The positive directions for element forces are shown in Figure 3-7. The following element forces, either real or complex (depending on the rigid format), are output on request:

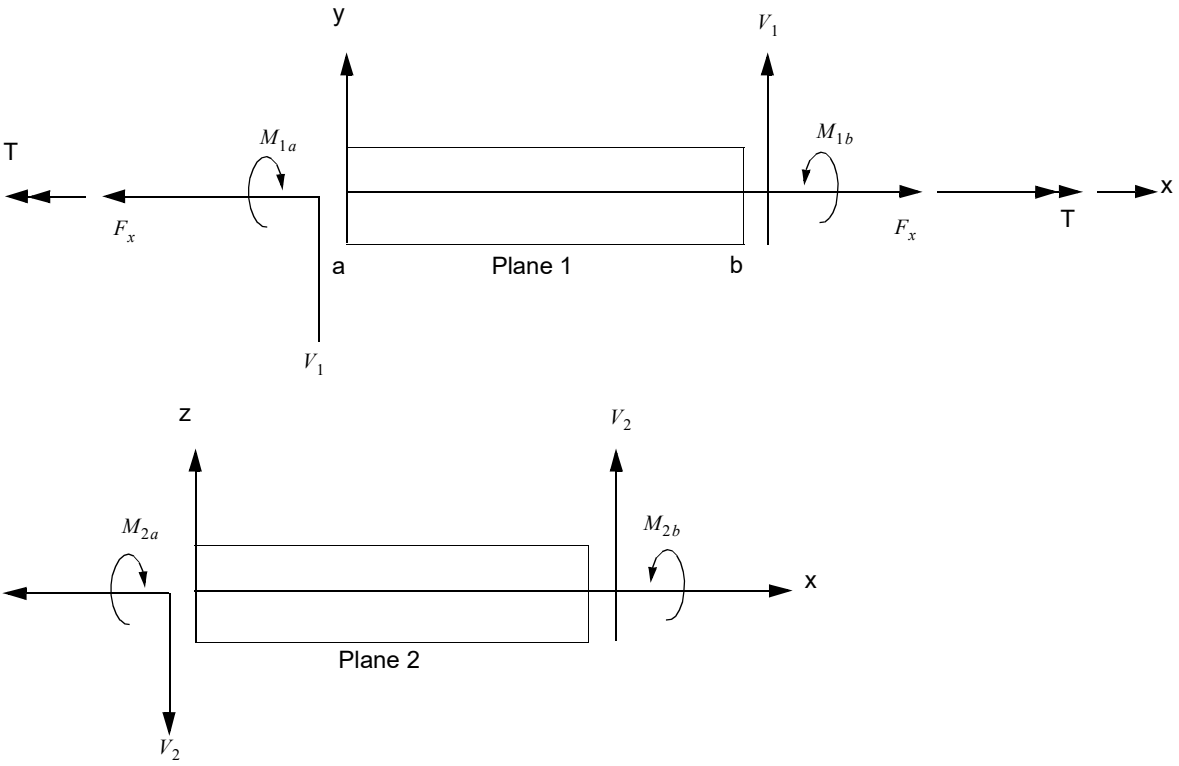


Figure 3-7 CBAR Element Forces

where M_{1a}, M_{1b}, M_{2a}, and M_{2b} are the bending moments at both ends in the two reference planes, V_1 and V_2 are the shear forces in the two reference planes, F_x is the average axial force, and T is the torque about the x-axis.

The following real element stresses are output on request:

- Average axial stress.
- Extensional stress due to bending at four points on the cross section at both ends. (Optional, calculated only if the user enters stress recovery points on the PBAR entry.)
- Maximum and minimum extensional stresses at both ends.
- Margins of safety in tension and compression for the whole element. (Optional, calculated only if the user enters stress limits on the MAT1 entry.)

Tensile stresses are given a positive sign and compressive stresses a negative sign. Only the average axial stress and the extensional stresses due to bending are available as complex stresses. The stress recovery coefficients on the PBAR entry are used to locate points on the cross section for stress recovery. The subscript 1 is associated with the distance of a stress recovery point from Plane 2. The subscript 2 is associated with the distance from Plane 1.

CBAR element force and stress data recovery with distributed loads (PLOAD1) and distributed mass (coupled mass) effects included may be obtained at intermediate as well as end points from the dynamic solution sequences. The following items must be in the input data:

- A LOADSET in Case Control which selects an LSEQ entry referencing PLOAD1 entries in the Bulk Data Section.
- Use PARAM,COUPMASS to select the coupled mass option for all elements.

The use of the BAROR Bulk Data entry avoids unnecessary repetition of input when a large number of bar elements either have the same property identification number or have their reference axes oriented in the same manner. BAROR defines default values on the CBAR entry for the property identification number and the orientation vector for the reference axes. The default values are used only when the corresponding fields on the CBAR entry are blank.

Bend Element (CBEND)

The bend element is defined with a CBEND entry and its properties are defined with a PBEND entry. The BEND element is a one-dimensional bending element with a constant radius of curvature. The CBEND element may be used to analyze either curved beams or pipe elbows. The bend element includes extension, torsion, bending in two perpendicular planes, and the associated transverse shear. The bend element cross section properties are assumed to be constant along the length of the element and the principal axes coincide with the element coordinates shown in Figure 3-8. Structural and nonstructural mass for the bend element is obtained only as consistent mass. The lumped mass option is not available.

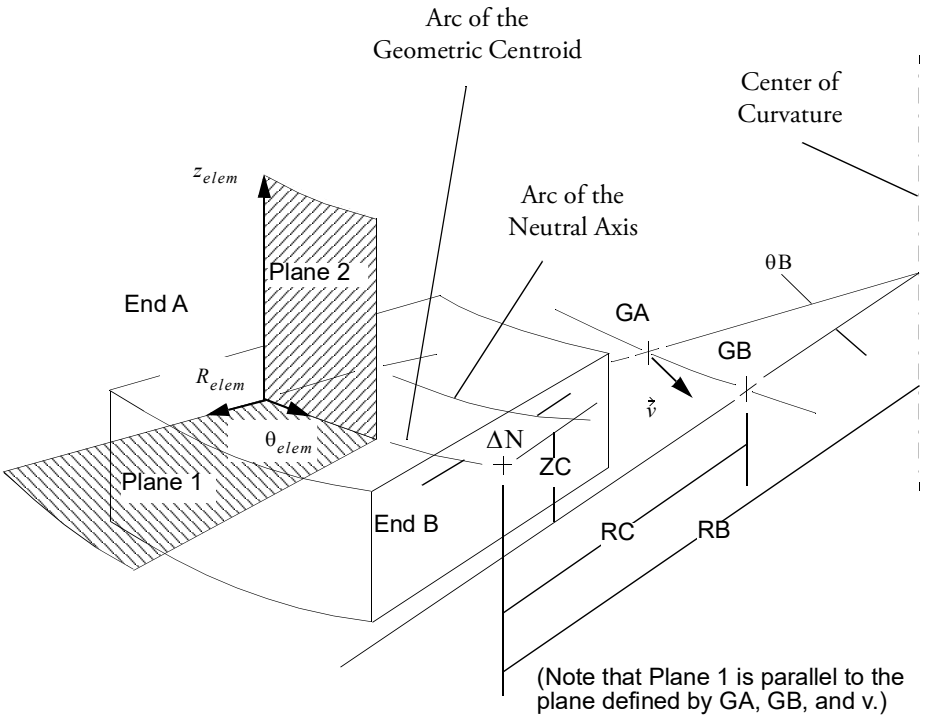


Figure 3-8 CBEND Element Coordinate System

Offsets of the ends of the element from the grid points are the same at both ends. The offsets are measured in the element coordinate system as shown in Figure 3-8. The element coordinate system is defined by one of four methods that the user specifies in the GEOM field on the CBEND entry. The *Z*-direction of the element coordinate system is defined by the cross product $\vec{v}_{ab} \times \vec{v}$ of the vector $\vec{v}_{ab}$ connecting grid point GA to grid point GB and the vector $\vec{v}$ for GEOM = 1. For GEOM = 2, 3, or 4, the *Z*-direction is defined by the cross product $\vec{v} \times \vec{v}_{ab}$. The center of curvature and intersection of the tangent lines from end A and end B are located using the data required for each of the four options. The *R*-direction is obtained by the vector extending from the center of curvature to end *A*. The θ-direction is the cross product of $\vec{Z} \times \vec{R}$. When $\theta = 0$, end *A* of the element is indicated and $\theta = \theta_B$ represents end B. Plane 1 of the element lies in the $R\theta$ plane of the element coordinates. Plane 1 is parallel to the plane defined by GA, GB and the vector $\vec{v}$, but it is offset by ZC in the *Z*-direction. Plane 2 lies in the θZ plane and is offset from GA and GB by RC in the *R*-direction. The subscripts 1 and 2 refer to forces and geometric properties associated with bending in Planes 1 and 2, respectively. These reference planes are the principal planes of the element cross section.

The neutral axis radial offset shown in Figure 3-8 from the geometric centroid due to bending of a curved beam with a constant radius of curvature is defined as follows:

$$\Delta N = \frac{R_B}{1 + \frac{A R_R^2}{Z}} \tag{3-5}$$

where:

R_B = the bend radius
A = the cross section area
Z = $\int \frac{r^2 dA}{1 + \frac{R}{R_B}}$
r = a local variable aligned with R_{elem} direction

The user can use the default provided with the general format or he can calculate and input a value using the above formula. For the circular section format, the neutral axis offset is automatically calculated with analytical expressions for hollow and solid circular cross-sectional elements.

The flexibility factors which multiply the bending terms of the flexibility matrix are selected by the FSI field on the hollow circular section format of the property entry. The options available are

for FSI = 1:

$k_p = 1.0$

for FSI = 2:

$$k_p = \frac{1.65r^2}{R_B t}\left[\frac{1}{1 + 6\frac{Pr}{Et}\left(\frac{r}{t}\right)^{4/3}\left(\frac{R_B}{r}\right)^{1/3}}\right] \quad (3\text{-}6)$$

for FSI = 3:

$$k_p = \frac{1.66}{R_B t}\left[\frac{1}{1 + 1.75\lambda^{4/3}\, exp(-1.5\Psi^{-1/4})}\right] \quad (3\text{-}7)$$

where:

λ =

Ψ = $\frac{PR_B^2}{Ert}$

r = mean cross-sectional radius

t = wall thickness

P = internal pressure

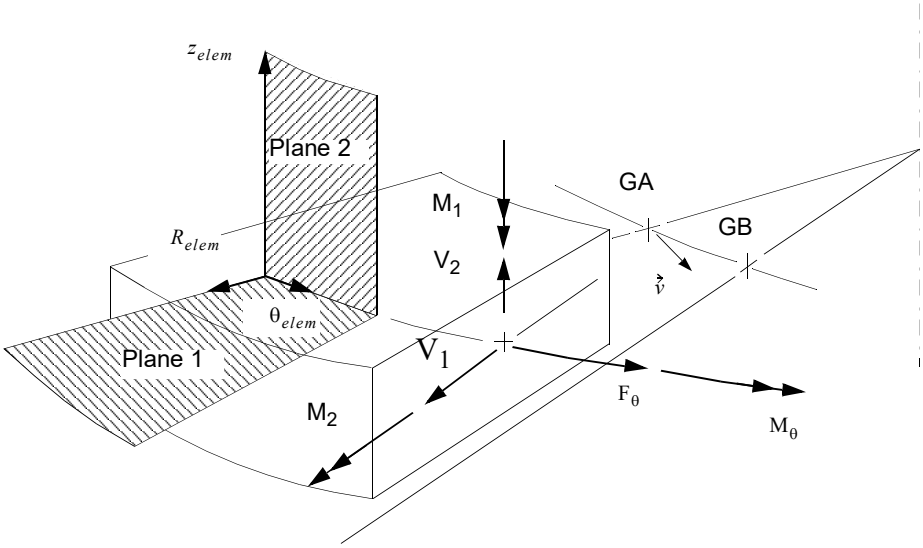
E = Young's modulus

ν = Poisson's ratio

The positive sign conventions for internal element forces are shown in Figure 3-9. The following element forces, either real or complex (depending on the rigid format), are output on request at both ends:

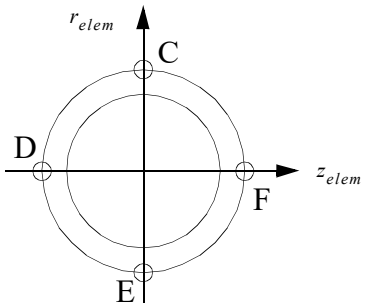
- Bending moments in the two reference planes, M_1 and M_2.
- Shears in the two reference planes, V_1 and V_2.
- Average axial force, F_θ.
- Torque about the bend axis, M_θ.

Figure 3-9 CBEND Element Internal Forces and Moments


z_{elem}
Plane 2
R_{elem}
θ_{elem}
Plane 1
M_1
V_2
V_1
M_2
GA
GB
$\vec{v}$
F_θ
M_θ

The following real element stress data are output on request:

- Real longitudinal stress at the four points which are the same at both ends for the general cross-sectional property entry format. If the circular cross-sectional property and format is used, the stress points are automatically located at the points indicated on the following figure.



- Maximum and minimum longitudinal stresses.
- Margins of safety in tension and compression for the element if the user enters stress limits on the MAT1 entry.
- Whenever the pipe format is used, the stress data are modified to account for stress intensification resulting from internal pressurization and curvature of the element. The internal pressure is prescribed on the property entry. The methods used to calculate the stress intensification factor are selected through the FSI parameters; see the Remarks under the PBEND (p. 2741) in the .

Tensile stresses are given a positive sign and compressive stresses a negative sign. Only the longitudinal stresses are available as complex stresses. The stress recovery coefficients on the general form of the PBEND entry are used to locate points on the cross section for stress recovery. The subscript 1 is associated with the distance of a stress recovery point from Plane 2. The subscript 2 is associated with the distance from Plane 1. If zero value stress recovery coefficients are used, the axial stress is output.

Rod Elements (CROD, CONROD, CTUBE)

The rod element is defined with a CROD entry and its properties with a PROD entry. The rod element includes extensional and torsional properties. The CONROD entry is an alternate form that includes both the connection and property information on a single entry. The tube element is a specialized form that is assumed to have a circular cross section. The tube element is defined with a CTUBE entry, and its properties with a PTUBE entry. The structural and nonstructural mass of the rod are lumped at the adjacent grid points unless coupled mass is requested with the PARAM,COUPMASS (see Parameters (p. 785) in the *MSC Nastran Quick Reference Guide*). Theoretical aspects of the rod element are treated in Section 5.2 of *The NASTRAN Theoretical Manual*.

The *x*-axis of the element coordinate system is defined by a line connecting end a to end b, as shown in Figure 3-10. The axial force and torque are output on request in either real or complex form. The positive directions for these forces are indicated in Figure 3-10.

The following real element stresses are output on request:

- Axial stress
- Torsional stress
- Margin of safety for axial stress
- Margin of safety for torsional stress

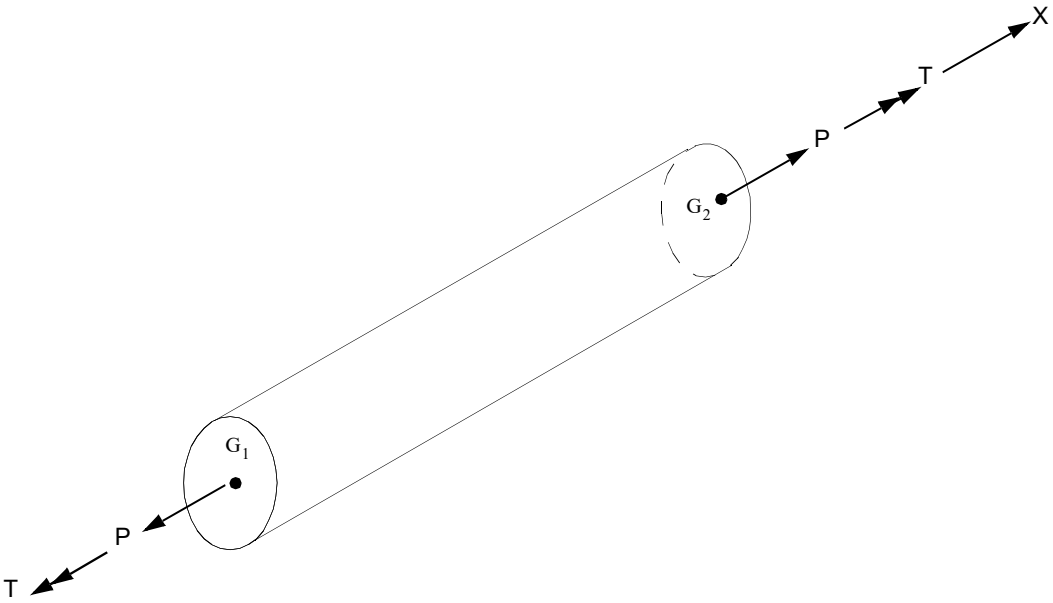


Figure 3-10 Rod Element Coordinate System and Element Forces

Positive directions are the same as those indicated in Figure 3-10 for element forces. Only the axial stress and the torsional stress are available as complex stresses.

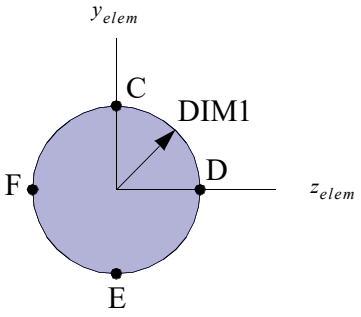
Another kind of rod element is the viscous damper. It has extensional and torsional viscous damping properties rather than stiffness properties. The viscous damper element is defined with a CVISC entry and its properties with a PVISC entry. This element is used in the formulation of dynamic matrices. The mode displacement method (PARAM,DDRMM,-1) must be selected for element force output.

Supplied Beam and Bar Libraries

The Bulk Data entries PBARL and PBEAML provide libraries of cross sections. For open sections and the HAT1, thin-walled theory is assumed.

The following pages provide equations that define the PBEAM and PBAR geometric property entries in terms of entries on the PBEAML or PBARL. Symbols absent for a particular cross section are normally set to zero.

A	Cross-sectional area.
y_c	Distance to centroid along Y element axis.
z_c	Distance to centroid along Z element axis.
y_s	Distance to shear center along Y element axis.
z_s	Distance to shear center along Z element axis.
I_1	Moment of inertia about the Z element axis at the centroid. $I_1 = I_{(zz)_{elem}}$
I_2	Moment of inertia about the Y element axis at the centroid. $I_2 = I_{(YY)_{elem}}$
I_{12}	Product moment of inertia at the centroid. $I_{12} = I_{(ZY)_{elem}}$
J	Torsional Stiffness Constant.
C, D, E, F	Location of the stress recovery points in the element coordinate system relative to the shear center. For the PBARL the locations must be changed to be relative to the centroid. This can be done by adding y_{na}, z_{na} to the listed equations.
K_1, K_2	Shear stiffness factor for plane 1 and plane 2.
I_w	Warping coefficient for the cross section relative to the shear center.
y_{na}, z_{na}	Coordinates of the centroid relative to the shear center.



r = DIM1

TYPE = "ROD"

$$A = \pi r^2$$

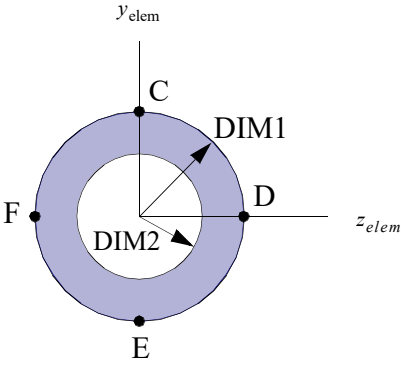
$$\left.\begin{matrix} I_1 \\ I_2 \end{matrix}\right\} = \frac{\pi}{4} r^4$$

$$J = \frac{\pi}{2} r^4$$

$$C = r, 0$$
$$D = 0, r$$
$$E = -r, 0$$
$$F = 0, -r$$

$$\left.\begin{matrix} K_1 \\ K_2 \end{matrix}\right\} = 0.9$$

Figure 3-11 Geometric Property Formulas for a ROD



$$r_0 = \text{DIM1}$$
$$r_i = \text{DIM2}$$

TYPE = "TUBE"

$$A = \pi(r_0^2 - r_i^2)$$

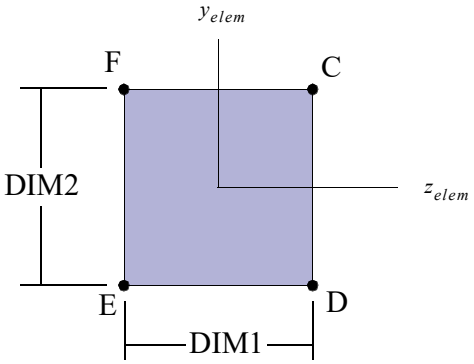
$$\left.\begin{matrix} I_1 \\ I_2 \end{matrix}\right\} = \frac{\pi}{4}(r_0^4 - r_i^4)$$

$$J = \frac{\pi}{2}(r_0^4 - r_i^4)$$

$$C = r_0, 0$$
$$D = 0, r_0$$
$$E = -r_0, 0$$
$$F = 0, -r_0$$

$$\left.\begin{matrix} K_1 \\ K_2 \end{matrix}\right\} = 0.5$$

Figure 3-12 Geometric Property Formulas for a TUBE Cross Section



b = DIM1
h = DIM2

TYPE = "BAR"

$$A = bh$$

$$I_1 = \frac{bh^3}{12}$$

$$I_2 = \frac{b^3h}{12}$$

$$I_{12} = 0.$$

$$J = bh^3\left[\frac{1}{3} - .21\frac{h}{b}\left(1 - \frac{h^4}{12b^4}\right)\right]$$

$$C = \frac{h}{2}, \frac{b}{2}$$

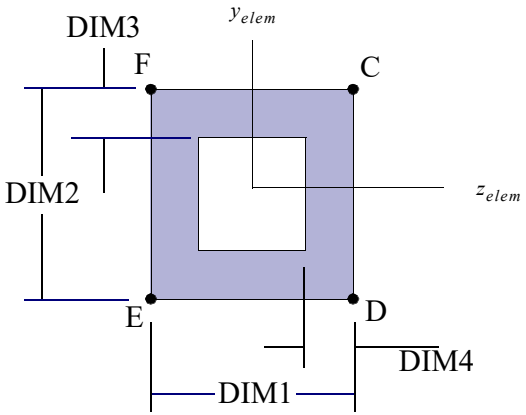
$$D = -\frac{h}{2}, \frac{b}{2}$$

$$E = -\frac{h}{2}, -\frac{b}{2}$$

$$F = \frac{h}{2}, -\frac{b}{2}$$

$$\left.\begin{matrix} K_1 \\ K_2 \end{matrix}\right\} = \frac{5}{6}$$

Figure 3-13 Geometric Property Formulas for a BAR



b = DIM1
h = DIM2
t_1 = DIM3
t_2 = DIM4
$b_i = b - 2t_2$
$h_i = h - 2t_1$

TYPE = “BOX”

$$A = bh - b_i h_i$$

$$I_1 = \frac{bh^3}{12} - \frac{b_i h_i^3}{12}$$

$$I_2 = \frac{hb^3}{12} - \frac{h_i b_i^3}{12}$$

$$I_{12} = 0.$$

$$J = \frac{2t_2 t_1 (b - t_2)^2 (h - t_1)^2}{bt_2 + ht_1 - t_2^2 - t_1^2}$$

$$C = \frac{h}{2}, \frac{b}{2}$$

$$D = -\frac{h}{2}, \frac{b}{2}$$

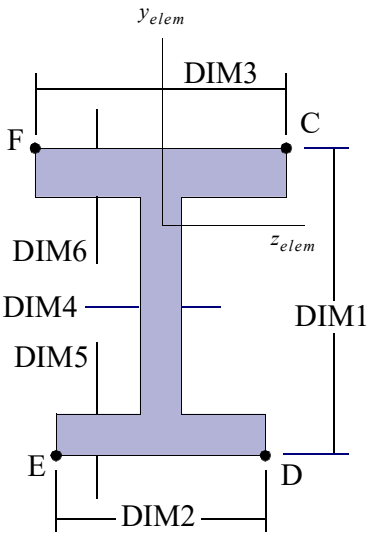
$$E = -\frac{h}{2}, -\frac{b}{2}$$

$$F = \frac{h}{2}, -\frac{b}{2}$$

$$K_1 = \frac{h_i t_2}{A} \cdot 2$$

$$K_2 = \frac{b_i t_1}{A} \cdot 2$$

Figure 3-14 Geometric Property Formulas for a BOX



$$a = \text{DIM2}$$
$$b = \text{DIM3}$$
$$h = \text{DIM1}$$
$$t_a = \text{DIM5}$$
$$t_b = \text{DIM6}$$
$$t_w = \text{DIM4}$$
$$h_w = h - (t_a + t_b)$$
$$h_f = h - 0.5(t_a + t_b)$$

TYPE = "I"

$$A = t_a a + h_w t_w + b t_b$$

$$y_c = \frac{1}{A}\left(\frac{1}{2} h_w (h_w + t_a) t_w + h_f t_b b\right)$$

$$y_s = \frac{t_b h_f b^3}{(t_b b^3 + t_a a^3)}$$

$$y_{na} = y_c - y_s$$

$$I_1 = \frac{b t_b^3}{12} + \frac{a t_a^3}{12} + \frac{t_w h_w^3}{12} + (h_f - y_c)^2 b t_b + y_c^2 a t_a + (y_c - 0.5(h_w + t_a))^2 h_w t_w$$

$$I_2 = \frac{b^3 t_b}{12} + \frac{t_a a^3}{12} + \frac{h_w t_w^3}{12}$$

$$I_{12} = 0.$$

$$J = \frac{1}{3}(t_b^3 b + t_a^3 a + t_w^3 h_f)$$

Note that I_c and y_s are relative to the center of flange defined by Dim1 and Dim4.

$$C = h_f - y_s + .5t_b, .5b$$
$$D = -y - .5t_a, .5a$$
$$E = -y - .5t_a, -.5a$$
$$F = h_f - y_s + .5t_b, -.5a$$

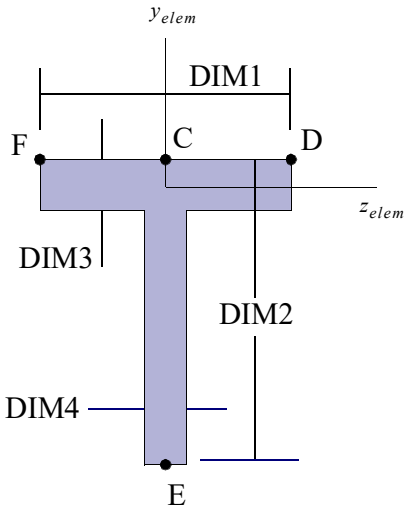
$$K_1 = \frac{t_w h_w}{A}$$

$$K_2 = \frac{5(a t_a + b t_b)}{6A}$$

$$I_w = \frac{h^2 t_b t_a b^3 a^3}{12(t_b b^3 + t_a a^3)}$$

Note that I_w and y_s are based on thin walled formulations.

Figure 3-15 Geometric Property Formulas for an I Section



TYPE = "T"

d = DIM1
h = DIM2 – $0.5t_f$
t_f = DIM3
t_w = DIM4
h_w = DIM2 – t_f

$$A = dt_f + h_w t_w$$

$$y_{na} = -\frac{h_w t_w (h_w + t_f)}{2A}$$

$$I_1 = \frac{dt_f^3}{12} + \frac{t_w h_w^3}{12} + h_w t_w (y_{na} + .5(h_w + t_f))^2 + dt_f y_{na}^2$$

$$I_2 = \frac{t_f d^3}{12} + \frac{h_w t_w^3}{12}$$

$$I_{12} = 0.$$

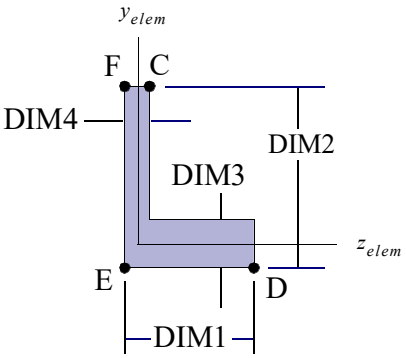
$$J = \frac{1}{3}(t_f^3 d + t_w^3 h)$$

$C = .5t_f, 0$
$D = .5t_f, .5d$
$E = -h, 0$
$F = .5t_f, -.5d$

$$K_1 = \frac{h_w t_w}{A}$$

$$K_2 = \frac{t_f d}{A}$$

Figure 3-16 Geometric Property Formulas for a T Section



$b = \text{DIM1} - 0.5t_2$

$h = \text{DIM2} - 0.5t_1$

$t_1 = \text{DIM3}$

$t_2 = \text{DIM4}$

$h_2 = \text{DIM2} - t_1$

$b_1 = \text{DIM1} - t_2$

TYPE = "L"

$$A = (b + .5t_2)t_1 + h_2t_2$$

$$y_c = \frac{t_2h_2(h_2 + t_1)}{2A}$$

$$z_c = \frac{t_1b_1(b_1 + t_2)}{2A}$$

$$I_1 = \frac{t_1^3(b + .5t_2)}{12} + t_1(b + .5t_2)y_c^2 + \frac{t_2h_2^3}{12} + h_2t_2(0.5(h_2 + t_1) - y_c)^2$$

$$I_2 = \frac{t_2^3h_2}{12} + t_2h_2z_c^2 + \frac{t_1(b + .5t_2)^3}{12} + t_1(b + .5t_2)(.5b_1 - z_2)^2$$

$$I_{12} = z_cy_ct_1t_2 - b_1t_1y_c(.5(b_1 + t_2) - z_c) - h_2t_2z_c(.5(h_2 + t_1) - y_c)$$

$$J = \frac{1}{3}(t_3^3b + t_2^3h)$$

$$C = h, .5t_2$$

$$D = -.5t_1, b$$

$$E = .5t_1, -.5t_2$$

$$F = h, -.5t_2$$

$$K_1 = \frac{h_2 t_2}{A}$$

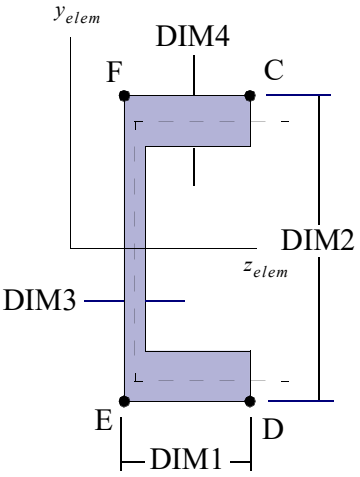
l

$$K_2 = \frac{b_1 t_1}{A}$$

$$y_{na} = y_c$$

$$z_{na} = z_c$$

Figure 3-17 Geometric Property Formulas for an L Section



$$b = \text{DIM1} - .5t_w$$
$$h = \text{DIM2} - t_f$$
$$t_w = \text{DIM3}$$
$$t_f = \text{DIM4}$$
$$b_f = \text{DIM1} - t_w$$
$$h_w = \text{DIM2} - 2.t_f$$

TYPE = "CHAN"

$$A = 2t_f b_f + (h + t_t)t_w$$

$$z_c = \frac{b_f t_f (b_f + t_w)}{A}$$

$$z_s = \frac{b^2 t_f}{\left(2bt_w + \frac{1}{3}ht_f\right)}$$

$$I_1 = \frac{h^2 t_f b_f}{2} + \frac{b_f t_f^3}{6} + \frac{(h + t_f)^3 t_w}{12}$$

(continued)

$$I_2 = \frac{(h + t_f)t_w^3}{12} + \frac{b_f^3 t_f}{6} + .5(b_f + t_w)^2 b_f t_f - z_c^2 A$$

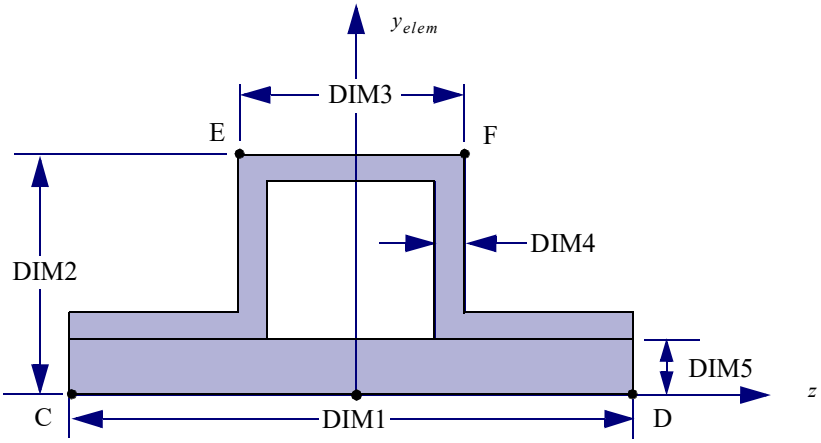
$$J = \frac{1}{3}(2bt_f^3 + ht_w^3)$$

$$I_w = \frac{t_f b^3 h^2}{12}\left(\frac{2t_w h + 3t_f b}{t_w h + 6t_f b}\right)$$

$$C = .5(h + .5t_f), z_s + b$$
$$D = -.5(h + .5t_f), z_s + b$$
$$E = -.5(h + .5t_f), z_s - .5t_w$$
$$F = .5(h + .5t_f), z_s - .5t_w$$
$$K_1 = \frac{t_w h_w}{A}$$
$$K_2 = \frac{2t_f b_f}{A}$$

$$z_{na} = z_c + z_s$$

Note that z_c, z_s are distances measured relative to an origin positioned at the center of the web.



b_1 = DIM1

h = DIM2

b_2 = DIM3

t_2 = DIM4

t_1 = DIM5

TYPE="HAT1"

Figure 3-18 Geometric Property Formulas for a Closed-hat Section

$$A = b_1 t_1 + t_2(b_1 - b_2) + 2t_2(h - t_1) + t_2(b_2 - 2t_2)$$

$$y_c = \frac{(b_1 t_1^2/2) + t_2(b_1 - b_2)(t_1 + t_2/2) + t_2(h^2 - t_1^2) + t_2(b_2 - 2t_2)(h - t_2/2)}{A}$$

$$y_s = \frac{\psi}{\psi_1} + \frac{t_2}{2}$$

where:

$$\psi_1 = \frac{t_2(b_1-b_2+t_2)^3}{6} + \frac{t_2(b_1-b_2+t_2)(b_1+b_2-t_2)^2}{16} + \frac{t_2(h-t_1-t_2)(b_2-t_2)^2}{2} + \frac{t_2(b_2-t_2)^3}{12} + \frac{t_1 b_1^3}{12}$$

$$\begin{aligned}\psi = {} & (b_2-t_2)(h-t_1/2-t_2/2)\left[\frac{t_1(b_1-b_2+t_2)(b_1+b_2-t_2)}{24} + \frac{t_1 b_1^2}{12} + \psi_3\right] \\ & + (b_2-t_2)(h-t_1-t_2)\left[\frac{t_2(b_1-b_2+t_2)(b_1+b_2-t_2)}{8} + \frac{t_2(h-t_1-t_2)(b_2-t_2)}{4} + \psi_3\right] + \frac{t_1 t_2 b_1^3}{12} \\ & - \frac{t_1 t_2(b_1-b_2+t_2)(b_1+b_2-t_2)(b_2-t_2)}{8} + \frac{t_2^2 b_1^3}{12} - \frac{t_1 t_2 b_1^2(b_2-t_2)}{12} \\ & + \frac{t_1 t_2(b_2-t_2)(b_1-b_2+t_2)(b_1+b_2-t_2)}{12} - \frac{t_2^2(b_2-t_2)(b_1-b_2+t_2)(b_1+b_2-t_2)}{8} - \frac{t_2^2 b_1^2(b_2-t_2)}{12} \\ & + \frac{t_2^2(b_2-t_2)(b_1-b_2+t_2)(b_1+b_2-t_2)}{12}\end{aligned}$$

and

$$\psi_3 = \frac{t_1 t_2 \psi_2}{2t_1(h-t_1-t_2) + (b_2-t_2)(t_1+t_2)}$$

where:

$$\begin{aligned}\psi_2 = {} & \frac{(b_1-b_2+t_2)(b_1+b_2-t_2)(h-t_1-t_2)}{4} + \frac{(h-t_1-t_2)^2(b_2-t_2)}{2} + \frac{(b_2-t_2)^3}{12} \\ & + \frac{(b_1-b_2+t_2)(b_1+b_2-t_2)(b_2-t_2)}{6} + \frac{(h-t_1-t_2)(b_2-t_2)^2}{2} + \frac{b_1^2(b_2-t_2)}{12}\end{aligned}$$

$$y_{na} = y_s - (h - y_c)$$

$$\begin{aligned}I_1 = {} & \frac{b_1 t_1^3}{12} + b_1 t_1(y_c - t_1/2)^2 + \frac{(b_1-b_2)t_2^3}{12} \\ & + (b_1-b_2)t_2(y_c-(t_1+t_2/2))^2 \\ & + \frac{t_2(h-t_1)^3}{6} + 2t_2(h-t_1)\left(y_c - \left(\frac{h-t_1}{2}\right) + t_1\right)^2 \\ & + \frac{(b_2-2t_2)t_2^3}{12} + (b_2-2t_2)t_2(y_c-(h-t_2/2))^2\end{aligned}$$

$$I_2 = \frac{t_1 b_1^3}{12} + \frac{t_2(b_1 - b_2)^3}{48} + \frac{t_2(b_1 - b_2)(b_1 + b_2)^2}{16} + \frac{(h - t_1)t_2^3}{6} + \frac{(h - t_1)t_2(b_2 - t_2)^2}{2} + \frac{2t_2(b_2/2 - t_2)^3}{3}$$

$$I_{12} = 0$$

$$J = \frac{4t_1 t_2(b_2 - t_2)^2(h - t_1/2 - t_2/2)^2}{(b_2 - t_2)(t_1 + t_2) + t_1(2h - t_1 - t_2)} + (b_1 - b_2)(t_1 + t_2)^3/3$$

$$C = y_s - h\ ,\ -b_1/2$$

$$D = y_s - h\ ,\ b_1/2$$

$$E = y_s\ ,\ -b_2/2$$

$$F = y_s\ ,\ b_2/2$$

$$K_1 = \frac{1}{n}\sum_{i=1}^{n} \frac{2(h - t_1)t_2}{mA}$$

$$K_2 = \frac{1}{n}\sum_{i=1}^{n} \frac{1 - 2(h - t_1)t_2}{mA}$$

where n is the number of sections and $m = 2$ for end A or end B and $m = 1$ for any intermediate station.

Adding Your Own Beam Cross-Section Library

The standard cross sections provided by MSC should be adequate in the majority of cases. If these standard sections are not adequate for your purposes, you can add your own library of cross sections to suit your needs. To add your own library, you need to write few simple subroutines in FORTRAN and link them to MSC Nastran through inter-process communications.

The MSC Nastran 2016 Utilities Guide will describe to you the current server requirements and provide you with the location of starter subroutines as described below.

This process requires writing and/or modifying up to eight basic subroutines:

1. BSCON -- Defines the number of dimensions for each of the section types.

2. BSBRP -- Calculates section properties based on section dimensions.
3. BSGRQ -- Defines NSECT, the number of section types, and NDIMAX, the maximum number of dimensions (including nonstructural mass) required by any of the sections.
4. BSBRT -- Provides the name, number of dimensions and number of design constraints for each section type.
5. BSBRI -- Provides information for the calculation of gradients of section properties with respect to section dimensions.
6. BSBRG -- Calculates any nonlinear gradients of section properties with respect to section dimensions.
7. BSBRC -- Defines constraints in the design of section dimensions.
8. BSMS -- A utility routine; handles errors that occur in the beam library.

BSCON and BSBRP are always required. BSGRQ, BSBRT, and BSBRI are required if you wish to perform sensitivity and/or optimization tasks using the beam library. BSBRG is required if you are providing nonlinear analytical sensitivities in the design task, and BSBRC is an optional routine that can be provided to help the optimizer to stay within physical design constraints. BSMSG handles any error messages you feel are appropriate.

BSCON SUBROUTINE

This routine provides the number of fields in the continuation lines to be read from the Bulk Data entries PBARL and PBEAML for each cross section in the library. The value of the ENTYP variable may be 0, 1, or 2. When ENTYP = 0, the value returned is the number of DIMi. When ENTYP = 1, the value returned includes both the DIMi and NSM fields. The value of 1 is used for PBARL only. When ENTYP = 2, the value returned includes the DIMi, NSM, SO, and XIXB fields for 11 different stations. The value of 2 applies to PBEAML only.

The calling sequence and example routine for the standard MSC library is given below.

```
      SUBROUTINE BSCON(GRPID,TYPE,ENTYP,NDIMI,ERROR)
C ---------------------------------------------------------------------
C   Purpose
C     To get the number of maximum fields in continuation entries for
C     each section in the library.
C
C   Arguments:
C
C    GRPID   input  integer      Integer id of this group or group name.
C                                Not used, reserved for future use.
C    TYPE    input  character*8  Name of cross section
C    ENTYP   input  integer      0: dimensions only without NSM
C                                1: PBARL, total # of data items for 2:PBEAML
C    NDIMI   output integer      Number of dimi fields for the 'ENTYP'
C                                section
C    ERROR   output integer      Error code
C
C   Called by BCCON
C ---------------------------------------------------------------------
C===  Argument Type Declaration
      INTEGER      GRPID,ENTYP,NDIMI,ERROR
      CHARACTER*8  TYPE

C===  Default to 'nothing wrong'
```

```
      ERROR = 0

C===  Dimensions vary with section type
      IF    ( TYPE.EQ.'ROD     ') THEN
              NDIMI = 1
      ELSEIF( TYPE.EQ.'TUBE    ' .OR. TYPE.EQ.'BAR     ') THEN
              NDIMI = 2
      ELSEIF( TYPE.EQ.'HEXA    ') THEN
              NDIMI = 3
      ELSEIF( TYPE.EQ.'BOX     ' .OR. TYPE.EQ.'T       ' .OR.
     +        TYPE.EQ.'L       '                         .OR.
     +        TYPE.EQ.'CHAN    ' .OR. TYPE.EQ.'CROSS   ' .OR.
     +        TYPE.EQ.'H       '                         .OR.
     +        TYPE.EQ.'I1      ' .OR. TYPE.EQ.'T1      ' .OR.
     +        TYPE.EQ.'CHAN1   '                         .OR.
     +        TYPE.EQ.'Z       ' .OR. TYPE.EQ.'CHAN2   ' .OR.
     +        TYPE.EQ.'T2      ' .OR. TYPE.EQ.'HAT     ') THEN
              NDIMI = 4
      ELSEIF( TYPE.EQ.'I       ' .OR. TYPE.EQ.'BOX1    ') THEN
              NDIMI = 6
      ELSE
C===  Set error code if invalid name for the section
              ERROR = 5150
              RETURN
      ENDIF

C===  Number of data items to be read on PBARL entry is DIMi
C===  plus the NSM field

      IF (ENTYP.EQ.1) NDIMI = NDIMI+1

C===  Number of data items to be read on PBEAML entry is DIMi
C===  plus the NSM, SO and X/XB fields for eleven different
C===  stations.

      IF (ENTYP.EQ.2) NDIMI = (NDIMI+3)*11

C-----------------------------------------------------------------
      RETURN
      END
```

BSBRP Subroutine

Finite element analysis requires section properties such as area, moment of inertia, etc., instead of section dimension. Therefore, the dimensions specified on PBARL and PBEAML need to be converted to equivalent properties usually specified on PBAR and PBEAM entries. The images of all these entries are stored in EPT data block as records.

BSBRP subroutine is the interface of your properties evaluator with MSC Nastran. You may use your own naming convention for the subroutines that calculate the cross-section properties from the dimensions. The calling tree used for the MSC standard library is shown in Figure 3-19.

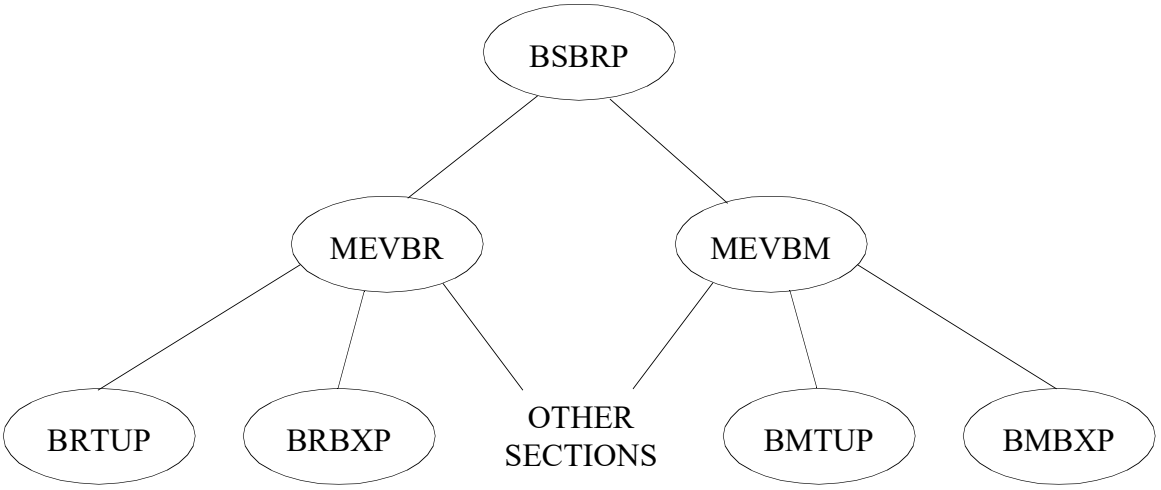


Figure 3-19 Calling Tree Generate Property Data

The BSBRP calls the bar evaluator routine MEVBR for the BAR element and the beam evaluator routine MEVBM for the BEAM element. The evaluators in turn call the routines for each section. The routines are named BRXXP and BMXXP, where XX is a two-letter identifier for the section. For example, the routines for the TUBE section are called BRTUP and BMTUP. The details for various routines are given in Listing 3-1.

Listing 3-1 BSBRP Subroutines

```
SUBROUTINE BSBRP(GRPID,ENTYP,TYPE,IDI,NID,IDO,NIDO,DIMI,NDIMI,
     +                 DIMO,NDIMO,ERROR)
C=======================================================================
C   PURPOSE:  This is the interface for the properties evaluator with
C             MSC Nastran.
C
C   ARGUMENTS
C       GRPID  input   integer      The ID of group name
C       ENTYP  input   integer      1: PBARL, 2: PBEAML
C       TYPE   input   character*8  Arrays for cross section types
C       IDI    input   integer      Array containing the integer words
C                                   in the PBARL or PBEAML EPT record
C       NID    input   integer      Dimension of the IDI array.  It is
C                                   equal to two.
C       IDO    output  integer      Array containing the integer words
C                                   in the PBAR or PBEAM EPT record.
C       NIDO   output  integer      Dimension of the IDO array.  It is
C                                   equal to two for PBAR and four for
C                                   the PEBAM EPT record.
C       DIMI   input   real         Array containing the floating words
C                                   in the PBARL or PBEAML EPT record
C                                   for the 'TYPE' section
C       NDIMI  input   integer      Dimension of the DIMI array.
C       DIMO   output  real         Array containing the real words for
```

```
C                                     the PBAR or PBEAM EPT record.
C       NDIMO  input   integer        Dimension of the DIMO array.  It is
C                                     equal to 17 for the PBAR and 193 for
C                                     the PBEAM EPT record.
C       ERROR  output  integer        Error code
C
C-----------------------------------------------------------------------
C   CALLED BY:
C            BCBRP
C-----------------------------------------------------------------------
C   CALLS:
C         MEVBR ,MEVBM
C-----------------------------------------------------------------------
C   IMPLICIT DECLARATIONS
      IMPLICIT INTEGER (I-N)
      IMPLICIT DOUBLE PRECISION  (A-H,O-Z)
C.......................................................................
C   EXPLICIT DECLARATIONS
      INTEGER ENTYP,ERROR,GRPID
      CHARACTER*8 TYPE
C-----------------------------------------------------------------------
C   DIMENSION STATEMENTS
      INTEGER IDI(NID), IDO(NIDO)
      DOUBLE PRECISION DIMI(NDIMI), DIMO(NDIMO)
C=======================================================================
C=== ENTYP=1, FOR PBAR1; 2, FOR PBEAM1
      IF (ENTYP.EQ.1) THEN
         CALL MEVBR(GRPID,TYPE,IDI,NID,IDO,NIDO,DIMI,NDIMI,DIMO,NDIMO
     +               ,ERROR)
      ELSE IF (ENTYP.EQ.2) THEN
         CALL MEVBM(GRPID,TYPE,IDI,NID,IDO,NIDO,DIMI,NDIMI,DIMO,NDIMO
     +               ,ERROR)
      END IF
C-----------------------------------------------------------------------
      RETURN
      END
```

MEVBR and MEVBM Subroutines

MEVBR and MEVBM are the branched routines for the various sections, and convert the section dimensions to section properties for Bar and Beam elements. You may rename these routines as you like or move the function of these routines to BSBRP. These routines call the BRXXP routines where XX is the two-letter keyword for various section types. The MEVBR routine for the MSC standard library is given in Listing 3-2.

Listing 3-2 MEVBR Subroutine

```
SUBROUTINE MEVBR(GRPID,TYPE,ID,NID,IDO,NIDO,DIMI,NDIMI,DIMO,
     +                  NDIMO,ERROR)

C=======================================================================
C   Purpose
C           Call the default type subroutine to convert PBAR1 to PBAR
C
C   Arguments
C
```

```
C   GRPID input   int     ID of group
C   TYPE  input   char    Type of cross section
C   ID    input   int     Array of values PID, MID contained in PBAR1
C                         entries
C   NID   input   int     Size of ID array, NID=2 for PBAR1 entry
C   IDO   output  int     Array of integer values contained in PBAR
C                         entries
C   NIDO  output  int     Size of IDO array, NIDO=2 for PBAR entry
C   DIMI  input   flt     Dimension values of cross section
C   NDIMI input   int     Size of DIMI array
C   DIMO  output  flt     Properties of cross section
C   NDIMO output  flt     Size of DIMO array
C   ERROR output  int     Type of error
C
C   Method
C             Call the subroutine with respect to  the section type
C
C   Called by
C             BSBRP
C
C   CALLS
C             BRRDPD,BRTUP,BRBRPD,BRBXP,BRIIPD,BRTTPD,BRLLPD,BRCHPD
C-----------------------------------------------------------------------
      IMPLICIT INTEGER (I-N)
      IMPLICIT DOUBLE PRECISION  (A-H,O-Z)

C     Calling  sequence arguments
      INTEGER          ERROR,GRPID,ID(NID),IDO(NIDO)
      CHARACTER*8      TYPE
      DOUBLE PRECISION DIMI(NDIMI),DIMO(NDIMO)

C=======================================================================

C     Clear the output array before usage
      CALL    ZEROD ( DIMO, NDIMO )
      CALL    ZEROI (  IDO, NIDO  )
      ERROR = 0
C
      IF    ( TYPE.EQ.'ROD     ') THEN
              CALL BRRDPD(ID,NID,IDO,NIDO,DIMI,NDIMI,DIMO,NDIMO,ERROR)
      ELSEIF( TYPE.EQ.'TUBE    ') THEN
              CALL BRTUP(ID,NID,IDO,NIDO,DIMI,NDIMI,DIMO,NDIMO,ERROR)
      ELSEIF( TYPE.EQ.'BAR     ') THEN
              CALL BRBRPD(ID,NID,IDO,NIDO,DIMI,NDIMI,DIMO,NDIMO,ERROR)
      ELSEIF( TYPE.EQ.'BOX     ') THEN
              CALL BRBXP(ID,NID,IDO,NIDO,DIMI,NDIMI,DIMO,NDIMO,ERROR)
      ELSEIF( TYPE.EQ.'I       ') THEN
              CALL BRIIPD(ID,NID,IDO,NIDO,DIMI,NDIMI,DIMO,NDIMO,ERROR)
      ELSEIF( TYPE.EQ.'T       ') THEN
              CALL BRTTPD(ID,NID,IDO,NIDO,DIMI,NDIMI,DIMO,NDIMO,ERROR)
      ELSEIF( TYPE.EQ.'L       ') THEN
              CALL BRLLPD(ID,NID,IDO,NIDO,DIMI,NDIMI,DIMO,NDIMO,ERROR)
      ELSEIF( TYPE.EQ.'CHAN    ') THEN
              CALL BRCHPD(ID,NID,IDO,NIDO,DIMI,NDIMI,DIMO,NDIMO,ERROR)
      ELSEIF( TYPE.EQ.'CROSS   ') THEN
              CALL BRCRPD(ID,NID,IDO,NIDO,DIMI,NDIMI,DIMO,NDIMO,ERROR)
      ELSEIF( TYPE.EQ.'H       ') THEN
```

```
                CALL BRHHPD(ID,NID,IDO,NIDO,DIMI,NDIMI,DIMO,NDIMO,ERROR)
        ELSEIF( TYPE.EQ.'T1      ') THEN
                CALL BRT1PD(ID,NID,IDO,NIDO,DIMI,NDIMI,DIMO,NDIMO,ERROR)
        ELSEIF( TYPE.EQ.'I1      ') THEN
                CALL BRI1PD(ID,NID,IDO,NIDO,DIMI,NDIMI,DIMO,NDIMO,ERROR)
        ELSEIF( TYPE.EQ.'CHAN1   ') THEN
                CALL BRC1PD(ID,NID,IDO,NIDO,DIMI,NDIMI,DIMO,NDIMO,ERROR)
        ELSEIF( TYPE.EQ.'Z       ') THEN
                CALL BRZZPD(ID,NID,IDO,NIDO,DIMI,NDIMI,DIMO,NDIMO,ERROR)
        ELSEIF( TYPE.EQ.'CHAN2   ') THEN
                CALL BRC2PD(ID,NID,IDO,NIDO,DIMI,NDIMI,DIMO,NDIMO,ERROR)
        ELSEIF( TYPE.EQ.'T2      ') THEN
                CALL BRT2PD(ID,NID,IDO,NIDO,DIMI,NDIMI,DIMO,NDIMO,ERROR)
        ELSEIF( TYPE.EQ.'BOX1    ') THEN
                CALL BRB1PD(ID,NID,IDO,NIDO,DIMI,NDIMI,DIMO,NDIMO,ERROR)
        ELSEIF( TYPE.EQ.'HEXA    ') THEN
                CALL BRHXPD(ID,NID,IDO,NIDO,DIMI,NDIMI,DIMO,NDIMO,ERROR)
ELSEIF( TYPE.EQ.'HAT     ') THEN
                CALL BRHTPD(ID,NID,IDO,NIDO,DIMI,NDIMI,DIMO,NDIMO,ERROR)
        ELSE
                ERROR = 5150
        END IF
C-------------------------------------------------------------------------
        RETURN
        END
```

BRXXP and BMXXP Subroutines

The purpose of the BRXXP and BMXXP routines is to calculate the properties from the section dimensions. For each cross section, subroutines are required to convert the images of PBARL and PBEAML records to the images of PBAR and PBEAM records in the EPT data block.

BRTUP Subroutine

BRTUP is an example routine that shows how to convert PBARL EPT record to PBAR EPT record for the Tube section. First, the details of the PBARL and the PBAR record are shown, and then the routine itself is given.

PBARL Record

The PBARL record in the EPT data block is a derived from the PBARL Bulk Data entry and is given below.

Table 3-4 PBARL (9102, 91, 52)

Word	Name	Type	Description
1	PID	I	Property identification number.
2	MID	I	Material identification number.
3	Group	Char	Group Name.
4	Group	Char	Group Name.
5	TYPE	Char4	Cross-section Type.

Table 3-4 PBARL (9102, 91, 52)

Word	Name	Type	Description
6	TYPE	Char4	Cross-section Type.
7	Dim1	RS	Dimension 1.
8	Dim2	RS	Dimension 2.
n+7-1	Dim n	RS	Dimension n (note that the final dimension is the nonstructural mass).
n+7	Flag	I	-1. Flag indicating end of cross-section dimensions.

PBAR Record

The PBAR record in the EPT data block is derived from the PBAR Bulk Data entry and consists of 19 words. It is a replica of the Bulk Data entry, starting with PID field. The word 8 in the record is set to 0.0 since the field 9 in the first line of the PBAR Bulk Data entry is not used. The details of the PBAR record are given in Table 3-5.

Table 3-5 PBAR (52, 20, 181)

Word	Name	Type	Description
1	PID	I	Property identification number.
2	MID	I	Material identification number.
3	A	RS	Area of cross-section.
4	I1	RS	Area moment of inertia for bending in plane 1.
5	I2	RS	Area moment of inertia for bending in plane 2.
6	J	RS	Torsional constant.
7	NSM	RS	Nonstructural mass per unit length.
8	FE	RS	Not used. Set to 0.0.
9	C1	RS	Stress recovery location.
10	C2	RS	Stress recovery location.
11	D1	RS	Stress recovery location.
12	D2	RS	Stress recovery location.
13	E1	RS	Stress recovery location.
14	E2	RS	Stress recovery location.
15	F1	RS	Stress recovery location.

Table 3-5 PBAR (52, 20, 181) (continued)

Word	Name	Type	Description
16	F2	RS	Stress recovery location.
17	K1	RS	Area factor of shear for plane 1.
18	K2	RS	Area factor of shear for plane 2.
19	I12	RS	Area product of inertia.

Listing 3-3 BRTUP Subroutine

```
      SUBROUTINE BRTUP(ID,NID,IDO,NIDO,DIMI,NDIMI,DIMO,NDIMO,ERROR)
C=======================================================================
C   Purpose:
C      Convert PBAR1(entity type : TUBE) to PBAR
C
C   Arguments:
C
C   DIMI  input  flt Array of dimension values for cross-section
C                    (SEE FIG. 5 IN MEMO SSW-25, REV. 4, DATE 8/16/94)
C   NDIM  input  int Size of DIMI array
C   DIMO  output flt Array of property values for cross-section
C   NDIMO output int Size of DIMO array
C   ERROR output int Type of error
C
C   DISCRIPTION FOR DIMO ARRAY:
C       DIMO (1)  = A
C       DIMO (2)  = I1
C       DIMO (3)  = I2
C       DIMO (4)  = J
C       DIMO (5)  = NSM
C       DIMO (6)  = FE
C       DIMO (7)  = C1
C       DIMO (8)  = C2
C       DIMO (9)  = D1
C       DIMO (10) = D2
C       DIMO (11) = E1
C       DIMO (12) = E2
C       DIMO (13) = F1
C       DIMO (14) = F2
C       DIMO (15) = K1
C       DIMO (16) = K2
C       DIMO (17) = I12
C
C   Method
C
C   Called by:
C             MEVBR
C-----------------------------------------------------------------------
      IMPLICIT INTEGER (I-N)
      IMPLICIT DOUBLE PRECISION  (A-H,O-Z)

C     Calling sequence arguments
```

```
      INTEGER          ID(NID),IDO(NIDO),ERROR
      DOUBLE PRECISION DIMI(NDIMI),DIMO(NDIMO)
C=======================================================================
C
C=== WRITE THE PART OF INTEGER
      DO 30 II = 1,NID
         IDO(II) = ID(II)
 30   CONTINUE
C
      DIM1 = DIMI(1)
      DIM2 = DIMI(2)
      DIMO(1) = PI*(DIM1*DIM1-DIM2*DIM2)
      DIMO(2) = PI*(DIM1**4-DIM2**4)/4.D0
      DIMO(3) = DIMO(2)
      DIMO(4) = PI*(DIM1**4-DIM2**4)/2.D0
      DIMO(5) = DIMI(3)
      DIMO(6) = 0.D0
      DIMO(7) = DIM1
      DIMO(8) = 0.D0
      DIMO(9) = 0.D0
      DIMO(10) = DIM1
      DIMO(11) = -DIM1
      DIMO(12) = 0.D0
      DIMO(13) = 0.D0
      DIMO(14) = -DIM1
      DIMO(15) = 0.5D0
      DIMO(16) = 0.5D0
      DIMO(17) = 0.D0

      IF ( DIMI(1).LE.DIMI(2) ) ERROR = 5102

C-----------------------------------------------------------------------
      RETURN
      END
```

BMTUP Subroutine

BMTUP is an example routine that shows how to convert PBEAML EPT record to PBEAM EPT record for the Tube section. First, the details of the PBEAML and the PBEAM records are shown, and then the routine itself is given.

PBEAML Record

The PBEAML record in the EPT data block is a derived from the PBEAML Bulk Data entry and is given in Table 3-6.

Table 3-6 PBEAML (9202, 92, 53)

Word	Name	Type	Description
1	PID	I	Property ID
2	MID	I	Material ID
3	Group	Char	Group Name
4	Group	Char	Group Name

Table 3-6 PBEAML (9202, 92, 53) (continued)

Word	Name	Type	Description
5	TYPE	Char4	Cross-section Type
6	TYPE	Char4	Cross-section Type
7	SO	RS	Stress output request flag, 1=yes,=no
8	XXB	RS	X/XB - parametric location of the station
9	Dim1	RS	Dimension 1
10	Dim2	RS	Dimension 2
n+9-1	Dim n	RS	Dimension n (note that the final dimension is the nonstructural mass)
n+9	Flag	I	-1. Flag indicating end of cross-section dimensions
Words 7 through ndim+11 repeat 11 times.			

PBEAM Record

The PBEAM record in EPT data block consists of 197 words. The first five words and the last 16 words are common to all the 11 stations. Each of the 11 stations have their own 21 unique words. The details of the PBEAM record are given in Table 3-7.

Table 3-7 PBEAM (5402, 54, 262)

Word	Name	Type	Description
1	PID	I	Property identification number.
2	MID	I	Material identification number.
3	N	I	Number of intermediate stations.
4	CCF	I	Constant cross-section flag. 1 = constant, 2 = variable.
5	X	RS	Unused.
6	SO	RS	Stress output request. 1.0 = yes, 0.0 = no.
7	XXB	RS	Parametric location of the station. Varies between 0. and 1.0.
8	A	RS	Area.
9	I1	RS	Moment of inertia for bending in plane 1.
10	I2	RS	Moment of inertia for bending in plane 2.
11	I12	RS	Area product of inertia.
12	J	RS	Torsional constant.

Table 3-7 PBEAM (5402, 54, 262) (continued)

Word	Name	Type	Description
13	NSM	RS	Nonstructural mass.
14	C1	RS	Stress recovery location.
15	C2	RS	Stress recovery location.
16	D1	RS	Stress recovery location.
17	D2	RS	Stress recovery location.
18	E1	RS	Stress recovery location.
19	E2	RS	Stress recovery location.
20	F1	RS	Stress recovery location.
21	F2	RS	Stress recovery location.
			Words 6 through 21 repeat 11 times.
182	K1	RS	Area factor for shear for plane 1.
183	K2	RS	Area factor for shear for plane 2.
184	S1	RS	Shear-relief coefficient for plane 1.
185	S2	RS	Shear-relief coefficient for plane 2.
186	NSIA	RS	Nonstructural mass moment of inertia at end A.
187	NSIB	RS	Nonstructural mass moment of inertia at end B.
188	CWA	RS	Warping coefficient for end A.
189	CWB	RS	Warping coefficient for end B.
190	M1A	RS	Y-coordinate of center of gravity for nonstructural mass at end A.
191	M2A	RS	Z-coordinate of center of gravity for nonstructural mass at end A.
192	M1B	RS	Y-coordinate of center of gravity for nonstructural mass at end B.
193	M2B	RS	Z-coordinate of center of gravity for nonstructural mass at end B.
194	N1A	RS	Y-coordinate for neutral axis at end A.
195	N2A	RS	Z-coordinate for neutral axis at end A.
196	N1B	RS	Y-coordinate for neutral axis at end B.
197	N2B	RS	Z-coordinate for neutral axis at end B.

Listing 3-4 BMTUP Subroutine

```
      SUBROUTINE BMTUP(ID,NID,IDO,NIDO,DIMI,NDIMI,DIMO,NDIMO,ERROR)
C=======================================================================
C   Purpose
```

```
C      Convert PBEAM1(entity type : TUBE) to PBEAM
C
C   Arguments
C
C   ID     input  int  Contain the integer information PID, MID
C   NID    input  int  Size of ID array, NID = 2
C   DIMI   input  flt  Dimension values of cross section
C                      ( See  FIG.5 IN MEMO SSW-25, REV. 4, DATE 8/16/94)
C   NDIMI input  int  Size of DIMI array
C   IDO   output int  Contain the integer information PID,MID,N,CCF
C   NIDO  output int  Size of IDO array, NIDO = 4
C   DIMO  output flt  Properties of cross section
C   NDIMO output int  Size of DIMO array
C   ERROR output int  Type of error
C
C   Description for DIMO array
C       DIMO (1)    = X
C       DIMO (2)    = SO
C       DIMO (3)    = XXB
C       DIMO (4)    = A
C       DIMO (5)    = I1
C       DIMO (6)    = I2
C       DIMO (7)    = I12
C       DIMO (8)    = J
C       DIMO (9)    = NSM
C       DIMO (10)   = C1
C       DIMO (11)   = C2
C       DIMO (12)   = D1
C       DIMO (13)   = D2
C       DIMO (14)   = E1
C       DIMO (15)   = E2
C       DIMO (16)   = F1
C       DIMO (17)   = F2
C    DIMO(2) thru DIMO(17) repeat 11 times
C       DIMO (178) = K1
C       DIMO (179) = K2
C       DIMO (180) = S1
C       DIMO (181) = S2
C       DIMO (182) = NSIA
C       DIMO (183) = NSIB
C       DIMO (184) = CWA
C       DIMO (185) = CWB
C       DIMO (186) = M1A
C       DIMO (187) = M2A
C       DIMO (188) = M1B
C       DIMO (189) = M2B
C       DIMO (190) = N1A
C       DIMO (191) = N2A
C       DIMO (192) = N1B
C       DIMO (193) = N2B
C
C   Method
C               Simply calculate the properties and locate that data
C               to the image of PBEAM entries
C   Called by
C               MEVBM
C-----------------------------------------------------------------------
```

```
      IMPLICIT INTEGER (I-N)
      IMPLICIT DOUBLE PRECISION  (A-H,O-Z)

C     Calling  sequence arguments
      INTEGER          ERROR,ID(NID),IDO(NIDO)
      DOUBLE PRECISION DIMI(NDIMI),DIMO(NDIMO)

C     Local variables
      INTEGER NAM(2)

C     NASTRAN common blocks
      COMMON /CONDAD/PI

C     Local data
      DATA NAM/4HBMTU,4HPD  /
C=====================================================================
C
C=== WRITE THE PART OF INTEGER
      DO 30 II = 1,NID
         IDO(II) = ID(II)
 30   CONTINUE
C---------------------------------------------------------------------
C=== DETECT HOW MANY STATION , CONSTANT OR LINEAR BEAM.
C---------------------------------------------------------------------
      ISTATC = NDIMI/11
      DO 35 II = 0,10
         NW = II*ISTATC
         IF (DIMI(3+NW).EQ.0.D0) THEN
            IDO(3) = II-2
            IDO(4) = 1
            IF (DIMI(3).NE.DIMI(3+NW-ISTATC)) IDO(4)=2
            GO TO 40
         END IF
 35   CONTINUE
 40   DIMO(1) = 0.D0
      DO 100 L1 = 0,10
         LC = 16*L1
         NW = L1*ISTATC
         IF (DIMI(3+NW+ISTATC).EQ.0.D0) LC = 160
         DIM1 = DIMI(3+NW)
         DIM2 = DIMI(4+NW)

         IF ( DIM1.LE.DIM2 ) ERROR = 5102

         DIMO(2+LC) = DIMI(1+NW)
         DIMO(3+LC) = DIMI(2+NW)
         DIMO(4+LC) = PI*(DIM1*DIM1-DIM2*DIM2)
         DIMO(5+LC) = PI*(DIM1**4-DIM2**4)/4.D0
         DIMO(6+LC) = DIMO(5+LC)
         DIMO(7+LC) = 0.D0
         DIMO(8+LC) = PI*(DIM1**4-DIM2**4)/2.D0
         DIMO(9+LC) = DIMI(5+NW)
         DIMO(10+LC) = DIM1
         DIMO(11+LC) = 0.D0
         DIMO(12+LC) = 0.D0
         DIMO(13+LC) = DIM1
         DIMO(14+LC) = -DIM1
```

```
         DIMO(15+LC) = 0.D0
         DIMO(16+LC) = 0.D0
         DIMO(17+LC) = -DIM1
         IF (LC.EQ.160) GO TO 110
 100  CONTINUE
 110  DIMO(178) = 0.5D0
      DIMO(179) = 0.5D0
C-----------------------------------------------------------------------
 300  RETURN
      END
```

BSGRQ Subroutine

For optimization of PBARL entries, you need to provide overall information such as number of cross sections in your library and the maximum number of fields in the continuation lines. BSGRQ provides the information and is required only if you wish to perform sensitivity or optimization with the section dimensions. The calling sequence and example routine is given in Listing 3-5.

Listing 3-5 BSGRQ Routine

```
      SUBROUTINE BSGRQ(GRPID,NSECT,NDIMAX,ERROR)
C
C ======================================================================
C   PURPOSE:
C     PROVIDE OVERALL CHARACTERISTICS OF A BEAM SECTION LIBRARY
C ----------------------------------------------------------------------
C   ARGUMENTS:
C
C      GRPID  input  integer  Group name.  Not used, reserved for future
c                             use.
C      NSECT  output integer  Number of different section types
C      NDIMAX output integer  Maximum number of dimension for any
C                             section type
C      ERROR  output integer  Indicates if an error has occurred. The
C                             code returned indicates the type of error
C ----------------------------------------------------------------------
C   CALLED BY:
C      BCGRQ routine
C----------------------------------------------------------------------
C
C===  Set number of section in the library
          nsect  = 19
C===  Set the maximum number of DIMi fields (including 1 for nonstructured mass)
required by any one section
          ndimax = 7
C----------------------------------------------------------------------
      RETURN
      END
```

Note that the arguments GRPID and ERROR are not used. GRPID is reserved for future use. You may use the ERROR argument to send an error code which could later be used to print an error message.

BSBRT Subroutine

The BSBRT routine provides the name, number of fields in the continuation line in the PBARL Bulk Data entry, and the number of constraints for each section in the library. As an example, the name of the tube section, shown in Figure 3-20, in the MSC standard library is “TUBE”, the number of dimensions for the tube section is three (OUTER RADIUS, INNER RADIUS, and NSM), and there is one physical constraint.

The physical constraint is that the inner radius (DIM2) can not be greater than outer radius (DIM1). It is necessary to specify the constraints so that the optimization of the section dimension in SOL200 does not result into an inconsistent shape.

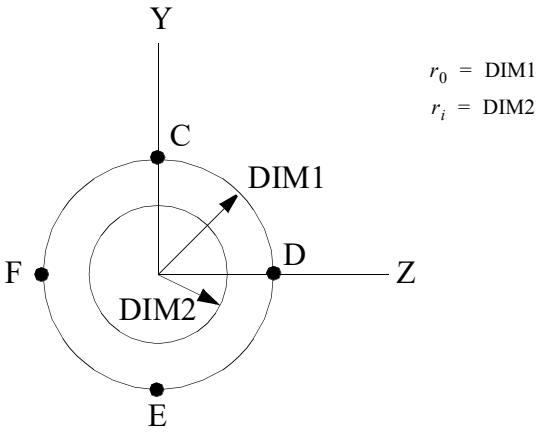


Figure 3-20 Tube Section

This routine is called during optimization process. The calling sequence and example routine is given in Listing 3-6.

Listing 3-6 BSBRT Subroutine

```
      SUBROUTINE BSBRT(GRPID,ENTYP,TYPE,NDIM,NCONST,NSECT,ERROR)
C ======================================================================
C   Purpose
C          Get the name, number of  fields in the continuation line and
C          number of constraints
C          to be used by optimization routines
C
C   Arguments
C
C       GRPID  input  integer      The ID of group name
C       ENTYP  input  integer      1: PBARL, 2: PBEAML
C       TYPE   output character*8  Arrays for cross section types
C       NDIM   output integer      Number of dimensions for isect
C                                  section type, including 1 for
                                   nonstructural mass
C       NCONST output integer      Number of dimensional constraints
C                                  imposed by isect section type
C       NSECT  input  integer      Number of sections
C       ERROR  output integer      Error code
```

```
C
C   Called by
C              BCBRT routine
C ----------------------------------------------------------------------
C===  Argument Type Declaration
      INTEGER      GRPID,ENTYP,NDIM(NSECT),NCONST(NSECT)
      CHARACTER*8  TYPE(NSECT)

C===  Local variables
      INTEGER NAM(2)
C ======================================================================
C===  Currently, only PBARL is supported for optimization. Based on
C===  ENTYP the library may have different number of DIMi fields and
C===  number of constraints. Currently, GRPID and ENTYP are not being
C===  used. So just set them to default values, even though they are
C===  input type.
     GRPID = 1
      ENTYP = 1

C ----------------------------------------------------------------------
C===  Set the name, number of fields in the continuation line for the PBARL entry,
C===  and number of constraints in the TYPE, DIMI and NCONST arrays,
      respectively.
C===   Note that the value in DIMI is one more (for NSM field) than DIMi fields
C===   Make sure names are all capitals, even if they are lower case in the
       input data file.
      TYPE(1) = 'ROD'
      NDIM(1) = 2
      NCONST(1) = 0
C
      TYPE(2) = 'TUBE'
      NDIM(2) = 3
      NCONST(2) = 1
C
      TYPE(3) = 'BAR'
      NDIM(3) = 3
      NCONST(3) = 0
C
      TYPE(4) = 'BOX'
      NDIM(4) = 5
      NCONST(4) = 2
C
      TYPE(5) = 'I'
      NDIM(5) = 7
      NCONST(5) = 3
C
      TYPE(6) = 'T'
      NDIM(6) = 5
      NCONST(6) = 2
C
      TYPE(7) = 'L'
      NDIM(7) = 5
      NCONST(7) = 2
C
      TYPE(8) = 'CHAN'
      NDIM(8) = 5
      NCONST(8) = 2
```

```
C
      TYPE(9) = 'CROSS'
      NDIM(9) = 5
      NCONST(9) = 1
C
      TYPE(10) = 'H'
      NDIM(10) = 5
      NCONST(10) = 1
C
      TYPE(11) = 'T1'
      NDIM(11) = 5
      NCONST(11) = 1
C
      TYPE(12) = 'I1'
      NDIM(12) = 5
      NCONST(12) = 1
C
      TYPE(13) = 'CHAN1'
      NDIM(13) = 5
      NCONST(13) = 1
C
      TYPE(14) = 'Z'
      NDIM(14) = 5
      NCONST(14) = 1
C
      TYPE(15) = 'CHAN2'
      NDIM(15) = 5
      NCONST(15) = 2
C
      TYPE(16) = 'T2'
      NDIM(16) = 5
      NCONST(16) = 2
C
      TYPE(17) = 'BOX1'
      NDIM(17) = 7
      NCONST(17) = 2
C
      TYPE(18) = 'HEXA'
      NDIM(18) = 4
      NCONST(18) = 1
C
      TYPE(19) = 'HAT'
      NDIM(19) = 5
      NCONST(19) = 2

C
C =====================================================================
      RETURN
      END
```

BSBRI Subroutine

The BSBRI subroutine is required if optimization is to be performed. Its function is to provide information required in the calculation of the sensitivities (gradients) of the bar properties with respect to the bar dimensions.

Two basic types of information are provided. The first is the SENTYP array, which indicates how each section property varies as a function of each dimension. Values in the SENTYP array can be either: 0 for no variation; 1 for a linear variation; 2 for a nonlinear variation; or 3 for an unknown variation. The SENTYP = 3 option is to be used when you know that the property is a function of the design dimension, but analytical gradient information is not being provided using the BSBRG subroutine. In this case, MSC Nastran will calculate the gradients for you using central differencing techniques.

The second piece of information is the ALIN array. This array provides any linear sensitivity data. For example, the C1 stress recovery location for the TUBE section is $1.0 \cdot \text{DIM1}$ so that this sensitivity of this stress recovery point with respect to the first dimension is 1.0.

You may use your own naming convention for the subroutines that specify the section sensitivity data. The calling tree used for the MSC standard library is shown in Figure 3-21.

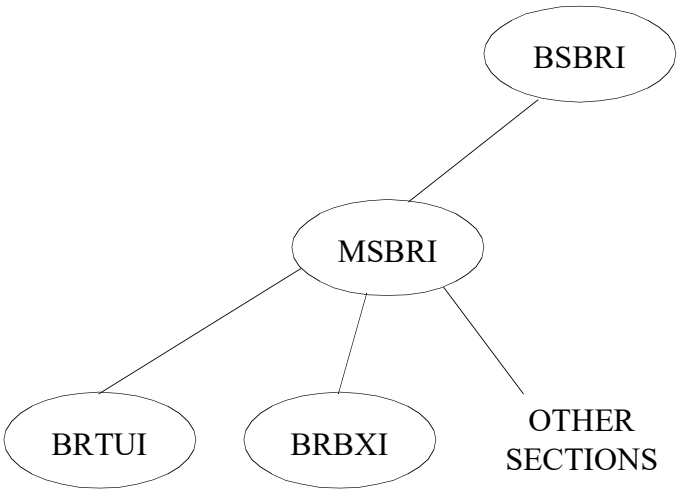


Figure 3-21 Calling Tree to Generate Sensitivity Data

BSBRI calls the bar evaluator routine MSBRI/S for the BAR element while the corresponding beam evaluator routine is absent since the PBEAM1 dimensions cannot be designed. The evaluator in turn calls the routines for each section. The routines are named BRXXID, where XX is a two-letter identifier for the section. For example, the routine for the TUBE section is called BRTUI. The details for various routines are given in Listing 3-7.

Listing 3-7 BSBRI Subroutine

```
      SUBROUTINE BSBRI(GRPID,ENTYP,SECTON,SENTYP,ALIN,NDIM,NPROP,
     1                 ERROR)

C
C ======================================================================
C   Purpose
C     set up section dependent information for a particular cross
C     section type
C
```

```
C   Arguments
C
C       GRPID  input  integer      ID of the group
C       ENTYP  input  integer      1: PBAR1, 2: PBEAM1
C       SECTON input  character*8  Section type
C       SENTYP output integer      Type of sensitivity, 0: invariant,
C                                  1: linear, 2: nonlinear, 3: calculated by
                                      finite difference
C       ALIN   output double       Matrix providing the linear
C                                  factors for sensitive relationships
C       NDIM   input  integer      Number of dimensions
C       NPROP  input  integer      Number of properties
C       ERROR  output integer      Type of error
C
C   Method
C               Simply transfer control based on entry type
C
C   Called by
C               BCBRID
C
C   Calls
C               MSBRI, MSBMID
C ------------------------------------------------------------------------
      IMPLICIT INTEGER (I-N)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

C     Calling sequence arguments
      INTEGER          ENTYP,ERROR,GRPID,SENTYP(NPROP,NDIM)
      CHARACTER*8      SECTON
      DOUBLE PRECISION ALIN(NPROP,NDIM)
C ========================================================================
      GRPID = 1
      IF(ENTYP.EQ.1) THEN
        CALL MSBRI(SECTON,SENTYP,ALIN,NDIM,NPROP,ERROR)
      ELSE
         ERROR = 5400
      END IF

C
      RETURN
      END
```

MSBRI Subroutine

MSBRI is a branched routine for providing information on the calculation of sensitivities for each of the bar types. You may rename this routine as you like or move its function to BSBRI. MSBRI calls the BRXXID routines, where XX is the two-letter keyword for various section types. The routine MSBRI for the MSC standard library is given in Listing 3-8.

Listing 3-8 MSBRI Subroutine

```
      SUBROUTINE MSBRI(SECTON,SENTYP,ALIN,NDIM,NPROP,ERROR)
C
C ========================================================================
C   Purpose
C     To set up section dependent information for PBAR1 cross section
C
```

```
C   Arguments
C
C       SECTON input  character*8 Name of section type
C       SENTYP output integer     Type of sensitivity, 0: invariant,
C                                 1: linear, 2: nonlinear
C       ALIN   output double      Matrix providing the linear factors
C                                 for sensitive relationships
C       NDIM   input  integer     No. of dimensions
C       NPROP  input  integer     No. of properties in EPT datablock
C       ERROR  output integer     Type of error
C
C    Method
C               Simply transfer the section dependent information of
C               the 19 kinds
C    Called by
C               BSBRI
C
C    Calls
C               BRRDID, BRBRID, BRTUI, BRBXI, BRIIID, BRLLID, BRTTID,
C               BRCHID, BRCRID, BRHHID, BRT1ID, BRI1ID, BRC1ID, BRZZID,
C               BRC2ID, BRT2ID, BRB1ID, BRHXID, BRHTID
C               ZEROI, ZEROD (Nastran utility)
C ------------------------------------------------------------------------

      IMPLICIT INTEGER (I-N)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

C     Calling sequence arguments
      CHARACTER*8       SECTON
      INTEGER           ERROR,SENTYP(NPROP,NDIM)
      DOUBLE PRECISION ALIN(NPROP,NDIM)
C ========================================================================
C
      CALL ZEROI( SENTYP(1,1), NPROP*NDIM )
      CALL ZEROD( ALIN(1,1), NPROP*NDIM )
      ERROR = 0

C
      IF(SECTON.EQ.'ROD') THEN
         CALL BRRDID(SENTYP,ALIN,NDIM,NPROP,ERROR)
      ELSE IF(SECTON.EQ.'BAR') THEN
         CALL BRBRID(SENTYP,ALIN,NDIM,NPROP,ERROR)
      ELSE IF(SECTON.EQ.'TUBE') THEN
         CALL BRTUI(SENTYP,ALIN,NDIM,NPROP,ERROR)
      ELSE IF(SECTON.EQ.'BOX') THEN
         CALL BRBXI(SENTYP,ALIN,NDIM,NPROP,ERROR)
      ELSE IF(SECTON.EQ.'I') THEN
         CALL BRIIID(SENTYP,ALIN,NDIM,NPROP,ERROR)
      ELSE IF(SECTON.EQ.'L') THEN
         CALL BRLLID(SENTYP,ALIN,NDIM,NPROP,ERROR)
      ELSE IF(SECTON.EQ.'T') THEN
         CALL BRTTID(SENTYP,ALIN,NDIM,NPROP,ERROR)
      ELSE IF(SECTON.EQ.'CHAN') THEN
         CALL BRCHID(SENTYP,ALIN,NDIM,NPROP,ERROR)
      ELSE IF(SECTON.EQ.'CROSS') THEN
         CALL BRCRID(SENTYP,ALIN,NDIM,NPROP,ERROR)
      ELSE IF(SECTON.EQ.'H') THEN
```

```
         CALL BRHHID(SENTYP,ALIN,NDIM,NPROP,ERROR)
      ELSE IF(SECTON.EQ.'T1') THEN
         CALL BRT1ID(SENTYP,ALIN,NDIM,NPROP,ERROR)
      ELSE IF(SECTON.EQ.'I1') THEN
         CALL BRI1ID(SENTYP,ALIN,NDIM,NPROP,ERROR)
      ELSE IF(SECTON.EQ.'CHAN1') THEN
         CALL BRC1ID(SENTYP,ALIN,NDIM,NPROP,ERROR)
      ELSE IF(SECTON.EQ.'Z') THEN
         CALL BRZZID(SENTYP,ALIN,NDIM,NPROP,ERROR)
      ELSE IF(SECTON.EQ.'CHAN2') THEN
         CALL BRC2ID(SENTYP,ALIN,NDIM,NPROP,ERROR)
      ELSE IF(SECTON.EQ.'T2') THEN
         CALL BRT2ID(SENTYP,ALIN,NDIM,NPROP,ERROR)
      ELSE IF(SECTON.EQ.'BOX1') THEN
         CALL BRB1ID(SENTYP,ALIN,NDIM,NPROP,ERROR)
      ELSE IF(SECTON.EQ.'HEXA') THEN
         CALL BRHXID(SENTYP,ALIN,NDIM,NPROP,ERROR)
      ELSE IF(SECTON.EQ.'HAT') THEN
         CALL BRHTID(SENTYP,ALIN,NDIM,NPROP,ERROR)
      ELSE
         ERROR = 5410
      END IF

C
      RETURN
      END
```

BRTUI Subroutine

BRTUI is an example routine that shows how to define the sensitivity type of each of the bar properties for the tube and the subset of the sensitivities that are linear.

Listing 3-9 BRTUI Subroutine

```
      SUBROUTINE BRTUI(SENTYP,ALIN,NDIM,NPROP,ERROR)
C
C ======================================================================
C   Purpose
C     To set up section dependent information for rod cross section
C
C   Arguments
C
C       SENTYP output integer  Type of sensitivity, 0: invariant,
C                              1: linear, 2: nonlinear
C       ALIN   output double   Matrix providing the linear factors for
C                              sensitive relationships
C       NDIM   input  integer  No. of dimensions
C       NPROP  input  integer  No. of properties in EPT datablock
C       ERROR  output integer  Type of error
C
C   Method
C               Simply provides the information
C
C   Called by
C               MSBRI
C ----------------------------------------------------------------------
```

```
      IMPLICIT INTEGER (I-N)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

C     Calling sequence arguments
      INTEGER          ERROR,SENTYP(NPROP,NDIM)
      DOUBLE PRECISION ALIN(NPROP,NDIM)

C     Local variables
      INTEGER NAM(2)

C     Local data
      DATA NAM/4HBRTU,4HID  /
C =====================================================================
C
      ALIN( 7,1) =  1.0D0
      ALIN(10,1) =  1.0D0
      ALIN(11,1) = -1.0D0
      ALIN(14,1) = -1.0D0
      ALIN( 5,3) =  1.0D0
C
      SENTYP( 1,1) = 2
      SENTYP( 1,2) = 2
      SENTYP( 2,1) = 2
      SENTYP( 2,2) = 2
      SENTYP( 3,1) = 2
      SENTYP( 3,2) = 2
      SENTYP( 4,1) = 2
      SENTYP( 4,2) = 2
      SENTYP( 7,1) = 1
      SENTYP(10,1) = 1
      SENTYP(11,1) = 1
      SENTYP(14,1) = 1
      SENTYP( 5,3) = 1

C
      RETURN
      END
```

BSBRG Subroutine

The BSBRG subroutine is required if optimization is to be performed and analytical sensitivities are needed (SENTYP = 2 in subroutine BSBRI). Its function is to provide the nonlinear gradients of the bar properties with respect to the bar dimensions. You may use your own naming convention for the subroutines that calculate the section gradients from the dimensions. The calling tree used for the MSC standard library is shown in Figure 3-22.

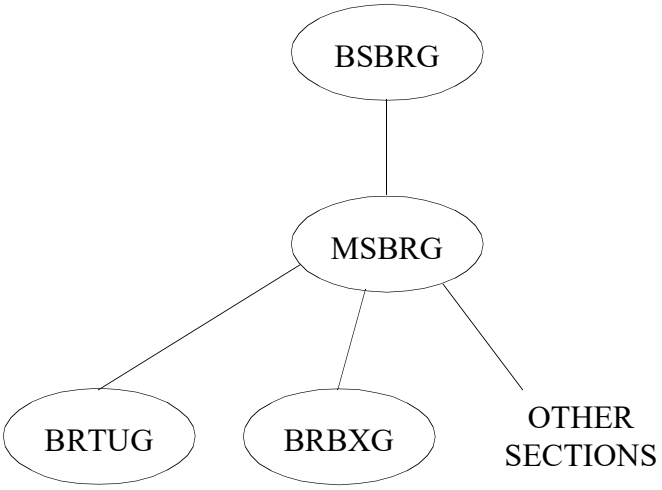


Figure 3-22 Calling Tree to Generate Nonlinear Sensitivity Data

BSBRG calls the bar evaluator routine MSBRG/S for the BAR element. The evaluator, in turn, calls the routines for each section. The routines are named BRXXGD, where XX is a two-letter identifier for the section. For example, the routine for the TUBE section is called BRTUG. The details for various routines are given in Listing 3-10.

Listing 3-10 BSBRG Subroutine

```
      SUBROUTINE BSBRG(GRPID,ENTYP,TYPE,DIMI,NDIMI,ANONL,NPROP,ERROR)
C
C ======================================================================
C   Purpose
C     To get the nonlinear factors of sensitivities for default sections
C
C   Arguments
C
C       GRPID input  integer      ID of group name
C       ENTYP input  integer      1: PBAR1, 2: PBEAM1
C       TYPE  input  character*8  Type name of cross-section
C       DIMI  input  double       Array from EPT record
C       NDIMI input  integer      Number of dimensions (Plus NSM)
C       ANONL output double       Array providing the nonlinear factors
C                                 for sensitivity relationships
C       NPROP input  integer      Number of properties in PBAR entries
C       ERROR output integer      Type of error
C
C   Method
C             Simply transfer control based on entry types
C   Called by
C             BCBRGD
C
C   Calls
C             MSBRG
```

```
C ---------------------------------------------------------------------
      IMPLICIT INTEGER (I-N)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

C     Calling sequence arguments
      INTEGER          GRPID,ENTYP,NDIMI,NPROP,ERROR
      CHARACTER*8      TYPE
      DOUBLE PRECISION ANONL(NPROP,NDIMI), DIMI(NDIMI)

C     Local variables
      INTEGER NAM(2)

C     Local data
      DATA NAM/4HBSBR,4HGD  /
C =====================================================================

      GRPID = 1
      IF(ENTYP .EQ. 1) THEN
         CALL MSBRG(TYPE,DIMI,NDIMI,ANONL,NPROP,ERROR)
     ELSE
         ERROR = 5300
      END IF
C ---------------------------------------------------------------------
      RETURN
      END
```

MSBRG Subroutine

MSBRG is a branched routine for providing information on the calculation of nonlinear gradients for each of the bar types. You may rename this routine as you like or move its function to BSBRG. MSBRG calls the BRXXGD routines, where XX is the two-letter keyword for various section types. The routine MSBRG for the MSC standard library is given in Listing 3-11.

Listing 3-11 MSBRG Subroutine

```
      SUBROUTINE MSBRG(TYPE,DIMI,NDIMI,ANONL,NPROP,ERROR)
C
C =====================================================================
C   Purpose
C     To get the nonlinear factors of sensitivities for PBAR1 entries
C
C   Arguments
C
C       TYPE  input  character*8  Type name of cross-section
C       DIMI  input  double       Array from EPT record
C       NDIMI input  integer      Number of dimensions (Plus NSM)
C       ANONL output double       Array providing the nonlinear factors
C                                 for sensitivity relationships
C       NPROP input  integer      Number of properties in PBAR entries
C       ERROR output integer      Type of error
C
C   Method
C             Simply transfer information based on cross-section type
C   Called by
C             BSBRG
C
C   Calls
```

```
C             BRRDGD,BRBRGD,BRBXG,BRTUG,BRIIGD,BRTTGD,BRLLGD,BRCHGD
C             BRCRGD,BRHHGD,BRT1GD,BRT2GD,BRI1GD,BRC1GD,BRC2GD,BRZZGD
C             BRHXGD,BRB1GD,BRHTGD,ZEROD(Nastran utility)
C ----------------------------------------------------------------------
      IMPLICIT INTEGER (I-N)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

C     Calling sequence arguments
      INTEGER          NDIMI,NPROP,ERROR
      CHARACTER*8      TYPE
      DOUBLE PRECISION ANONL(NPROP,NDIMI), DIMI(NDIMI)
C ======================================================================

      CALL ZEROD( ANONL(1,1), NPROP*NDIMI )
      ERROR = 0

      IF(TYPE.EQ.'ROD')  THEN
            CALL  BRRDGD(DIMI,NDIMI,ANONL,NPROP,ERROR)
      ELSE IF(TYPE.EQ.'BAR')  THEN
             CALL BRBRGD(DIMI,NDIMI,ANONL,NPROP,ERROR)
      ELSE IF(TYPE.EQ.'BOX')  THEN
             CALL BRBXG(DIMI,NDIMI,ANONL,NPROP,ERROR)
      ELSE IF(TYPE.EQ.'TUBE')  THEN
             CALL BRTUG(DIMI,NDIMI,ANONL,NPROP,ERROR)
      ELSE IF(TYPE.EQ.'I')  THEN
             CALL BRIIGD(DIMI,NDIMI,ANONL,NPROP,ERROR)
      ELSE IF(TYPE.EQ.'L')  THEN
             CALL BRLLGD(DIMI,NDIMI,ANONL,NPROP,ERROR)
      ELSE IF(TYPE.EQ.'T')  THEN
             CALL BRTTGD(DIMI,NDIMI,ANONL,NPROP,ERROR)
      ELSE IF(TYPE.EQ.'CHAN')  THEN
             CALL BRCHGD(DIMI,NDIMI,ANONL,NPROP,ERROR)
      ELSE IF(TYPE.EQ.'CROSS') THEN
             CALL BRCRGD(DIMI,NDIMI,ANONL,NPROP,ERROR)
      ELSE IF(TYPE.EQ.'H') THEN
             CALL BRHHGD(DIMI,NDIMI,ANONL,NPROP,ERROR)
      ELSE IF(TYPE.EQ.'T1') THEN
             CALL BRT1GD(DIMI,NDIMI,ANONL,NPROP,ERROR)
      ELSE IF(TYPE.EQ.'I1') THEN
             CALL BRI1GD(DIMI,NDIMI,ANONL,NPROP,ERROR)
      ELSE IF(TYPE.EQ.'CHAN1') THEN
             CALL BRC1GD(DIMI,NDIMI,ANONL,NPROP,ERROR)
      ELSE IF(TYPE.EQ.'Z') THEN
             CALL BRZZGD(DIMI,NDIMI,ANONL,NPROP,ERROR)
      ELSE IF(TYPE.EQ.'CHAN2') THEN
             CALL BRC2GD(DIMI,NDIMI,ANONL,NPROP,ERROR)
      ELSE IF(TYPE.EQ.'T2') THEN
             CALL BRT2GD(DIMI,NDIMI,ANONL,NPROP,ERROR)
      ELSE IF(TYPE.EQ.'BOX1') THEN
             CALL BRB1GD(DIMI,NDIMI,ANONL,NPROP,ERROR)
      ELSE IF(TYPE.EQ.'HEXA') THEN
             CALL BRHXGD(DIMI,NDIMI,ANONL,NPROP,ERROR)
      ELSE IF(TYPE.EQ.'HAT') THEN
             CALL BRHTGD(DIMI,NDIMI,ANONL,NPROP,ERROR)
      ELSE
         ERROR = 5310
      ENDIF
```

```
C
      RETURN
      END
```

BRTUG Subroutine

BRTUG is an example routine that shows how to define the nonlinear gradients of the TUBE section as a function of the outer and inner radius of the tube.

Listing 3-12 BRTUG Subroutine

```
      SUBROUTINE BRTUG(DIMI,NDIMI,ANONL,NPROP,ERROR)
C
C ======================================================================
C   Purpose
C     To get the nonlinear factors of sensitivities for TUBE section
C
C   Arguments
C
C       DIMI  input  double       Array of EPT records (Dimi+NSM)
C       NDIMI input  integer      Number of dimensions (Plus NSM)
C       ANONL output double       Array providing the nonlinear factors
C                                 for sensitivity relationships
C       NPROP input  integer      Number of properties in PBAR entries
C       ERROR output integer      Type of error
C
C   Method
C             Simply calculates the nonlinear factors
C   Called by
C             MSBRG
C ----------------------------------------------------------------------
      IMPLICIT INTEGER (I-N)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

C     Calling sequence arguments
      INTEGER          NDIMI,NPROP,ERROR
      DOUBLE PRECISION ANONL(NPROP,NDIMI), DIMI(NDIMI)

C     Nastran common blocks
      COMMON /CONDAD/ PI
C ======================================================================

      DIM1       = DIMI(1)
      DIM2       = DIMI(2)
      PDIM13     = PI*DIM1**3
      PDIM23     = PI*DIM2**3
C
      ANONL(1,1) = 2*PI*DIM1
      ANONL(1,2) = -2*PI*DIM2
      ANONL(2,1) = PDIM13
      ANONL(2,2) = -PDIM23
      ANONL(3,1) = PDIM13
      ANONL(3,2) = -PDIM23
      ANONL(4,1) = 2*PDIM13
      ANONL(4,2) = -2*PDIM23
```

```
C
      RETURN
      END
```

BSBRC Subroutine

The BSBRC subroutine allows you to place constraints on values the beam dimensions can take during a design task. It is not needed unless optimization is used and, even then, is available only to impose conditions on the dimensions to keep the optimization process from selecting physically meaningless dimensions.

For example, the optimizer might select a TUBE design with the inner radius greater than the outer radius because this allows for a negative area and therefore a negative weight (something a weight minimization algorithm loves!) These constraints are not the same as the PMIN and PMAX property limits that are imposed on the DVPREL1 entry. Instead, these are constraints that occur between or among section dimensions. A DRESP2 entry could be used to develop the same design constraints, but the subroutine reduces the burden on the user interface, the primary goal of the beam library project. The calling tree used for the MSC standard library, which is shown in Figure 3-23.

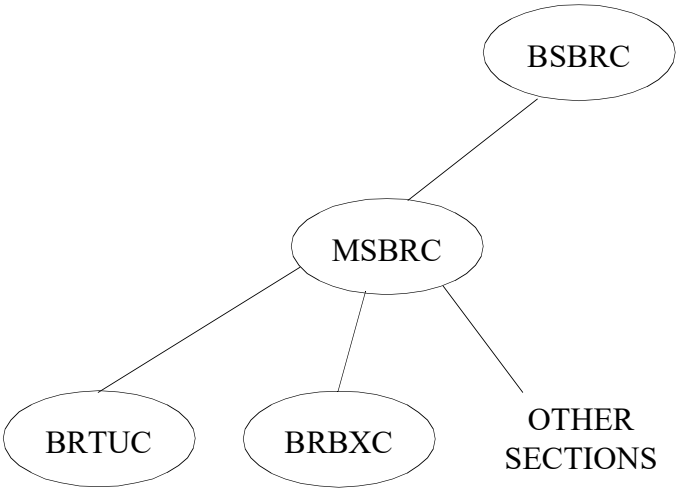


Figure 3-23 Calling Tree to Generate Beam Dimension Constraints

BSBRC calls the bar evaluator routine MSBRC. The evaluator in turn calls the routines for each section that requires constraints. The routines are named BRXXCD, where XX is a two-letter identifier for the section. For example the routine for the TUBE section is called BRTUC. The details for various routines are given in Listing 3-13.

Listing 3-13 BSBRC Subroutine

```
      SUBROUTINE BSBRC(GRPID,ENTYP,TYPE,AFACT,NCONST,NDIMI,ERROR)
C
C ======================================================================
C   Purpose
C           To get constraint information for default types
C
```

```
C   Arguments
C
C       GRPID  input integer     ID of the group name
C       ENTYP  input integer     1: PBAR1, 2: PBEAM1
C       TYPE   input character*8 Section type
C       NCONST input integer     Number of constraints
C                                for the section type
C       NDIMI  input integer     Number of dimensions
C                                for the section type
C       AFACT  output double     The factor for the NDIMI dimension in
C                                the constraint relation. Dimensions are
C                                NCONST by NDIMI.
C       ERROR  output integer    type of error
C
C   Method
C            Simply transfers control based on PBAR1 or PBEAM1 entries
C
C   Called by
C            BCBRCD
C
C   Calls
C            MSBRC
C ----------------------------------------------------------------------
      IMPLICIT INTEGER (I-N)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

C     Calling sequence arguments
      CHARACTER*8          TYPE
      INTEGER              GRPID,ENTYP,NCONST,NDIMI,ERROR
      DOUBLE PRECISION     AFACT(NCONST,NDIMI)
C ======================================================================
C
      GRPID = 1
      IF(ENTYP.EQ.1) THEN
         CALL MSBRC(TYPE,AFACT,NCONST,NDIMI,ERROR)
      END IF
C ----------------------------------------------------------------------
      RETURN
      END
```

MSBRC Subroutine

MSBRC is a branched routine for providing information on the calculation of gradients for each of the bar types. You may rename this routine as you like or move its function to BSBRC. MSBRC calls the BRXXCD routines, where XX is the two-letter keyword for various section types. The routine MSBRC for the MSC standard library is given in Listing 3-14.

Listing 3-14 MSBRC Subroutine

```
      SUBROUTINE MSBRC(TYPE,AFACT,NCONST,NDIMI,ERROR)
C=====================================================================
C   Purpose
C           To get constraint information for PBAR1 types
C
C   Arguments
C
C       TYPE   input character*8 Section type
```

```
C       AFACT  output double     The factor for the NDIMI
C                                dimension in the constraint
C                                relation. Dimensions are NCONST
C                                by NDIMI.
C       NCONST input integer     Number of constraints
C                                for the section type
C       NDIMI  input integer     Number of dimensions
C                                for the section type
C       ERROR  output integer    type of error
C
C   Method
C              Simply transfers control based on group name
C
C   Called by
C              BSBRCT
C
C   Calls
C              BRTUC, BRBXC, BRIICD, BRTTCD, BRLLCD, BRCHCD,
C              BRCRCD, BRHHCD, BRT1CD, BRI1CD, BRC1CD, BRZZCD,
C              BRC2CD, BRT2CD, BRB1CD, BRHXCD, BRHTCD, ZEROD
C   (Nastran utility)
C ---------------------------------------------------------------------
      IMPLICIT INTEGER (I-N)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

C     Calling sequence arguments
      CHARACTER*8          TYPE
      INTEGER              NCONST,NDIMI,ERROR
      DOUBLE PRECISION     AFACT(NCONST,NDIMI)

C ======================================================================

       CALL ZEROD( AFACT(1,1), NCONST*NDIMI )
       ERROR = 0

       IF(TYPE.EQ.'TUBE') THEN
          CALL BRTUC(AFACT,NCONST,NDIMI,ERROR)
       ELSE IF(TYPE.EQ.'BOX') THEN
          CALL BRBXC(AFACT,NCONST,NDIMI,ERROR)
       ELSE IF(TYPE.EQ.'I') THEN
          CALL BRIICD(AFACT,NCONST,NDIMI,ERROR)
       ELSE IF(TYPE.EQ.'L') THEN
          CALL BRLLCD(AFACT,NCONST,NDIMI,ERROR)
       ELSE IF(TYPE.EQ.'T') THEN
          CALL BRTTCD(AFACT,NCONST,NDIMI,ERROR)
       ELSE IF(TYPE.EQ.'CHAN') THEN
          CALL BRCHCD(AFACT,NCONST,NDIMI,ERROR)
       ELSE IF(TYPE.EQ.'CROSS') THEN
          CALL BRCRCD(AFACT,NCONST,NDIMI,ERROR)
       ELSE IF(TYPE.EQ.'H') THEN
          CALL BRHHCD(AFACT,NCONST,NDIMI,ERROR)
       ELSE IF(TYPE.EQ.'T1') THEN
          CALL BRT1CD(AFACT,NCONST,NDIMI,ERROR)
       ELSE IF(TYPE.EQ.'I1') THEN
```

```
            CALL BRI1CD(AFACT,NCONST,NDIMI,ERROR)
         ELSE IF(TYPE.EQ.'CHAN1') THEN
            CALL BRC1CD(AFACT,NCONST,NDIMI,ERROR)
         ELSE IF(TYPE.EQ.'Z') THEN
            CALL BRZZCD(AFACT,NCONST,NDIMI,ERROR)
         ELSE IF(TYPE.EQ.'CHAN2') THEN
            CALL BRC2CD(AFACT,NCONST,NDIMI,ERROR)
         ELSE IF(TYPE.EQ.'T2') THEN
            CALL BRT2CD(AFACT,NCONST,NDIMI,ERROR)
         ELSE IF(TYPE.EQ.'BOX1') THEN
            CALL BRB1CD(AFACT,NCONST,NDIMI,ERROR)
         ELSE IF(TYPE.EQ.'HEXA') THEN
            CALL BRHXCD(AFACT,NCONST,NDIMI,ERROR)
         ELSE IF(TYPE.EQ.'HAT') THEN
            CALL BRHTCD(AFACT,NCONST,NDIMI,ERROR)
         ELSE
            ERROR = 5500
         END IF

C
         RETURN
         END
```

BRTUC Subroutine

BRTUC is an example routine that shows how to define the constraints for a bar section. This example routine is for the TUBE section and imposes a single constraint that $-DIM1 + DIM2 < 0.0$, where DIM1 is the outer radius and DIM2 is the inner radius of the tube. The constraints should always be specified so that the specified linear combination of dimensions is less or equal to zero when the constraint is satisfied.

Listing 3-15 BRTUC Subroutine

```
      SUBROUTINE BRTUC(AFACT,NCONST,NDIMI,ERROR)
C
C ======================================================================
C   Purpose
C           To get constraint information for TUBE type
C
C   Arguments
C
C     AFACT   output double   The factor for the NDIMI dimension in the
C                             constraint relation. Dimensions are NCONST
C                             by NDIMI.
C     NCONST  input  integer  Number of constraints for the section type
C     NDIMI   input  integer  Number of dimensions for the section type
C     ERROR   output integer  type of error
C
C   Method
C             Simply transfers constraint information
C
C   Called by
C             MSBRC
C ----------------------------------------------------------------------
      IMPLICIT INTEGER (I-N)
```

```
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

C     Calling sequence arguments
      INTEGER              NCONST,NDIMI,ERROR
      DOUBLE PRECISION     AFACT(NCONST,NDIMI)
C =====================================================================
C
      IF(NCONST.NE.1) THEN
         ERROR = 5502
         RETURN
      END IF
C
      AFACT(1,1) = -1.0D0
      AFACT(1,2) =  1.0D0

C
      RETURN
      END
```

BSMSG Subroutine

The Error handling is performed by a subroutine called bsmsg.f. This routine has the following parameters:

```
SUBROUTINE BSMSG(GRPID,ERRCOD,MXLEN,Z,ERROR)
C =====================================================================
C   Purpose
C     To handle the error messages for User Defined Group.
C ---------------------------------------------------------------------
C   Arguments
C
C   GRPID  input  int    ID of group or group name - not used
C   ERRCOD input  int    Error message number if any found
C   MXLEN  input int    Maximum length of the message that can be
C   passed
C   Z      output char   Array to contain the message return
C   ERROR  output int    The code returned indicates the type of error
C
C   Called by
C             BCMSG routine
C--------------------------------
```

The purpose of this subroutine is to return an error String associated with an error code. The error codes are to be returned by the other 7 “BS...” routines and as such the “BSMSG” routine is used as the repository of all of the error messages for the Beam Library applications.

For example, suppose that “BSBRP” application returns an error code of 5103 when a certain error condition occurs. Then, the Beam Library Client routines will expect that there will be a String returned from the “BSMSG” routine, which corresponds to this error code. These error messages will be printed in the “*.f06” file to guide the user as to what the error could have been and how to fix it. The string may be as long as 160 characters for this release of MSC Nastran.

The following is an example of BSMSG code construct:

```
      IF(ERRCOD .EQ. 5103) THEN
      Z(1:MXLEN) =  'This is a User Specified Error Message ....'//
     &              'Messages could be 160 characters long ...'
```

```
      ERROR = ERRCOD
      .....
```

Again, it is highly recommended that you use the example Beam Server files as a template to generate your BSMSG routines.

Linking Your Library to MSC Nastran

Once you have created the eight "BS..." routines, these routines may be linked with MSC Nastran Beam Server Library to build a beam server executable. It is highly recommended that you study, build, and use the example "Beam Server" before you build your own version of the beam server.

The MSC Nastran special library contains a main routine as well as the communications routines that allow MSC Nastran to communicate with the user-defined beam server.

The MSC Nastran user may connect up to 10 beam servers in a single job execution. This connection is made using the concept of evaluator groups described in the remainder of this section. For each group, the user specifies on the PBEAML/PBARL entry referring to an external beam server, MSC Nastran will start and communicate with the beam server.

You may define as many beam evaluators as required using the "CONNECT" FMS commands. Only 10 of these evaluators, however, may be referenced in the groups on the PBARL or PBEAML Bulk Data entries.

The PBARL/PBEAML entries specifies a "Group" name on the fourth field of the first entry. This group name is associated with an "Evaluator" class using a "Connect" command in the FMS section. Finally, the "Evaluator" class is associated with an executable using the MSC Nastran configuration file specified via the "gmconn" key word on the nastran command line.

The following example shows the mappings mentioned:

1. Group is referenced on the PBARL/PBEAML entry (or entries).

   PBARL,39,6,LOCSERV,I_SECTION (Specify the "Group" name.)
2. The "Group" is associated with an "Evaluator" class.

   CONNECT,BEAMEVAL,LOCSERV,EXTBML (Associate the "Group" name with an "Evaluator" class.)
3. The external evaluator connection file associates the "Evaluator" class with a server executable. The following statement must be specified in the connection file:

   EXTBML,-,*beam_server_pathname*
4. Refer to the external evaluator connection file on the command line using the "gmconn" keyword

   nastran myjob ... gmconn=*external_evaluator_pathname.*
5. In the example Beam Library section that standard input (FORTRAN unit 5) or standard output (FORTRAN unit 6) are not used as these I/O channels are reserved by the Inter-Process Communications (IPC) subsystem.

Example of Building and Linking a Beam Server

As an example of building and linking a beam server executable, the sample beam server will be modified. Complete instructions on building and using a beam server are provided in the *MSC Nastran 2020 Utilities Guide* for your system.

- Make a copy of the beam server sample source.

- Edit the source for the BRTUP subroutine; this routine describes the equations that convert the PBARL dimensions into the standard PBAR dimensions for a tube cross section.
- Add an extra multiplication of 3.0 to the DIMO(2) equation to increase the calculated moments of inertia.

Since the formulation of this bar section has been changed, the sensitivities for optimization will also change. Rather than calculate what the new sensitivities should be, the MSC Nastran can calculate them using central differencing techniques. To permit this, edit the source file for the BRTUI subroutine and change all occurrences of SENTYP = 2 to SENTYP = 3.

Build your new beam server using the instructions detailed in the *MSC Nastran 2016 Utilities Guide*. Once you have built the beam server executable, you must create an external evaluator connection to point to your executable. Typically, this file would be kept in the user's home directory, but for this example it will remain in the current directory. Edit the new file bmconfig.fil. Put the following line in the file:

`LOCBMLS,-,`*`pathname`*

where LOCBMLS is the evaluator referenced in the SAMPLE data file included with the beam server. Remember, this file can contain references to any number of beam servers.

To run the sample job, type in the following command:

```
nastran sample scr=yes bat=no gmconn=bmconfig.fil
```

Common problems which may occur when attempting to run an external beam library job are generally indicated in the F06 by USER FATAL MESSAGE 6498. If this message includes the text “No such group defined,” the PBARL/PBEAML selected a group not defined on a CONNECT entry. If UFM 6498 includes the text “No such evaluator class,” either the “gmconn” keyword was not specified or the CONNECT entry selected an evaluator not defined in the configuration file.

If the job was successful, you can look at the Design Variable History and see that the results for the variable mytubeor are different than the results for tubeor. These variables refer to the outer radius of tube sections from equivalent models. One model used the MSC tube section while the other used the tube section in your modified beam server.

Support for Offsets in Element Coordinates for CBAR & CBEAM Elements

Bar elements (`CBAR` and `CBEAM` Bulk Data entries) have been used for many years to model structural members that can transmit axial loads, shear forces and bending moments. The offset feature of these elements allows the analyst to position the beam axis relative to the line connecting the two grid points used in the definition of the element.

Theory

Structural stiffeners such as I-beams, wide flange beams, channel sections, z-sections and hat sections are typically modeled using the bar and beam finite elements available in MSC Nastran. When these members are used to support flat or curved panel structures, it is often desirable to include the offset effects of the member neutral/shear axis location when modeling the structure. These effects are most generally prevalent when the grid points are located at the mid-plane or outer surface of the panels. Under these circumstances, when a beam element is attached to these points, the line connecting the beam element end grid points does not correctly represent the element neutral axis location relative to the panel surface. To handle this situation, the `CBAR` and `CBEAM` Bulk Data entries provide for the specification of offset vectors for each end of the element that locate the positions of the element end points with respect to the locations of the element connection grid points.

Until now, the components of the offset vectors had to be specified in the principal directions of the output coordinate systems of the connection grid points. The program would then transform each of the offset vectors so that the components were measured in the basic coordinate reference system of the model. The locations of the end grid points are also converted to this basic system if necessary. Adding the offsets to the coordinates of the grid points produces the element end point locations. The vector between the two end points together with the orientation vector can be used to generate the element coordinate system. Thus, when defining element offset vectors, the analyst must keep in mind the fact that the vectors must be specified in the output coordinate system directions of the connection grid points. For example, in the following figure, the local x direction at grid point GA is in the opposite direction of the local y direction at grid point GB and the local y at grid point GA and the local x at grid point GB are in the same direction. Let us assume that the element x-axis is offset 0.5 units in the local x direction at grid point GA from the line connecting grid points GA and GB. Then, in order to define the neutral axis of the element properly, the analyst must define the End A offset vector as Wa = {0.5, 0.0, 0.0} and at the same time specify the End B offset vector as Wb = {0.0, -0.5, 0.0}.

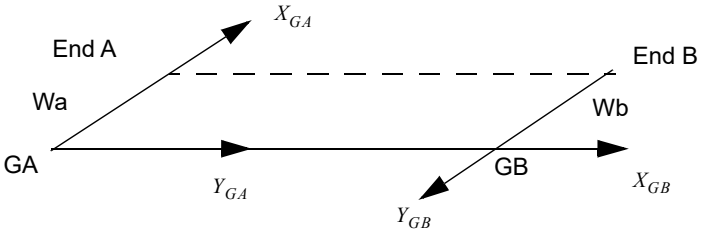


Figure 3-24

If we use the concept of the local element coordinate system, then no discussion of the local output coordinate systems employed at the connection grid points is necessary. For this case, the element x-axis is from GA to GB and the orientation vector is in the direction of X_{GA}, defining the direction of the element- y-axis and the x-y plane. The offset vector components will be the same at both ends and will be specified as Wa = Wb = {0.0, 0.5, 0.0}. The program will take care of any necessary transformations between the element system, the basic system and the local systems at the end grid points.

Inputs

The specification of the offset vectors for the two ends of the bar and beam elements is still accomplished using fields four through nine of the first continuation entry for the CBAR and CBEAM Bulk Data entries respectively. The orientation (reference) vector is specified using fields 6, 7 and 8 of the bulk data entries. How MSC Nastran interprets the contents of these fields depends upon the offset and reference vector specification codes supplied in field nine of the main (parent) bulk data entry. Each of the entries is described by one character. There must be three characters in the string if it is present. The first character is associated with the end A offset vector. The second character is associated with the end B offset vector. The third character is associated with the orientation vector. The character “G” signifies that the input is specified in the Global coordinate reference system. The character “E” signifies that the input is specified in the Element local coordinate reference system. The character “B” signifies that the input is specified in the Basic coordinate reference system. Offset vectors can be defined in Global (default) or Element systems. The orientation vector can be defined in Global (default) or Basic systems. The following table describes the various valid character combinations that can be entered in this field.

String	End A Offset	End B Offset	Orientation Vector
GGG	Global	Global	Global
GGB	Global	Global	Basic
GEG	Global	Element	Global
GEB	Global	Element	Basic
EGG	Element	Global	Global
EGB	Element	Global	Basic
EEG	Element	Element	Global
EEB	Element	Element	Basic

The default entry for this field is “GGG”. This indicates that all vectors are defined in the global reference system. This is the historical specification for the vector inputs. Any attempt to specify invalid combinations results in a bulk data entry input error message. For example, a value of EEE indicates offset vectors for end A and end B, as well as the orientation vector, are all specified in the local Element reference system. This usage results in a fatal error because the orientation vector can be specified only in Global or Basic systems but not the Element local system.

Outputs

There are no special output considerations when the offset vectors are specified in the local element coordinate system.

Guidelines and Limitations

This feature is not applicable to the CBEAM entry when it is acting as a p-element beam.

Example

A simple model is presented to demonstrate the specification of the offset vectors in the element coordinate system. The input file contains two CBAR elements that use offsets. One of the CBARs uses the original method of specifying the offsets in the global system components at the connection grid points. The second of the CBARs uses the new technique of specifying the offsets in the element coordinate system. The displacements and element forces should be the same for the two elements if the offset vectors are “equivalent”.

Model Description

Two CBAR elements are used to demonstrate the use of the offset vectors in the element coordinate system. The grid points that define the ends of the elements are situated along the basic y-axis. The output coordinate system at each grid point is the basic coordinate system. The global system is equivalent to the basic system. The orientation vector for both elements places the element x-y plane at a thirty degree cant to the basic y-z plane toward the negative x-direction. Element 101 defines the offsets to be -0.50 units in the global (basic) x-direction and 0.866 units in the global (basic) z-direction. This is standard legacy usage of the offset vectors. Element 111 defines the offsets to be 1.0 units in the y-direction of the local

element coordinate system. One end of each bar is completely constrained resulting in a cantilever beam model. Three subcases are defined to apply unit loads to the free end of the beam in each of the basic directions. Element forces are recovered for the CBAR elements for each subcase.

Input File

```
id test,cbar
sol 101
cend
$
title = v2004 release notes demonstration problem
subtitle = test cbar offset vector specification options
label= two equivalent cbars
disp=all
spcf=all
elfo=all
spc = 100
$
$  unit load in basic x-direction
$
subcase 100
load=100
$
$  unit load in basic y-direction
$
subcase 200
load = 200
$
$  unit load in basic z-direction
$
subcase 300
load = 300
begin bulk
$
$===============================================================
$   offset bar example.
$   grid output cstm is basic
$   offset in element coordinate system is {0.0,1.0,0.0}(t)
$   which is equivalent to {-0.5,0.0,0.866025}(t) in basic
$   element force results should be the same for both elements
$
grid,1001,,0.0,11.0,0.0
grid,1002,,0.0,20.0,0.0
$
$  Standard legacy input.
$  Element 101 specifies offsets and orientation vector in the
$  global coordinate system.
$
cbar,101,1101,1001,1002,-0.50,0.0,.866025,,+br101
+br101,,,-0.50,0.0,0.866025,-0.50,0.0,0.86602
$
$  New feature option input.
$  Element 111 specifies offsets in the local element system and
$  orientation vector in the global coordinate system.
$
cbar,111,1101,1001,1002,-0.50,0.0,.866025,eeg,+br111
```

```
+br111,,,0.0,1.0,0.0,0.0,1.0,0.0
$
force,100,1002,,1000.0,1.0,0.0,0.0
force,200,1002,,1000.0,0.0,1.0,0.0
force,300,1002,,1000.0,0.0,0.0,1.0
spc1,100,123456,1001
$
$=============================================================
$   common data
$
pbar,1101,1,1.0,1.0,1.0,1.0
mat1,1,10.7+6,,.33
enddata
```

Surface Elements

The surface elements consist of the following:

- Shear panel
- Shell
- Conical shell

Shear Panel Element (CSHEAR)

The shear panel is defined with a CSHEAR entry and its properties with a PSHEAR entry. A shear panel is a two-dimensional structural element that resists the action of tangential forces applied to its edges, and the action of normal forces if effectiveness factors are used on the alternate form of the PSHEAR Bulk Data entry. The structural and nonstructural mass of the shear panel is lumped at the connected grid points. Details of the shear panel element are discussed in Section 3.0 of *The NASTRAN Theoretical Manual.*

The element coordinate system for a shear panel is shown in Figure 3-25. The labels G1, G2, G3, and G4 refer to the order of the connected grid points on the CSHEAR entry. The element forces are output on request in either real or complex form.

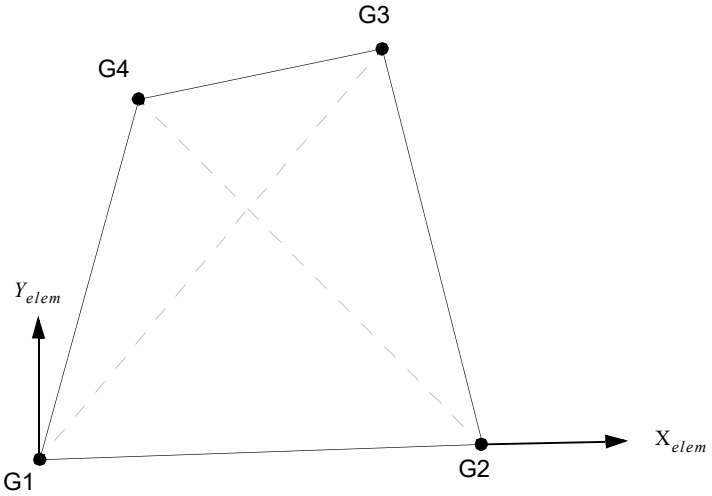


Figure 3-25 Shear Panel Connection and Coordinate System

The positive directions for these forces are indicated in Figure 3-26. These forces consist of the forces applied to the element at the corners in the direction of the sides, kick forces at the corners in a direction normal to the plane formed by the two adjacent edges, and shear flows (force per unit length) along the four edges. The shear stresses are calculated at the corners in skewed coordinates parallel to the exterior edges. The average of the four corner stresses and the maximum stress are output on request in either the real or complex form. A margin of safety is also output when the stresses are real.

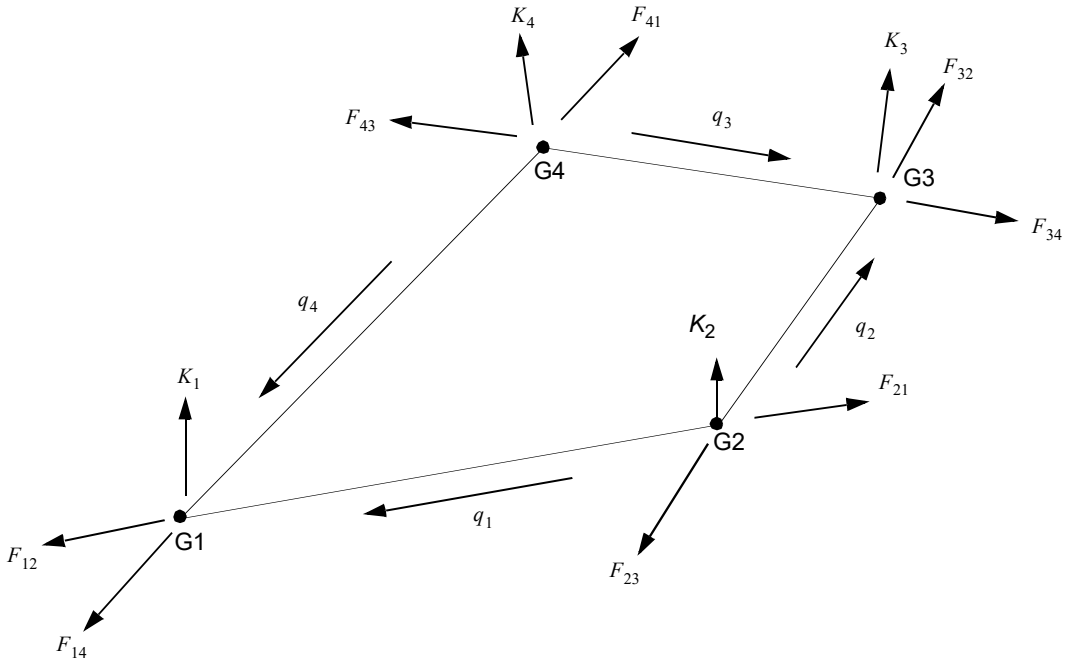


Figure 3-26 CSHEAR Element Corner Forces and Shear Flows

Shell Elements (CTRIA3, CTRIA6, CTRIAR, CQUAD4, CQUAD8, CQUADR)

MSC Nastran includes two different shapes of isoparametric shell elements (triangular and quadrilateral) and two different stress systems (membrane and bending). There are in all a total of six different forms of shell elements that are defined by connection entries as follows:

- CTRIA3 -- Isoparametric triangular element with optional coupling of bending and membrane stiffness.
- CTRIA6 -- Isoparametric triangular element with optional coupling of bending and membrane stiffness and optional midside nodes.
- CTRIAR -- Isoparametric triangular element with no coupling of bending and membrane stiffness; the membrane stiffness formulation includes rotation about the normal to the plane of the element.
- CQUAD4 -- Isoparametric quadrilateral element with optional coupling of bending and membrane stiffnesses.
- CQUAD8 -- Isoparametric quadrilateral element with optional coupling of bending and membrane stiffness and optional midside nodes.
- CQUADR -- Isoparametric quadrilateral element with no coupling of bending and membrane stiffnesses; the membrane stiffness formulation includes rotation about the normal to the plane of the element.

The CQUAD4, CQUAD8, CTRIA6, and CTRIA3 elements may be used to model hyperelastic materials; see CQUAD4 and CQUADR Coordinate Systems, 125.

The properties for the above elements are defined on the PSHELL entry. Anisotropic material may be specified for all shell elements. Transverse shear flexibility may be included for all bending elements on an optional basis. Structural mass is calculated from the membrane density and thickness. Nonstructural mass can be specified for all shell elements. Lumped mass procedures are used unless coupled mass is requested with the parameter COUPMASS. Differential stiffness matrices are generated for all shell elements. Plane strain analysis may be requested for all shell elements.

The element coordinate systems for triangular and quadrilateral shell elements are shown in Figure 3-27 through Figure 3-30. The integers 1 through 8 refer to the order of the connected grid points on the connection entries defining the elements. The angle, THETA, is the orientation angle for material properties. The element coordinate system for the CQUAD8 and CTRlA6 elements is a Cartesian system defined locally for each point (ξ, η) with the following orientation:

- The plane containing x_{elem} and y_{elem} is tangent to the surface of the element.
- For the CQUAD8 element, x_{elem} and y_{elem} are obtained by doubly bisecting the lines of constant ξ and η.
- For the CTRIA6 element, x_{elem} is tangent to the line of constant η.
- x_{elem} increases in the general direction of increasing ξ and y_{elem} of η.

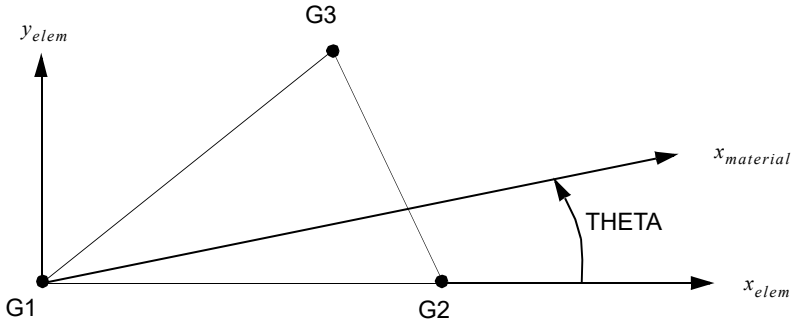


Figure 3-27 CTRIA3 and CTRIAR Coordinate Systems

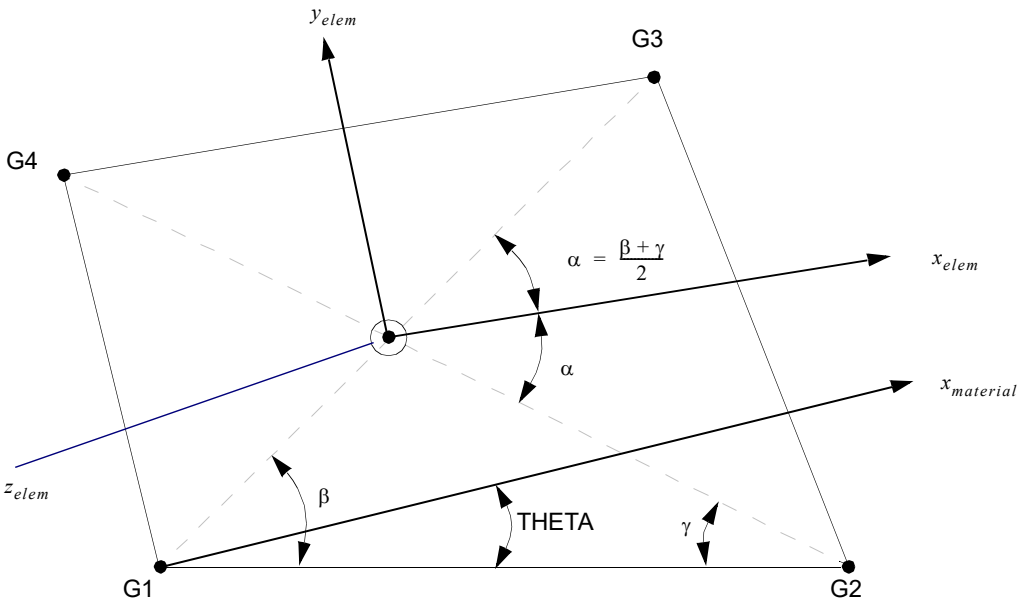


Figure 3-28 CQUAD4 and CQUADR Coordinate Systems

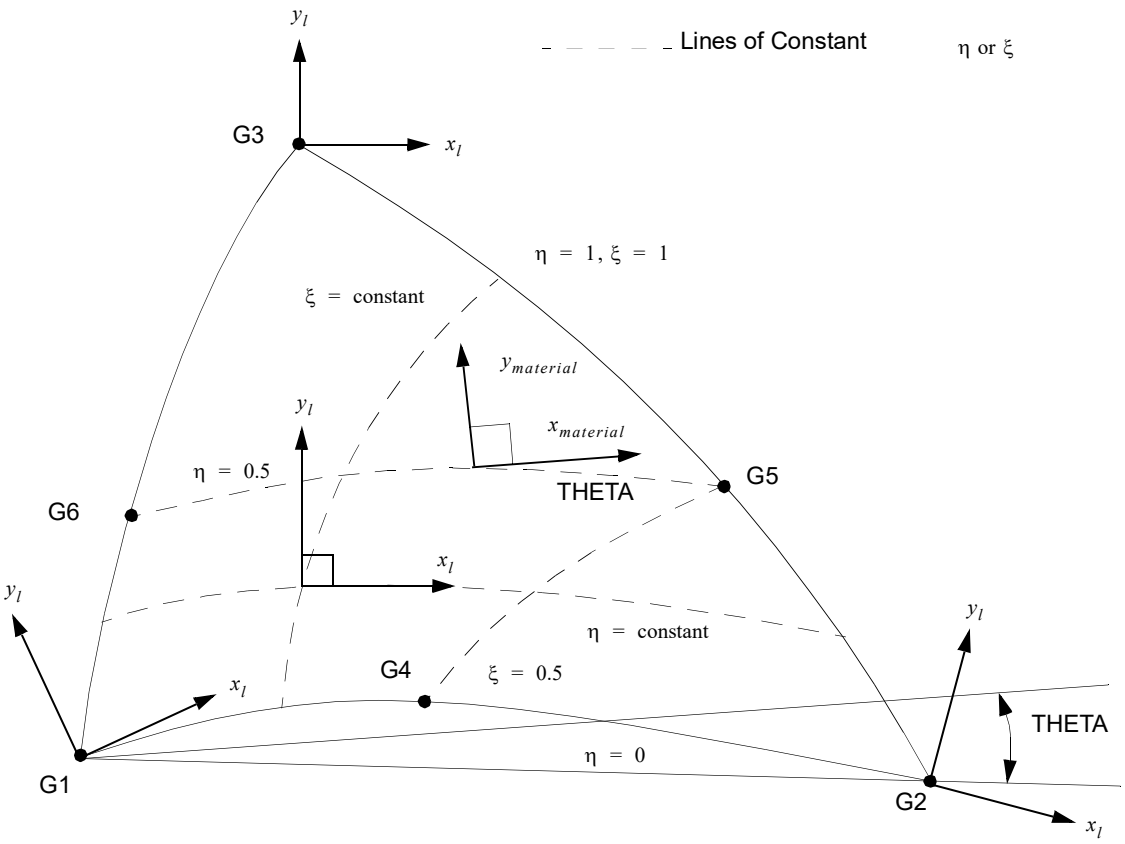


Figure 3-29 CTRIA6 Coordinate System

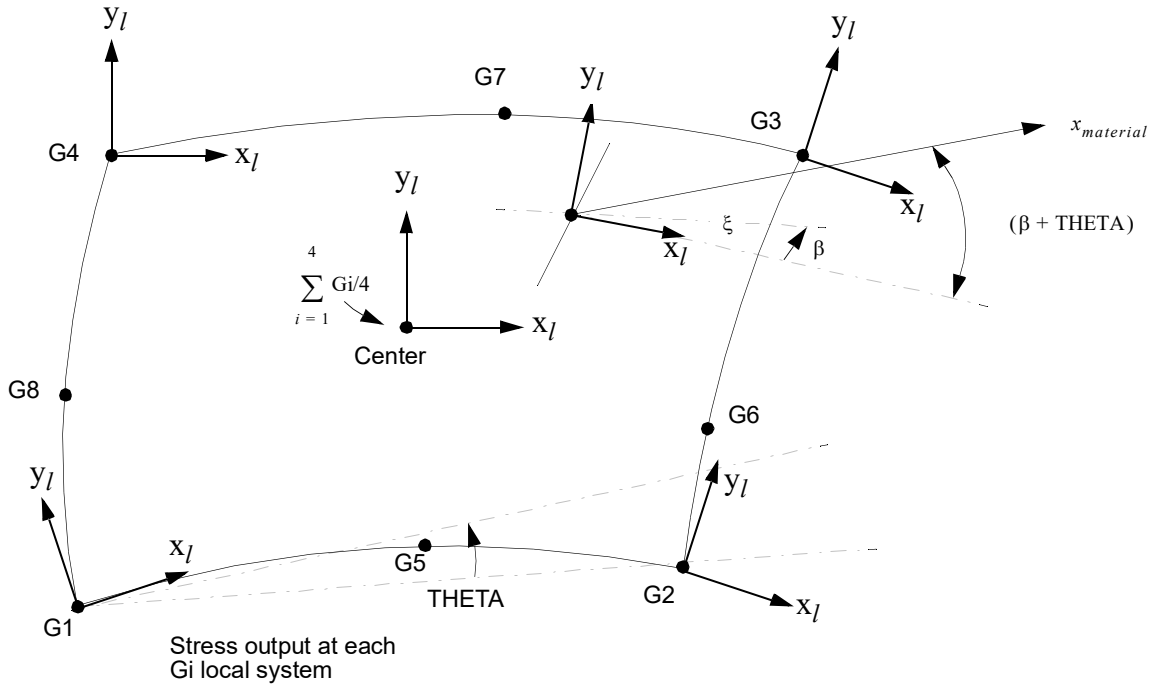


Figure 3-30 `CQUAD8` Coordinate System

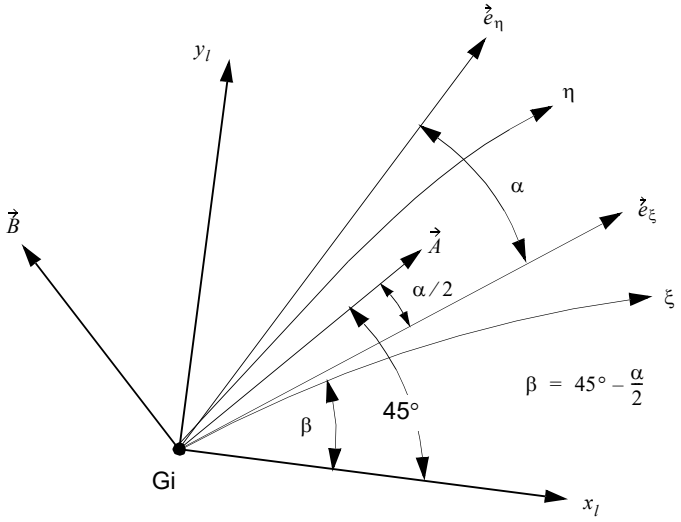


Figure 3-31 Figure 3-29 CQUAD8 Coordinate System (continued)

The CQUAD4, CQUAD8, CTRIA3, and CTRIA6 elements may be offset relative to the mean plane of their connected grid points. There are three commonly used techniques to define these offsets in MSC Nastran. These are generically denoted as the ZOFFS, MID4, and RBAR methods and are described below.

It is generally recommended the ZOFFS be used if one has a sufficiently fine mesh in the region in which offsets are to be defined. In regions of a finite element model in which the analyst has determined that a coarse mesh is satisfactory, the use of RBARs to define offsets is generally more accurate.

Irrespective of which of the three methods is used to define the offset for the element, it is required that values be specified for both MID1 and MID2 on the PSHELL entry referenced by the offset element. The mass properties of an offset element are not modified to reflect the existence of the offset when the ZOFFS and MID4 methods are used. Therefore, if the weight or mass properties of an offset element are to be used in an analysis, the RBAR method must be used to represent the offset.

The CQUADR and CTRIAR elements may be offset by the RBAR method only, but no membrane-bending coupling is computed.

For the CQUAD4 and the CTRIA3, the forces are evaluated at the centroid of the element. For the CQUAD8, CTRIA6, CQUADR, and CTRIAR, the forces are evaluated at the centroid and at the vertices. The positive directions for shell element forces in the element coordinate system are shown in Figure 3-32 and Figure 3-33. The following element forces per unit of length, either real or complex, are output on request:

- Bending moments on the *x* and *y* faces.
- Twisting moment.
- Shear forces on the *x* and *y* faces.
- Normal forces on the *x* and *y* faces.

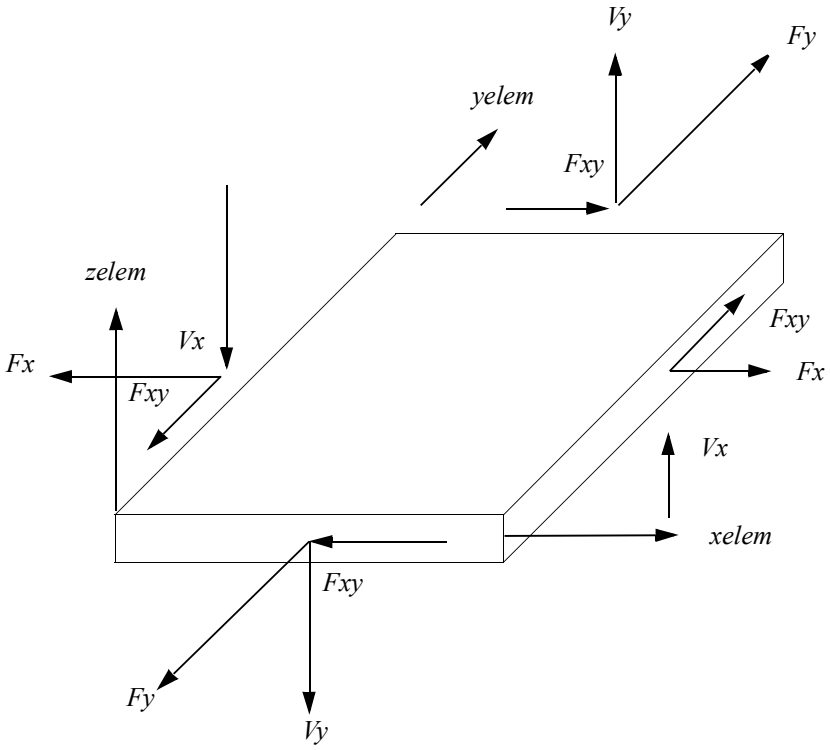


Figure 3-32 Forces in Shell Elements

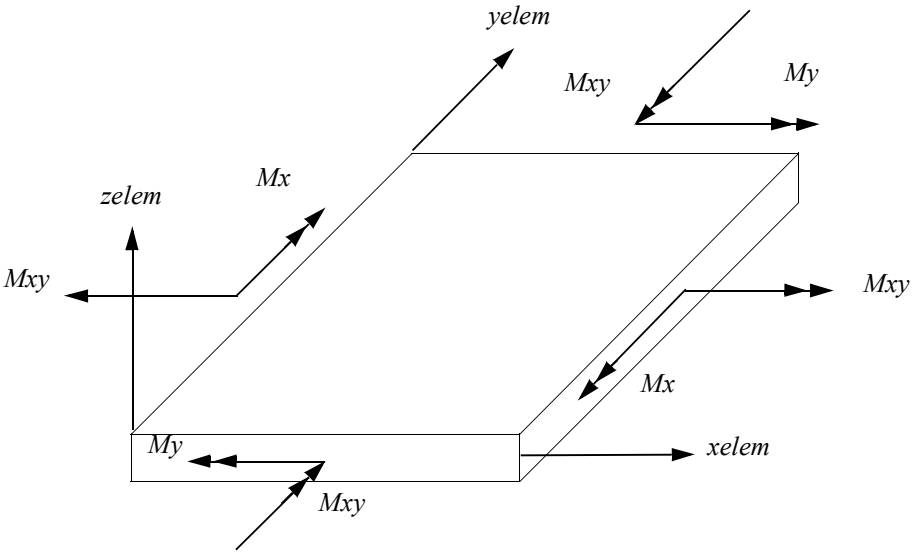


Figure 3-33 Moments in Shell Elements

For the CQUAD4 and CTRIA3 elements, the stresses are evaluated at the centroid of the element. For the CQUAD4, stress at the corner can be requested if STRESS(CORNER) is requested. For the CQUAD8, CTRIA6, CQUADR, and CTRIAR elements, the stresses are evaluated at the centroid and at the vertices. The positive directions for the stresses are shown in Figure 3-34. The stresses are calculated in the element coordinate system. The following real stresses are output on request:

- Normal stresses in the *x* and *y* directions.
- Shear stresses on the *x* face in the *y* direction.
- Angle between the *x* axis and the major principal axis.
- Major and minor principal stresses.
- von Mises equivalent stress if STRESS(VONM) is requested or maximum shear stress if STRESS(SHEAR) is requested.

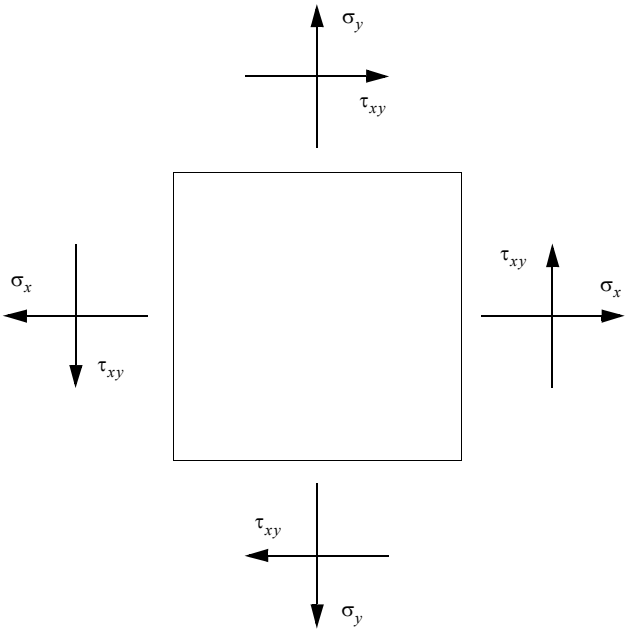


Figure 3-34 Stresses in Shell Elements

The von Mises equivalent stress for plane strain analysis is defined as follows:

$$\bar{\tau}_v = [1/2\{(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2\} + 3\tau_{xy}^2]^{1/2} \tag{3-8}$$

For plane stress analysis, $\sigma_z = 0$. Only the normal stresses and shearing stresses are available in the complex form.

The von Mises equivalent strain is defined as

$$\bar{\varepsilon}_v = \left[\frac{4}{9}(\varepsilon_x^2 + \varepsilon_y^2 - \varepsilon_x\varepsilon_y) + \frac{1}{3}\gamma_{xy}^2\right]^{1/2} \tag{3-9}$$

where the strain components are defined as

$$\varepsilon_x = \frac{\partial u}{\partial x};\ \varepsilon_y = \frac{\partial v}{\partial y};\ \gamma_{xy} = \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \tag{3-10}$$

and the curvatures are defined as

$$\chi_x = \frac{\partial^2 w}{\partial x^2};\ \chi_y = \frac{\partial^2 w}{\partial y^2};\ \chi_{xy} = 2\frac{\partial^2 w}{\partial x \partial y} \tag{3-11}$$

The maximum shear stress is

$$\hat{\tau} = \left[\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \tau_{xy}^2\right]^{1/2} \tag{3-12}$$

The maximum shear strain is

$$\hat{\gamma} = [(\varepsilon_x - \varepsilon_y)^2 + \gamma_{xy}^2]^{1/2} \tag{3-13}$$

The stresses are calculated at two specified points on the cross section. The distances to the specified points are given on the property entries. The default distance is one-half the thickness. The positive directions for these fiber distances are defined according to the right-hand sequence of the grid points specified on the connection entry.

In addition, interpolated grid point stresses and mesh stress discontinuities are calculated in user-specified coordinate systems for grid points which connect the shell elements. See Element Data Recovery Resolved at Grid Points, 435 for a description of the Case Control Section to request grid point stresses. Only real stresses are available at the grid points. Mesh stress discontinuities are available in linear static analysis only.

Grid point stresses are computed by

$$\sigma_g = \sum_{e=1}^{N_e} W_{ge}\sigma_{ge} \tag{3-14}$$

where σ_{ge}, a grid point stress component, multiplied by W_{ge}, the interpolation factor, and summed for all elements, N_e, connected to the grid point.

The stress discontinuity for one component and one element is

$$\delta_{ge} = \sigma_{ge} - \sigma_g \tag{3-15}$$

The discontinuity from all elements for one component is then obtained by

$$\delta_g = \sqrt{\sum_{e=1}^{N_e} (W_{ge}\delta_{ge})^2} \tag{3-16}$$

The total discontinuity at a grid point from all elements and all components N_c defines the error and is obtained by

$$\delta_g = \sqrt{\frac{1}{N_c}\sum_{c=1}^{N_c} \delta_{gc}^2} \tag{3-17}$$

δ_{gc} and δ_g are requested and printed with the GPSDCON Case Control command.

The total discontinuity at an element for one component over all of its grid points N_g is

$$\delta_{ec} = \sqrt{\frac{1}{N_g}\sum_{g=1}^{N_g}\delta_g^2} \tag{3-18}$$

and the error for an element is

$$\delta_{e} = \sqrt{\frac{1}{N_g}\sum_{g=1}^{N_g}\delta_g^2} \tag{3-19}$$

δ_{ec} and δ_{e} are requested and printed with the ELSDCON Case Control command.

The CQUAD4 element is intended for use when the surfaces are reasonably flat and the geometry is nearly rectangular. For these conditions, the quadrilateral elements eliminate the modeling bias associated with the use of triangular elements and the quadrilaterals give more accurate results for the same mesh size. If the surfaces are highly warped, curved, or swept, triangular elements should be used. Under extreme conditions, quadrilateral elements will give results that are considerably less accurate than triangular elements for the same mesh size. Quadrilateral elements should be kept as nearly square as possible, because the accuracy tends to deteriorate as the aspect ratio of the quadrilateral increases. Triangular elements should be kept as nearly equilateral as practicable because the accuracy tends to deteriorate as the triangle becomes obtuse and the ratio of the longest to the shortest side increases.

The CQUADR and CTRIAR elements are improved shell elements that, when compared to the CQUAD4 and CTRIA3, respectively, are much less sensitive to high aspect ratios and values of Poisson's ratio near 0.5. In these elements, a rotational stiffness is computed about the normal to the element at the vertices and used in the formulation of the element stiffness. It should be noted that this degree of freedom must not be constrained unless at a prescribed boundary. The CQUADR and CTRIAR element stiffness formulations do not include coupling of membrane and bending stiffness. These elements are not implemented for differential and nonlinear stiffness and heat transfer analysis.

The CQUAD8 and CTRIA6 elements should not be used with all midside nodes deleted since they are excessively stiff in this configuration. A user warning message is produced when all midside nodes are deleted. Use CQUAD4 and CTRIA3 for this condition.

QUADR and TRIAR Elements

Traditional Mindlin shell elements, such as QUAD4, have 5 degrees of freedom per grid point--three translations and two bending rotations. The stiffness for the rotational degree of freedom normal to the element (the drilling degree of freedom) is zero. This creates modeling difficulties which may eventually lead to poor solutions. Many methods--such as AUTOSPC and K6ROT--have been devised to remedy this deficiency. None of them, however, supply physical stiffness to the drilling degree of freedom to resist the drilling moments.

Benefits

The results for the new QUADR are compared with the QUAD4 and the old QUADR.

Comparing with the QUAD4, the new QUADR has the following benefits:

- It has six degrees of freedom at each grid point as opposed to the QUAD4, which has five degrees of freedom. This removes the necessity of using such methods as the AUTOSPC and K6ROT--which sometimes cast doubts on results--to account for the normal rotational DOF in the QUAD4.
- When using AUTOSPC or K6ROT for the QUAD4, the stiffness at the drilling degree of freedom is not an actual physical stiffness. Therefore, drilling loads cannot be correctly transferred between elements, which lead to modeling difficulties. For QUADR, the stiffness at the drilling degree of freedom is an actual physical stiffness and, therefore, the drilling loads can be transferred correctly between elements. The ability of the QUADR to resist the drilling moments correctly is demonstrated in Example 3.
- The membrane behaviors of the new QUADR is far superior than those of the QUAD4 and approaches that of the QUAD8, as demonstrated by Examples 2 and 6.

Many capabilities that are available to the QUAD4 are not included in the old QUADR. For the new QUADR, all the capabilities in QUAD4 have been incorporated into the QUADR. In comparison with the old QUADR, the current QUADR has the following benefits:

- An improved theoretical formulation to make the current QUADR a more robust element (see the **Theory** Section).
- Coupled with the shell normal, the current QUADR is well suited for modeling shell problems (see the Section on **Shell Models and Shell Normal**).
- The rotational mass is implemented for the drilling degree of freedom (see the Section **Mass Properties**.)
- The consistent surface loads and edge loads are implemented (see the Section **Consistent Surface and Edge Loads**).
- The differential stiffness matrix is implemented for the new QUADR so that buckling and other analyses requiring the differential stiffness can be performed.
- Coupling between bending and membrane is implemented.
- Composite material is implemented for layered stress/strain output.
- Offset is allowed for the new QUADR.
- The QUADR is implemented into the solution sequence SOL 200, optimization.
- QUADR is supported in heat transfer analysis.
- A System cell 370 is available for converting between QUAD4/TRIA3 and QUADR/TRIAR.
- Nonlinear material and geometric analysis will be implemented in a future version.

Theory

Allman and Cook were the first to propose a successful membrane element with drilling DOFs. On the other hand, Hughes and Brezzi presented a variational formulation, which was first suggested by Reissner. This variational principle uses the drilling degrees of freedom as independent rotation field; i.e., it is based on the separate kinematic variables of displacements and rotations. Ibrahimbegavic, Taylor and Wilson combined the Allman type of interpolation and the variational principle to construct a quadrilateral shell element. In MSC Nastran, membrane elements with drilling degrees of freedom were first introduced in 1988. A summary of the theory for the membrane portion of the shell element is given below.

Allman type of interpolation fields assumes the normal displacements to be quadratic and the tangential displacements to be linear. For example, along the edge from grid 1 to 2 (see Figure 3-35), the displacement in the middle of the edge is

$$\mathbf{u}_m = \frac{1}{2}(\mathbf{u}_1 + \mathbf{u}_2) + \frac{1}{8} l_1 (\theta_2 - \theta_1)\mathbf{n}_1 \tag{3-20}$$

where all bold letters indicate vectors with x and y-components. $\mathbf{u}_1$ and $\mathbf{u}_2$ are the displacements at grids 1 and 2, respectively; l_1 is the length of the edge; θ_1 and θ_2 are the drilling rotations at grid 1 and 2, respectively; $\mathbf{n}_1$ is the vector normal to the edge in the x-y plane; and $\mathbf{u}_m$ is the interpolated displacement vector in the middle of the edge.

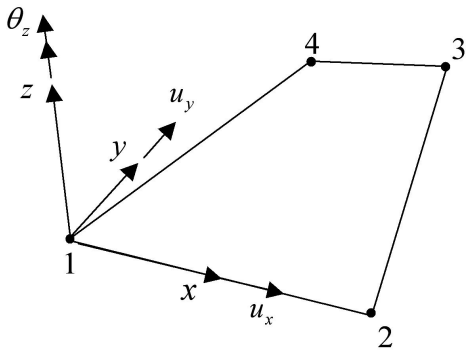


Figure 3-35 In-Plane displacements and drilling rotation

Hughes and Brezzi presented two variational formulations: one is the mixed type and the other is the displacement type, which is the one adopted here. The displacement type variational formulation states that the function to be minimized for elements with drilling degrees of freedom is

$$\Pi = \frac{1}{2}\int_A \varepsilon^T \mathbf{G}\varepsilon t dA + \frac{1}{2}\gamma' G \int_A (\Omega - \theta)^2 t dA \tag{3-21}$$

$$\Omega = \frac{1}{2}\left(\frac{\partial u_y}{\partial x} - \frac{\partial u_x}{\partial y}\right) \tag{3-22}$$

where:

γ	=	non-dimensional constant, for most problems $\gamma = 1.0$
G	=	the shear modulus, A and t are the area and thickness
Ω	=	the rotational field given by Equation (3-22)
θ	=	the interpolated drilling field
ε	=	the strain vector
$\mathbf{G}$	=	the stress-strain relationship defined by $\sigma = \mathbf{G}\varepsilon$

The first term is the normal strain energy and the second term is the contribution due to the drilling degrees of freedom.

In the variational formulation given by Equation (3-21), the drilling field and the displacement field are interpolated independently. For the drilling field, the interpolation function used is the linear QUAD4 shape function. For displacement field, the Allman type interpolation functions given by Equation (3-20) are used. They are given by the following equations:

$$\theta(\xi, \eta) = \sum_{i=1}^{4} N_i(\xi, \eta)\theta_i$$

$$\mathbf{u}(\xi, \eta) = \sum_{i=1}^{4} N_i(\xi, \eta)\mathbf{u}_i + \sum_{i=1}^{4} N_{i+4}(\xi, \eta)\frac{l_i}{8}(\theta_{i+1} - \theta_i)\mathbf{n}_i c_r + N_0(\xi, \eta)\mathbf{u}_0 \tag{3-23}$$

where:

$$i = 1, 2, 3, 4$$
$$\theta_5 = \theta_1$$

The N_is in above equations are shape functions for QUAD4 and QUAD8 elements. They are defined by:

$$N_i = \frac{1}{4}(1 + \xi_i\xi)(1 + \eta_i\eta) \quad \text{for i = 1, 2, 3, 4}$$
$$N_i = \frac{1}{2}(1 - \xi^2)(1 + \eta_i\eta) \quad \text{for i = 5, 7} \tag{3-24}$$
$$N_i = \frac{1}{2}(1 + \xi_i\xi)(1 - \eta^2) \quad \text{for i = 6, 8}$$

The N_0 in Equation (3-23) is a bubble function or an auxiliary strain function. The $\mathbf{u}_0$ is the displacement corresponding to the bubble function or the auxiliary function. The $\mathbf{n}_i$ is the outward normal vector to element boundary and l_i is its length.

With the shape functions defined by and Equation (3-23), and the variational principle defined by Equation (3-21), the standard finite element theory can be used to obtain the element forces and stiffness.

With the formulation given above, the singularity in the 6th degree of freedom of element such as QUAD4 is replaced with a true stiffness. In addition, the drilling rotation becomes a true load carrying degree of freedom. We will show that the membrane behavior is significantly improved. In fact, the accuracy of QUADR approaches that of the QUAD8.

Shell Models and Shell Normal

The shell element described in the previous **Theory** Section is for a flat element. In using flat shell elements for modeling a curved shell, local normals of adjacent shell elements have different directions. In MSC Nastran, an algorithm has been developed to generate a unique normal at a grid point. The unique normal is the average normal at each grid point so that the angles between the average normal and all the local normals of the adjacent shell elements are minimized. This unique normal is called the **Shell Normal**, which is requested by the PARAM, SNORM or the Bulk Data entry SNORM.

The new QUADR/TRIAR, formulation is more robust, is less sensitive to the shell normal, and is well suited for solving shell problems. The following list illustrates the effect of the shell normal to the solution of shell problems. Some of the problems are the MacNeal/Hard test problems.

- Twist beam (MacNeal/Harder test, see example below) -- no effect, excellent solution.
- Scordelis-Lo roof (MacNeal/Harder test, see example below) -- no effect, excellent solution.
- Spherical shell (MacNeal/Harder test, see example below) -- the solution is slightly stiffer with the shell normal than without the shell normal. The solution for the coarse model with shell normal converges rapidly to that of the fine mesh without shell normal.
- Cylindrical shell with internal pressure (see example below) and without the shell normal, solution converges with the fine mesh. Withe the shell normal, the solution is excellent even with a coarse mesh.
- Pinched cylinder (not presented) -- almost no effect, excellent solution. The pinched cylinder is a cylinder with two rigid end diaphragms. This cylinder is subject to a pair of concentrated forces, which pinch the cylinder.
- Raasch hook (see example below) -- no effect, excellent solution.
- We can conclude from the results that the shell normal is beneficial for membrane-dominant shell problems, and unfavorable effect for bending-dominant shell problems. To provide good solutions for a wide class of shell problems, the default value for PARAM, SNORM is set to 20.0. Also from the above results, we can conclude that, with or without the shell normal, the new QUADR/TRIAR is well suited for modeling shell problems.

Shell normal yields wrong results for elements with offsets. Therefore, the shell normal should be turned off globally (PARAM,SNORM,0.0) or turned off locally with the Bulk Data entry, SNORM.

Mass Properties

In MSC Nastran, two mass formulations are available--the lumped mass and the coupled mass. Lumped mass is the default formulation. Coupled mass formulation is requested by the parameter PARAM, COUPMASS, 1.

For the QUADR/TRIAR, the mass properties are as follows:

- Lumped mass -- only the translational masses are computed. No rotational mass is computed for the drilling degrees of freedom.
- Coupled mass -- in addition to the translational masses, the rotational masses are computed for the drilling degrees of freedom.

Consistent Surface and Edge Loads

`PLOAD4`, the most commonly used surface load, may have both normal and tangential components relative to the surface of an element. For the QUADR/TRIAR element, the consistent loads due to the surface load contain both concentrated forces at the connected grid points and the drilling moments at the drilling degrees of freedom. For the new QUADR/TRIAR, the consistent surface loads have been implemented.

For QUADR/TRIAR--because of the presence of drilling degrees of freedom--it is important to compute the consistent grid point forces for any distributed load along the edge of the element. This point is best illustrated by the following example. A single QUADR element is loaded with uniform tensile edge load along the opposite edge as shown in Figure 3-36A. If we lump the uniform load into grid point forces, as shown in Figure 3-36B, then we will get the non-uniform

displacement solution, instead of the correct uniform displacement solution. This error can be corrected if we compute the consistent grid point forces for the distributed edge loads. The consistent grid point forces consist of the lumped grid forces and a pair of drilling moments, as shown in Figure 3-36C.

For QUADR/TRIAR only, the consistent edge loads have been implemented. User can request the consistent edge loads using the PLOAD4 Bulk Data entry.

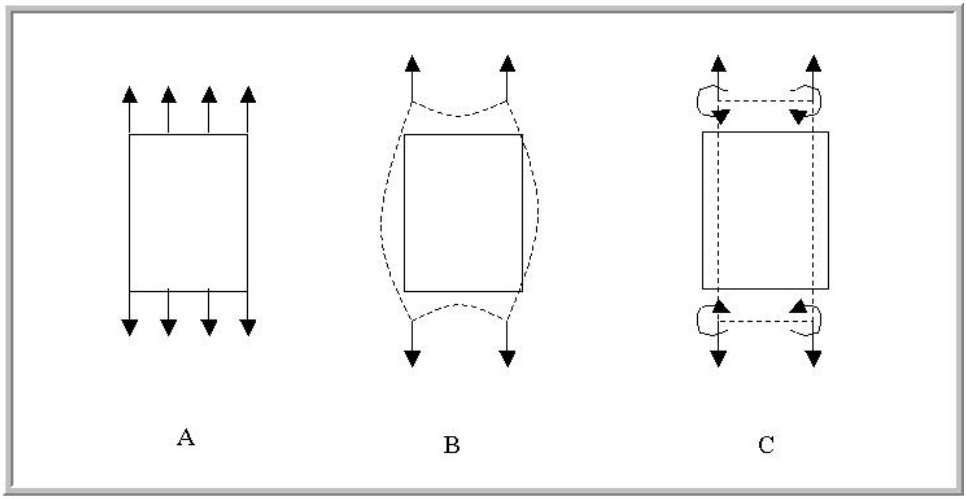


Figure 3-36 Consistent Edge Loads

User Interfaces

Parameters SNORM and Bulk Data Entry SNORM

The parameter SNORM is used for requesting unique shell normals for all grid points which are connected to the shell elements, such as QUADR, in a model. The default shell normal value is PARAM, SNORM, 20.0. For relationships between a shell model and the shell normal, please see the discussions in the **Shell Normal and Shell Models Section**.

The Bulk Data entry, SNORM requests a local shell normal at a grid point. It will override the shell normals requested by PARAM, SNORM, and can also be used to turn off the local default shell normal.

Bulk Data Entries CQUADR, CTRIAR, and PSHELL

The QUADR/TRIAR elements are requested by the existing CQUADR, CTRIAR, and PSHELL Bulk Data entries. For the new QUADR/TRIAR, there is no modification to the PSHELL entry. For the CQUADR and CTRIAR entries, two new fields are added:

- ZOFFS -- this entry is used to input the offsets from the surface of the grid points to the element plane.
- TFLAG -- this flag allows user to input the relative values for the variable grid point thickness, instead of absolute values. See Section for further details.

Bulk Data Entry PLOAD4

The PLOAD4 entry is used to define a pressure load for a 3-D or 2-D element. For the QUADR/TRIAR only, this entry is extended to accommodate consistent edge load. See the **Consistent Surface and Edge Loads** Section for further details. Two new fields are added to PLOAD4 for this purpose:

- SORL -- selects whether the load is a surface or an edge load.
- SORL=SURF selects the surface load and SORL=LINE selects the consistent edge load. SORL=SURF is the default.
- LDIR -- defines the direction of the consistent edge load (SORL=LINE). It has the value of X, Y, Z, TANG, or NORM. See PLOAD4 (p. 2865) in the for their meanings. The default is NORM.

For example, the uniform extensile edge loads shown in Figure 3-36 can be specified by the following PLOAD4 entries:

1	2	3	4	5	6	7	8	9	10
PLOAD4	10	1	20.0						
					LINE				

The edge loads have a magnitude of 20.0 and the direction is NORM, i.e., it is in the mean plane of the element, normal to the edge, and pointing outward from the element.

NASTRAN System (370) -- QRMETH

The new default QUADR/TRIAR formulation is recommended. However, a new system cell 370 (QRMETH) is available for reverting back to the old formulation.

A companion version of the QUAD4/TRIA3 is also developed along with the new QUADR/TRIAR. We will call this version the new formulation for QUAD4/TRIA3, and the existing QUAD4 and TRIA3 the old formulation. The performances of the old and the new formulations are about the same.

The default for the QUADR/TRIAR uses the new formulation. The default for the QUAD4/TRIA3 uses the old formulation.

In addition to the selection of the new or old formulation for QUADR/TRIAR, QRMETH provides the following options:

- 0 -- selects the new QUADR/TRIAR formulation.
- 1 -- selects the old QUADR/TRIAR formulation.
- 2 -- converts QUADR/TRIAR into QUAD4/TRIA3 using the new QUAD4/TRIA3 formulation.
- 3 -- converts QUADR/TRIAR into QUAD4/TRIA3 using the old QUAD4/TRIA3 formulation.
- 4 -- selects the new QUAD4/TRIA3 formulation.
- 5 -- converts QUAD4/TRIA3 into QUADR/TRIAR using the new QUADR/TRIAR formulation.

The default is for QRMETH is 0.

Converting a QUAD4 model to QUADR

The following examples will illustrate that the QUADR is superior to the QUAD4. User may want to convert a QUAD4 model to QUADR by using system cell QRMETH. However, the following should be noted:

- Using QRMETH=5 will convert all QUAD4/TRIA3 in a model to QUADR/TRIAR.

- At the boundaries of the model, the drilling degrees of freedom must be constrained if the user wants fixed boundaries. For example, in the example **Static Analysis of Straight Beam with Coarse Mesh**, the drilling degrees of freedom must be constrained at the left end.
- At internal grid points of a model, user must remove any SPC or the PS field on the GRID Bulk Data entries which constrain the drilling degrees of freedom because the QUADR/TRIAR supply stiffness for these degrees of freedom. On the other hand, if user uses K6ROT or AUTOSPC to manage the drilling degrees of freedom, these will be automatically converted with QRMETHD=5. No user action is required.

Examples

Example 1 - Static Analysis of Straight Beam with Coarse Mesh

For membrane behavior, the performances of the QUADR element are substantially better than those of the QUAD4 element. This can best be illustrated by the static analysis of a straight beam with coarse mesh, shown in Figure 3-37. The load is an in plane shear load acting at the tip of the beam. This test is taken from the MacNeal-Harder tests and is designed to exhibit in plane behavior of shell elements for different element shapes. The results for this test are shown in Table 3-8. The displacements shown are at the tip of the beam, normalized to the theoretical value.

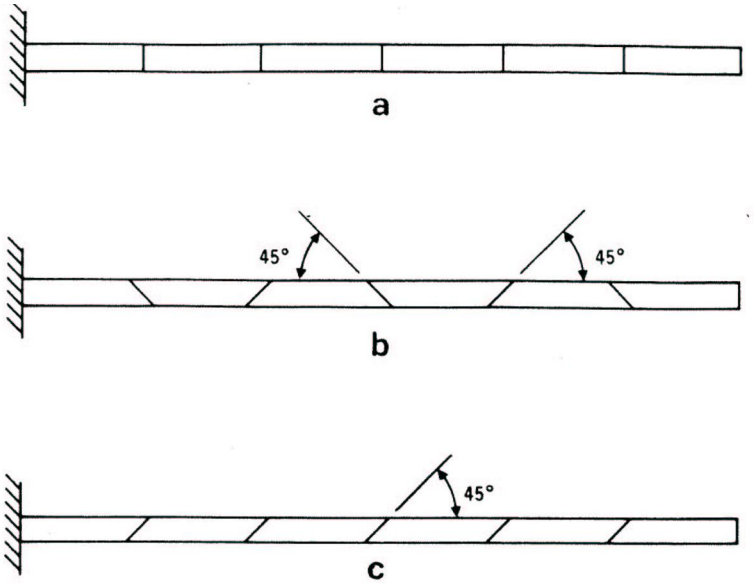


Figure 3-37 Straight Beam with Coarse Mesh

Table 3-8 Normalized Displacements for In-plane Shear Load at Tip

Element	Rectangle	Trapezoid	Parallelogram
QUAD4	0.9929	0.0515	0.6232
QUADR	0.9926	0.9613	0.9491

From the results, we see that the QUADR provides excellent results for both regular and irregular element shapes, while QUAD4 fails for the irregular shapes. We may argue that, in real life, we don't use element mesh as coarse as the one in this test. Next, we will look at the results of a fine element mesh, which is discussed in next sub-example.

Example 2 - Normal Modes Analysis of Straight Beam with Fine Mesh

For this example, a refined mesh of Example 1 is used in a normal mode analysis. Table 3-9 summarizes the results of the first mode using various element shapes.

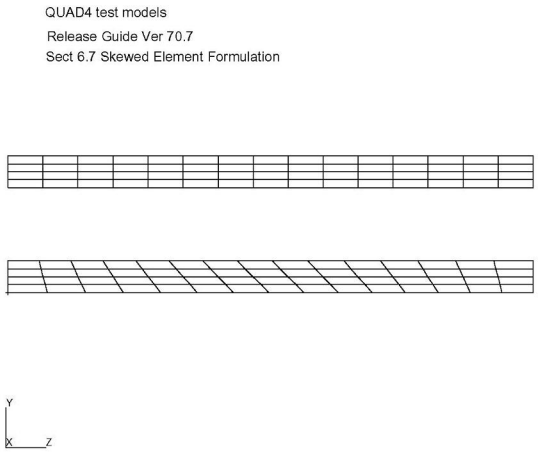


Figure 3-38 Straight beam with fine mesh

Table 3-9 In-Plane Bending Frequency of Straight Beam with Fine Mesh

Element	Rectangle (hz)	Trapezoid (hz)	Parallelogram (hz)
QUAD4	9.39	15.89	9.41
QUADR	9.39	9.42	9.39

Once again, the QUAD4 element exhibits locking effects for trapezoidal shapes. Even for finer meshes, the results of the QUADR elements are still noticeable better than the results of the QUAD4 element in the case of distorted element shapes.

Example 3 - Static Analysis of Straight Beam with Drilling Moments

This example shows that the QUADR element provides the true physical stiffness for the drilling degrees of freedom and hence able to resist the actual drilling loads. A straight beam is modeled by six QUADR elements and is subject to a pair of end drilling moments as shown in Table 3-9.

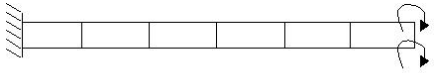


Figure 5. Straight Beam Subject to Tip Drilling Moments

Figure 3-39 Straight Beam Subject to Tip Drilling Moments

The tip displacements of the straight beam for a static analysis are

- Theory = -5.40e-2
- Finite element solution = -5.46e-2

The above results show that the QUADR can predict correctly the response due to applied drilling moments.

Example 4 - Static Analysis of Cylindrical Shell with Internal Pressure

The purpose of this example is to illustrate the benefits of the shell normal in a shell structure modeled by QUADR elements. The shell normal is discussed in the **Shell Models and Shell Normal** Section. A cylinder of height 1000.0, radius 1000.0, and thickness 10.0, is subject to internal pressure of 1.0, as shown in Figure 3-40. The cylinder is free at both edges. Due to symmetry, only half of the height and a quarter of the cylinder is modeled. The radial displacement at the free edge is given in Table 3-10.

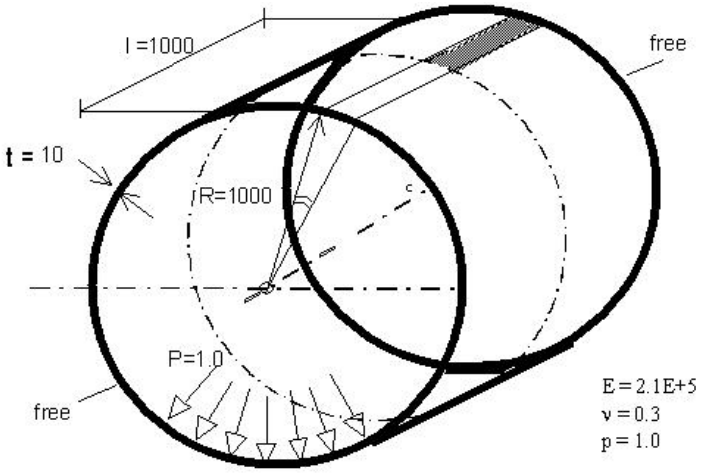


Figure 6. Cylindrical Shell with Internal Pressure

Figure 3-40 Cylindrical Shell with Internal Pressure

Table 3-10 Radial Displacement at Free Edge for Pressurize Cylinder

Mesh	Theory	With SNORM	No SNORM
4x4	0.4762	0.4670	0.6733
8x8	0.4762	0.4739	0.5251
16x16	0.4762	0.4756	0.4882

From Table 3-10, we can see that the results with the shell normal are substantially better than those without the shell normal.

Example 5 - Raasch Hook

The Raasch hook is a curved strip clamped at one end and loaded at the other end with an in-plane shear load as shown in Figure 3-41. A static analysis is performed on the Raasch hook for cases with and without the shell normal. The tip displacements in the direction of the shear force are summarized in Table 3-11 using the new QUADR with various mesh size.

From Table 3-11, we can see that, with or without the shell normal, the results for the new QUADR are excellent. Using the old QUADR, the solution converges to the wrong results if shell normals are not used.

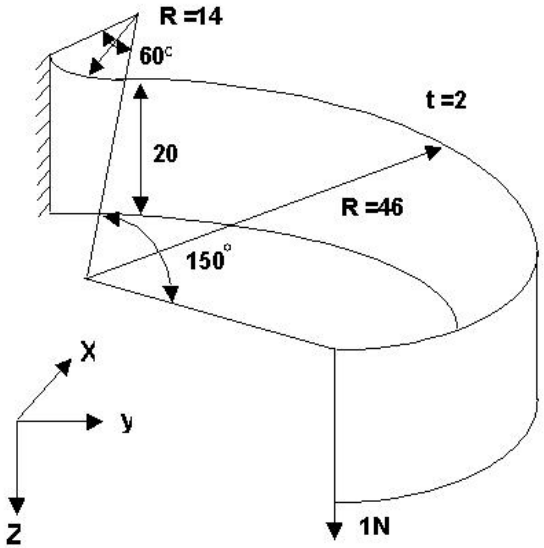


Figure 7. Raasch Hook

Figure 3-41 Raasch Hook

Table 3-11 Tip Displacement for Raasch Hook

Mesh	Theory	With SNORM	No SNORM
1x9	5.012	4.600	4.601
3x17	5.012	4.890	4.888
5x34	5.012	4.916	4.915
10x68	5.012	4.970	4.971
20x136	5.012	5.011	5.012

Example 6 - MacNeal and Harder Tests

MacNeal and Harder suggested a series of test problems for evaluating the accuracy of finite element. We have run these test problems for QUADR, QUAD4 and QUAD8 for comparisons. The results are summarized in Table 3-12 and a score is given to each test problem based on how well it compares to the theoretical solution. The scoring criteria is shown:

- A -- error is less than and equal to 2%.
- B -- error is between 2% and 10%.
- C -- error is between 10% and 20%.
- D -- error is between 20% and 50%.
- F -- error is greater than 50%.

From Table 3-12, we see that QUADR got the best scores among the three elements. It does not have a score below B. Between QUADR and QUAD4, the results of QUADR are better than those of the QUAD4. Between QUADR and QUAD8, the accuracy of the QUADR approaches that of QUAD8 for membrane problems and is better than that of QUAD8 for bending problems.

For QUADR, the tests are run with the default value of the shell normal equal to 20.0. Except for the spherical shell (14), the shell normal has no effect on the tests. For the spherical shell, it will score an A if shell normal is turned off (PARAM, SNORM, 0.0.)

Table 3-12 MacNeal and Harder Tests for QUADR, QUAD4, and QUAD8

MacNeal/Harder Test Description	Element Loading		Element Shapes	QUADR	QUAD4	QUAD8
	In-Plane	Out-of-Plane				
1. Patch Test	X		Irregular	A	A	C
2. Patch Test		X	Irregular	A	A	D
3. Straight Beam, Extension	X		All	A	A	A
4. Straight Beam, Bending	X		Regular	A	A	A

Table 3-12 MacNeal and Harder Tests for QUADR, QUAD4, and QUAD8

MacNeal/Harder Test Description	Element Loading		Element Shapes	QUADR	QUAD4	QUAD8
	In-Plane	Out-of-Plane				
5. Straight Beam, Bending	X		Irregular	B	F	B
6. Straight Beam, Bending		X	Regular	A	A	A
7. Straight Beam, Bending		X	Irregular	A	A	A
8. Straight Beam, Twist	X	X	All	B	B	B
9. Curved Beam	X		Regular	A	C	A
10. Curved Beam		X	Regular	B	B	B
11. Twisted Beam	X	X	Regular	A	A	A
12. Rectangular Plate (N=4)		X	Regular	A	B	B
13. Scordelis-Lo Roof (N=4)	X	X	Regular	B	B	A
14. Spherical Shell (N=8)	X	X	Regular	B	A	C
15. Thick Walled Cylinder	X		Regular	B	B	B

Conical Shell Element (RINGAX)

The properties of the conical shell element are assumed to be symmetrical with respect to the axis of the shell. However, the loads and deflections need not be axisymmetric because they are expanded in Fourier series with respect to the azimuthal coordinate. Due to symmetry, the resulting load and deformation systems for different harmonic orders are independent, a fact that results in a large time saving when the use of the conical shell element is compared with an equivalent model constructed from plate elements. Theoretical aspects of the conical shell element are provided in Section 5.9 of *The NASTRAN Theoretical Manual*.

The conical shell element cannot be combined with other types of elements. This is primarily because the Fourier coefficients are stored on internally generated pseudo grid points. The unconventional nature of this element results in its capabilities being limited, as summarized in Table 3-1. The existence of a conical shell problem is defined by the AXIC entry. This entry also indicates the number of harmonics desired in the problem formulation. Only a limited number of Bulk Data entries are allowed when using conical shell elements. The list of allowable entries is given on AXIC (p. 1245) in the . This element is not allowed in the superelement solution sequences.

The geometry of a problem using the conical shell element is described with RINGAX entries instead of GRID entries. The RINGAX entries describe concentric circles about the basic z-axis, with their locations given by radii and z-coordinates as shown in Figure 3-42. The degrees of freedom defined by each RINGAX entry are the Fourier coefficients of the motion

with respect to angular position around the circle. For example, the radial motion u_r at any angle ϕ is described by the equation

$$u_r(\phi) = \sum_{n=0}^{N} u_r^2 \cos n\phi + \sum_{n=0}^{N} u_r^{2*} \sin n\phi \qquad (3\text{-}25)$$

where u_r^n and u_r^{n*} are the Fourier coefficients of radial motion for the n-th harmonic. For calculation purposes, the series is limited to N harmonics as defined by the AXIC entry. The first sum in the above equation describes symmetric motion with respect to the ϕ plane. The second sum with the "starred" (*) superscripts describes the antisymmetric motion. Thus each RINGAX entry will produce six times N degrees of freedom for each series.

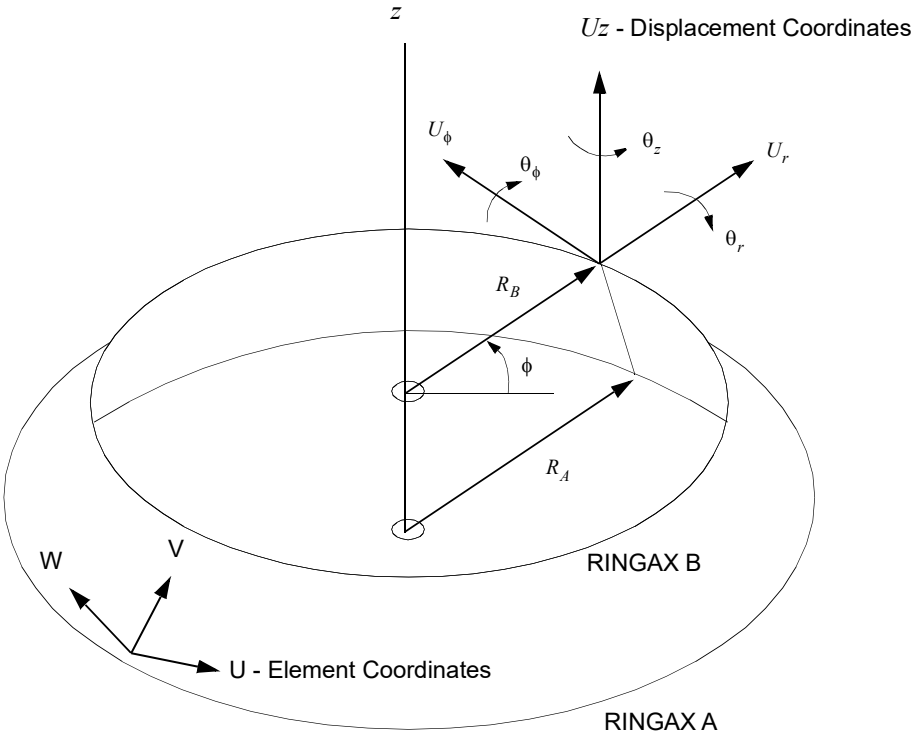


Figure 3-42 Geometry for Conical Shell Element

The selection of symmetric or antisymmetric solutions is controlled by the AXISYM command in the Case Control Section. For general loading conditions, a combination of the symmetric and antisymmetric solutions must be made, using the Case Control command, SYMCOM (Case) (p. 610) in the *MSC Nastran Quick Reference Guide*. If the AXISYM command is not present, stress and element force requests will be ignored.

Since the user is rarely interested in applying loads in terms of Fourier harmonics and interpreting data by manually performing the above summations, special data entries which automatically perform these operations are provided. The POINTAX entry is used like a GRID entry to define physical points on the structure for loading and output. Sections of the circle may be defined by a SECTAX entry which defines a sector with two angles and a referenced RINGAX entry. The POINTAX and SECTAX entries define six degrees of freedom each. The basic coordinate system for these points is a cylindrical system (r, ϕ, z), and their applied loads must be described in this coordinate system. Since the displacements of these points are dependent on the harmonic motions, they may not be constrained in any manner.

The conical shell element is connected to two RINGAX rings with a CCONEAX entry. The properties of the conical shell element are described on the PCONEAX entry. The RINGAX points must be placed on the neutral surface of the element, and the points for stress calculation must be given on the PCONEAX entry relative to the neutral surface. Up to fourteen angular positions around the element may be specified for stress and force output. These values will be calculated midway between the two connected rings.

The structure defined with RINGAX and CCONEAX entries must be constrained in a special manner. All harmonics may be constrained for particular degree of freedom on a ring by using permanent single-point constraints on the RINGAX entries. Specified harmonics of each degree of freedom on a ring may be constrained with an SPCAX entry. The entry is the same as the SPC entry except that a harmonic must be specified. The MPCAX, OMITAX, and SUPAX data entries correspond to the MPC, OMIT, and SUPORT data entries except that harmonics must be specified. SPCADD and MPCADD entries may be used to combine constraint sets in the usual manner.

The stiffness matrix includes five degrees of freedom per grid circle per harmonic when transverse shear flexibility is included. Since the rotation about the normal to the surface is not included, the sixth degree of freedom must be constrained to zero when the angle between the meridional generators of two adjacent elements is zero. Since only four independent degrees of freedom are used when the transverse shear flexibility is not included, the fifth and sixth degrees of freedom must be constrained to zero for all rings. These constraints can be conveniently specified on the RINGAX entry.

The conical shell structure may be loaded in various ways. Concentrated forces may be described by FORCE and MOMENT entries applied to POINTAX points. Pressure loads may be input by the PRESAX data entry which defines an area bounded by two rings and two angles. Temperature fields are described by a paired list of angles and temperatures around a ring as required by the TEMPAX entry. Direct loads on the harmonics of a RINGAX point are given by the FORCEAX and MOMAX entries. Since the basic coordinate system is cylindrical, the loads are given in the r, ϕ, and z directions. The value of a harmonic load F_n is the total load on the whole ring of radius r. If a sinusoidal load-per-unit length of maximum value is given, the value on the FORCEAX entry must be

$$F_n = 2\pi r a_n \qquad n = 0 \tag{3-26}$$

$$F_n = \pi r a_n \qquad n = 0 \tag{3-27}$$

Displacements of rings and forces in conical shell elements can be requested in two ways:

- The harmonic coefficients of displacements on a ring or forces in a conical element
- The displacements at specified points or the average value over a specified sector of a ring. The forces in the element at specified azimuths or average values over specified sectors or a conical element.

Harmonic output is requested by ring number for displacements and conical shell element number for element forces. The number of harmonics that will be output for any request is a constant for any single execution. This number is controlled by the Case Control command, HARMONICS (p. 417) in the *MSC Nastran Quick Reference Guide*.

The following element forces per unit of width are output either as harmonic coefficients or at specified locations on request

- Bending moments on the u and v faces
- Twisting moments
- Shearing forces on the u and v faces.

The following element stresses are calculated at two specified points on the cross section of the element and output as harmonic coefficients or at specified locations on request:

- Normal stresses in u and v directions
- Shearing stress on the u face in the v direction
- Angle between the u-axis and the major principal axis
- Major and minor principal stresses
- Maximum shear stress

Shell Elements Corner Thickness Specification

The CQUAD4, CQUAD8, CQUADR, CTRIA3, CTRIA6, and CTRIAR elements allow the user to specify either corner element thickness or a single thickness on the PSHELL entry. A TFLAG field has been added to the connection entries of each of the above shell elements that allows the user to specify the corner thickness as a fraction relative to the T value of the PSHELL entry.

Benefits

In forming shell like structures, quite often the taper at the corners is really just a function of some average thickness. With this option, the average thickness can be specified as T on the PSHELL and the corner thickness will be computed as a fraction of the T on the PSHELL. In SOL 200, the T on the PSHELL can be specified on the DVPREL1 with average thickness specified using a DESVAR entry and the shell will be tapered accordingly.

Theory

If TFLAG=1 the following calculation is used for each corner:

$$T_i = T_i \times T$$

Inputs

If the fraction relative to the T value of the PSHELL is desired, TFLAG=1 should be specified on the element connection entry.

Outputs

Standard shell element output is obtained.

Guidelines and Limitations

This option should be useful in design optimization.

Example

The following example input demonstrates the form and use of the new entries:

Input File

```
diag 8,15,56
sol 200
cend
$
title = Cantilever Plate: Optimization
subtitle = Relative Thickness
$
analysis = statics
$
desobj = 1000
dessub = 2000
$
dsaprt
$
load = 10
spc  = 20
$
disp   = all
stress = all
$
begin bulk
$
param, grdpnt, 0
$
dresp1, 1000, weight, weight
$
dresp1, 2000, disp, disp,,, 2,, 5
dconstr, 2000, 2000, -0.5
$
doptprm, desmax, 15
$
$ Design Model: Relative Thickness
$
desvar, 1, t1, 1., .001
desvar, 2, t2, 1., .001
desvar, 3, t3, 1., .001
$
dvprel1, 100, pshell, 1, t
       ,   1, 0.1
dvprel1, 200, pshell, 2, t
       ,   2, 0.2
dvprel1, 300, pshell, 3, t
```

```
       ,   3, 0.1
$
$ Test Plate: Sheet Metal Thickness Variation
$
$ Relative Thicknesses on CQUAD4/CTRIA3 Entries
$
$ ------------------------------------------------------------
$
grid,   1,,    0.,   0.,   0.
=   ,*(1),, *(1.), ==
=(3)
grid,  11,,    0.,   1.,   0.
=   ,*(1),, *(1.), ==
=(2)
grid,  21,,    0.,   2.,   0.
=   ,*(1),, *(1.), ==
=(3)
$
cquad4,  1,  1,  1,  2, 12, 11,
      ,   ,  1, 1., 1.1, 1.1, 1.
cquad4,  2,  1,  2,  3, 13, 12,
      ,   ,  1, 1.1, 1.2, 1.3, 1.1
cquad4,  3,  1,  3,  4, 14, 13,
      ,   ,  1, 1.2, 1.1, 1.6, 1.3
$
pshell, 1, 1, 0.1, 1
$
cquad4, 11,  2, 11, 12, 22, 21
      ,   ,  1, 1.0, 1.1, 1.1, 1.0
cquad4, 12,  2, 12, 13, 23, 22
      ,   ,  1, 1.1, 1.0, 1.2, 1.1
cquad4, 13,  2, 13, 14, 24, 23
      ,   ,  1,  1.0, 0.8, 1.1, 1.2
$
pshell, 2, 1, 0.2, 1
$
ctria3, 21,  3,  4,  5, 14
      ,   ,  1, 1.1, 1.0, 1.6
ctria3, 22,  3,  5, 25, 14
      ,   ,  1, 1.0, 2.0, 1.6
ctria3, 23,  3,  14, 25, 24
      ,   ,  1, 1.6, 2.0, 2.2
$
pshell, 3, 1, 0.1, 1
$
mat1, 1, 10000.,, 0.3, 1.
$
force, 10,  5,, 10., 0., -1.
force, 10, 25,, 10., 0., -1.
$
spc1, 20, 12,   1, 11, 21
spc1, 20, 3456, 1, thru, 25
$
enddata
```

Partial Output File

```
 S U M M A R Y   O F   D E S I G N   C Y C L E   H I S T O R Y
                          ************************************************************

                                                (HARD CONVERGENCE ACHIEVED)

                                                (SOFT CONVERGENCE ACHIEVED)


                                        NUMBER OF FINITE ELEMENT ANALYSES COMPLETED          5
                                        NUMBER OF OPTIMIZATIONS W.R.T. APPROXIMATE MODELS    4


                                                OBJECTIVE AND MAXIMUM CONSTRAINT HISTORY
      ------------------------------------------------------------------------------------------------------------------------
                                OBJECTIVE FROM          OBJECTIVE FROM          FRACTIONAL ERROR          MAXIMUM VALUE
        CYCLE                      APPROXIMATE               EXACT                    OF                       OF
        NUMBER                     OPTIMIZATION             ANALYSIS             APPROXIMATION             CONSTRAINT
      ------------------------------------------------------------------------------------------------------------------------

        INITIAL                                          1.299167E+00                                     -4.583377E-02

              1             1.088284E+00               1.088204E+00              7.350594E-05            -1.224685E-02

              2             1.067696E+00               1.067703E+00             -6.364063E-06             1.266837E-03

              3             1.065613E+00               1.065610E+00              2.349259E-06             2.949357E-03

              4             1.065610E+00               1.065610E+00              0.000000E+00             2.949357E-03
*                           ************************************************************
------------------------------------------------------------------------------------------------------------------------
1    CANTILEVER PLATE: OPTIMIZATION                                          MARCH  18, 2003  MSC Nastran  3/18/03   PAGE    40
     RELATIVE THICKNESS
0
                                                DESIGN VARIABLE HISTORY
 ------------------------------------------------------------------------------------------------------------------------------
---
  INTERNAL |   EXTERNAL   |            |
   DV. ID. |    DV. ID.   |    LABEL   |   INITIAL    :      1       :      2       :      3       :      4       :      5       :
 ------------------------------------------------------------------------------------------------------------------------------
---
         1 |          1   |  T1        |   1.0000E+00 :   1.2538E+00 :   1.3250E+00 :   1.3398E+00 :   1.3398E+00 :
         2 |          2   |  T2        |   1.0000E+00 :   8.2646E-01 :   7.5902E-01 :   7.5085E-01 :   7.5085E-01 :
         3 |          3   |  T3        |   1.0000E+00 :   3.8964E-01 :   3.8075E-01 :   3.7388E-01 :   3.7388E-01 :
 *** USER INFORMATION MESSAGE 6464 (DOM12E)
     RUN TERMINATED DUE TO HARD CONVERGENCE TO AN OPTIMUM AT CYCLE NUMBER =        4.
```

SHELL Structural Damping

If PARAM,SHLDAMP,DIFF is present in the run, a structural damping matrix K^4 is formed for each shell element based on a relationship of the following form:

$$K^4 = ge_{mid1} \times membrane \cdot stiffness + ge_{mid2} \times bending \cdot stiffness + ge_{mid3} \times transverse \cdot stiffness + ge_{mid4} \times combined \cdot stiffness$$

Solid Elements (CTETRA, CPENTA, CHEXA)

MSC Nastran includes three different solid polyhedron elements which are defined on the following Bulk Data entries:

1. CTETRA -- Four-sided solid element with 4 to 10 grid points. See Figure 3-43.
2. CPENTA -- Five-sided solid element with 6 to 15 grid points. See Figure 3-44.
3. CHEXA -- Six-sided solid element with 8 to 20 grid points. See Figure 3-45.

These elements can be used with all other MSC Nastran elements except the axisymmetric elements. Connections are made only to displacement degrees of freedom at the grid points.

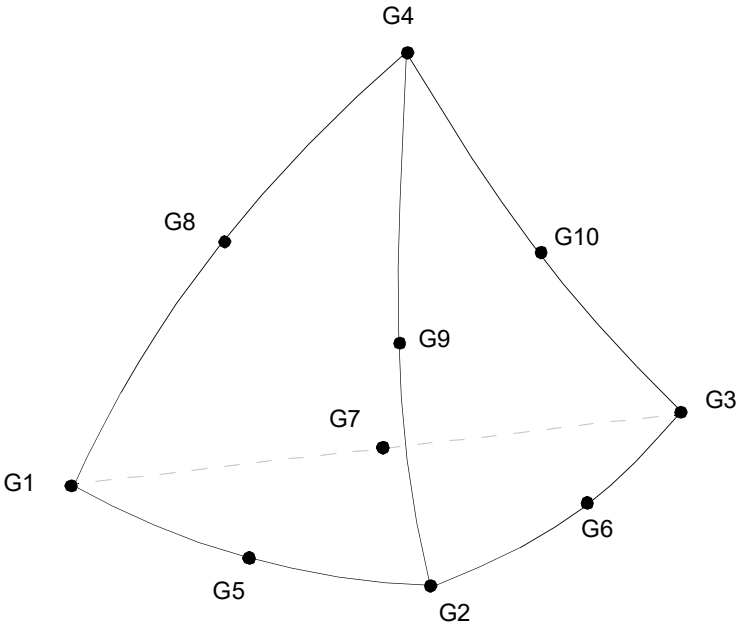


Figure 3-43 The CTETRA Element

Three types of integration are available for the CPENTA and CHEXA elements: reduced shear integration with and without the use of bubble functions and standard isoparametric integration. Only standard isoparametric integration is available for the CTETRA element. The type of integration is selected on the PSOLID entry.

Reduced shear integration minimizes shear locking and will not cause zero energy modes. If reduced shear integration is selected with bubble functions, it minimizes Poisson's ratio locking which occurs in nearly incompressible materials and in elements under bending. In plastic analysis, bubble functions are necessary to reduce locking caused by the plastic part in the material law. Therefore, reduced shear integration with bubble functions is the default option for the CPENTA and CHEXA elements. However, the use of bubble functions requires more computational effort.

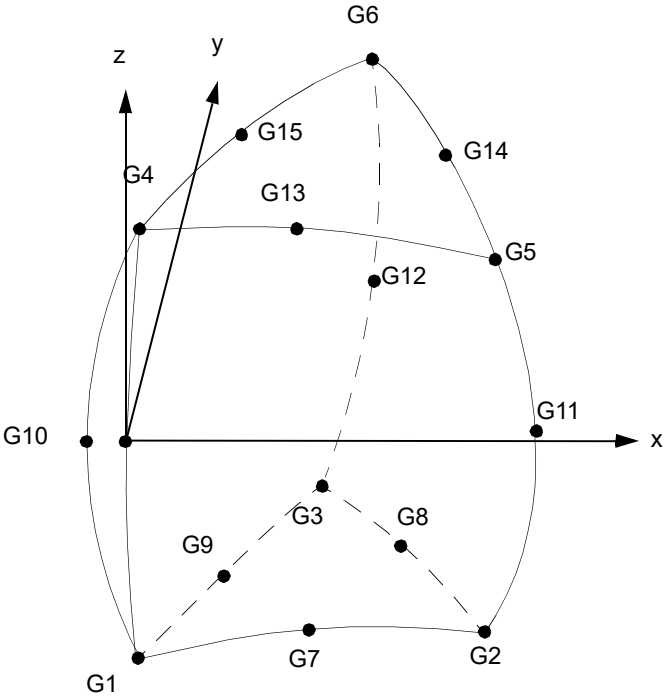


Figure 3-44 The CPENTA Element

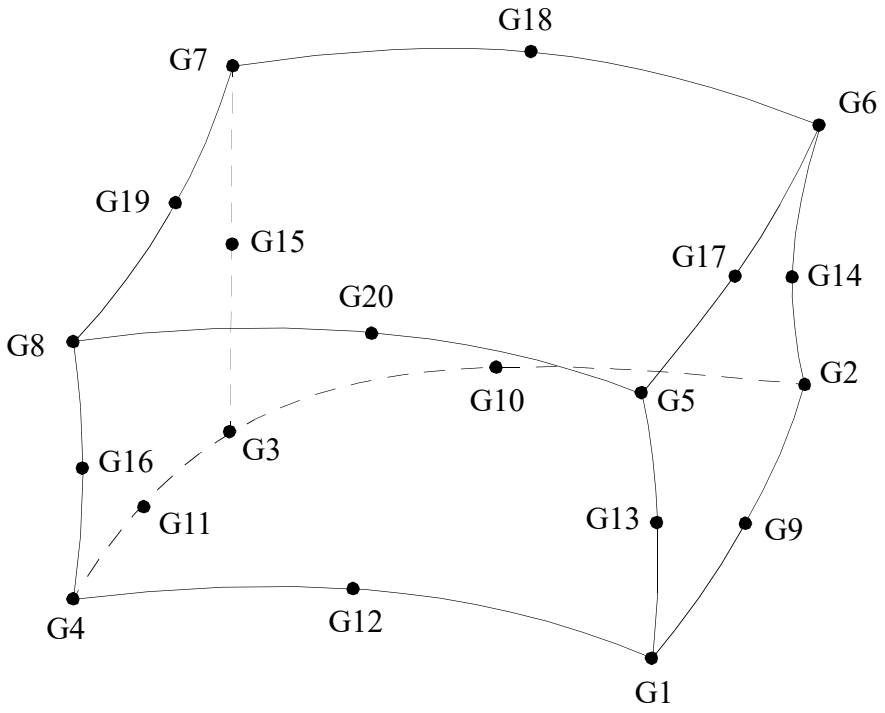


Figure 3-45 The CHEXA Element

In standard isoparametric integration, the number of Gauss points, or integration network, may be changed to underintegrate or overintegrate the solid elements. Underintegration may cause zero energy modes and overintegration results in an element which may be too stiff. Standard isoparametric integration is more suited to nonstructural problems.

The element coordinate systems are described under the description of the CHEXA, CPENTA, and CTETRA entries, respectively. These elements may use anisotropic materials as defined on the MAT9 Bulk Data entry. The material reference and integration network is specified on the PSOLID entry.

Structural mass is calculated for all solid elements. The default mass procedure is lumped mass. The coupled mass procedure may be requested with PARAM,COUPMASS (p. 804) in the *MSC Nastran Quick Reference Guide*.

The solid elements can be used to define fluid elements for coupled fluid-structural analysis. See Coupled Fluid-Structure Interaction, 663.

The solid elements can be used in differential stiffness analysis and buckling analysis. A geometric and material nonlinear stiffness formulation is available for these elements only if there are no midside nodes.

The following stresses and strains are output on request:

- Normal: σ_x, σ_y, σ_z, and ε_x, ε_y, ε_x

  Shear: τ_{xy}, τ_{yz}, τ_{zx}, and γ_{xy}, γ_{yz}, γ_{zx}

Principal with magnitude and direction

Mean pressure $p_o = \frac{1}{3}(\sigma_x + \sigma_y + \sigma_z)$

Octahedral shear stress or

$$\bar{\tau} = \frac{1}{3}[(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6\tau_{yz}^2 + 6\tau_{zx}^2 + 6\tau_{xy}^2]^{1/2} \quad (3\text{-}28)$$

the von Mises equivalent stress

$$\bar{\tau}_v = \left(\frac{3}{\sqrt{2}}\right)\bar{\tau}_o \quad (3\text{-}29)$$

- Octahedral shear strain or

$$\bar{\varepsilon}_o = \left[\frac{1}{9}\{(\varepsilon_x - \varepsilon_y)^2 + (\varepsilon_y - \varepsilon_z)^2 + (\varepsilon_z - \varepsilon_x)^2\} + \frac{1}{6}(\gamma_{xy}^2 + \gamma_{yz}^2 + \gamma_{zx}^2)\right]^{1/2} \quad (3\text{-}30)$$

the von Mises equivalent strain

$$\bar{\varepsilon}_v = \left[\frac{2}{9}\{(\varepsilon_x - \varepsilon_y)^2 + (\varepsilon_y - \varepsilon_z)^2 + (\varepsilon_z - \varepsilon_x)^2\} + \frac{1}{3}(\gamma_{xy}^2 + \gamma_{yz}^2 + \gamma_{zx}^2)\right]^{1/2} \quad (3\text{-}31)$$

where the strain components are defined as

$$\varepsilon_x = \frac{\partial u}{\partial x};\ \varepsilon_y = \frac{\partial v}{\partial y};\ \varepsilon z = \frac{\partial w}{\partial z} \quad (3\text{-}32)$$

$$\gamma_{xy} = \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right);\ \gamma_{yz} = \left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}\right);\ \gamma_{zx} = \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}\right) \quad (3\text{-}33)$$

In nonlinear nonhyperelastic (small strain) analysis, the mechanical strain σ/E is output rather than the total strain defined above. The hyperelastic elements output total strains. See Fully Nonlinear Hyperelastic Elements, 579.

By default, the stresses and strains are evaluated in the basic coordinate system at each of the corner points and the centroid of the element. The stresses and strains may also be computed in the material coordinate system as defined in the CORDM field on the PSOLID entry.

In addition, interpolated grid point stresses and mesh stress discontinuities are calculated in user-specified coordinate systems for grid points which connect solid elements. See Element Data Recovery Resolved at Grid Points, 435 for a description of the Case Control Section to request grid point stresses and mesh stress discontinuities. Only real stresses are available at the grid points. Mesh stress discontinuities are available in linear static analysis only. For further discussion see the end of Shell Elements (CTRIA3, CTRIA6, CTRIAR, CQUAD4, CQUAD8, CQUADR), 123.

Rigid Elements and Multipoint Constraints (R-type, MPC)

The multipoint constraint, or MPC entry, provides the capability to model rigid bodies and to represent other relationships which can be treated as rigid constraints. The MPC entry provides considerable generality but lacks user convenience. Specifically, the user must supply all of the coefficients in the equations of constraint defined through the MPC entry.

To enhance user convenience, nine rigid body elements are available in MSC Nastran. See Table 3-13. These elements require only the specification of the degrees of freedom that are involved in the equations of constraint. All coefficients in these equations of constraint are calculated internally in MSC Nastran.

Table 3-13 Rigid Element and MPC Entries

Name	Description	Finite Rotation	m = Dependent Degrees of freedom
RROD	A open-ended rod which is rigid in extension	Y	$m = 1$
RBAR	Rigid bar with six degrees of freedom at each end.	Y	$1 \le m \le 6$
RJOINT	Rigid joint with six degrees of freedom at each end.	Y	$1 \le m \le 6$
RTRPLT	Rigid triangular plate with six degrees of freedom at each vertex.	Y	$1 \le m \le 12$
RBE2	A rigid body connected to an arbitrary number of grid points. The independent degrees of freedom are the six components of motion at a single grid point. The dependent degrees of freedom at the other grid points all have the same user-selected component numbers.	Y	$m \ge 1$
RBE1	A rigid body connected to an arbitrary number of grid points. The independent and dependent degrees of freedom can be arbitrarily selected by the user.	Y	$m \ge 1$

Table 3-13 Rigid Element and MPC Entries (continued)

Name	Description	Finite Rotation	m = Dependent Degrees of freedom
RBE3	Defines a constraint relation in which the motion at a "reference" grid point is the least square weighted average of the motions at other grid points. The element is useful for "beaming" loads and masses from a "reference" grid point to a set of grid points.	Y	$1 \le m \le 6$
RSPLINE	Defines a constraint relation whose coefficients are derived from the deflections and slopes of a flexible tubular beam connected to the referenced grid points. This element is useful in changing mesh size in finite element models.	N	$m \ge 1$
RSSCON	Define a multipoint constraint relation which models a clamped connection between shell and solids.	N	$m \ge 5$
MPC	Rigid constraint that involves user-selected degrees of freedom at both grid points and at scalar points. The coefficients in the equation of constraint are computed and input by the user.	N	$m = 1$

Any combination of the above elements may be used in an MSC Nastran analysis in any of the structural solution sequences, however, these elements should be used with care in geometric nonlinear analysis (see the *MSC Nastran Handbook for Nonlinear Analysis*). The nine rigid elements are ignored in the heat transfer solution sequences. See Rigid Element Enhancements, 159 for nonlinear enhancements to the rigid element.

User Input

A Bulk Data entry is associated with each of the nine rigid elements (see the Bulk Data Entries (p. 1119) in the *MSC Nastran Quick Reference Guide*). Before proceeding with a discussion of input requirements, it is important to understand how MPC and rigid element entries are treated.

Each MPC entry generates a single equation of the form

$$A_1 u_1 + A_2 u_2 + \ldots A_n u_n = 0 \qquad (3\text{-}34)$$

where u_1, u_2, ..., u_n are user-designated degrees of freedom (grid point plus component) and A_1, A_2, ..., A_n are coefficients which are user-supplied. The first named degree of freedom is placed in the u_m set (degrees of freedom eliminated by multipoint constraints). A rigid element (RROD, RBAR, RTRPLT, RBE1, RBE3, RSPLINE, RBE2, RJOINT, or RSSCON) generates a number of equations of the form of (3-34). The user designates the degrees of freedom to be placed in the u_m set (one for each equation of constraint). In some cases (RBAR and RTRPLT), the designation can be

accomplished by a default option. The coefficients in (3-34) are computed automatically from geometric data and for the RBE3 element. For the RSSCON, the M-set degrees of freedom are automatically assigned. The complete u_m set consists of the first named terms on the MPC entries plus the designated degrees of freedom on the rigid element entries. The user has complete control over the membership of this set.

Input Requirements

The following table lists the procedural requirements and rules that the user must follow when using MPCs and the (RBAR, RTRPLT, RBE1, RBE2, RBE3, RSSCON, and RSPLINE) rigid elements in an analysis.

1. A member of the u_m set cannot also be a member of any other user defined set. User-defined sets include:

   u_s -- degrees of freedom eliminated by SPCi entries, AUTOSPC, and PS field on GRID entries

   u_o -- degrees of freedom specified on OMITi Bulk Data entries

   u_r -- degrees of freedom specified on SUPORT and SUPORT1 Bulk Data entries

   u_a -- members of the analysis set specified on ASETi Bulk Data entries or exterior degrees of freedom in superelement analysis

2. A degree of freedom cannot be designated as a member of the u_m set more than once.

   A fatal error will result if, for example, the same degree of freedom is designated as dependent by two rigid elements or it the first-named degree of freedom on any MPC entry is also a designated member of u_m on a rigid element entry.

3. The user-selected independent degrees of freedom u_n, for the RBAR, RTRPLT, RBE1, RBE2, RBE3 and RSPLINE rigid elements must be sufficient to define any general rigid body motion of the element.

   These degrees of freedom are independent only for the particular element, and they may be declared dependent by other rigid element, MPCs, SPCs, OMITs, or SUPORTs. As far as a particular rigid element is concerned, it is always acceptable to select all six independent degrees of freedom at one grid point. This may not, however, be a good choice when the total problem requirements are considered.

   For six of the elements under discussion (RBAR, RTRPLT, RBE1, RBE3, RSPLINE, and RBE2), the user lists the degrees of freedom in u_m and u_n. The remaining degrees of freedom at the grid points to which the rigid element is jointed are not involved with the rigid element. This lack of connection represents either a sliding or rotating joint release, or both. The rigid rod element (RROD) is an exception because once a component of translation is placed in u_m, all of the five remaining components of translation will automatically be placed in u_n. The rotational degrees of freedom are not involved in the RROD element.

4. The user must avoid over-constraining the structure when two or more rigid elements are used.

   A structure is over-constrained when the degrees of freedom, which remain after the members of u_m have been selected, are insufficient to represent a general rigid body motion of the structure as a whole.

Consider, for example, a number of RBAR elements connected together to form a rigid ring. Let the grid points be numbered from 1 to N and assume that the u_m degrees of freedom for each rigid element are placed at the higher numbered grid point so that the only degrees of freedom which remain independent as each element is added to the ring are those at grid point 1. The addition of the last rigid element between grid points N and 1 will remove even those independent degrees of freedom and thereby over-constrain the structure.

5. For the RSSCON, the shell degrees of freedom are placed in u_m. The translational degrees of freedom of the shell edge are connected to the translational degrees of freedom of the upper- and lower-edge of the solid. The shell's two rotational degrees of freedom are also connected to the translational degrees of freedom of the upper and lower edge of the solid. The RSSCON only impresses a rigid constraint on the shell's two rotational degrees of freedom.
6. Nonlinear forces in dynamic analysis cannot be applied to members of the u_m set.

Rigid Element Enhancements

Rigid elements, such as RBAR, RBE1, … etc., have been available in MSC Nastran since the inception of MSC Nastran. However, they contain limitations, which are discussed below.

In the "Theory" section, first, an introduction is given to the formulation of large rotation theory used by the Lagrange rigid element, followed by a summary of theoretical formulation for each type of the rigid element is given.

Finally, the "User Interface" and "Examples" sections demonstrate the usage of the new Lagrange rigid elements.

Types of Rigid Element

In MSC Nastran, two types of rigid elements are available -- the linear rigid element and the Lagrange rigid element.

Linear Rigid Elements

Linear rigid elements have been available since the inception of MSC Nastran. The linear rigid elements use the linear elimination technique and are not "real" finite elements. Instead, they are internally represented by a set of MPC equations. By using these MPC equations, the dependent degrees of freedom (the m-set) are eliminated from the solution set (the l-set). These types of rigid elements have the following limitations. They:

- Do not compute thermal load.
- Do not have differential stiffness matrix so that the solutions are incorrect for the buckling analysis or other solution sequences where the differential stiffness matrix is required.
- Use the small rotation theory in the geometrical nonlinear analysis so that the solutions are incorrect in this type of analysis.
- Use the elimination method for solution resulting in very dense stiffness matrices. These dense matrices cannot take advantages of the sparse matrix algorithm.

The Lagrange rigid element, which is discussed in next sub-section, does not have the above limitations.

Lagrange Rigid Elements

These Lagrange rigid elements are “real” finite elements, similar to, for example, QUAD4 element. Instead of using MPC equations, the element stiffness matrix is computed for each Lagrange rigid element. All the limitations for the linear rigid element are removed, i.e., they

- Include thermal load effect.
- Include the differential stiffness.
- Support for both the small rotation and the large rotation. Use the large rotation theory in the geometrical nonlinear analysis (PARAM,LDISP,1).
- Take advantages of the sparse matrix algorithm if the augmented Lagrange multiplier method, defined below, is used for solution.

For each Lagrange rigid element, a number of Lagrange multiplier degrees of freedom are created by MSC Nastran internally. For example, one to six Lagrange multiplier degrees of freedom are created for RBAR and one Lagrange multiplier degree of freedom is created for RROD. For each Lagrange rigid element, the independent degrees of freedom, the dependent degrees of freedom, and the Lagrange multiplier degrees of freedom are first placed in the solution set (the l-set). Two methods are available for solution:

Augmented Lagrange Multiplier Method

For this method, the solution is obtained with the independent degrees of freedom, the dependent degrees of freedom, and the Lagrange multipliers degrees of freedom in the solution set. Thus, the sparse characteristic of the stiffness matrix is maintained and the sparse matrix algorithm can be utilized.

Lagrange Elimination Method

For this method, additional operations of eliminating both the dependent and Lagrange multiplier degrees of freedom from the solution set are performed after the global stiffness matrix is assembled. The solution is performed on the independent set (the n-set), creating dense matrices after elimination. Once again, these dense matrices cannot take advantage of the efficiency features of the sparse method.

Method Selection

For most problems, the augmented Lagrange multiplier method is the preferred method. The tangent stiffness matrix using this method, however, is not positive definite, and can potentially pose numerical difficulties. It is for this reason that the Lagrange eliminator method is introduced as a backup method.

For linear rigid elements, the dependent degrees of freedom are eliminated and placed in the mp-set. For the augmented Lagrange multiplier rigid elements, both the dependent degrees of freedom and the internally created Lagrange multipliers are placed in the l-set. For the Lagrange elimination rigid elements, both the dependent degrees of freedom and the Lagrange multipliers are eliminated and placed in the mr-set. The mp-set and the mr-set are subsets of the m-set. Please see Degree-of-Freedom Set Definitions (p. 1102) in the *MSC Nastran Quick Reference Guide*.

The Lagrange elements are only available for linear static analysis (SOL 101), normal modes analysis (SOL 103), buckling analysis (SOL 105), and general nonlinear analysis (SOL 400). Furthermore, the Lagrange elimination method is not implemented for the following reasons:

- For nonlinear analysis, the tangent stiffness matrix must be updated at each nonlinear iteration for the Lagrange elimination method. (See The Lagrange Elimination Method, 165.) This implies that only the full Newton-Raphson method can be used, which conflicts with the general iteration solution scheme employed in MSC Nastran.
- The denseness of the tangent stiffness matrix after elimination poses an inefficiency problem for the sparse solver.
- For nonlinear analysis, only stiffness matrix decomposition is required and the eigenvalue analysis is not required. For stiffness matrix decomposition, the method for solving a non-positive definite matrix is well established, and hence removes the need for a backup method for the Lagrange multiplier method.

There are some additional limitations for SOL 400. Please see Limitations on SOL 400, 169.

Theory

In this section, we give introduction to the theoretical formulation of the rigid elements. Rigid elements use large rotation theory and hence warrant a discussion of large rotation. In the following sections, the method for representing large rotation for rigid elements is presented first, followed by the solution methods for the linear elimination method, the Lagrange multiplier method, and the Lagrange multiplier method with elimination. Also, only the solution methods for the nonlinear analysis are discussed.

Rotation Vector

Mathematically, there are many ways of representing large rotation; the most commonly used methods are Euler angles, Gimbal angles, Rodrigues parameters, quaternion (Euler parameters), and rotation vector methods. MSC Nastran implemented the Gimbal angles and the rotation vector method into the geometric nonlinear analysis. The Gimbal angles have inherent limitations. They are non-invariant parameters, which means that their values are intimately associated with the coordinate system used. On the other hand, the rotation vector gives a simple geometric meaning of rotations and it is an invariant parameter. Thus, the Lagrange elements use the rotation vector for representing large rotations, which is described in the next paragraph. Please note that the Lagrange elements do not have an option to use the Gimbal angles.

The most general motions of a rigid body can be represented by a translation of a base point and rotation about the base point. Therefore, the motion of a GRID point rigid body can be decomposed into a translation of the GRID point and a rotation defined by:

$$\Psi = \psi \mathbf{p}$$

where:

$$\psi = \|\Psi\| = (\Psi^T \Psi)^{\frac{1}{2}}$$

$$\mathbf{p} = \frac{\Psi}{\psi} = \begin{Bmatrix} p_x \\ p_y \\ p_z \end{Bmatrix} \tag{3-35}$$

Thus ψ is the magnitude or the angle of the rotation and $\mathbf{p}$ is the direction of the rotation.

It is well known that large rotations cannot be treated as vectors. Consequently, the assumption that rotations are linear, which is one of the basic assumptions of linear analysis, does not provide a unique solution when a series of large rotations is applied in different sequences. However, the effect of large rotation can be expressed in terms of the rotation matrix $\mathbf{R}(\Psi)$. Any vector $\mathbf{v}$ can be rotated into new position v_r given by the equation:

$$\mathbf{v}_r = \mathbf{R}(\Psi)\mathbf{v}$$

$$\mathbf{R}(\Psi) = \begin{bmatrix} cos\psi + p_x^2(1-cos\psi) & -p_z sin\psi + p_x p_y(1-cos\psi) & p_y sin\psi + p_x p_z(1-cos\psi) \\ p_z sin\psi + p_x p_y(1-cos\psi) & cos\psi + p_y^2(1-cos\psi) & -p_x sin\psi + p_y p_z(1-cos\psi) \\ -p_y sin\psi + p_x p_z(1-cos\psi) & p_x sin\psi + p_y p_z(1-cos\psi) & cos\psi + p_z^2(1-cos\psi) \end{bmatrix} \tag{3-36}$$

The rotational matrix is an orthogonal matrix, which has the property:

$$\mathbf{R}\mathbf{R}^T = \mathbf{I} \quad \text{or} \quad \mathbf{R}^{-1} = \mathbf{R}^T$$

A very important matrix associated with the rotation vector Ψ is the skew-symmetric matrix Ψ, which is defined by:

$$\hat{\Psi} = \begin{bmatrix} 0 & -\psi_z & \psi_y \\ \psi_z & 0 & -\psi_x \\ -\psi_y & \psi_x & 0 \end{bmatrix} \tag{3-37}$$

The rotation vector Ψ and its skew-symmetric matrix have the relationship:

$$\hat{\Psi}\Psi = 0 \quad \text{and} \quad \hat{\Psi}\mathbf{v} = \Psi \times \mathbf{v} \quad \text{for any vector } \mathbf{v}$$

Ψ is the axial vector of the skew-symmetric matrix $\hat{\Psi}$. In term of the skew-symmetric matrix and the unit vector $\mathbf{p}$ of rotation, the rotation matrix can be rewritten as:

$$\mathbf{R}(\Psi) = \mathbf{p}\mathbf{p}^T + cos\psi(\mathbf{I} - \mathbf{p}\mathbf{p}^T) + sin\psi\hat{\mathbf{p}} \tag{3-38}$$

where $\hat{\mathbf{p}}$ is the skew-symmetric matrix of the unit vector $\mathbf{p}$.

Composite Rotations -- Left Rotation or Right Rotation

A composite rotation is successive application of two or more rotations. For large rotation, this series of rotations is no longer additive. In addition, they are not commutative, i.e., the order of application of the rotation is important. For example, in a geometric nonlinear analysis, an incremental rotation $\Delta\theta_g$ is computed for a load increment or a nonlinear iteration. The composite rotation of $\Delta\theta_g$ and the previous rotation Ψ_i can be expressed by the equation:

$$\mathbf{R}_{i+1}(\Psi_{i+1}) = \Delta\mathbf{R}_l(\Delta\theta_g)\mathbf{R}_i(\Psi_i) \tag{3-39}$$

Let us examine the coordinate system used for (3-39). Both Ψ_{i+1} and Ψ_i are computed in terms of the global coordinate system at the GRID point. Note that the global coordinate system at any GRID point is fixed in space. Here we must be very careful about the coordinate system used for the incremental rotation $\Delta\theta_g$. In order for (3-39) to be true, we must

compute the incremental rotation in terms of the same coordinate system used by Ψ_{i+1} and Ψ_i, i.e., the global coordinate system at the GRID point. The corresponding incremental rotation matrix $\Delta\mathbf{R}_l$ computed is called ***the left rotation***.

There is another way to compute the incremental rotation matrix. Let us rotate the global coordinate system at the GRID point by Ψ_i, i.e.

$$\mathbf{m}_x = \mathbf{R}_i(\Psi_i)\mathbf{g}_x \tag{3-40}$$

where $\boldsymbol{g}_x$, $x = 1, 2, 3$, are the unit vectors of the global coordinate system **g** at the GRID point and $\mathbf{m}_x$ are the unit vectors of the rotated coordinate system **m**. The coordinate system **m** is called the material coordinate system at the GRID point. Please note that the material coordinate system is ***not*** fixed in the space and it is rotated with the GRID point. Now, let us compute the incremental rotation $\Delta\theta_m$ in terms of the material coordinate system. Then, the composite rotation of the incremental rotation and the previous rotation can be expressed by:

$$\mathbf{R}_{i+1}(\Psi_{i+1}) = \mathbf{R}_i(\Psi_i)\Delta\mathbf{R}_r(\Delta\theta_m) \tag{3-41}$$

The rotation matrix $\Delta\mathbf{R}_r$ so computed is called ***the right rotation***. It can be shown that the right rotation and the left rotation are related by the equations:

$$\Delta\theta_m = \mathbf{R}_i^T\Delta\theta_g \tag{3-42}$$

$$\Delta\mathbf{R}_r = \mathbf{R}_i^T\Delta\mathbf{R}_l\mathbf{R}_i \tag{3-43}$$

The right rotation has the following advantages:

- The representation of inertia forces is greatly simplified.
- The stiffness properties of each elastic member may be described in a rigorous manner, including the geometric stiffening effects.

To improve the geometric nonlinear analysis, the rigid elements are formulated in terms of the right rotation approach. However, all the existing elements, CQUAR4, CTRIA3, and CBEAM, are formulated in terms of the left rotation approach. In order to make the existing elements compatible with the rigid elements, the existing elements are modified in terms of the right rotation approach.

The Linear Elimination Method

For the linear elimination method, all rigid elements are reduced using MPC constraint equations given by:

$$\mathbf{u}_m = \mathbf{G}_m\mathbf{u}_n \tag{3-44}$$

where $\mathbf{u}_m$ are the dependent degrees of freedom, $\mathbf{u}_n$ are the independent degrees of freedom, and $\mathbf{G}_m$ is the constraint matrix. During the solution process, the dependent degrees of freedom are eliminated from the solution set (the l-set). This method is valid only for small rotation because the relationship given by (3-44) is linear. Also, the thermal loads and the differential stiffness matrix are not computed. For large rotation, the relationships between dependent degrees of freedom and independent degrees of freedom are nonlinear. This elimination method can be modified to solve the nonlinear rigid element problem. However, we use the augmented Lagrange multiplier method instead, which is discussed in next sub-section.

The Augmented Lagrange Multiplier Method

The equilibrium equation for a static nonlinear solution of a constrained structural model using the Lagrange multiplier method is given by:

$$\begin{aligned} \mathbf{F} + \mathbf{H}^T\lambda &= \mathbf{p} \\ \Phi(\mathbf{u}) &= \mathbf{q} \end{aligned} \tag{3-45}$$

where **F** is the internal force vector, λ is a vector of Lagrange multipliers, **p** is the external force vector, **u** is the displacement vector, Φ is a vector of nonlinear constraints on the displacements generated by the rigid elements, **q** is the external enforced displacements or effect due to thermal loads, and **H** is the displacement gradient given by:

$$\mathbf{H} = \frac{\partial}{\partial \mathbf{u}}(\Phi(\mathbf{u}) - \mathbf{q}) = \frac{\partial \Phi}{\partial \mathbf{u}} \tag{3-46}$$

The explicit form for Φ and **H** are dependent on the individual rigid element type. For example, the forms are different for the RBAR and the RBE3.

The equilibrium for (3-45) is obtained by minimizing the functional:

$$\Pi = \Pi_k + k(\Phi - \mathbf{q})^T\lambda + \frac{1}{2}(\Phi - \mathbf{q})^T p(\Phi - \mathbf{q}) \tag{3-47}$$

The first term Π_k of (3-47) contains the regular strain energy and external work done to the system. For rigid elements, this term is zero. The second term is contribution due to the Lagrange multipliers, where k is a scale factor introduced to balance the magnitude of the Lagrange multipliers with other terms in the stiffness matrix. The last term is the added penalty function term and p is the penalty function. The reason for adding the penalty function term is to reduce numerical difficulty during matrix decomposition.

The constrained equilibrium equation can be also solved by the ***penalty function method.*** For this method, the magnitude of the penalty function must be very large in comparison to the tangential stiffness matrix in order to obtain an approximation solution of the original constrained equation. When the penalty function approaches infinity, the solution becomes the exact solution of the original constrained equation. The shortcoming of this method is that, when the penalty function approaches infinity, the tangential stiffness matrix becomes ill conditioned. On the other hand, in the augmented Lagrangian method as given by (3-47), the penalty function p needs not to be large. It only needs to be a similar magnitude as the other terms in the tangential stiffness matrix for enforcing the positive definiteness of the stiffness matrix.

For the nonlinear analysis, the next step is to derive the Newton-Raphson iteration equation from (3-47). Let $(\mathbf{u}^*, \lambda^*)$ be an approximation solution to (3-47). By using the Newton-Raphson method, a corrected solution can be obtained in the form:

$$(u, \lambda) = (\mathbf{u}^* + \Delta\mathbf{u}, \lambda^* + \Delta\lambda) \tag{3-48}$$

$$\boldsymbol{K}_{ee}\begin{Bmatrix} \Delta\mathbf{u} \\ \Delta\lambda \end{Bmatrix} = \begin{Bmatrix} \mathbf{e}_u \\ \mathbf{e}_\lambda \end{Bmatrix} \tag{3-49}$$

where $\mathbf{K}_{ee}$ is the tangential stiffness matrix and the right hand side is the error vector. The first and second variations of (3-47) are

$$\delta\Pi = \delta\mathbf{u}^T\mathbf{H}^T(p(\Phi - \mathbf{q}) + k\lambda) + \delta\lambda^T k(\Phi - \mathbf{q}) \tag{3-50}$$

$$d(\delta\Pi) = \delta\mathbf{u}^T\mathbf{H}^T(p\mathbf{H}d\mathbf{u} + kd\lambda) + \delta\lambda^T k\mathbf{H}d\mathbf{u} + \delta\mathbf{u}^T\frac{\partial}{\partial\mathbf{u}}(\mathbf{H}^T(p(\Phi - \mathbf{q}) + k\lambda))d\mathbf{u} \tag{3-51}$$

From (3-51), we obtain the corresponding element tangential stiffness matrix as:

$$\mathbf{K}_{ee} = \mathbf{K}_{ee}^e + \mathbf{K}_{ee}^d = \begin{bmatrix} \mathbf{H}^T p\mathbf{H} & k\mathbf{H}^T \\ k\mathbf{H} & 0 \end{bmatrix} + \begin{bmatrix} \frac{\partial}{\partial\mathbf{u}}(\mathbf{H}^T(p(\Phi - \mathbf{q}) + k\lambda)) & 0 \\ 0 & 0 \end{bmatrix} \tag{3-52}$$

In the above equation, the first term is the element stiffness and the second term is the differential stiffness for the rigid element. Next, we need to compute the error vector for the rigid element in a nonlinear iteration. For the approximate solution $(\mathbf{u}^*, \lambda^*)$, the error vector can be obtained from (3-50) as

$$\begin{Bmatrix} \mathbf{e}_u \\ \mathbf{e}_\lambda \end{Bmatrix} = \begin{Bmatrix} -\mathbf{H}^T(\mathbf{u}^*)(k\lambda^* + p(\Phi(\mathbf{u}^*) - \mathbf{q})) \\ k(\mathbf{q} - \Phi(\mathbf{u}^*)) \end{Bmatrix} \tag{3-53}$$

With the tangential stiffness matrix and the error vector defined, (3-48) and (3-49) are used to obtain a better approximation. This process continues until the solution converges.

The Lagrange Elimination Method

The stiffness matrix for the Lagrange multiplier method given by (3-52) is not positive definite. For most problems, this method is appropriate. However, for certain type of problems, this method may create numerical difficulties. Therefore, the Lagrange multiplier method with elimination, called the Lagrange elimination method, is implemented as a backup method for the Lagrange multiplier method. For reasons given in Types of Rigid Element, 159, the Lagrange elimination method is implemented only for the linear static analysis, the normal modes analysis, and the buckling analysis. However, for completeness, the theory given below is still based on the nonlinear analysis.

The Lagrange elimination method is summarized as follows:

- Compute the rigid element tangential stiffness matrix and error vector by using the Lagrange multiplier method as given by (3-52) and (3-53) with the penalty function p=0.
- Assemble the element stiffness and error vector into global stiffness and error vector, which will create a system equation similar to (3-49).
- Eliminate the Lagrange multipliers and the dependent degrees of freedom from the tangential stiffness matrix and the error vector.
- Construct and solve the Newton-Raphson equation in terms of the independent degrees of freedom.

The method of elimination is derived as follows. Let *g* denote the total displacement degrees of freedom for the structural model, *m* the dependent degrees of freedom, and *n* the independent degrees of freedom. Then the displacement gradient defined by (3-46) can be partitioned into m-set and n-set as

$$\mathbf{H}_{mg}\Delta\mathbf{u}_g = \begin{bmatrix} \mathbf{H}_{mm} & \mathbf{H}_{mn} \end{bmatrix} \begin{Bmatrix} \Delta\mathbf{u}_m \\ \Delta\mathbf{u}_n \end{Bmatrix} \tag{3-54}$$

And the constraint equation for the dependent degrees of freedom can be computed as

$$\begin{aligned} \Delta\mathbf{u}_m &= \mathbf{G}_{mn}\Delta\mathbf{u}_n \\ \mathbf{G}_{mm} &= -\mathbf{H}_{mm}^{-1}\mathbf{H}_{mn} \end{aligned} \tag{3-55}$$

Now, the global tangential stiffness matrix in g-set, similar to (3-49), (3-52), and (3-53) can be expanded into m-set and n-set as

$$\begin{bmatrix} \bar{\mathbf{K}}_{nn} & \mathbf{K}_{nm} & \mathbf{G}_{mn}^T \\ \mathbf{K}_{nm}^T & \mathbf{K}_{mm} & -\mathbf{I} \\ \mathbf{G}_{mn} & -\mathbf{I} & 0 \end{bmatrix} \begin{Bmatrix} \Delta\mathbf{u}_n \\ \Delta\mathbf{u}_m \\ \Delta\lambda_m \end{Bmatrix} = \begin{Bmatrix} \mathbf{p}_n - \mathbf{F}_n(\mathbf{u}_g^*) - \mathbf{G}_{mn}^T(\mathbf{u}_g^*)\lambda_m^* \\ \mathbf{p}_m - \mathbf{F}_m(\mathbf{u}_g^*) + \lambda_m^* \\ -\mathbf{H}_{mm}^{-1}(\mathbf{q}_m - \Phi(\mathbf{u}_g^*)) \end{Bmatrix} = \begin{Bmatrix} \bar{\mathbf{e}}_n \\ \mathbf{e}_m \\ \bar{\mathbf{q}}_m \end{Bmatrix} \tag{3-56}$$

The right hand side of above equation is the error vector. Elimination of m-set displacements and the Lagrange multipliers from above equation yields the tangent stiffness matrix in terms of independent degrees of freedom, the n-set, as:

$$\mathbf{K}_{nn}\Delta\mathbf{u}_n = \mathbf{e}_n \tag{3-57}$$

$$\mathbf{K}_{nn} = \bar{\mathbf{K}}_{nn} + \mathbf{K}_{nm}\mathbf{G}_{mn} + \mathbf{G}_{mn}^T\mathbf{K}_{nm}^T + \mathbf{G}_{mn}^T\mathbf{K}_{mm}\mathbf{G}_{mn} \tag{3-58}$$

$$\mathbf{e}_n = \bar{\mathbf{e}}_n + \mathbf{G}_{mn}^T\mathbf{e}_m + (\mathbf{K}_{nm} + \mathbf{G}_{mn}^T\mathbf{K}_{mm})\bar{\mathbf{q}}_m \tag{3-59}$$

$$\Delta\mathbf{u}_m = \mathbf{G}_{mn}\Delta\mathbf{u}_n - \bar{\mathbf{q}}_m \tag{3-60}$$

$$\Delta\lambda_m = \mathbf{K}_{nm}^T\Delta\mathbf{u}_n + \mathbf{K}_{mm}\Delta\mathbf{u}_m - \mathbf{e}_m \tag{3-61}$$

(3-57), (3-58), and (3-59) can be used in the Newton-Raphson iterations, and (3-60) and (3-61) can be used to recover the displacements in m-set and the Lagrange multipliers. In general, the error vector given by (3-57) is dependent on the tangential stiffness matrix. In order to compute the correct error vector, the stiffness matrix must be updated at each nonlinear iteration. This means only the full Newton-Raphson method can be used to obtain the solution for the Lagrange elimination method.

User Interfaces

Case Control Command RIGID

Control Case command RIGID selects the type of rigid element. It has the following format:

$$\text{RIGID} = \begin{Bmatrix} \text{LINEAR} \\ \text{LAGR} \\ \text{LGELIM} \end{Bmatrix}$$

LINEAR will select the linear rigid elements, LAGR will select the Lagrange rigid element with the Lagrange multiplier method, and LGELIM will select the Lagrange element with the Lagrange elimination method.

If RIGID command does not exist in the user Case Control file, the linear rigid element will be used for all solution sequences except SOL 400. For SOL 400, the default rigid element type is the Lagrange rigid element with the Lagrange multiplier method.

Parameters LANGLE, LMFACT, PENFN, and ORIGID

The parameter LANGLE selects the method to represent the lager rotations, which has the following values:

LANGLE,1 -- use the Gimbal angle method.

LANGLE,2 -- use the left rotation method.

LANGLE,3 -- use the right rotation method.

See sub-section, Composite Rotations -- Left Rotation or Right Rotation, 162 for a description of the rotation methods. LANGLE=1 or 2 is the existing method. LANGLE=3 is the new method implemented for the Lagrange rigid elements and is the only method available for the Lagrange rigid elements. Therefore, if there are Lagrange rigid elements in a structural model, MSC Nastran will automatically use LANGLE=3, and ignore user input on the LANGLE parameter.

The parameters LMFACT and PENFN are the scale factor k and penalty function p (see (3-47)) for the Lagrange rigid element with the Lagrange multiplier method. The purpose of LMFACT and PENFN is to make the values of the stiffness matrix for the Lagrange rigid elements about the same relative magnitude as those of the other elements in a structural model. Too small a value produces inaccurate results and too large a value poses numerical difficulties. Usually, the same value is assign to both LMFACT and PENFN. Under special requirement, user may assign different values to LMFACT and PENFN. For example, if $\text{LMFACT} \neq 0.0$ and PENFN = 0.0, then the solution method becomes the pure Lagrange multiplier method instead of the augmented Lagrange multiplier method. However, user must exercise caution when assigning different values to LMFACT and PENFN. The default value is 1.0e+5 for all solution sequences except SOL 400. For the nonlinear solution sequence SOL 400, MSC Nastran will compute the appropriate default value in nonlinear iterations.

The parameter ORIGID is the offset for the identification number of the Lagrange multiplier. MSC Nastran internally generates the Lagrange multiplier degrees of freedom for the Lagrange elements. The identification number for the Lagrange multiplier is ORIGID+MGID, where MGID is the grid point identification number of the dependent grid point of the rigid element. The default value for ORIGID is 101000000 if there are no other internally generated grid points. This value is modifiable using system cell 178

Bulk Data Entries

Existing rigid elements -- RBAR, RBE1, RBE2, RBE3, RROD, and RTRPLT -- can be used as either the linear rigid element or the Lagrange element, which is selected by the Case Control command, RIGID. In addition, there are three new Bulk Data entries, RBAR1, RTRPLT1, and RJOINT, which are discussed in next two sub-sections.

For each Lagrange rigid element, a number of Lagrange multiplier degrees of freedom are created internally by MSC Nastran. The number of the Lagrange multiplier degrees of freedom is equal to the dependent degrees of freedom defined by the rigid element. The identification numbers of Lagrange multiplier are determined by the parameter ORIGID.

All of the rigid elements described above can either be the linear rigid element or the Lagrange rigid element. However, the input rules are not the same for these two types of rigid element. The following paragraphs discuss the differences.

Except for the RBE3 and RROD, the major difference in the input format between the linear rigid element and the Lagrange rigid element is the selection of the independent degrees of freedom. This can be illustrated by the Bulk Data entry for RBAR:

RBAR	EID	GA	GB	CNA	CNB	CMA	CMB	ALPHA	

The independent degrees of freedom are selected by CNA and CNB. For the linear rigid element, the independent degrees of freedom can be assign to both CNA and CNB: for example CNA=1236, CNB=34, as long as the total number equals to six and they can jointly represent any general rigid body motion. However, for the Lagrange rigid element, all six independent degrees of freedom must be assigned to a single grid point, i.e., “CNA=123456, CNB=blank”, or “CNA=blank, CNB=123456”. The same rule applies to the RBE1, RBE2, and RTRPLT elements.

For the RBE3, the REFC degrees of freedom can be any combination of the integer 1 through 6 for the linear rigid element. For the Lagrange rigid element, RFEC must be 123, 456, or 123456.

For the RROD element, the user must select one dependent degree of freedom by inputting either CMA or CMB for the linear rigid element. However, for Lagrange rigid element, user can leave both fields blank and let MSC Nastran to select the best component as the dependent degree of freedom. In fact, this is the recommended method.

For all rigid elements, there is a new field ALPHA. This is the thermal coefficient of expansion. For the Lagrange rigid elements, if ALPHA is given and the thermal loads are requested by the Case Control command TEMPERATURE(INITIAL) and TEMPERATURE(LOAD), the thermal load effect will be computed for the rigid elements. The temperature loads are taken as the average temperature given by the independent grid point and the dependent grid point. For example, the temperature load for the BAR element is taken as the average temperature of grid points GA and GB. For the linear rigid element, no temperature effect is computed and the ALPHA field is ignored.

RBAR1 and RTRPLT1

RBAR1 and RTRPLT1 are two new Bulk Data entries and are alternative input formats for RBAR and RTRPLT. The Bulk Data entry for RBAR1 is shown below:

RBAR1	EID	GA	GB	CMB	ALPHA				

GA is the independent grid point and all six degrees of freedom are independent. The dependent degrees of freedom are selected by CMB of the dependent grid point GB. This is a much simpler input format than RBAR and is preferred input format for the Lagrange rigid element.

The format for RTRPLT1 is shown below:

RTRPLT1	EID	GA	GB	GC	CMB	CMC	ALPHA		

RTRPLT1 defines one independent grid point GA and two dependent grid points GB and GC with dependent degrees of freedom CMB and CMC, respectively. Between RTRPLT1 and RTRPLT, RTRPLT1 is the preferred input format for the Lagrange rigid element.

RJOINT

RJOINT is a new rigid element. The format for RJOINT is as follows:

RJOINT	EID	GA	GB	CB					

GA is the independent grid point and all six degrees of freedom are independent. GB is the dependent grid point. The length between points GA and GB must be zero. Since its length is zero, the thermal load effect is not applicable.

If CB=123456 or blank, then the grid point GB is constrained to move with the grid point GA and two the grid points move as a single point. If any degree of freedom is released on CB, then RJOINT becomes a mechanical joint. A mechanical joint is a mechanical system that has two bodies jointed at a point. The two bodies can rotate relatively about one, two or three axis of a local coordinate system at the point. For example:

- Hinge - A hinge is a mechanical joint that rotates freely about one axis about the local coordinate system. It can be simulated by RJOINT with one rotational degree of freedom released, i.e., CB = 12356, 12346, or 12345.
- Universal joint - A universal joint is a mechanical joint that rotates freely in two axes. It can be simulated by RJOINT with two rotational degrees of freedom released, i.e., CB = 1234, 1235, or 1236.
- Spherical joint - A spherical joint is a mechanical joint that rotates freely about all three axes. It can be simulated by a zero length RJOINT with all rotational degrees of freedom released, i.e. CB = 123.
- Prismatic joint - A prismatic joint is a mechanical system with two blocks that are constrained to have the same rotations, but translate relative with each other along a local axis. It can be simulated by RJOINT with one translational degree of freedom released, i.e., CB = 23456, 13456, 12456.
- Cylindrical joint - A cylindrical joint is a mechanical system that allows two grid points have relative translation along a moving axis and, at the same, have relative rotation about the same axis. It can be simulated by RJOINT with one translational degree of freedom and one rotational degree of freedom released, i.e. CB = 2356, 1346, 1245.

Guidelines for the RJOINT element:

- The theory for the RJOINT is formulated such that a consistent mechanical joint is created even if user requests different global coordinate systems at grid points GA and GB. If different global coordinate systems are used, the degrees of freedom to be released is determined by the coordinate system at GB.
- For linear or nonlinear static analysis, the joints must constrained by other elements in the structural model. Otherwise a singular stiffness matrix may be produced, leading to failure in decomposition.

Limitations on SOL 400

As described in sub-section, Composite Rotations -- Left Rotation or Right Rotation, 162, the Lagrange rigid elements are formulated in terms of the right rotation method and the existing elements, `CQUAD4`, `CTRIA3`, and `CBEAM`, are formulated in the left rotation method. In order to make these existing elements compatible with the Lagrange rigid elements, the existing elements must be modified to formulate in the right rotation method. This work is under development for the initial release of MSC Nastran 2004. Therefore, for the initial release, SOL 400 has the following limitations:

- If Lagrange rigid elements are used in a model that contains `CQUAD4`, `CTRIA3`, or `CBEAM` elements, then the problem will converge only for truly 2D motion. If there are follower force effects, the out-of-plane motion may need to be constrained in this case. Full 3D motion is supported with the Lagrange rigid elements if the model does not contain the above elements.

Examples

In this section, four simple examples are included to illustrate the usage of the Lagrange rigid elements. These four examples are

- Linear static analysis with thermal loads
- Nonlinear geometric analysis with thermal loads
- Buckling analysis
- Normal modes analysis with differential stiffness

Example 1: Linear Static Analysis with Thermal Loads (nlrgd11a.dat)

- To illustrate the effect of thermal loads on the Lagrange rigid elements in a static analysis, a simple beam is modeled by a RBAR1 element, which has a length of 1.0. At left end of the beam, a very soft rotational spring is connected to the normal rotation and other five degrees-of freedom are fixed. At right end, all six degrees of freedom are free and a vertical force of 100.0 is acting on it. This beam is subjected to a thermal load, which will double its length to 2.0. This problem is solved by the Lagrange multiplier method (RIGID=LAGR) with the linear static analysis (SOL 101).
- The input file for this problem is shown below.

```
ID RIGID,NLRGD10A
SOL 101
CEND
TITLE = LINEAR STATIC ANALYSIS - THERMAL LOAD ON RBAR
SUBTIT= LAGRANGE MULTIPLIER METHOD
RIGID = LAGR
TEMP(INIT) = 10
SUBCASE 1
  TEMP(LOAD) = 20
  LOAD = 100
  SPC  = 10
  DISP = ALL
  SPCF = ALL
  FORC = ALL
BEGIN BULK
FORCE,100,2,0,500.0,0.0,1.0,0.0
CELAS2,101,1000.0,1,6
GRID,1,,1.0,0.0,0.0
GRID,2,,2.0,0.0,0.0
RBAR1,3,1,2,123456,2.0-2
TEMP,10,1,0.0
TEMP,10,2,0.0
TEMP,20,1,50.0
TEMP,20,2,50.0
SPC1,10,12345,1
ENDDATA
```

Results for nlrgdlla.dat:

Description	Value
Vertical displacement	0.5
Horizontal displacement	1.0
Angle of rotation	28.6°
Initial beam length	1.0
Final beam length	2.06

These results show that horizontal thermal expansion is correctly computed as 1.0. However, because of the soft rotational spring on the left end of beam, this problem is a large displacement problem; therefore, the vertical and horizontal displacements, angle of rotation, and the final beam length are incorrect.

Example 2: Geometric Nonlinear Analysis with Thermal Loads (nlrgd11b.dat)

This problem is to illustrate the use of the Lagrange rigid elements in geometric nonlinear analysis. The model and loading for this problem is identical to nlrgd11a, except that the geometric nonlinear analysis (SOL 400 and LGDISP,1) is used. Note that the RIGID Case Control command in the is not needed for SOL 400 as the default for SOL 400 is RIGID=LAGR.

The input file for this problem follows.

```
ID RIGID,NLRGD11B
SOL 400
CEND
TITLE = GEOMETRIC NONLINEAR ANALYSIS - THERMAL LOADS ON RBAR
SUBTIT= LAGRANGE MULTIPLIER METHOD
TEMP(INIT) = 10
NLPARM = 10
SUBCASE 1
  TEMP(LOAD) = 20
  LOAD = 100
  SPC  = 10
  DISP = ALL
  SPCF = ALL
  FORC = ALL
BEGIN BULK
NLPARM,10,10
PARAM,LGDISP,+1
.
.
.
ENDDATA
```

Results for nlrgd11b.dat:

Description	Value
Vertical displacement	1.344
Horizontal displacement	0.481
Angle of rotation	42.22º
Initial beam length	1.0
Final beam length	2.0

Now the final beam length is 2.0, which is expected, and all above results are correct when large displacement theory is used.

Example 3: Buckling Analysis (nlrgd10d.dat)

To illustrate the use of the Lagrange rigid elements in a buckling analysis, two simple columns are modeled by RBAR elements and subject to compressive loads of 100.0 at top of the column. At bottom of the columns, they are connected to a single grid point and are constrained by very soft rotational spring in both horizontal directions. Also, at bottom, the three translations and axial rotation are fixed. The buckling analysis (SOL 105) and the Lagrange elimination method (RIGID=LGELIM) are used to obtain the solution.

The input file for this problem is shown below.

```
ID RIGID,NLRGD10D
SOL 105
CEND
TITLE = BUCKLING ANALYSIS - RBAR
SUBTI = LAGRANGE ELIMINATION METHOD
DISP = ALL
SPC  = 10
RIGID = LGELIM
SUBCASE 1
$ STATIC PRELOAD CASE
  LOAD = 100
SUBCASE 2
$ BUCKLING CASE
  METHOD = 10
BEGIN BULK
EIGRL,10,,,10
FORCE,100,4 ,0,-100.0,1.0,0.0,0.0
FORCE,100,41,0,-100.0,1.0,0.0,0.0
CELAS2,101,100.0,3,5
CELAS2,102,120.0,3,6
GRID, 3 ,,2.0,0.0,0.0
GRID, 4 ,,4.0,0.0,0.0
GRID, 41,,4.0,0.0,0.0
RBAR, 4,3,4
RBAR, 3,3,41,123456, , ,123456
SPC1,10,1234,3
```

```
ENDDATA
```

The buckling load factors calculated are 0.25 and 0.30.

Example 4: Normal Modes Analysis with Differential Stiffness (nlrgd14c.dat)

This problem illustrates the effect of differential stiffness matrix with the Lagrange rigid elements for normal modes analysis. The model of this problem is similar to that of nlrgd10a, except that concentrated masses are added to the top of the columns and tensile forces of 100.0 units are applied. The normal modes analysis (SOL 103) using the Lagrange multiplier method (RIGID=LAGR) are used to obtain the solution.

Two problems are investigated--one without and one with the effects of the differential stiffness matrix included.

The input file for the problem including the differential stiffness effect is shown below. For the problem without the differential stiffness, just remove subcase 1.

```
ID RIGID,NLRGD14C
SOL 103
CEND
TITLE = NORMAL MODES ANALYSIS WITH DIFFERENTIAL STIFFNESS - RBAR
SUBTI = LAGRANGE MULTIPLIER METHOD
DISP = ALL
SPC  = 10
RIGID = LAGR
SUBCASE 1
$ STATIC PRELOAD CASE
  LOAD = 100
SUBCASE 2
$ NORMAL MODES CASE
  STATSUB = 1
  METHOD = 10
BEGIN BULK
EIGRL,10,,,10
CONM2,201,4 ,,10.
CONM2,202,41,,10.
FORCE,100,4 ,0,+100.0,1.0,0.0,0.0
FORCE,100,41,0,+100.0,1.0,0.0,0.0
CELAS2,101,100.0, 3,5
CELAS2,102,120.0, 3,6
GRID, 3 ,,2.0,0.0,0.0
GRID, 4 ,,4.0,0.0,0.0
GRID, 41,,4.0,0.0,0.0
RBAR, 4,3,4
RBAR, 3,3,41,123456, , ,123456
SPC1,10,1234,3
ENDDATA
```

Results for nlrgd11b.dat:

Description	Value
Eigenvalues without preload	1.25, 1.50
Eigenvalues with preload	6.25, 6.50

From the above results, we can see clearly that the differential stiffness can affect the results substantially.

Scalar Elements (CELASi, CMASSi, CDAMPi)

Scalar elements are connected between pairs of degrees of freedom (at either scalar or geometric grid points) or between one degree of freedom and ground. Scalar elements are available as springs, masses, and viscous dampers. Scalar spring elements are useful for representing elastic properties that cannot be conveniently modeled with the usual structural elements (elements whose stiffnesses are derived from geometric properties). Scalar masses are useful for the selective representation of inertia properties, such as occurs when a concentrated mass is effectively isolated for motion in one direction only. The scalar damper is used to provide viscous damping between two selected degrees of freedom or between one degree of freedom and ground. It is possible, using only scalar elements and constraints, to construct a model for the linear behavior of any structure. However, use of scalar elements with offsets will cause incorrect results in buckling analysis and differential stiffness because the large displacement effects are not calculated. Offsets will also cause internal constraints in linear analysis, i.e., hidden constraints to ground. Therefore, it is expected that these elements will be used only when the usual structural elements are not satisfactory. Scalar elements are useful for modeling part of a structure with its vibration modes or when trying to consider electrical or heat transfer properties as part of an overall structural analysis. The reader is referred to Sections 5.5 and 5.6 of *The NASTRAN Theoretical Manual* for further discussions on the use of scalar elements.

The most general definition of a scalar spring is given with a CELAS1 entry. The associated properties are given on the PELAS entry. The properties include the magnitude of the elastic spring, a damping coefficient, and a stress coefficient to be used in stress recovery. The CELAS2 defines a scalar spring without reference to a property entry. The CELAS3 entry defines a scalar spring that is connected only to scalar points and the properties are given on a PELAS entry. The CELAS4 entry defines a scalar spring that is connected only to scalar points and without reference to a property entry. No damping coefficient or stress coefficient is available with the CELAS4 entry. Element force F is calculated from the equation.

Static:

$$F = k(u_1 - u_2)$$

Frequency:

$$F = (1 + i(g + g_e))k\overline{U} \;;\; \overline{U} = (u_1 - u_2)_{\text{real}} + i(u_1 - u_2)_{img} \tag{3-62}$$

Transient:

$$F = (u_1 - u_2)k + \left(\frac{g}{W3} + \frac{g_e}{W4}\right)k(\dot{u}_1 - \dot{u}_2) \tag{3-63}$$

where k is the stiffness coefficient for the scalar element and u_1 is the displacement of the first degree of freedom listed on its connection entry. Element stresses are calculated from the equation

$$\sigma = S \cdot F \tag{3-64}$$

where S is the stress coefficient on the connection or property entry and F is as defined above.

Scalar elements may be connected to ground without the use of constraint entries. Grounded connections are indicated on the connection entry by leaving the appropriate scalar identification number blank. Since the values for scalar elements are not functions of material properties, no references to such entries are needed.

The CMASS1, CMASS2, CMASS3, and CMASS4 entries define scalar masses in a manner similar to the scalar spring definitions. The associated PMASS entry contains only the magnitude of the scalar mass.

The CDAMP1, CDAMP2, CDAMP3, CDAMP4, and CDAMP5 entries define scalar dampers in a manner similar to the scalar spring definitions. The associated PDAMP entry contains only a value for the scalar damper. The mode displacement method (PARAM,DDRMM,-1) must be selected for element force output.

For the CDAMP, the force equations are:

Frequency:

$$F = i\omega B(u_1 - u_2)$$

Transient:

$$F = B(\dot{u}_1 - \dot{u}_2)$$

Axisymmetric Solid Elements (CTRIAX6, CTRIAX, CQUADX)

The axisymmetric elements CTRIAX6, CTRIAX, and CQUADX define a solid ring by sweeping a surface defined on a plane through a circular arc. The CTRIAX and CQUADX elements are described in Hyperelastic Elements, 197. The CTRIAX6 element is described as follows.

The triangular ring element (CTRIAX6) is a linear isoparametric element with an axisymmetric configuration that is restricted to axisymmetric applied loading. It is used for the modeling of axisymmetric, thick-walled structures of arbitrary profile. This element is not designed to be used with any other elements. Otherwise, CTRIAX6 is used in a conventional manner, and except for its own connection entry (and a pressure load entry, PLOADX1) it does not require special bulk data entries.

The coordinate system for the CTRIAX6 element is shown in Figure 3-46. Cylindrical anisotropy is optional. Orientation of the orthotropic axes in the (r, z) plane is specified by the angle θ. Deformation behavior of the element is described in terms of translations in the r and z directions at each of the six grid points. All other degrees of freedom should be constrained.

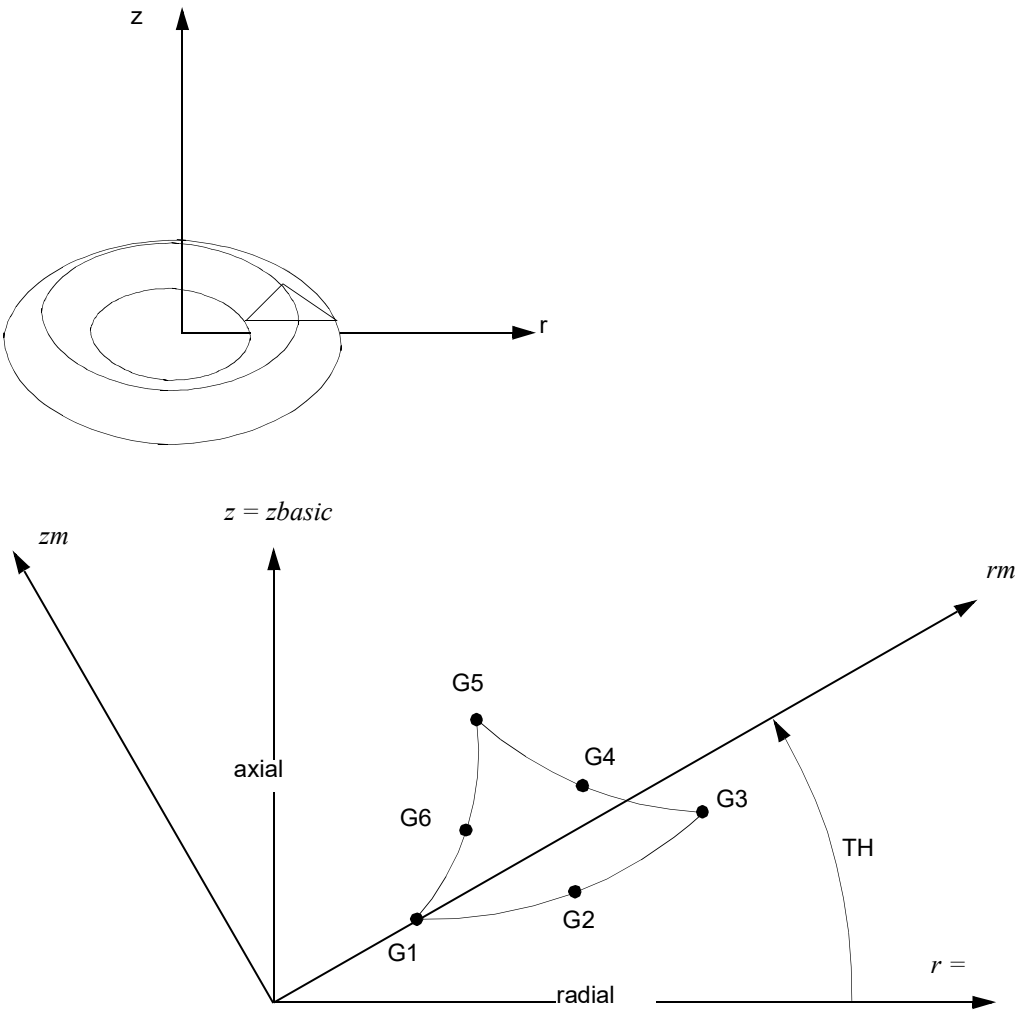


Figure 3-46 CTRIAX6 Element Coordinate System

The isoparametric triangular ring element is defined with a CTRIAX6 entry. No property entry is used for this element. The material property reference is given on the connection entry.

The following stresses, evaluated at the three vertex grid points and at the centroid of the element, are output on request:

- σ_r -- stress in r_m direction of material coordinate system.
- σ_θ -- stress in azimuthal direction.
- σ_z -- stress in z_m direction of material coordinate system.

- τ_{rz} -- shear stress in material coordinate system.
- Maximum principal stress.
- Maximum shear stress.
- von Mises equivalent or octahedral shear stress.

Crack Tip Elements (CRAC2D, CRAC3D)

MSC Nastran crack tip elements include both two-dimensional (CRAC2D) and three-dimensional (CRAC3D) types. ADUM8 and ADUM9 Bulk Data entries are needed with these elements. The formats of these entries are as follows:

1	2	3	4	5	6	7	8	9	10
ADUM8	18	0	5	0	CRAC2D				
ADUM9	64	0	6	0	CRAC3D				

The two geometries of the CRAC2D element are shown in Figure 3-47 and Figure 3-48. For the quadrilateral crack element, corner and crack grid points 1 through 10 are required while midside grid points 11 through 18 are optional. The CRAC3D element geometry is depicted in Figure 3-49. For the brick crack element, grid points 1 through 10 and 19 through 28 are required; whereas grid points 11 through 18 and 29 through 64 are optional.

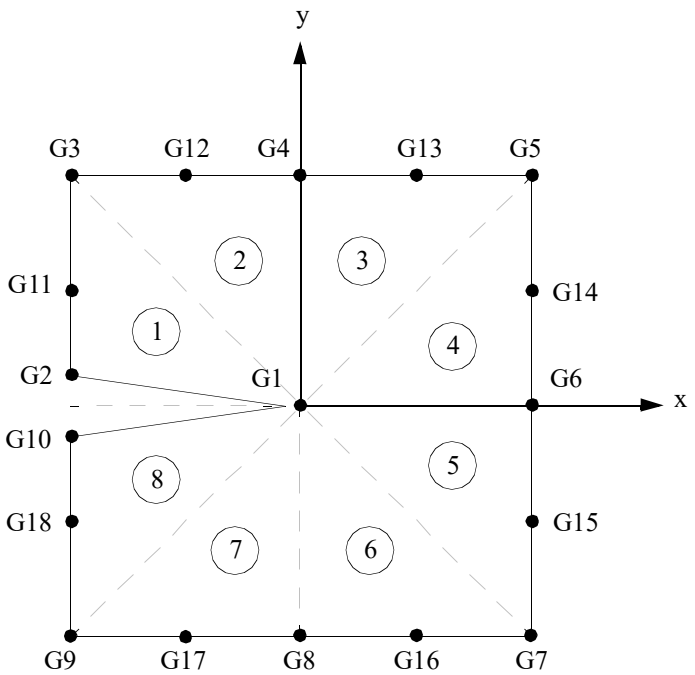


Figure 3-47 Quadrilateral Crack Element

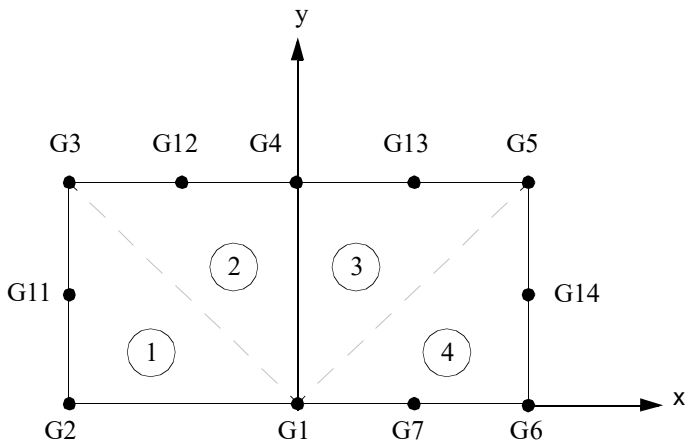


Figure 3-48 Symmetric Half-Crack Option

The CRAC2D and CRAC3D entries are used to define the element geometry, while properties are defined on PRAC2D and PRAC3D entries. Both elements can be modeled with temperature-independent anisotropic materials. The 2-D element may be either plane stress or plane strain. The element will generate either coupled or lumped mass matrices.

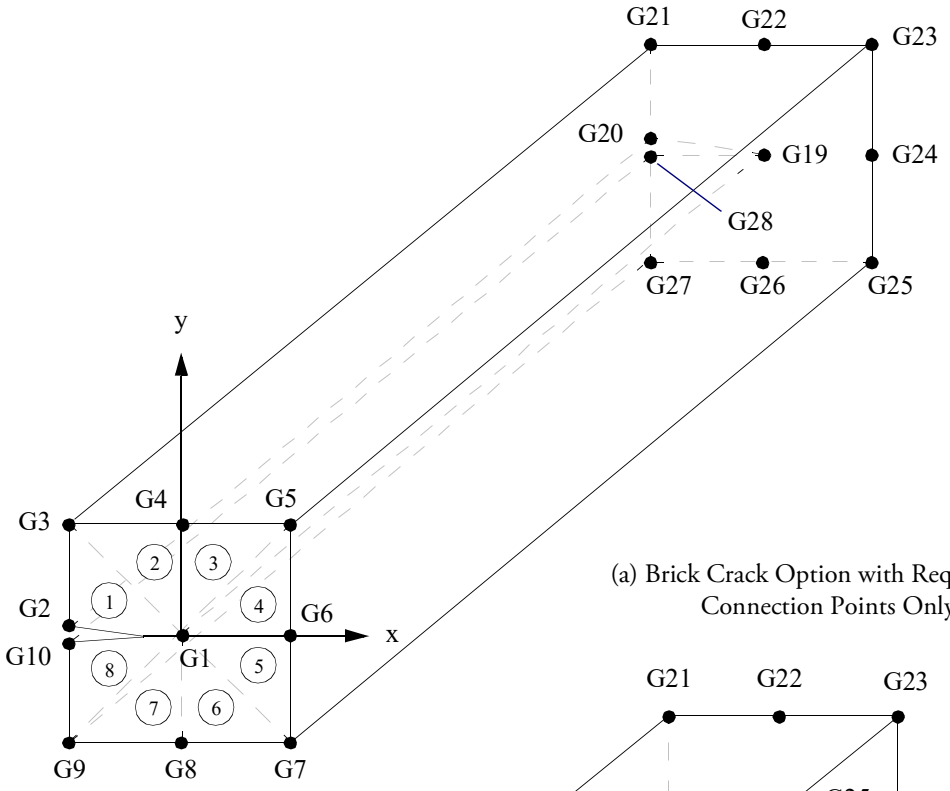
Figure 3-47 depicts the quadrilateral and symmetric half options for the CRAC2D element. The element is divided, by the program, into basic triangular elements 1 through 8, whereas the symmetric half-crack option is divided into triangles 1 through 4. Grid points 1 through 7 are required for the symmetric half-crack option, while grid points 11 through 14 are optional. Stresses and the local coordinates of these stresses for the quadrilateral option are computed at the origin of the natural coordinates of triangles 4 and 5. The stresses and coordinates are then averaged and reported. Stress intensity factors, K_I and K_{II}, are computed for triangles 1 and 8, averaged and reported. For the symmetric half-crack option, coordinates and stresses are reported at the origin of the natural coordinates of triangle 4 while the stress intensity factor K_I only is reported for triangle 1.

Interpretation of CRAC2D Element Stress Output (Dummy Element Format)

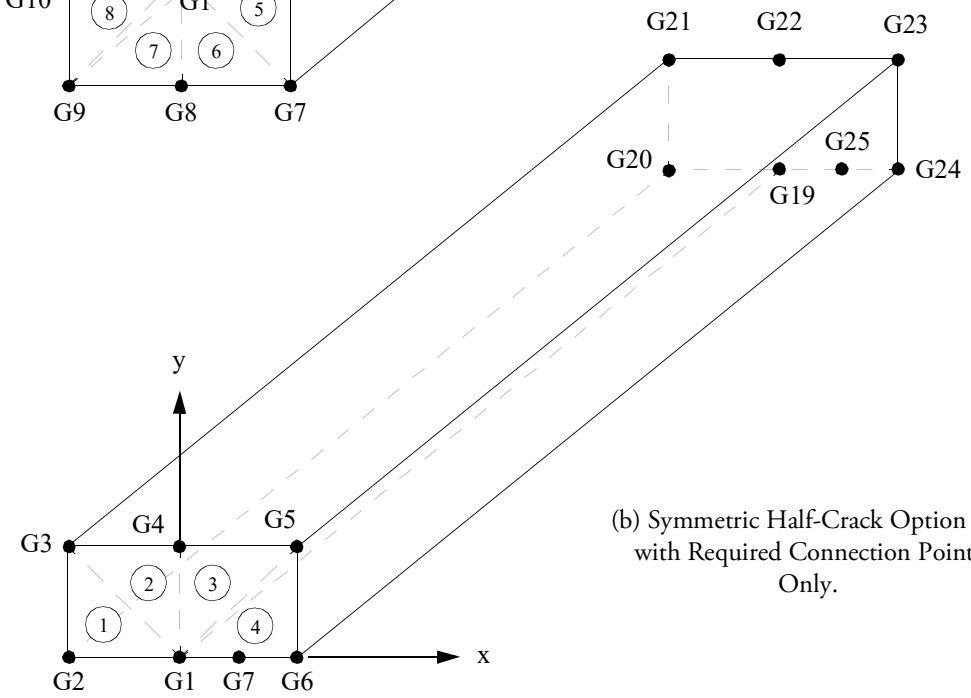
S1	S2	S3	S4	S5	S6	S7	S8	S9
x	y	σ_x	σ_y	τ_{xy}	K_I	K_{II}	0	0
where *x* and *y* are the element coordinates where stresses are reported. K_I and K_{II} are stress intensity factors.								

Figure 3-49 depicts the 3-D brick and symmetric half-crack options for the CRAC3D element. The brick element is divided, by the program, into basic wedge elements labeled 1 through 8, whereas the symmetric half-crack option is divided into wedges 1 through 4. Grid points 1 through 7 and 19 through 25 are required for symmetric crack option. Stresses and the local coordinates of these stresses for the brick element option are computed at the origin of the natural coordinates of wedges 4 and 5. The stresses and coordinates are then averaged and reported. Stress intensity factors, K_I and K_{II}, are computed for wedges 1 and 8, averaged, and reported. For the symmetric half-crack option, coordinates and stresses are

reported at the origin of the natural coordinates of wedge 4, whereas the stress intensity factor K_I only is reported for wedge 1.

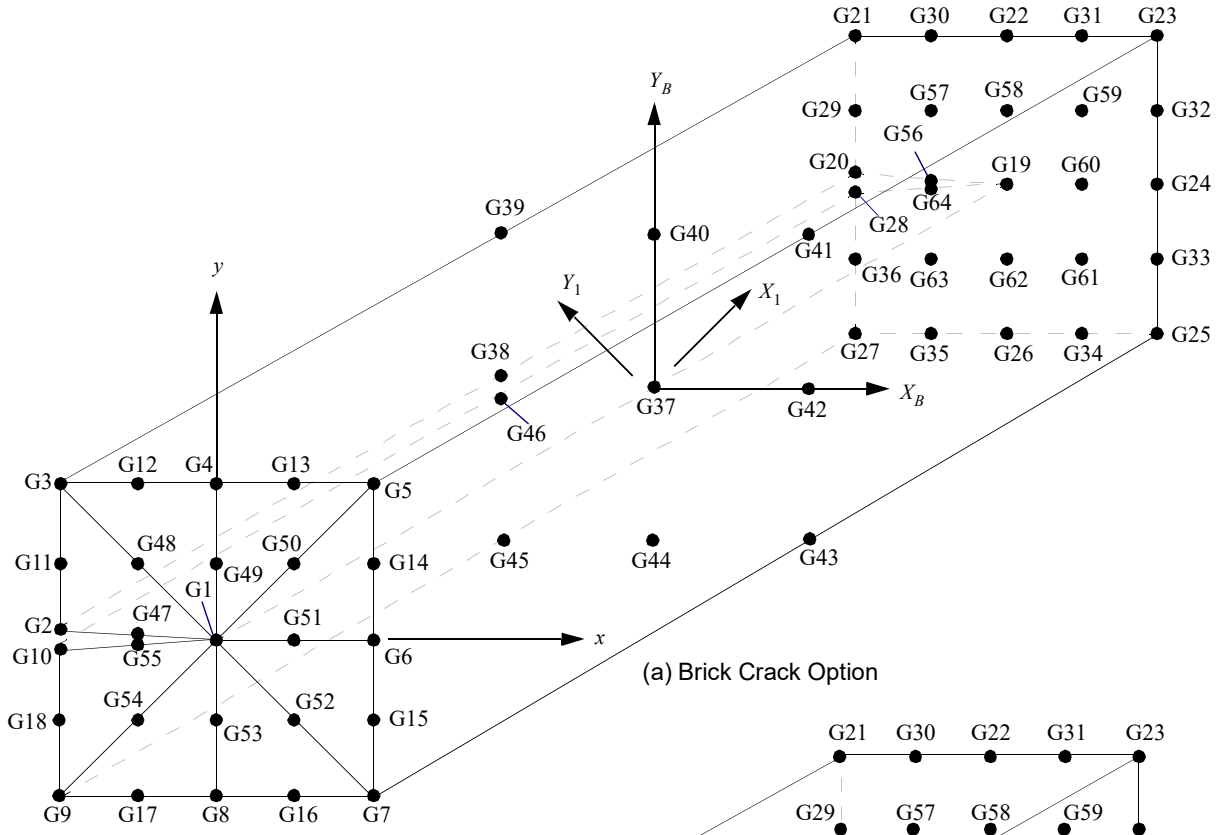


(a) Brick Crack Option with Required Connection Points Only.

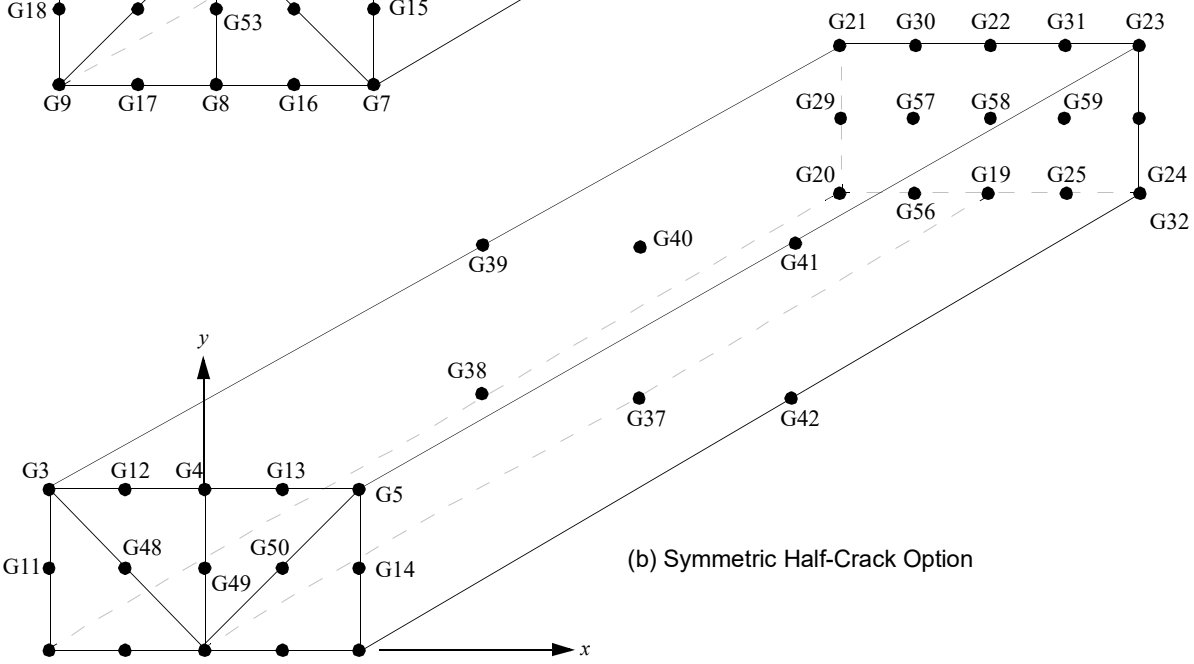


(b) Symmetric Half-Crack Option with Required Connection Points Only.

Figure 3-49 The CRAC3D Solid Crack Tip Element



(a) Brick Crack Option



(b) Symmetric Half-Crack Option

Figure 3-50 The CRAC3D Solid Crack Tip Element (continued)

Interpretation of CRAC3D Element Stress Output (Dummy Element Format)

S1	S2	S3	S4	S5	S6	S7	S8	S9
σ_x	σ_y	σ_z	τ_{xy}	τ_{yz}	τ_{zx}	K_I	K_{II}	K_{III}

Stresses are reported at the average of the origin of the natural coordinate of wedges 4 and 5, whereas for the symmetric crack option they are reported at the origin of the natural coordinate of wedge 4.

The CRAC2D element is based upon a 2-D formulation, but may be used in three-dimensional structures. However, the element should be planar. Any deviation from a planar element is checked, and if significant deviations arise, error messages will be issued.

The CRAC3D element is based upon a 3-D formulation. Both the faces (formed by grid points 1 through 18 and grid points 19 through 36) and the midplane (grid points 37 through 46) should be planar. Any significant deviation will be checked and error messages will be issued.

Gap and Line Contact Elements (CGAP, BCONP, BLSEG)

Gap Element

The gap and friction elements are specified on a CGAP entry. The element coordinate system and nomenclature are shown in Figure 3-51. CID is required, if it is used to define the element coordinate system. Otherwise, the *X*-axis of the element coordinate system, x_{elem}, is defined by a line connecting GA and GB of the gap element. The orientation of the gap element is determined by vector $\vec{v}$ similar to the definition of the beam element, which is in the direction from grid points GA to GO or defined by (X1, X2, X3).

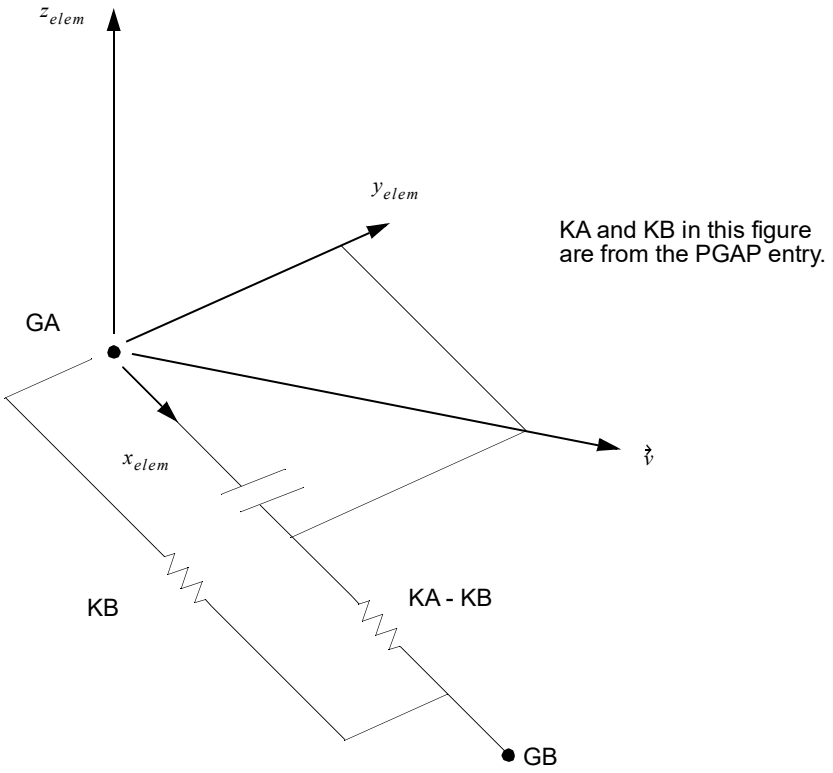


Figure 3-51 Gap Element Coordinate Systems

The properties for the gap elements are defined on the PGAP entry. The initial gap opening is defined by U_O. If the gap is closed ($U_A - U_B \geq U_O$), the axial stiffness (KA) has a very large value (relative to the adjacent structure). When the gap is open, there is a small stiffness KB in the axial direction.

MSC Nastran includes two types of gap elements: nonadaptive and adaptive. When the nonadaptive GAP element is used, the anisotropic coefficients of friction (μ_1 and μ_2) are specified for the frictional displacements. Also, the anisotropic coefficients of friction are replaced by the coefficients of static and dynamic friction μ_s and μ_k. On the PGAP continuation entry, the allowable penetration limit T_{max} should be specified because there is no default. In general, the recommended allowable penetration T_{max} is about 10% of the element thickness for plates or the equivalent thickness for other elements that are connected by GA and GB. When T_{max} is set to zero, the penalty values will not be adjusted adaptively.

Gap element forces (or stresses) and relative displacements are requested by the STRESS or FORCE Case Control command and computed in the element coordinate system. A positive axial force F_x indicates compression. For the element with friction, the magnitude of the slip displacement is always less than the shear displacement after the slip starts. For the element without friction, the shear displacements and slip displacements have the same value.

3-D Slideline Contact

Definition and Acronyms

The terminology associated with the slideline capability is as follows:

Two deformable bodies that come into contact are called the master and the slave. The modeling of interactions between the master and slave bodies requires you to define the contact regions in terms of slidelines. A master line is a list of grid points in the topological order on the master body and a slave line is a list of grid points in the topological order on the slave body. A slideline contact region consists of a master line and a slave line as shown in Figure 3-52. The grid points on the master line are called the master nodes and those on the slave line are called the slave nodes.

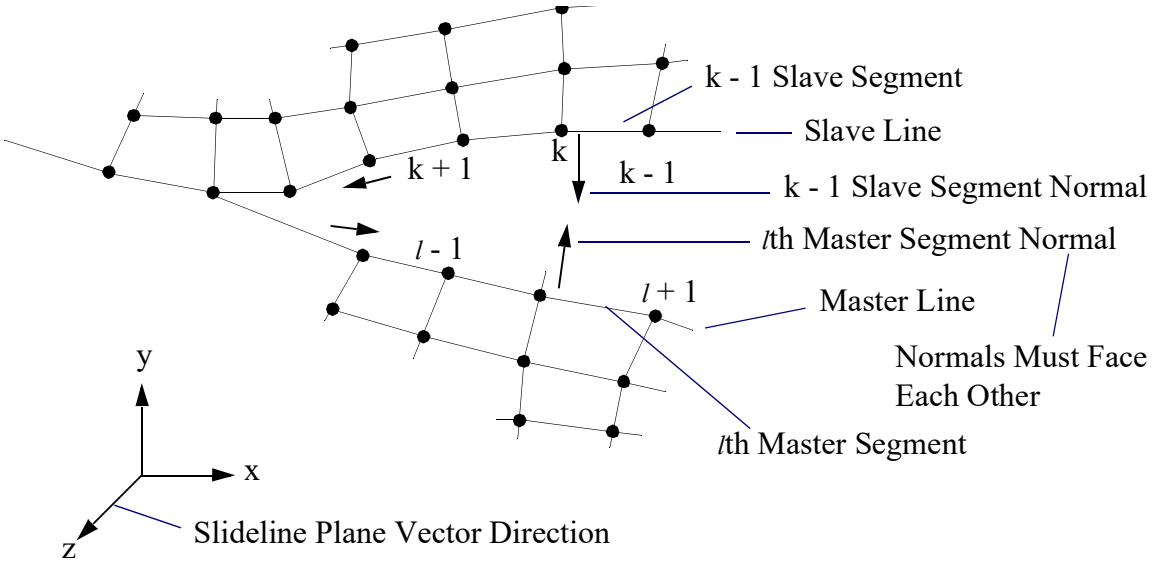


Figure 3-52 Typical Slideline Contact Region

- Arrows show positive direction for ordering nodes. Counterclockwise from master line to slave line.
- X-Y plane is the slideline plane. Unit normal in the Z-direction is the slideline plane vector.
- Slave and master segment normals must face each other; i.e., the normals must be in the opposite directions.
- A line segment joining two consecutive master nodes is called a master segment. The master and the slave nodes in a contact region must lie in a plane called the slideline plane (see Figure 3-52). The slideline plane is defined by a vector normal to the plane called the slideline plane vector. The master and the slave can have large relative motions within the slideline plane. However, relative motion outside the slideline plane is ignored, and therefore, motion must be small compared to a typical master segment. Thus, slideline contact is ideal for modeling interactions between two bodies that may come in contact in specified planes. MSC Nastran checks to make sure that in the initial geometry all the nodes of a contact region lie in the slideline plane and issues the appropriate messages if they are not. However, no such check is made during the analysis.

In general, contact is determined only for slave nodes and the master line. This may result in master nodes penetrating the slave line. However, the error involved depends only on the mesh discretization. At the expense of increase in computer runtime, you may wish to determine contact between master nodes and the slave line in addition to determining contact between slave nodes and the master line. This is called symmetric penetration. Thus for symmetric penetration, no distinction is made between slave and master.

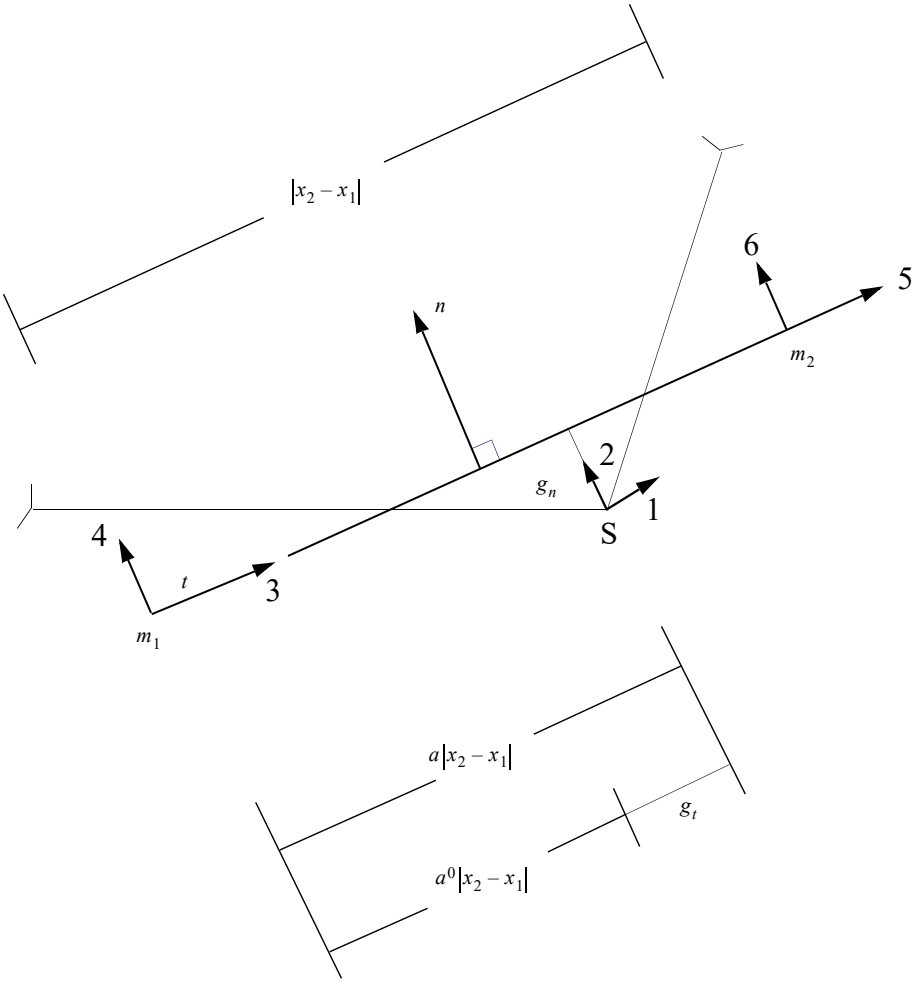
Initial penetration between two bodies is not allowed. The initial gaps between the slave nodes and the master line are calculated based on the coordinates specified for the slave and the master nodes. If the grid coordinates are not specified accurately enough (i.e., MSC Nastran will determine whether initial penetration exists), then MSC Nastran will adjust the coordinates automatically. In the case where the calculated initial penetration is less than ten percent of the length of the master segment to which it projects, MSC Nastran issues a warning message and automatically adjusts the coordinates. However, in the case where the initial penetration is larger than ten percent, the analysis terminates and a fatal message is issued. If the element normal of the master and slave segments are not pointing towards each other, MSC Nastran will stop and issue a fatal message similar to one issued in the case of initial penetration, even though the initial gaps may be acceptable.

Guidelines

MSC Nastran creates a local coordinate system for each master segment to determine contact between the slave nodes and the master line. The tangent direction for the master segment is given by the vector from its first node to the second node. It is very important that you specify the master nodes in the topological order. MSC Nastran then determines the normal direction to the master segment by the cross product between the slideline plane vector and the tangent vector. The normal direction must point from the master line to the slave line because MSC Nastran determines the contact between the slave nodes and the master line by measuring how close the slave nodes are to the master segment in that normal direction (see Figure 3-52).

For symmetric penetration, tangent and normal directions are also defined for the slave segment similar to the master segment. The normals for the master segment must point toward the slave and the normals for the slave must point toward the master (see Figure 3-52). This is generally accomplished by traversing from master line to slave line in a counter-clockwise fashion or clockwise fashion depending on whether the slideline plane vector forms a right-hand or a left-hand coordinate system with the slideline plane. It is your responsibility to make sure that the normals face each other. MSC Nastran checks indirectly to make sure that the normals face each other by calculating the initial gap between the two bodies. This is not a foolproof check, especially when there is no initial clearance.

Once MSC Nastran detects a contact between a node and a body, a three node slideline element as shown in Figure 3-53 is automatically created. The first node of this element is the slave node and the last two nodes are the master nodes of the master segment with which the slave node is in contact. The slave node is forced to slide on the master line after contact and must remain on the master line until a tensile force develops.



- S, m_1 , , m_2 = slave, master node 1 and master node 2, respectively
- a, a_0 = current and previous surface coordinate
- g_n = penetration of slave node into the master segment
- g_t = sliding of the slave node on the master segment
- n = normal direction for the master segment

Figure 3-53 Geometry and Definition of a Typical Three-Node Slideline Element

In MSC Nastran the contact and traction forces/stresses are always associated with slave nodes. In order to compute the contact and traction stresses, an area is associated with each slave node, based on the contributory length and the

width/thickness from the adjacent slave segments. Therefore, you need to specify widths associated with slave lines in a three-dimensional model or thickness associated with slave lines in a two dimensional model. You can also request output results for any number of slideline contact regions. The output for a slave node consists of (a) the slideline contact region identification number, (b) the master segment to which it projects, (c) the parametric surface coordinate to identify the exact projection of slave nodes relative to the master nodes of the master segment, and (d) a slip ratio to indicate whether the slave node is sticking, slipping or sliding.

Text Input

Table 3-14 shows how the slideline input data is given to MSC Nastran through executive and case control as well as the bulk data entries. For a complete list of these input entries, please refer to the *MSC Nastran Quick Reference Guide*.

Table 3-14 MSC Nastran Input for Nonlinear Slideline Contact

Slideline Input	MSC Nastran Input
Analysis Type and Diagnostics Request	Executive Control Parameters:
Nonlinear Statics	SOL 106
Nonlinear Dynamics	SOL 129
Slideline Diagnostics	DIAG 35 -- Provides extended slideline diagnostics output.
Slideline Output Request	Case Control Parameters:
	BOUTPUT -- Selects the contact regions for history output.
	Bulk Data Entries:
Contact Parameters	BCONP -- Defines the parameters for contact between two bodies.
Boundary Line Segments	BLSEG -- Defines the curve, which consists of a number of line segments via grid point numbers that may come in contact with other body.
Boundary Line Segments Width/Thickness	BWIDTH -- Defines width/thickness for line segments in 3-D/2-D slideline contact defined in the corresponding BLSEG Bulk Data entry.
Contact Friction	BFRIC -- Defines frictional properties between two bodies in contact.
Contact Output	BOUTPUT -- Defines the slave nodes at which the output is requested.
User Control and Results Processing; Parameters	ADPCON -- (Default = 1.0); Automatically calculated penalty values can be scaled by specifying a real value for the ADPCON. This is useful for changing the penalty values on restart. This will change the penalty value to be equal to a number calculated by MSC Nastran times SFAC value specified in BCONP Bulk Data entry times |ADPCON|. ADPCON is applied to all the contact regions.

The slideline contact formulation is based on the penalty method, which uses an artificial stiffness value. MSC Nastran will automatically calculate the appropriate numbers based on local stiffness. You have the option to scale the calculated penalty

values. The automatically calculated penalty values may sometimes cause convergence difficulties. However, you can overcome the convergence problems by restarting the analysis from any previous converged solution and scaling down the penalty value. Generally, we recommend decreasing the penalty values by one order of magnitude at a time.

Concentrated Mass Elements (CONM1, CONM2)

The concentrated mass elements are used to define a concentrated mass at a grid point. Two forms of input of concentrated mass are supported, the CONM1 and CONM2 elements.

The CONM1 allows a general 6×6 symmetric mass matrix in a specified coordinate system to be assigned to a geometric grid point. In addition, a less general form, the CONM2 element allows a concentrated mass about its center of gravity to be specified. CONM2 allows for the specification of the offset of the center of gravity of the concentrated mass relative to grid point location, a reference coordinate system, the mass and a 3×3 symmetric matrix of mass moments of inertia measured from its center of gravity. For additional information see the Bulk Data Entries (p. 1119) in the *MSC Nastran Quick Reference Guide*.

General Element Capability (GENEL)

The input required for the general element is the GENEL Bulk Data entry. The description of GENEL (p. 2070) in the contains a complete description of the input options and an example.

The principal application of the GENEL element is to represent the stiffness of a substructure which has an arbitrary number of connected grid point components and/or scalar points. The input data may be obtained either from another computer run or from test data.

The general element is a structural stiffness element connected to any number of degrees of freedom, as specified by the user. In defining the form of the externally generated data on the stiffness of the element, two major options are provided to the user.

1. Instead of supplying the stiffness matrix for the element directly, the user provides the deflection influence coefficients for the structure supported in a nonredundant manner. The associated matrix of the restrained rigid body motions may be input or may be generated internally by the program.
2. The stiffness matrix of the element may be input directly. This stiffness matrix may be for the unsupported body, containing all the rigid body modes, or it may be for a subset of the body's degrees of freedom from which some or all of the rigid body motions are deleted. In the latter case, the option is given for automatic inflation of the stiffness matrix to reintroduce the restrained rigid body terms, provided that the original support conditions did not constitute a redundant set of reactions. An important advantage of this option is that, if the original support conditions restrain all rigid body motions, the reduced stiffness matrix need not be specified by the user to high precision in order to preserve the rigid body properties of the element.

The defining equation for the general element when written in the flexibility form is

$$-\begin{Bmatrix} u_i \\ f_d \end{Bmatrix} = \left[\begin{array}{c|c} Z & S \\ \hline -S^T & 0 \end{array}\right] \begin{Bmatrix} f_i \\ u_d \end{Bmatrix} - \tag{3-65}$$

where:

$[Z]$ = is the matrix of deflection influence coefficients for coordinates $\{u_i\}$ when coordinates $\{u_d\}$ are rigidly restrained.

$[S]$ = is a rigid body matrix whose terms are the displacements $\{u_i\}$ due to unit motions of the coordinates $\{u_d\}$, when all $f_i = 0$.

$[f_i]$ = are the forces applied to the element at the $\{u_i\}$ full coordinates.

$[f_d]$ = are the forces applied to the element at the $\{u_d\}$ coordinates. They are assumed to be statically related to the $\{f_i\}$ forces, i.e., they constitute a nonredundant set of reactions for the element.

The defining equation for the general element when written in the stiffness form is

$$-\begin{Bmatrix} f_i \\ f_d \end{Bmatrix} = \begin{bmatrix} k & -k\,S \\ -S^T k & S^T k\,S \end{bmatrix} \begin{Bmatrix} u_i \\ u_d \end{Bmatrix} - \tag{3-66}$$

where all symbols have the same meaning as in (3-65) and $[k] = [Z]^{-1}$, when $[k]$ is nonsingular. Note, however, that it is permissible for $[k]$ to be singular. (3-66) derivable from (3-65) when $[k]$ is nonsingular.

Input data for the element consists of lists of the u_i and u_d coordinates, which may occur at either geometric or scalar grid points; the values of the elements of the $[Z]$ matrix, or the values of the elements of the $[k]$ matrix; and (optionally) the values of the elements of the $[S]$ matrix.

The user may request that the program internally generate the $[S]$ matrix. If so, the $\{u_i\}$ and coordinates occur only at geometric grid points, and there must be six or less $\{u_d\}$ coordinates that provide a nonredundant set of reactions for the element as a three-dimensional body.

The $[S]$ matrix is internally generated as follows. Let $\{u_b\}$ be a set of six independent motions (three translations and three rotations) along coordinate axes at the origin of the basic coordinate system. Let the relationship between $\{u_d\}$ and $\{u_b\}$.

$$\{u_d\} = [D_d]\{u_b\} \tag{3-67}$$

The elements of $[D_d]$ are easily calculated from the basic (x,y,z) geometric coordinates of the grid points at which the elements of $\{u_d\}$ occur, and the transformations between basic and global (local) coordinate systems. Let the relationship between $\{u_i\}$ and $\{u_b\}$ be

$$\{u_i\} = [D_i]\{u_b\} \tag{3-68}$$

where $[D_i]$ is calculated in the same manner as $[D_d]$. Then, if $[D_d]$ is nonsingular,

$$[S] = [D_i][D_d]^{-1} \tag{3-69}$$

Note that, if the set $\{u_d\}$ is not a sufficient set of reactions, $[D_d]$ is singular and $[S]$ cannot be computed in the manner shown. When $\{u_d\}$ contains fewer than six elements, the matrix $[D_d]$ is not directly invertible but a submatrix $[a]$ of rank r, where r is the number of elements of $\{u_d\}$, can be extracted and inverted.

A method which is available only for the stiffness formulation and not for the flexibility formulation will be described. The flexibility formulation requires that $\{u_d\}$ have six components. The method is as follows. Let $\{u_d\}$ be augmented by 6-r displacement components $\{u_d'\}$ which are restrained to zero value. We may then write

$$\begin{Bmatrix} u_d \\ d_d' \end{Bmatrix} = \begin{bmatrix} D_d \\ D_d' \end{bmatrix} \{u_b\} = [\bar{D}]\{u_b\} \tag{3-70}$$

The matrix $[D_d]$ is examined and a nonsingular subset $[a]$ with r rows and columns is found. $\{u_b\}$ is then reordered to identify its first r elements with $\{u_d\}$. The remaining elements of $\{u_b\}$ are equated to the elements of $\{u_d\}$. The complete matrix $[\bar{D}]$ then has the form

$$[\bar{D}] = \left[\begin{array}{c|c} a & b \\ \hline 0 & I \end{array}\right] \tag{3-71}$$

with an inverse

$$[\bar{D}]^{-1} = \left[\begin{array}{c|c} a^{-1} & -a^{-1}b \\ \hline 0 & I \end{array}\right] \tag{3-72}$$

Since the members of $\{u_d'\}$ are restrained to zero value,

$$\{u_b\} = [D_r]\{u_d\} \tag{3-73}$$

where $[D_r]$ is the $(6 \times r)$ partitioned matrix given by

$$[D_r] = \left[\begin{array}{c} a^{-1} \\ \hline 0 \end{array}\right] \tag{3-74}$$

The $[D_i]$ matrix is formed as before and the $[S]$ matrix is then

$$[S] = [D_i][D_r] \tag{3-75}$$

Although this procedure will replace all deleted rigid body motions, it is not necessary to do this if a stiffness matrix rather than a flexibility matrix is input. It is, however, a highly recommended procedure because it will eliminate errors due to nonsatisfaction of rigid body properties by imprecise input data.

The stiffness matrix of the element written in partitioned form is

$$[K_{ee}] = \begin{bmatrix} K_{ii} & K_{id} \\ - & - \\ K_{id}^T & K_{dd} \end{bmatrix} - \tag{3-76}$$

When the flexibility formulation is used, the program evaluates the partitions of $[K_{ee}]$ from $[Z]$ and $[S]$ as follows:

$$[K_{ii}] = [Z]^{-1} \tag{3-77}$$

$$[K_{id}] = [Z]^{-1}[S] \tag{3-78}$$

$$[K_{dd}] = [S]^T[Z]^{-1}[S] \tag{3-79}$$

If a stiffness matrix, $[k]$, rather than a flexibility matrix is input, the partitions of $[K_{ee}]$ are

$$[K_{ii}] = [k] \tag{3-80}$$

$$[K_{id}] = -[k][S] \tag{3-81}$$

$$[K_{dd}] = -[S]^T[k][S] \tag{3-82}$$

No internal forces or other output data are produced for the general element.

Direct Matrix Input on Grid Points (DMIG)

The Bulk Data entry DMIG can be used to input a stiffness (or mass) matrix which connects specified degrees of freedom. The matrix so defined will be added to the stiffness (or mass) matrix computed from finite element properties.

The DMIG entry includes provisions for unsymmetric terms and complex values, both of which are useful in dynamic analysis. These provisions should not be used in static or normal modes. Note that an entry in the Case Control Section is required (K2GG = NAME for a stiffness matrix or M2GG = NAME for a mass matrix). See Case Control Commands (p. 191) in the *MSC Nastran Quick Reference Guide* for general instructions regarding the Case Control Section.

The primary application of the DMIG Bulk Data entry is to enter stiffness and mass data for parts of the structure which are obtained from another computer run. The format is cumbersome (two matrix terms per continuation entry) and the matrix should be input to high precision (see Use of Parameters, 32 for a discussion of double-field entries). For stiffness matrices only, the GENEL Bulk Data entry is an alternative for manually inputting data.

p-Elements

p-elements are elements that have variable degrees of freedom. The user can specify the polynomial order for each element (Px, Py, and Pz) and the program will generate the degrees of freedom required. p-elements are not available for nonlinear analysis using SOLs 106, 129, 400, 600, and 700. They are also not available for cyclic symmetry using SOLs 114, 115, 116, and 117.

There are a total of six different forms of p-elements that are defined by connection entries:

CTETRA	Four-sided solid element specified by 4 corner grid points
CPENTA	Five-sided solid element specified by 6 corner grid points
CHEXA	Six-sided solid element specified by 8 corner grid points
CTRIA	Three-sided curved shell element specified by 3 corner grid points
CQUAD	Four-sided curved shell element specified by 4 corner grid points
CBEAM	A curved beam element specified by its 2 end grid points

The properties of the p-elements will be specified using the PSOLID, PSHELL, PBEAM, or PBEAML entry. These elements may use either isotropic materials as defined on the MAT1 entry or anisotropic materials as defined on the MAT9 entry. The material coordinate system can use the basic system (0), any defined system (integer > 0), or the element coordinate system (-1 or blank).

Geometry

The model geometry of the p-elements can be done by

1. Supplying additional POINT entries to define the geometry of the edges, i.e., in the `FEEDGE` entry, the user needs to specify GEOMIN = POINT. In this case, when the `FEEDGE` entry contains the ID of one point, then the `FEEDGE` is considered to have a quadratic geometry. When the `FEEDGE` entry contains the ID of two points, then the `FEEDGE` is considered to have a cubic geometry. If there are no additional points specified on the `FEEDGE` entry, then the `FEEDGE` is considered to have a linear geometry (see Figure 3-54).

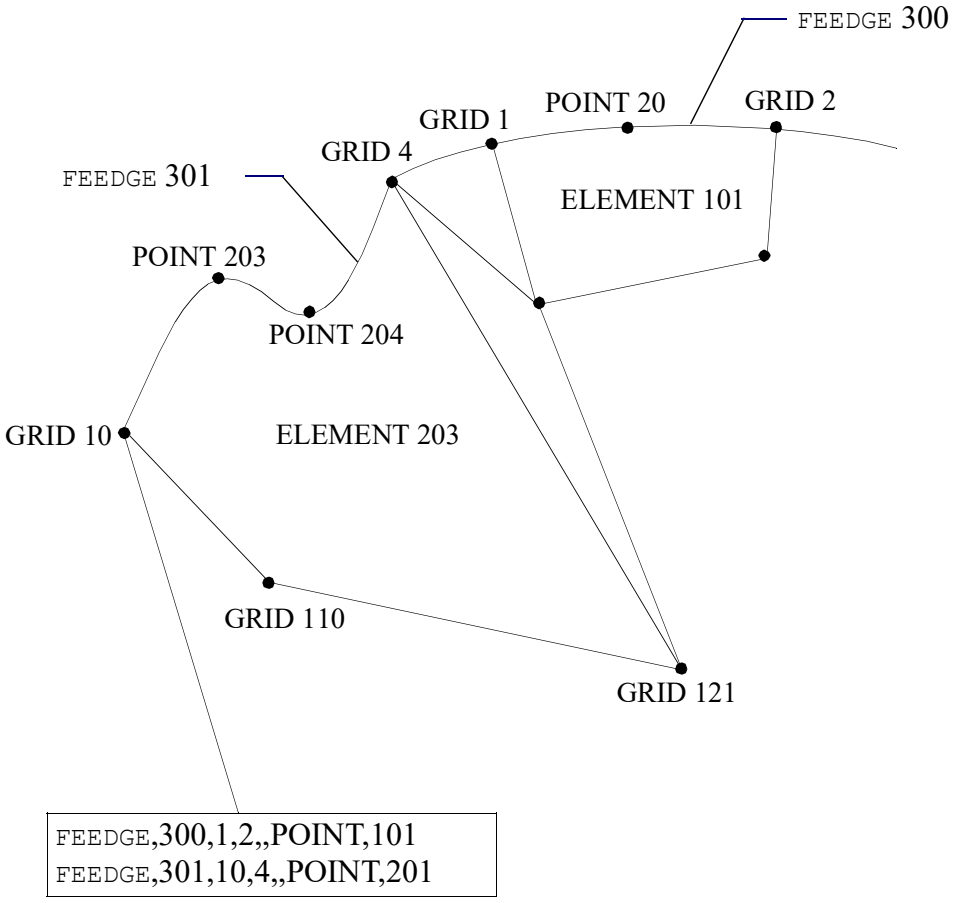


Figure 3-54 Specifying Geometry Using GEOMIN = POINT Method

2. Supplying the actual geometry, using the GMCURV and GMSURF entries, to define the geometry of the edges. The user needs to specify GEOMIN = GMCURV in the FEEDGE entry and providing a SURFID on the FEFACE entry (see Figure 3-55).

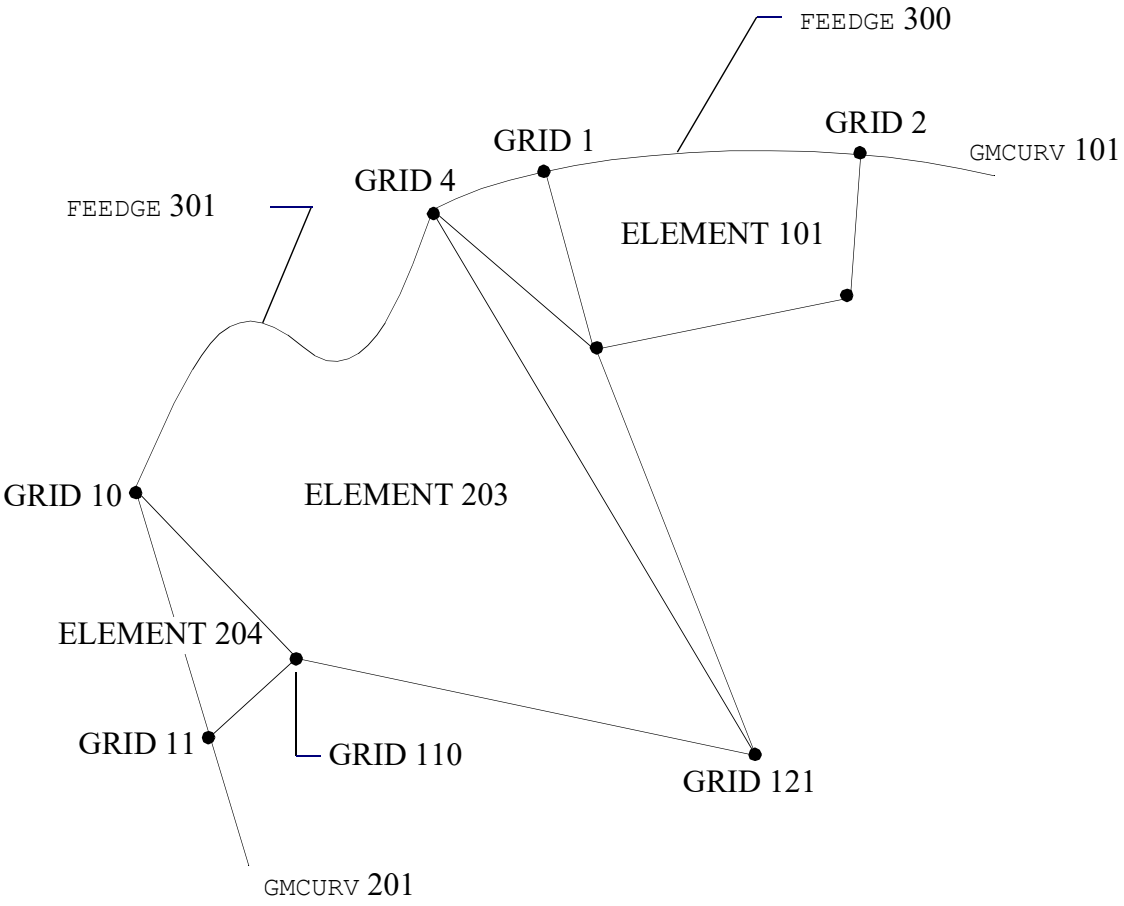


Figure 3-55 Specifying Geometry Using GEOMIN = GMCURV Method

In general, the geometry specification for any edge using a FEEDGE entry overrides the geometry specification for that edge via the FEFACE entry. In other words, the geometry of the edges belonging to both a FEEDGE and a FEFACE will be calculated using the data supplied on the FEEDGE entry. In addition, whenever two or more GMSURF are intersecting and FEFACE entries referring to the individual GMSURF entries are supplied, an additional FEEDGE entry must be supplied for the edges that are common to the multiple surfaces.

Bulk entries defining p-elements with p-valve greater than 1 are:

ADAPT	GMSPC
DEQATN	GMSURF
FEEDGE	OUTPUT
FEFACE	OUTRCV
GMBC	PSET
GMCORD	PVAL
GMCURV	TABLE3D
GMLOAD	TEMPF

Hyperelastic Elements

The hyperelastic elements are intended for fully nonlinear (finite deformation) analysis using SOL 106 or SOL 129 including the effect of large strain and large rotation (SOL 400 is recommended). Geometric nonlinearity is a subset of this type of analysis. In addition, the elements are especially designed to handle nonlinear elastic materials at the nearly incompressible limit. Volumetric locking avoidance is provided through a mixed formulation, based on a three field variational principle, with isoparametric displacement and discontinuous pressure and volumetric strain interpolations. Shear locking avoidance is provided through the use of second order elements. See Small Strain Elements, 575.

The hyperelastic elements are defined on the same connection entries as the other shell and solid elements and are distinguished by their property entries. A PLPLANE or PLSOLID entry defines a hyperelastic element. The hyperelastic material, which is characterized by a generalized Rivlin polynomial form of order 5, applicable to compressible elastomers, is defined on the MATHP entry. See Hyperelastic Material, 550 for a more detailed discussion.

Hyperelastic Solid Elements

The following elements are available:

1. CTETRA -- Four-sided solid element with 4 to 10 nodes.
2. CPENTA -- Five-sided solid element with 6 to 15 nodes.
3. CHEXA -- Six-sided solid element with 8 to 20 nodes.

See Figure 3-43, Figure 3-44, and Figure 3-45 on solid elements.

There is no element coordinate system associated with the hyperelastic solid elements. All output is in the basic coordinate system. The following quantities are output at the Gauss points:

- Cauchy stresses

  σ_x, σ_y, σ_z, τ_{xy}, τ_{yz}, τ_{zx}
- Pressure

$$p = \frac{1}{3}(\sigma_x + \sigma_y + \sigma_z)$$

Logarithmic strains

$$\varepsilon = \sum_{l=1}^{3} l n \gamma_l N_l N_l^T$$

- Volumetric strain

$$J - 1 = \frac{dV - dV_0}{dV_0}$$

See Small Strain Elements, 575 for definitions of these quantities.

Hyperelastic Plane Elements

These are plane strain elements defined on the following connectivities:

- CQUAD -- Quadrilateral element with 4 to 9 nodes. When the center node is missing, this element may also be specified on a CQUAD8 connectivity entry. When all edge nodes are missing, the CQUAD4 connectivity may be used.
- CTRIA3 -- Triangular element with 3 nodes.
- CTRIA6 -- Triangular element with 3 to 6 nodes.

Figure 3-56 shows the element connectivity for the CQUAD element.

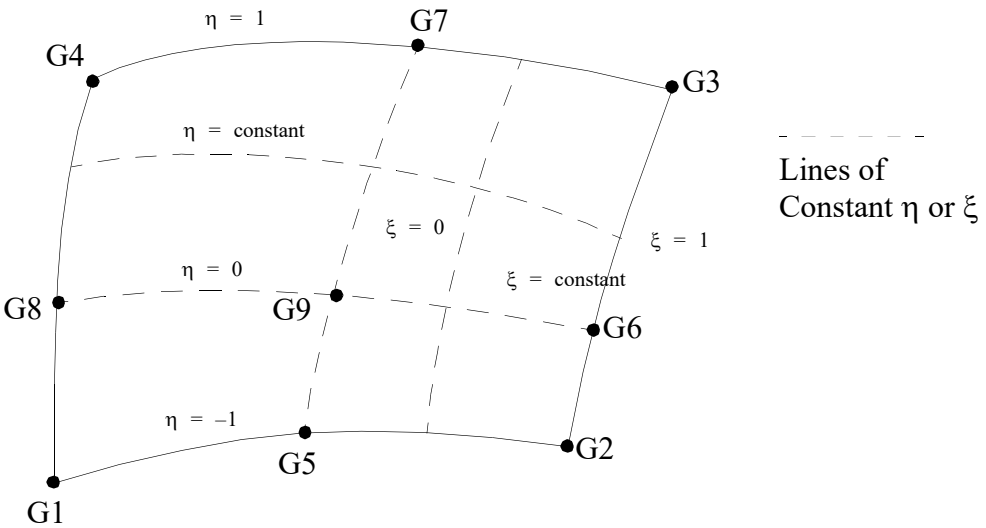


Figure 3-56 CQUAD Element

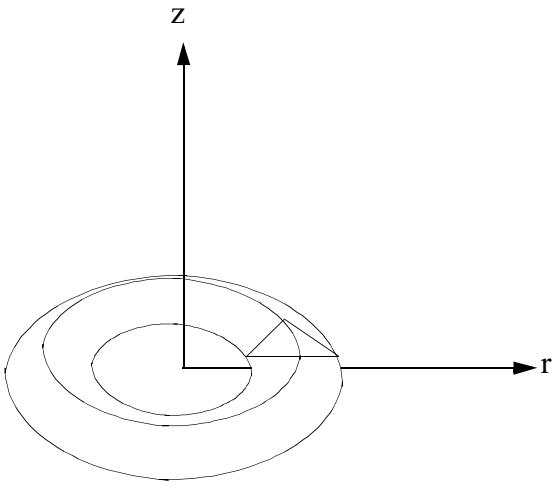
See Figure 3-27 through Figure 3-30 for the other elements. Note, however, that there is no element coordinate system associated with the hyperelastic plane elements. All output is in the CID coordinate system. Cauchy stresses σ_x, σ_y, σ_z, τ_{xy}, pressure $p = 1/3(\sigma_x + \sigma_y + \sigma_z)$, logarithmic strains and volumetric strain are output at the Gauss points.

The plane of deformation is the XY plane of the CID coordinate system, defined on the PLPLANE property entry. The model and all loading must lie on this plane, which, by default, is the XY plane of the basic coordinate system. The displacement along the Z axis of the CID coordinate system is zero or constant.

Hyperelastic Axisymmetric Elements

The axisymmetric hyperelastic elements are defined on the following connectivity entries:

- CQUADX -- Quadrilateral axisymmetric element with 4 to 9 nodes.
- CTRIAX -- Triangular axisymmetric element with 3 to 6 nodes.



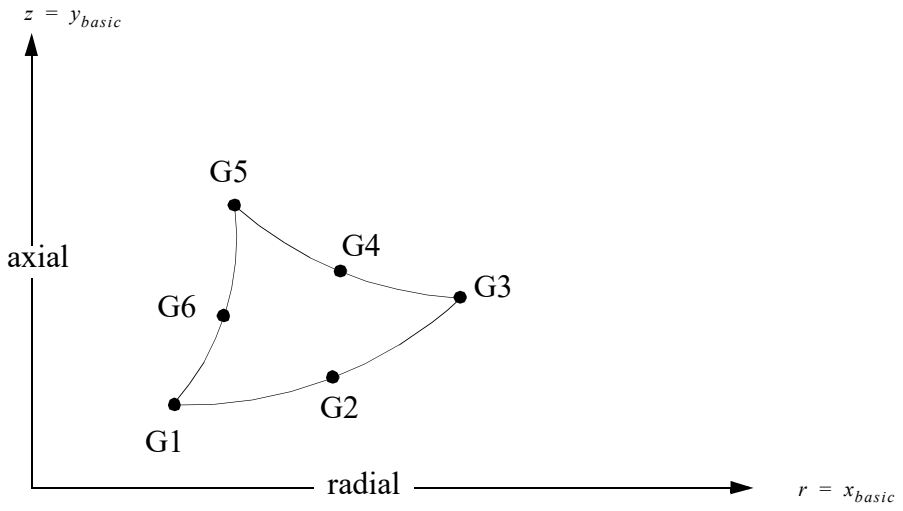


Figure 3-57 The CTRIAX Hyperelastic Coordinate System Element

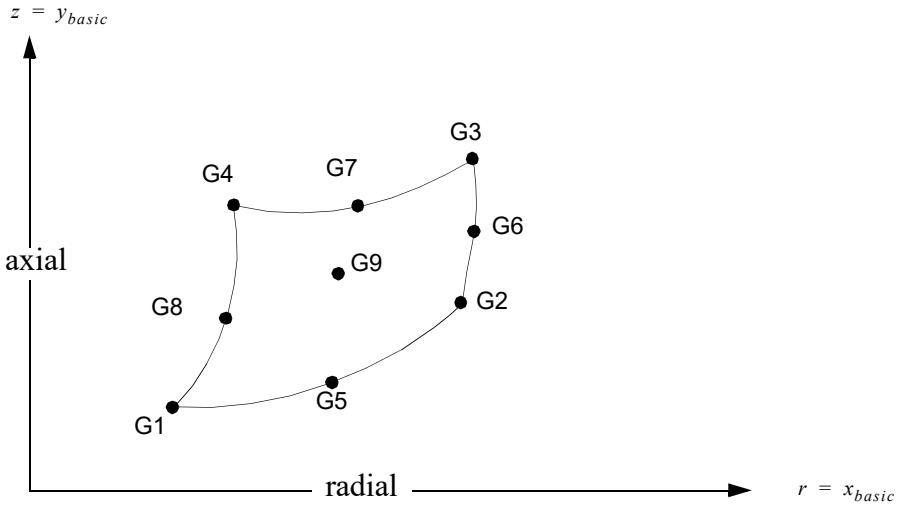


Figure 3-58 The CQUADX Hyperelastic Coordinate System Element

The plane of deformation is the XY plane of the basic coordinate system, and Y is the axisymmetry axis. The model and all loading must lie on this plane.

All output is in the basic coordinate system. Cauchy stresses σ_x (radial), pressure σ_y (axial), σ_z (circumferential), τ_{xy}, pressure $p = 1/3(\sigma_x + \sigma_y + \sigma_z)$, logarithmic strains and volumetric strain are output at the Gauss points.

Pressure loads, specified on the PLOAD4 and PLOADX1 Bulk Data entries respectively, with follower force characteristics, are available for the solid and axisymmetric elements. A pressure load may not be specified on the plane strain elements.

Temperature loads may be specified for all hyperelastic elements on the TEMP and TEMD entries. The hyperelastic material, however, may not be temperature dependent. Temperature affects the stress-strain relation as indicated in Hyperelastic Material, 550.

GPSTRESS and FORCE (or ELFORCE) output is not available for hyperelastic elements.

Bushing Elements

The bushing (generalized spring and damper) elements consist of the following:

- CBUSH
- CBUSH1D

Bush Element (CBUSH)

The generalized spring-damper element CBUSH is a structural scalar element connecting two noncoincident grid points, or two coincident grid points, or one grid point with an associated PBUSH entry. This combination is valid for any structural solution sequence. To make frequency dependent the PBUSH need only have an associated PBUSHT Bulk Data entry. The PBUSHT entry, for frequency dependency, is only used in SOL 108 and SOL 111. The PBUSHT entry can be used to define load-displacement dependency in SOL 106 and SOL 400.

In modal frequency response, the basis vectors (system modes) [ϕ] will be computed only once in the analysis and will be based on nominal values of the scalar frequency dependent springs. In general, any change in their stiffness due to frequency will have little impact on the overall contribution to the structural modes.

The stiffness matrix for the CBUSH element takes the diagonal form in the element system:

$$K_o = \begin{bmatrix} k_u & & & & & & \\ & k_v & & & & & \\ & & k_w & & & & \\ & & & k_{\theta_x} & & & \\ & & & & k_{\theta_y} & & \\ & & & & & k_{\theta_z} & \end{bmatrix}$$

For the B matrix replace the k terms with b.

When transformed into the basic system, there is coupling between translations and rotations, thus ensuring rigid body requirements.

The element axes are defined by one of the following procedures:

1. If a CID is specified then the element x-axis is along T1, the element y-axis is along T2, and the element z-axis is along T3 of the CID coordinate system. The options GO or (X1,X2,X3) have no meaning and will be ignored. Then $[T_{ab}]$ is computed directly from CID.
2. For noncoincident grids (GA ≠ GB), if neither options GO or (X1,X2,X3) is specified and no CID is specified, then the line AB is the element x-axis. No y-axis or z-axis need be specified. This option is only valid when K1 (or B1) or K4 (or B4) or both on the PBUSH entry are specified but K2, K3, K5, K6 (or B2, B3, B5, B6) are not specified. If K2, K3, K5, K6 (or B2, B3, B5, B6) are specified, a fatal message will be issued. Then $[T_{ab}]$ is computed from the given vectors like the beam element.

Direct Frequency Response

Nominal Values

The following matrices are formed only once in the analysis and are based on the parameter input to EMG of /' '/ implying that the nominal values only are to be used for frequency dependent springs and dampers.

$$K_{dd} = (1 + ig)K^1_{dd} + K^{2x}_{dd} + iK^4_{dd}$$

$$M_{dd} = M^1_{dd} + M^2_{dd}$$

$$B_{dd} = B^1_{dd} + B^{2x}_{dd}$$

ESTF

The following matrices are formed at each frequency in the analysis and are based on the parameter input to EMG of /'ESTF'/

$$\Delta K^f_{dd} = (1 + ig)\Delta K^{1f}_{dd}$$

$$\Delta K^{4f}_{dd} = g^f_e[\Delta k^f_j] \quad j = 1 \rightarrow 6$$

$$\Delta B^f_{dd} = \Delta B^{1f}_{dd}$$

ESTNF

The following matrices are formed at each frequency in the analysis and are based on the parameter input to EMG of /'ESTNF'/

$$\Delta K^{4f\,nominal}_{dd} = [\Delta g^f_e \; k^{f\,nominal}_j] \quad j = 1 \rightarrow 6$$

and j runs through the stiffness values specified for the CBUSH element or j = 1 for the CELAS1 and CELAS3 elements.

Then at each frequency form:

$$K_{dd} \leftarrow K_{dd} + \Delta K^f_{dd} + i\Delta K^{4f}_{dd} + i\Delta K^{4f\,nominal}_{dd}$$

$$B_{dd} \leftarrow B_{dd} + \Delta B^{f}_{dd}$$

Then the equation to be solved is:

$$[-\omega^2\ M_{dd} + i\omega\ B_{dd} + K_{dd}][u_d]\ =\ [P_d]$$

Modal Frequency Response

Basis Vector and Nominal Values

The basis vector matrix $[\phi]$ (system modes) is formed only once in the analysis using nominal values for frequency dependent elements.

The following matrices are formed only once in the analysis and are based on the parameter input to EMG of /' '/ implying that the nominal values only are to be used for frequency dependent springs and dampers.

$$K^2_{dd}\ =\ K^{2x}_{dd} + igK^1_{dd} + iK^4_{dd}$$

$$B^2_{dd}\ =\ B^1_{dd} + B^{2x}_{dd}$$

ESTF

The following matrices are formed at each frequency in the analysis and are based on the parameter input to EMG of /'ESTF'/

$$\Delta K^{2f}_{dd}\ =\ ig\Delta K^{1f}_{dd}$$

$$\Delta K^{4f}_{dd}\ =\ g^f_e[\Delta k^f_j] \quad j\ =\ 1 \rightarrow 6$$

$$\Delta B^{2f}_{dd}\ =\ \Delta B^{1f}_{dd}$$

ESTNF

The following matrices are formed at each frequency in the analysis and are based on the parameter input to EMG of /'ESTNF'/

$$\Delta K^{4f\,nominal}_{dd}\ =\ [\Delta g^f_e\ k^{f\,nominal}_j] \quad j\ =\ 1 \rightarrow 6$$

and j runs through the stiffness values specified for the CBUSH element or j = 1 for the CELAS1 and CELAS3 elements.

Then at each frequency form:

$$K^2_{dd} \leftarrow K^2_{dd} + \Delta K^f_{dd} + i\Delta K^{4f}_{dd} + i\Delta K^{4f\,nominal}_{dd}$$

$$B^2_{dd} \leftarrow B^2_{dd} + \Delta B^f_{dd}$$

The GKAM module will then produce:

$$K_{hh}\ =\ [k] + [\Phi]^T[K^2_{dd}][\Phi]$$

$$B_{hh} = [b] + [\Phi]^T [B^2_{dd}][\Phi]$$

$$M_{hh} = [m] + [\Phi]^T [M^2_{dd}][\Phi]$$

Then the equation to be solved is:

$$[-\omega^2 M_{hh} + i\omega B_{hh} + K_{hh}][u_h] = [P_h]$$

Element force calculation:

Frequency:

$$F_e = [(1 + ig_t)K_e + i\omega B_e]U_e$$

where

$$g_t = g + g_e$$

Static:

$$F_e = K_e U_e$$

Transient:

$$F_e = K_e U_e(t) + \left[B_e + \left(\frac{g}{w3} + \frac{g_e}{w4}\right)K_e\right]\dot{U}_e(t)$$

CBUSH Stiffness and Structural Damping Matrices

If both stiffness K and structural damping ge is specified each with their own frequency dependent tables, matrix terms of the following form are created:

$$K^f_j + i \cdot ge^f_j \cdot K^f_j$$

If stiffness K with frequency dependent tables and non frequency dependent structural damping ge are specified, matrix terms of the following form are created:

$$K^f_j + i \cdot ge^0_j \cdot K^f_j$$

If non frequency dependent stiffness K with frequency dependent structural damping ge are specified, matrix terms of the following form are created:

$$K^0_j + i \cdot ge^f_j \cdot K^0_j$$

In the above three expressions, the superscript f denotes frequency dependent and the superscript 0 denotes a nominal value. The subscript j implies the j-th degree of freedom of the CBUSH element. The real term goes into the element stiffness matrix and the imaginary term goes into the element K^4 matrix.

Bush1D Element (CBUSH1D)

The BUSH1D is a one dimensional version of the BUSH element (without the rigid offsets) and supports large displacements.

The element is defined with the CBUSH1D and a PBUSH1D entry. The user may define several spring or damping values on the PBUSH1D property entry. It is assumed that springs and dampers work in parallel. The element force is the sum of all springs and dampers.

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

Benefits

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

Output

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

Guidelines

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

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

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

Limitations

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

Interface Elements

The interface elements are primarily used when performing global local analyses. The current implementation is for p-elements.

For curve interfaces the following entries are used:

- GMBNDC -- Geometric Boundary - Curve
- GMINTC -- Geometric Interface - Curve
- PINTC -- Properties of Geometric Interface - Curve

For surface interfaces the following entries are used:

- GMBNDS -- Geometric Boundary - Surface
- GMINTS -- Geometric Interface - Surface
- PINTS -- Properties of Geometric Interface - Surface

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

Curve Interface Elements

The addition of interface elements allows dissimilar meshes to be connected over a common geometric boundary, instead of using transition meshes or constraint conditions. Primary applications where the analyst specifies the interface elements manually include: facilitating global-local analysis, where a patch of elements may be removed from the global model and replaced by a denser patch for a local detail, without having to transition to the surrounding area; and connecting meshes built by different engineering organizations, such as a wing to the fuselage of an airplane.

Primary applications where the interface elements could be generated automatically are related to automeshers, which may be required to transition between large and small elements between mesh regions; and h-refinement, where subdivided elements may be adjacent to undivided elements without a transition area.

Dissimilar meshes created by the analyst can occur with global-local analysis, where part of the structure is modeled as the area of primary interest in which detailed stress distributions are required, and part of the structure is modeled as the area of secondary interest through which load paths are passed into the area of primary interest.

Generally the area of primary interest has a finer mesh than the area of secondary interest and, therefore, a transition area is required. Severe transitions generally produce elements that are heavily distorted, which can result in poor stresses and poor load transfer into the area of primary interest. An example of using interface elements to avoid such transitions is shown in Figure 3-59. Similarly, a patch of elements may be removed from the global model and replaced by a denser patch for local detail.

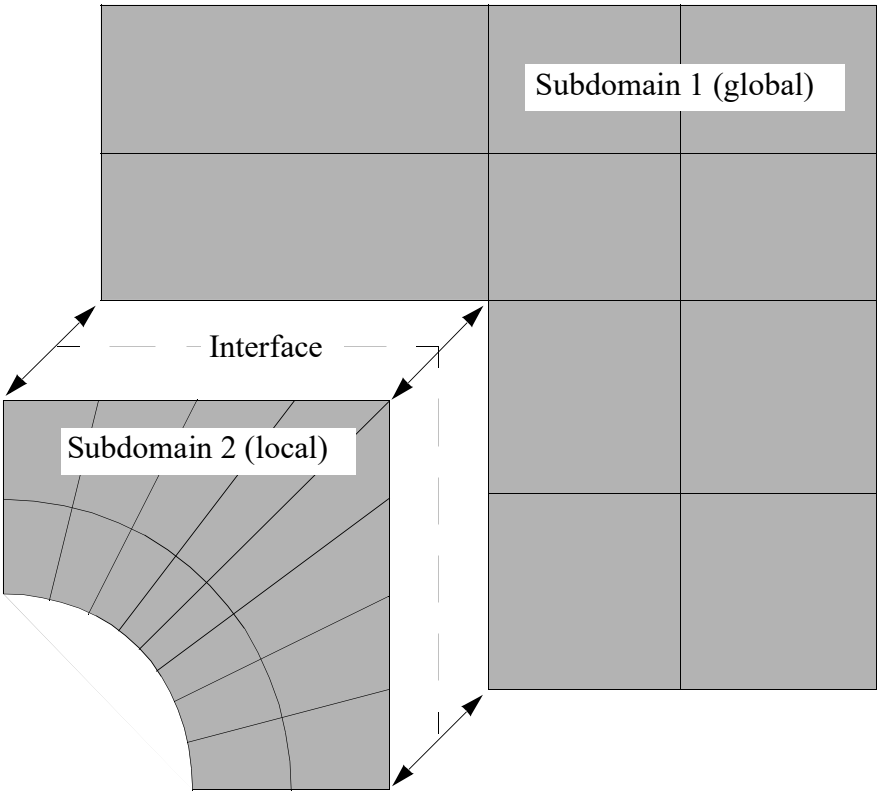


Figure 3-59 Example of Interface Elements (Exploded View)

In large system assemblies, different analysts or even different organizations may have created different components of the model, such as the wing and the fuselage of an airplane. Unless they have carefully coordinated their efforts, the finite element meshes of the different components may not match at the interfaces.

Dissimilar meshes generated by the analysis program can also arise with automeshers, which may be required to transition between large elements and small elements in a small area. Many automeshers generate tetrahedral meshes for solids, and

distorted tetrahedra may be more susceptible to poor results. Interface elements are particularly useful when a transition is needed between large and small elements within a small area.

When mesh refinement is performed, subdivided elements may be adjacent to undivided elements with no room for a transition area. Without some kind of interface element, the subdivision would need to be carried out to the model boundary or otherwise transitioned out.

It is important to note that the interface elements provide a tool for connecting dissimilar meshes, but they do not increase the accuracy of the mesh. As with any interface formulation, the hybrid variational technology, which imposes continuity conditions in a weak form, can not increase the accuracy of the adjacent subdomains. For instance, if a single element edge on one boundary is connected to many element edges on the other boundary, the analysis is going to be limited to the accuracy of the boundary containing the single element edge. This restriction should be considered when deciding how close to the areas of primary interest to put the interface elements.

Guidelines

- The interface elements use the geometry of the boundary with the least number of p-element edges, which consist of cubic polynomials. If the other boundaries have widely varying geometry, poor answers may result. Warnings may be issued, but no geometrical adjustment is performed.
- Sharp corners within the interface element may degrade accuracy. An alternative is to specify multiple interface elements.
- Connecting few elements to many may result in very high p-levels along the interface. The limiting factor on the accuracy will be those few elements.
- Interface elements will only connect common fields of different element types. No kinematic constraints are enforced. Shell p-elements have five fields, and beam p-elements have six. Solid p-elements have three fields, but connecting edges of solid elements may lead to stress singularities.
- Constraining the boundaries may lead to unstable results. The most common constraint on a boundary is at an end point. Cases with constraints include multiple interfaces at the same point and end points of different boundaries connected together. Such unstable results are indicated by high epsilons or nonphysical modes.
- Superelements have not been implemented.
- The value of epsilon and the shapes of modes of primary interest should be checked to detect unstable results.
- For the rank deficient matrices, the sparse solver should be used for linear statics and the Lanczos solver for normal modes. SYSTEM(166)=4 must also be set for normal modes.
- Since the boundaries are physically distinct, certain functions, such as shell normals and stress discontinuities in the error estimator, will not be applied across the interface.
- Contour plots may show differences across the interface because of the different view meshes, even though the solutions may be identical.

Surface Interface Elements

In finite element analysis the issue of connecting dissimilar meshes often arises, especially when refinement is performed. One method of connecting these dissimilar meshes is to use interface elements. An example in which patches of elements

have been removed from the global model and replaced by denser patches for local details is shown in Figure 3-60, where the boundaries of the patches are bold.

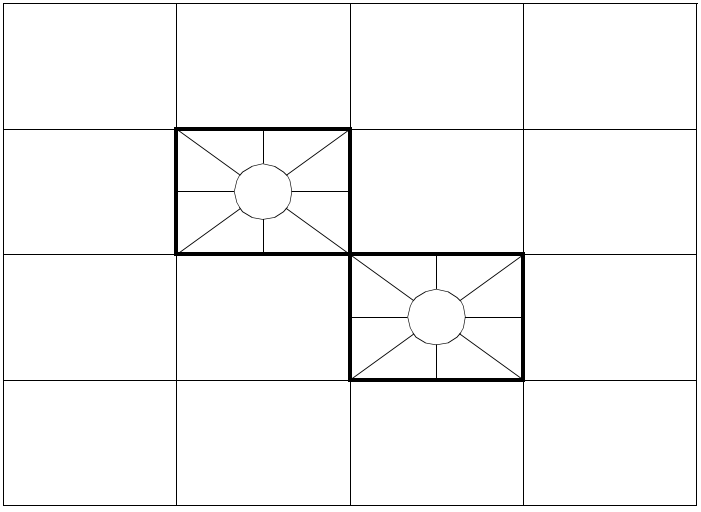


Figure 3-60 Example of Dissimilar Mesh from Global/Local Analysis

Implementation

Surface interface elements have been implemented for p-elements:

- The surface interface elements connect p-element faces. These can be either faces of solid p-elements or shell p-elements, and the faces may be either quadrilateral or triangular.
- The interface elements are geometry-based. They use the same geometry as the p-element boundaries.
- The interface elements connect only corresponding displacement fields. They do not have kinematic constraints, such as connecting shell rotations to solid translations.
- There are three methods of defining the subdomain boundaries of solid or shell p-element faces (GMBNDS). For the surface interface, each boundary may be defined using the GMSURF with which the finite element faces are associated; the `FEFACE`s defining the finite element faces; or in the most basic form, the GRIDs over the finite element faces.
- Once the boundaries have been defined, they must be associated with the interface elements. This is accomplished by referencing the boundaries in the interface element definition (GMINTS).
- Since the interface elements consist only of the differences in displacement components weighted by the Lagrange multipliers, there are no conventional element or material properties. The property Bulk Data entry (PINTS) specifies a tolerance for the interface elements, which defines the allowable distance between the subdomain boundaries; and a scaling factor, which may improve the conditioning of the Lagrange multipliers.

Output

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

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

Guidelines

- The interface elements use the geometry of the boundary with the least number of degrees of freedom, which consist of cubic polynomials. If the other boundaries have widely varying geometry, poor answers may result. Warnings may be issued, but no geometrical adjustment is performed.
- Sharp corners within the interface element may degrade accuracy. The preferred alternative is to specify multiple interface elements. An example of multiple interface elements is solved as the second example.
- Connecting few elements to many will not improve the accuracy along the interface. The limiting factor on the accuracy will be those few elements. For example, if one boundary has one element and the other boundary has four, the accuracy will be limited to that one element.
- Interface elements will only connect common displacement fields of different element types. No kinematic constraints are enforced. Shell p-elements have five fields, and solid p-elements have three. Therefore the rotational fields of the shell p-elements will not be connected to the solid p-elements.
- The sparse solver for linear statics and the Lanczos eigensolver for normal modes should be used. The sparse solver is the default solver for linear statics.
- The value of epsilon, which is the residual from the linear solution, and the shapes of modes of primary interest, which can best be evaluated graphically, should be checked to detect unstable results. Plots of displacements and stresses may also indicate unstable results. This would be visible as discontinuities in the displacements or stresses across the interface, which would imply a poor solution in that area.
- Since the boundaries are physically distinct, certain functions, such as shell normals and stress discontinuities in the error estimator, will not be applied across the interface.
- Contour plots may show differences across the interface because of the different view meshes. However, this is an indication of the results processing, not the original solution. A denser view mesh would reduce the differences.

Limitations

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

Connector Elements

Introduction

The Connector elements (CWELD, CFAST, CSEAM, and CINTC) can be used to address and solve many of the modeling issues associated with the other structural connector and fastener elements such as the bars, the rigid elements and MPC's. See Connector Elements in the *MSC Nastran Linear Static Analysis User's Guide* for the basic definition of these elements. The connector family of elements enables you to connect surfaces with differing mesh densities and using a subset of spot weld elements you can connect more than one element per surface.

CWELD Element

The CWELD /CFAST elements have been changed so that there is now a consistent formulation between linear and SOL 400 nonlinear analysis. Additionally, the CSEAM element is now supported in SOL 400 and has a consistent formulation between linear and SOL 400 nonlinear analysis.

PARAM, NEWWELD, YES (DEFAULT)

- The CWELD/CFAST elements provide the same element output formats in both linear and SOL 400 nonlinear solutions.
- In the CFAST/CSEAM/CWELD analysis, the auxiliary points generated are in the solution set.
- The CWELD/CFAST algorithm has been improved to find the Best Possible Projection with zero projection tolerance improvements.
- The improved CWELD with options "PARTPAT" and "ELPAT", and CFAST elements do not move GA and GB if both are supplied by the user, thus maintaining user mesh integrity.
- The 3x3 mesh limitation has been removed for the CWELD with options "PARTPAT" and "ELPAT" and the CFAST elements.
- There are no required changes in the user element input description of the CFAST/CSEAM/CWELD elements.
- The CFAST/CWELD/CBUSH provides nonlinear force output for SOL 400 "ANALYSIS=NLTRAN".
- MPC Force output is available for the connector element constraints.
- Besides global search algorithm control there is now local connector element connectivity control via the new CONCTL bulk data entry.
- A brief summary of connector projection results is output in the F06 file for each connector type.
- A new "SWLDPRM, CSVOUT, UNITNUM" entry will produce a comma separated file useful for reports.
- The CSEAM and CWELD (not by default) can now contribute mass to the structure.

Additional Information

- In SOL 400, for "ANALYSIS=NLSTAT" or "ANALYSIS=NLTRAN", the generated RBE3 constraints become Lagrange elements and may undergo large rotation. For "ANALYSIS=NLTRAN" with initial conditions (IC=n) in case control that cause large initial stresses in the structure at time t=0, the case control entry RIGID needs to have the value "RIGID=LINEAR" to insure convergence.
- For user desiring to post process the CFAST/CSEAM/CWELD connectors with their own methods, the following is useful:
  a. The GEOM2 table contains, after module MODGM2, a record ELCORR that correlates the CFAST/CSEAM/CWELD and its associated RBE3 elements. Also, this module will, for linear analysis and for nonlinear SOL 400 analysis run with RIGID=LINEAR, place the internal generated RBE3 into the GEOM4 table.
  b. In SOL 400 with "RIGID=LAGRANGE" (Default), internally generated RBE3 elements go into the GEOM2 (as do all other user specified rigid elements) not the GEOM4 table.
  c. The CWELD/CFAST/CBUSH force output for "ANALYSIS=NLTRAN" in SOL 400 is OP2 file output on OEFNLXX data block. If SCR=POST is run, then this force data is also written to the data base file OEFNL3 and op2 file OEFNL.op2 is also written.
- The DISPLACEMENT (CONNECTOR=) Case Control Command works in the same fashion for both the old connector formulation and the new connector formulation.
- Reminder: For the CFAST with option "ELEM" and the CWELD with option "ELPAT," the shell elements connected on each patch must have same property identification number of PSHELL entry.
- Reminder: If parameter OSWPPT is used to specify the offset for internally generated grid IDs, its value should be greater than the maximum identification number of GRID entries to avoid conflict IDs.

Formulation Changes

In the new consistent formulation for the CWELD/CFAST elements for linear analysis, RBE3 elements are written internally, and the auxiliary points are in the solution set and both are identifiable by the SWLDPRM, PRTSW entry.

- The auxiliary grids generated start with GRID ID 101000001. There are always four auxiliary grids for patch A and four auxiliary grids for patch B.
- The RBE3 elements generated start with 100001002. An RBE3 is generated for each auxiliary point for each patch A and B tying each patch grid to that auxiliary point. There is a RBE3 generated for GA tying GA to its patch auxiliary points and a RBE3 generated for GB tying GB to its patch auxiliary points.
- Both linear and nonlinear output is consistent.

The CWELD can connect congruent as well as non congruent meshes. The connectivity options can handle spot welds which connect more than one element per sheet and handle connections other than surface patches. The connector itself is considered to be rigid in user defined degrees of freedom.

The CWELD using the ELEMID and GRIDID format, (see Figure 3-61) is limited to connect one element per shell sheet in patch-to-patch type connections. The connectivity could be unsymmetric so that results became unsymmetric for a

model with symmetric mesh, loads, and boundary conditions. The two patch-to-patch type connections, formats ELPAT and PARTPAT,preserve symmetry and can connect more than one element per sheet, see Figure 3-62. In addition, the two element connectivity options could miss elements if the mesh is so fine that the spot weld area spans over more than one element, see Figure 3-63. The situation of meshes with element sizes smaller than spot weld sizes may occur in automotive models for crash analysis.

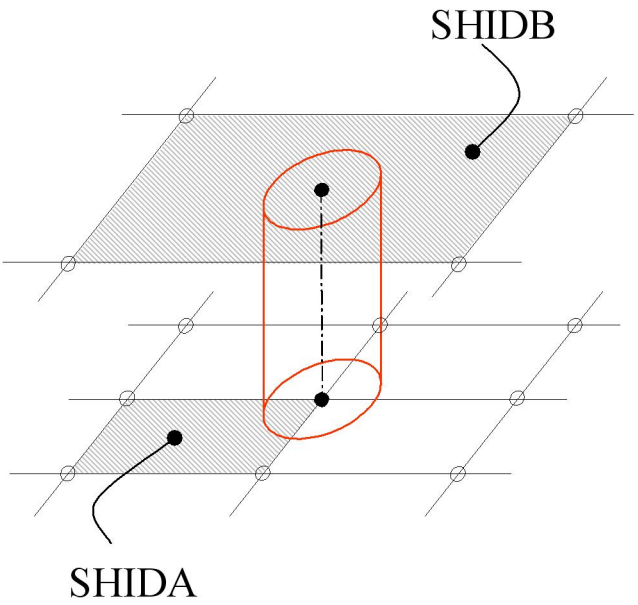


Figure 3-61 Connected Elements for Format ELEMID

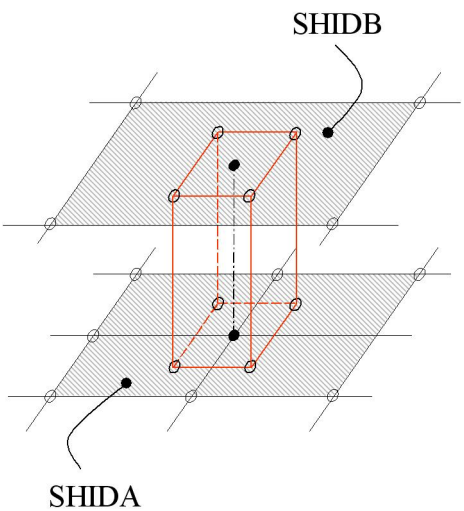


Figure 3-62 Connected Elements for Format ELPAT

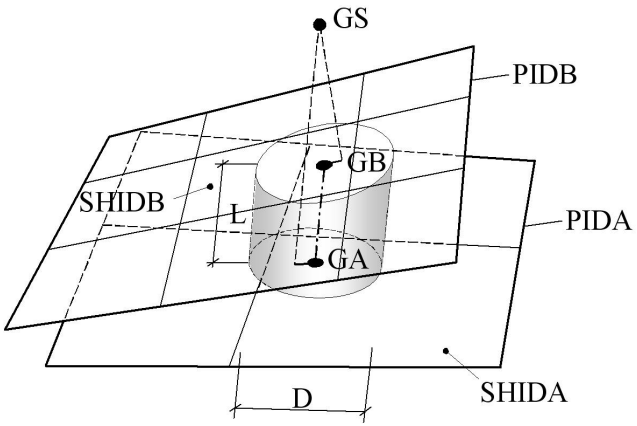


Figure 3-63 Spot Weld Covering a Patch of Elements

Starting in Nastran V2016.1 the CWELD has been enhanced by default: The original CWELD can be obtained with use of PARAM, OLDWELD, YES.

CFAST and the CWELD with options "PARTPAT" and "ELPAT" with an improved formulation has removed the restriction that a connector patch cannot span more than three elements. It will now span over a patch of as many elements that the value of diameter D of the patch encloses and for which projections can be found.

In the following Figure 3-64 (The example for this figure can be found at /tpl/connectr14/cei_103.dat), all element grids contained in the green circle region of say patch A will be used in a RBE3 connection in addition to the RBE3 connections generated for the four auxiliary points. The green circle passes through the four auxiliary points of the patch (The nine digit grid IDs.). The user-specified diameter D on the PWELD and

PFAST entries determine the locations of the four auxiliary points. (The green circle diameter is approximately 1.253D.). The element grids shown outside the green circle belong only to the respective auxiliary points.

For higher-order shell elements CQUAD8 or CTRIA6 with no missing midside nodes, the RBE3 relationships use only the midside nodes. If one or more midside node is missing (NEVER RECOMMENDED), then the corner nodes are used.

The diameter D on the PFAST/PWELD entry is used to determine the projection location of these auxiliary points as well as the stiffness properties of the patch to patch connection.

A single RBE3 then connects the four auxiliary points and the shell grids within the green circle to the connector grid GA=4000065.

Reminder: The CWELD ELEMID option still only connects by design two elements. The diameter is only used to compute the beam stiffness.

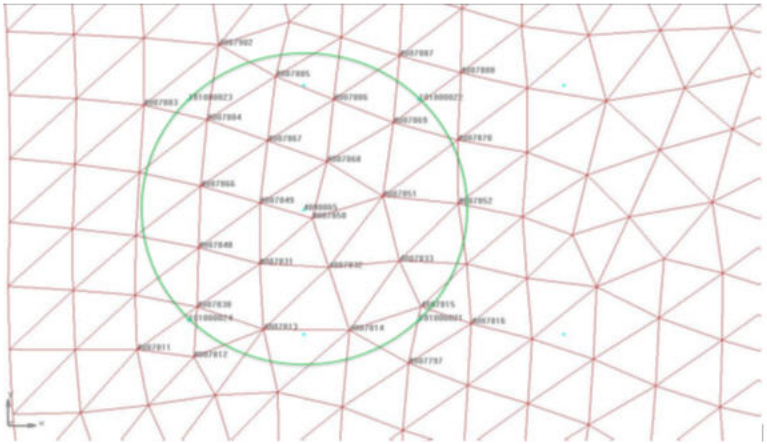


Figure 3-64 (a) Spot Weld Covering a Patch of Elements

The SWLDPRM, PRTSW entry will list the additional grids located within the green circle under FMESH SHELL A or B GRIDS where FMESH is the f06 file listing title of the additional shell grids connected in the RBE3 relationships for Finer MESH.

Table 3-15 shows the grids associated with auxiliary grid 101000023 of Figure 3-64 for its RBE3 generation. The WTi's are weight factors based on patch shape functions. Grids G1, G2, G3 are selected for RBE3 EID

100001026 because they are the shell grids of the triangular element that contains the projected auxiliary point.

Table 3-15 Auxiliary Grid 101000023 and Associated RBE3 100001026

RBE3	EID	GAH3	REFC	Weight	Ci	Gi
	100001026	101000023	123	WT1	123	G1=4007883
				WT2	123	G2=4007884
				WT3	123	G3=4007902

Table 3-16 shows the grids associated with grid GA=4000065 of Figure 2-1 through RBE3=100001022. G5 through Gn are the grids contained within the green circle. Grids G2=4007884, 4007869, 4007815, and 4007830 are NOT included in any of the G5-Gn entries because they are included in their associated auxiliary point RBE3 elements 100001026 because they are the shell grids of the triangular element that contains the projected auxiliary point.

Table 3-16 All GHAi + Patch A Green Circle Grids for RGE3 100001022

RBE3	EID	GRD_A	REFC	Weight	Ci	Gi
	100001022	4000065	123456	WT1	123	GAH1=101000021
				WT2	123	GAH2=101000022
				WT3	123	GAH3=101000023
				WT4	123	GAH4=101000024
				1.0	123	G5= 4007885
				...	...	...
				1.0	123	Gn=last

The CWELD element, using the ELEMID and GRIDID format is restricted to handling connectivities of shell elements or surface patches made up of grid point sequences. The point-to-patch type connectivity allows any element type, load or boundary condition to be connected to the CWELD. The element can now be used to model bearings of large masses, connectivities to springs, gaps, etc., see Figure 3-65.

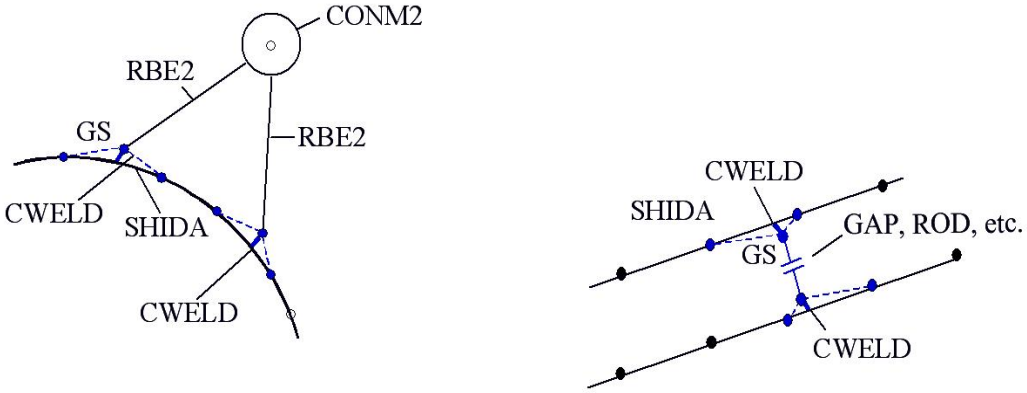


Figure 3-65 Modeling Options using the Point-to-patch Type Connectivity

Report Format

CFAST	ELEM	6000002												
FMESH SHELL A GRID IDS														
	4007813	4007814	4007831	4007832	4007833	4007848	4007849	4007850	4007851	4007852				
	4007866	4007867	4007868	4007870	4007885	4007886								
T_BE	MATRIX		0.00E+00	0.00E+00	-1.00E+00									
			0.00E+00	1.00E+00	0.00E+00									
			1.00E+00	0.00E+00	0.00E+00									
NGSMV	N_REDIA	FINAL_I	THETA_AI	THETA_H:	THETA_H:	THETA_H:	THETA_H:	PTOLAB	PTOLH1	PTOLH2	PTOLH3	PTOLH4	XL0	XL
0	0	4.43E+00	0	0	0	0	0	0.2	0	0	0	0		
	GRID ID	RBE3 EID	X	Y	Z	GRID ID	RBE3 EID	X	Y	Z	GRID ID	X	Y	Z
GAB5	4000065	1E+08	7.50E+01	7.50E+01	9.50E+01	4000066	1E+08	7.50E+01	7.50E+01	1.00E+02	4000065	7.50E+01	7.50E+01	9.50E+01
GAH12	1.01E+08	1E+08	7.94E+01	7.06E+01	9.50E+01	1.01E+08	1E+08	7.94E+01	7.94E+01	9.50E+01				
GAH34	1.01E+08	1E+08	7.06E+01	7.94E+01	9.50E+01	1.01E+08	1E+08	7.06E+01	7.06E+01	9.50E+01				
GBH12	1.01E+08	1E+08	7.94E+01	7.06E+01	1.00E+02	1.01E+08	1E+08	7.94E+01	7.94E+01	1.00E+02				
GBH34	1.01E+08	1E+08	7.06E+01	7.94E+01	1.00E+02	1.01E+08	1E+08	7.06E+01	7.06E+01	1.00E+02				
SHELL	EID	G1	G2	G3	G4	G5	G6	G7	G8					
AH1	6009263	4007816	4007815	4007797	0	0	0	0	0					
AH2	6009339	4007888	4007887	4007869	0	0	0	0	0					
AH3	6008594	4007883	4007884	4007902	0	0	0	0	0					
AH4	6008518	4007811	4007812	4007830	0	0	0	0	0					
BH1	1074	1093	1094	1100	1099	0	0	0	0					
BH2	1074	1093	1094	1100	1099	0	0	0	0					
BH3	1074	1093	1094	1100	1099	0	0	0	0					
BH4	1074	1093	1094	1100	1099	0	0	0	0					
CFAST	ELEM	6000003												
T_BE	MATRIX		0.00E+00	0.00E+00	-1.00E+00									
			0.00E+00	1.00E+00	0.00E+00									
			1.00E+00	0.00E+00	0.00E+00									
NGSMV	N_REDIA	FINAL_I	THETA_AI	THETA_H:	THETA_H:	THETA_H:	THETA_H:	PTOLAB	PTOLH1	PTOLH2	PTOLH3	PTOLH4	XL0	XL
0	0	4.43E+00	0	0	0	0	0	0.2	0	0	0	0		
	GRID ID	RBE3 EID	X	Y	Z	GRID ID	RBE3 EID	X	Y	Z	GRID ID	X	Y	Z
GAB5	4000074	1E+08	7.50E+01	7.50E+01	1.00E+02	4000075	1E+08	7.50E+01	7.50E+01	1.03E+02	4000074	7.50E+01	7.50E+01	1.00E+02
GAH12	1.01E+08	1E+08	7.94E+01	7.06E+01	1.00E+02	1.01E+08	1E+08	7.94E+01	7.94E+01	1.00E+02				
GAH34	1.01E+08	1E+08	7.06E+01	7.94E+01	1.00E+02	1.01E+08	1E+08	7.06E+01	7.06E+01	1.00E+02				
GBH12	1.01E+08	1E+08	7.94E+01	7.06E+01	1.03E+02	1.01E+08	1E+08	7.94E+01	7.94E+01	1.03E+02				
GBH34	1.01E+08	1E+08	7.06E+01	7.94E+01	1.03E+02	1.01E+08	1E+08	7.06E+01	7.06E+01	1.03E+02				
SHELL	EID	G1	G2	G3	G4	G5	G6	G7	G8					
AH1	1074	1093	1094	1100	1099	0	0	0	0					
AH2	1074	1093	1094	1100	1099	0	0	0	0					
AH3	1074	1093	1094	1100	1099	0	0	0	0					
AH4	1074	1093	1094	1100	1099	0	0	0	0					
BH1	6000470	4000627	4000628	4000652	4000651	4009715	4009814	4009815	4009810					
BH2	6000520	4000675	4000676	4000700	4000699	4009915	4010014	4010015	4010010					
BH3	6000518	4000673	4000674	4000698	4000697	4009907	4010006	4010007	4010002					
BH4	6000468	4000625	4000626	4000650	4000649	4009707	4009806	4009807	4009802					

Theory

Finding CWELD End Points GA and GB

If the CWELD end points GA and GB are not defined, then they are created from the locator point GS as follows. For the patch-to-patch type connectivity, the user supplied grid point GS is projected on to shell A and B according to Figure 3-66. The end points GA and GB are created by a normal projection of GS to shell A and B if GS lies between shell A and B. A two step projection is used if GS lies outside of shell A and B. First, the auxiliary points GA' and GB' are created by a normal projection of GS to shell A and B. From the midpoint GC' of section GA',GB' a second normal projection is done that produces the final end points GA and GB. All the points described in this paragraph do not have degrees of freedom attached.

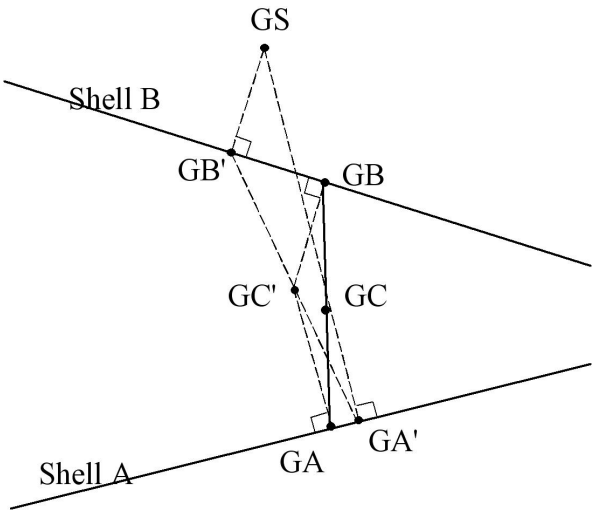


Figure 3-66 Projection of GS to Determine End Points GA and GB

Element Coordinate System

The element x-axis points from GA to GB.

$$e_1 = \frac{x_B - x_A}{\|x_B - x_A\|} \quad \text{element x-axis}$$

In case of zero length, the normal of shell A is taken. All vector components are in basic unless noted otherwise.

Find the smallest component j of e_1

$$e_1^j = \min_{i = 1, 2, 3} \{e_1^i\}.$$

In case of two equal components we take the one with the smaller i. The corresponding basic vector

$$b_j \text{, e.g., for } j = 3,\ b_3 = \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix}$$

provides a good directional choice for e_2. In addition, the vector e_2 must be orthogonal to e_1.

$$\tilde{e}_2 = b_j - \frac{e_1^T b_j}{e_1^T e_1} e_1 \qquad e_2 = \frac{\tilde{e}_2}{\|\tilde{e}_2\|} \quad \text{element y-axis}$$

and e_3 is just the cross product

$e_3 = e_1 \times e_2$ element z-axis

The final transformation matrix is

$$T_{be} = \left[e_1 \middle| e_2 \; e_3 \right]$$

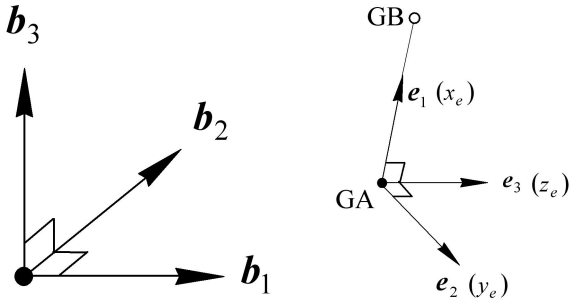


Figure 3-67 Element Coordinate System

Element Stiffness

The elastic part of the CWELD is a short beam from GA to GB with 2x6 degrees of freedom. The element is a shear flexible Timoshenko type beam. The Young's and shear modulus is taken from the material defined on the PWELD property entry. The cross sectional properties are calculated from the user supplied spot weld diameter D defined on the PWELD property entry,

E = Young's modulus

G = shear modulus = $\frac{E}{2(1+\nu)}$

A = area = $\pi D^2/4$

I_{zz} = I_{yy} moment of inertia = $\frac{\pi D^4}{64}$

J = $\frac{\pi D^4}{32}$

The length L of the beam element is from end points GA to GB. If the user defines the type as SPOT on the PWELD property entry, then the effective element length L_e is always

$$L_e = \frac{1}{2}(t_A + t_B)$$

regardless of the true length L. The user supplied values for the Young's and shear modulus E and G are scaled by the ratio of true length to effective length

$$\tilde{E} = E\frac{L}{L_e} \qquad \tilde{G} = G\frac{L}{L_e}$$

With the scaling, the spot weld stiffness is approximately constant for all elements. Extremely stiff elements from short lengths L and extremely soft elements from long lengths L are avoided. For zero length L, the beam element degenerates to 6 uncoupled springs. If the user does not define SPOT as the type, the true length is used if it is inside the range of

$$LDMIN \le \frac{L}{D} \le LDMAX$$

where *LDMIN* and *LDMAX* are user supplied parameters on the PWELD entry. By default $LDMIN = 0.2$ and $LDMAX = 5.0$. If the user does not want the true length adjusted at all, $LDMIN = 0.$ and $LDMAX =$ 1.e+12 may be specified.

Constraints Connecting GA and GB to Shell Grids

The end points GA and GB are connected to the shell grids of shell A and B, respectively, using constraint equations. The 3 translations at grid GA are connected to the 3 translations of the shell grid points using the interpolation functions of the corresponding shell surface. The 3 rotations at grid GA are connected to the 3 translations of the shell grid points using Kirchhoff conditions.

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix}_A = \sum N_I(\xi_A, \eta_A) \cdot \begin{Bmatrix} u \\ v \\ w \end{Bmatrix}_I$$

$$\theta_x^A = \frac{\partial w}{\partial y} = \sum N_{I,y} \cdot w_I$$

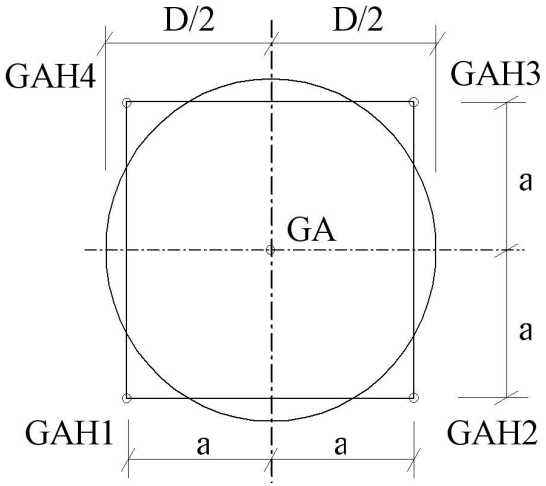
$$\theta_y^A = -\frac{\partial w}{\partial x} = -\sum N_{I,x} \cdot w_I$$

$$\theta_z^A = \frac{1}{2}\left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right) = \frac{1}{2}\left(\sum N_{I,x} \cdot v_I - \sum N_{I,y} u_I\right)$$

For the formats GRIDID and ELEMID on the CWELD entry, the grid points I are grid points of shell elements or grid points of solid element faces. For the formats ELPAT and PARTPAT, the grid points I are auxiliary points GAHI and GBHI, I=1,4, constructed according to Figure 3-68 and Figure 3-69. The auxiliary points are connected to shell element grids with a second set of constraints

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix}_I = \sum_K G_{IK} \cdot \begin{Bmatrix} u \\ v \\ w \end{Bmatrix}_K$$

where G_{IK} is a coefficient matrix derived from RBE3 type constraints. The grid points K are shell grid points.



$$a = \sqrt{\pi}\,\frac{D}{4}$$

Figure 3-68 Equivalent Square Cross Sectional Area for a Spot Weld of Diameter D

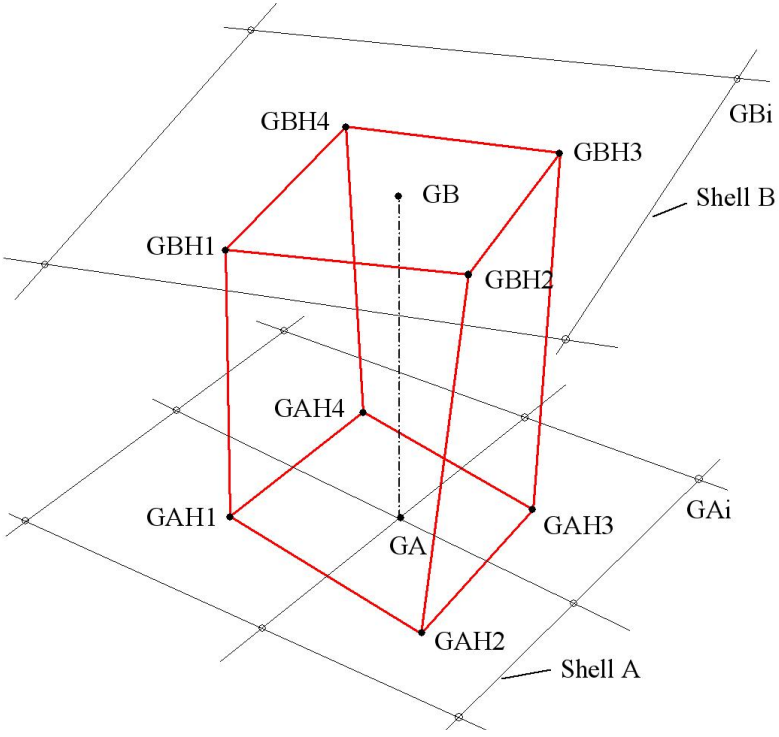


Figure 3-69 Auxiliary Points GAHI and GBHI for Formats ELPAT and PARTPAT

With the formats ELPAT and PARTPAT, the element can connect from one to 3x3 elements on shell A and B, respectively, see Figure 3-70.

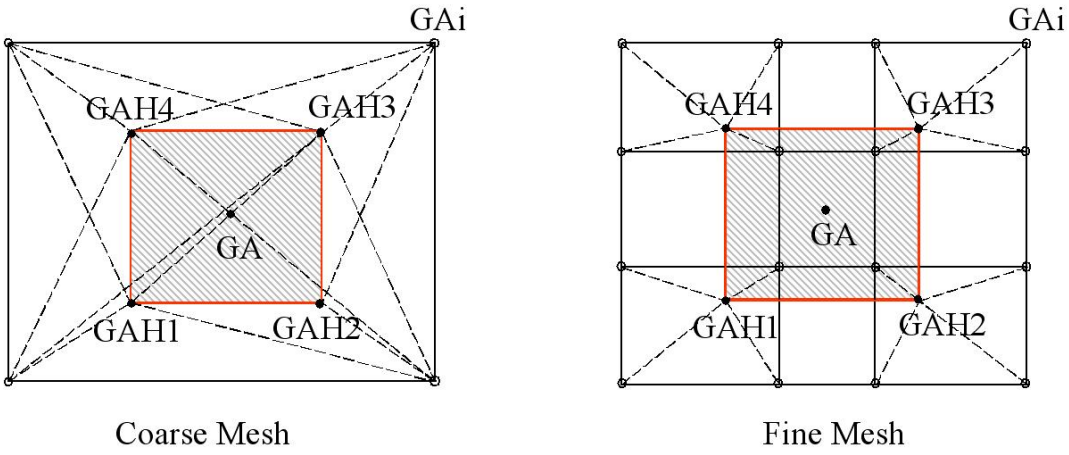


Figure 3-70 Connectivity for Formats ELPAT and PARTPAT

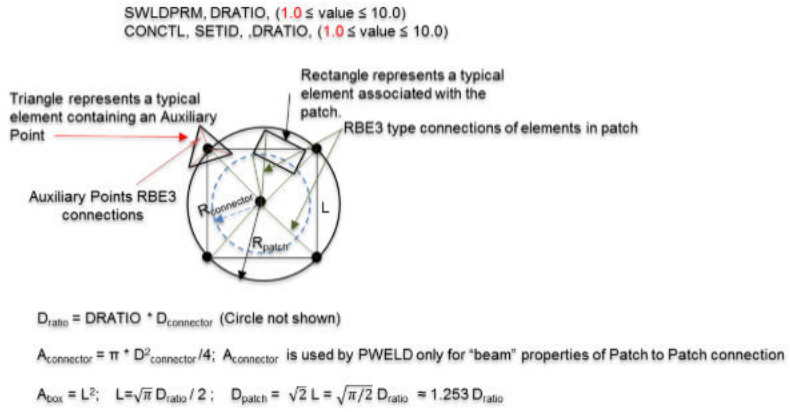
With the constraint equations above, the degrees of freedom of grid points GA, GB and the auxiliary points GAHI, GBHI are condensed out on the element level. The final stiffness matrix of the CWELD consists of translational degrees of freedom of grid points on shell A and B, respectively. There are no extra degrees of freedom generated except for the case where the user requests explicit constraints by setting MSET=ON on the PWELD entry. The flag MSET=ON works only for the formats ELEMID and GRIDID.

Connector Stiffness

Connector contribution to a structural model's overall stiffness is sensitive to the model's mesh size and the orientation of the connector relative to the mesh. Thus, the discretization process itself may cause, for example, a model using a fine mesh to be stiffer in torsion than a corresponding model using a coarse mesh. Also for production models that correlate well with test, refining the mesh may cause an inherent overall loss of stiffness due to mesh refinement and hence loss of correlation.

To allow the user some control over this loss of stiffness, the new connectors (CWELD with ELPAT or PARTPAT or CFAST) are provided with two options to provide additional connector stiffness. The two options may be used individually or in combination.

The first stiffening technique is activated by "SWLDPRM, DRATIO, (1.0 ≤ *value* ≤ 10.0)" or "CONCTL, SETID, , DRATIO, (1.0 ≤ *value* ≤ 10.0)". For this option the diameter, D_{ratio}, is defined as D_{ratio} = DRATIO * $D_{connector}$. This results in the diameter of the patch taking a value of $D_{patch} = \sqrt{\pi/2}\ D_{ratio}$. The default of DRATIO is a *value*=1.0 which implies the diameter of the patch is computed in the standard fashion. For the patch to patch connection for the "beam" properties of the CWELD, the area is still computed as $A_{connector} = \pi\ D^2_{connector}/4$ as defined in the PWELD entry.



A disadvantage of this method is that as DRATIO is increased using the global command SWLDPRM, DRATIO, value; some connector elements may begin to fail because they may no longer be able to find a patch projection.

To overcome this, the "SWLDPRM, NREDIA, Integer_value" can be increased to a value as high as Integer_value = 8 to allow failing welds to halve their patch diameters up to eight times.

If the "SWLDPRM, NREDIA, Integer_value" is not an approach the user wishes to pursue, then for these failing elements, the bulk data entry CONCTL, SETID, ,DRATIO, value can be used to define a set for failing connectors and set a value of DRATIO for these connectors that allows them to find a projection.

The second stiffening algorithm attempts, based on the diameter of the connector, to determine a measure of the mesh discretization.

This feature is activated by "SWLDPRM, SKIN, 1" or "CONCTL, SETID,, SKIN, 1". The default is a 0 which implies no stiffening. There is an associated stiffening factor "SWLDPRM, SCLSKIN" with value = 0.10 as default.

Depending on the complexity of the model and the overall mesh size and the number of connectors within the model and the diameter of the connectors relative to the mesh, the default value tends to stiffen a structural model from about 0.4% to about 4%. A value of SCLSKIN=10.0 stiffens coarser mesh models by about 10% to 11% and finer mesh models by about 2% to 6%.

The contribution of the stiffening algorithm to the overall stiffness of the FEM model eventually reaches a limit. For example, a very large value SCLSKIN=100 increases the stiffness of the models overall by only about 0.1% to 2% over the stiffness obtained for SCLSKIN=10.

For a correlated structural model evaluated at a specific mesh size, with an aim to refine the mesh for some portion of this model containing connectors, while leaving other portions containing connectors with an unmodified mesh, it is recommended that the "SKIN, 1" and "SCLSKIN, real value" be entered on the CONCTL bulk data entry referring to the connectors within the area of the refined mesh. Different refined mesh areas within the structural model can have different values of SCLSKIN associated to the specific connectors in each refined region.

For post processing of the SKIN option, for the affected shell elements, an updated EPT table is available after module MODGM2. It contains the PSKNSHL record that correlates the property data of the shells involved and a list of shell elements for each patch modified.

Connector Control:

To provide the user with better connector control, the following new Bulk Data entry has been introduced:

CONCTL Parameter SWLDPRM override for CFAST, CSEAM, and CWELD Connector Elements.

This entry provides local connector search algorithm control to override SWLDPRM values.

Please see CONCTL (p. 1596) in the *MSC Nastran Quick Reference Guide* for the complete Bulk Data entry.

Enhanced Algorithm

For the new connector logic, the search algorithm has been enhanced based on user inputs in an attempt to achieve the best possible connections. The new search tolerance starts with a zero projection tolerance. This may result in changes from the previous connector results using the old CWELD/CFAST elements.

The list below gives a brief summary of the highlights of the improved CWELD/CFAST algorithm.

- For the CFAST and the CWELD with options "PARTPAT" and "ELPAT", grids GA and GB internally keep the user-specified IDs and the user-specified locations. This change was primarily introduced because many users complained that the location of GA and GB represented their modeling procedures and desired mesh locations.
- For the CWELD with ELEMID/GRIDID option, grids GA and GB internally keep the user-specified IDs and the user-specified locations, but in the case when GA and GB are associated with shell patches, a duplicate internal grid is generated to avoid singularity of the generated RBE3.

  CWELD, 5646, 22, , ELEMID, 3276, 3115

  , 2191, 1941

  CTRIA3, 2191, 8, 3272, 3276, 3271

  Grid 3276 as input from standard mesh modeling procedures will automatically be placed in the independent degree of freedom set, or may have been placed by the analyst in a SPC or MPC set at generation time. In either case, the CWELD algorithm must create an internal constraint on this grid using a RBE3 element. This causes a set conflict which is avoided by generating an internal grid.
- For two stacked connectors having a common patch with a common grid, the program checks duplicated GA/GB and only a single RBE3 is generated for the common patch.

  CWELD, 11, 100, 9001, PARTPAT, 3001, 3002

  CWELD, 12, 100, 9002, PARTPAT, 3002, 3003

- If the user specifies both grids GA and GB, for the CFAST and the CWELD with options "PARTPAT" and "ELPAT", the SWLDPRM, GSMOVE entry is nonfunctional. In the CFAST and the CWELD with options "PARTPAT" and "ELPAT", if the user specifies both GA and GB they will not be moved. This may cause the CFAST/CWELD search algorithm to fail for some welds that had passed under the old CWELD/CFAST search algorithm. If this occurs, the user can do one of four things:
  a. Determine a better location for GA and GB of the failing welds so that they may project.
  b. Remove GA and GB from the CWELD/CFAST entry and replace with a GS entry allowing the CWELD/CFAST algorithm to move and project and generate internal GA and GB locations.
  c. Use "SWLDPRM, MOVGAB, 1" to generate internal GA and GB grids at the corrected locations for all CWELD/CFAST. The locations of the original user specified GA and GB are unchanged.
  d. Use the new CONCTL Bulk Data entry with "SWLDPRM, MOVGAB, 1" to allow local control of specific welds to correct the locations of grids GA and GB. (See Connector Control section below for detail.)

     CONCTL, 83, , MOVGAB, 1

     Where:

     SET3, 83, ELEM, 1345, 2678
- The maximum tolerance for SWLDPRM, PROJTOL has been relaxed.
  a. Regardless of the value of SWLDPRM PROJTOL, the algorithm starts by assuming a zero projection tolerance for the projections of GA/GB for the CWELD option "PARTPAT" or the CFAST option "PROP" and for GAHi/GBHi for the CWELD options "PARTPAT" and "ELPAT" and any CFAST option.
  b. The tolerance is increased by 0.02 until a projection is found or the PROJTOL value is reached.
  c. This can be turned off while computing the auxiliary grid projection onto EIDA/EIDB by setting PROJTOL= - value where 0.0 ≤ value ≤ 1.0. In this case, the projection calculation starts at tolerance = |PROJTOL|. For the rest of the projection search, the algorithm reverts back to (a) and (b) above.
- A brief "Connector Summary" of projection results is always output in the f06 file for each connector type: FST-ELEM, WLD-PARTPAT, WLD-ELEMID, etc.

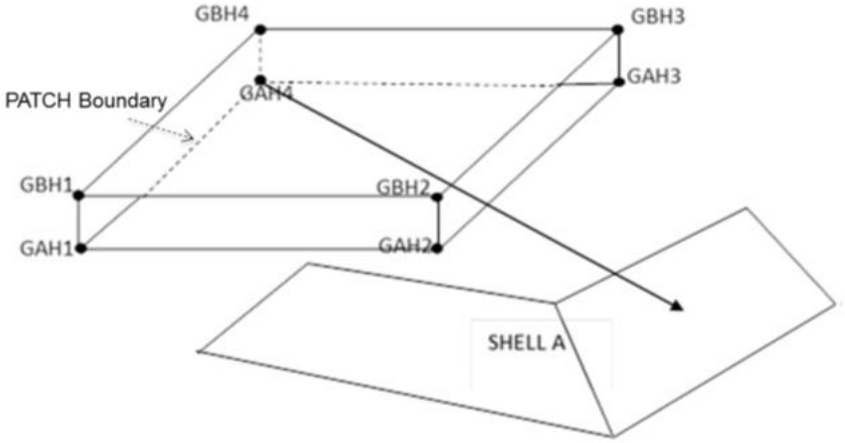
C O N N E C T O R S U M M A R Y					
Elem type	GOOD/BAD NUMBER FOUND	MAX ANGLE B/N SHELL NORMALS GAB/GH (GSPROJ) AT EID	MAX TIMES GS MOVED (GSMOVE) AT EID	MAX TIMES GS MOVED (GSMOVE) AT EID	MAX TIMES DIAMETER REDUCED (NREDIA) AT EID
FST-ELEM	G 26 B 0	GAB 0.0 (20.00) 6000000	GH 0.0 (0.02) 6000000	0 (0) 6000000	1 (4) 6000018

- For linear connectors, MPCFORCE output is available. In nonlinear SOL 400, the RBE3 elements generated become Lagrange elements if the default RIGID=LAGRANGE is used and are no longer in the MPC set; hence, there will be NO MPCFORCE output for RIGID=LAGRANGE.
- In the improved CFAST and CWELD, GA and GB are not moved and internal coincident grids are not generated at a new location; thus, two additional restrictions are required.
  a. There can be no user-supplied constraints on GA and GB. A fatal message will be issued if there are any.
  b. The CWELD length must be > 10-6. The point to patch option defined by ELEMID or GRIDID will, however, still create a new GS internally to obtain a minimum required length; i.e., LDMIN ≤ length/D ≤ LDMAX. For the point-to-patch connection, GS is used as GB. The algorithm will use the new GS as GB but keep the user-supplied GS unchanged. Since the point-to-patch is often used to "tack" two shell corners, the default LDMIN may cause the connector to be unstable. To avoid this, it is recommended that the user set LDMIN=1.E-6 on the appropriate PWELD entry.
- **Warning:** CWELD will not contribute to MASS by default even if its associated MATi entry has a nonzero density. To react to a nonzero density "SWLDPARM, WMASS, 1" is required. If mass is computed, the PARAM, COUPMASS effects the mass calculation.

Detailed Projection Algorithm for Best Possible Projection enhanced CWELD

- The Enhanced algorithm applies to the following projection calculations:
  a. Find projections of GA/GB for the "PARTPAT" format of CWELD and "PROP" format of CFAST.
  b. For "ELPAT" format of the CWELD and the "ELEM" format of the CFAST, the user has specified the specific shell elements and therefore no element search for GA/GB projections is required. Though no search is required, GA/GB, however, must project onto the user specified EIDA/EIDB.
  c. Find projections of the auxiliary grids GAHi/GBHi for the "PARTPAT" and "ELPAT" options on the CWELD and any CFAST option respectively.
- The projection algorithm searches for possible projections from shell elements with shell grids that are closest to GS. The closest grid may connect to several shell elements; hence, more than one shell element may get a projection from GS for curved patches. The shell elements with projections from GS are collected and the selection is based on the SHIDA/SHIDB pair with the smallest angle between their normal vectors.
  a. The old CWELD/CFAST algorithm used the first shell elements found to get a projection and used SWLDPRM, GMCHK, 1 and 2 to provide some control. These two options have no effect on the new formulation.

b. Backward connections sometimes occur if the patch is near the boundary of a structure, and there is a "vertical" flange associated with the patch elements. In this case, SWLDPRM, GMCHK, 3 may be used to prevent backward projection. (See Figure 3-64a.)



- The minimum angle selected above must be ≤ SWLDPRM GSPROJ if GSPROJ ≥ 0.0
- If the user has not specified BOTH GA and GB and the algorithm cannot find a GSPROJ satisfied projection, then for SWLDPRM GSMOVE entry, the point GS will be moved in an attempt to satisfy the projection requirement.
- Reminder: If user has specified both GA and GB and CFAST and the CWELD with options "PARTPAT" and "ELPAT" are used, then GSMOVE will be ignored and the connector will fail to connect if the user has taken the default "SWLDPRM, NREDIA, 0" for NREDIA. Failed connectors issue USER FATAL MESSAGE 7635.
- If the GSMOVE specification limit is reached for the CFAST or the CWELD with options "PARTPAT" and "ELPAT" and SWLDPRM NREDIA ≠ 0, then the diameter of the connector will be reduced by half to compute new locations of auxiliary grids. If necessary this is repeated until the NREDIA specified value is reached.
  a. When the NREDIA ≠ 0 is initiated, the GS at its current location is used for GSMOVE ≥ 0.
  b. When the NREDIA ≠ 0 is initiated, the GS at its original location is used for the new option GSMOVE < 0.

Outputs

Displacement Output of GA and GB are available for both the CWELD and CFAST Elements

The displacements of the projected grids GA and GB for CWELD elements with GRIDID, ELEMID, ELPAT or PARTPAT format and CFAST elements are computed to display the relationship between these elements and their connecting shell elements. The displacement output of GA and GB for CWELD element with GRIDID or ELEMID format and MSET=OFF are calculated from the constraint equations described in the **Basic Theory and Methods** section.

If GA or GB is not specified for CFAST elements or CWELD elements with ELPAT or PARTPAT option, the program will create a grid internally, with the grid ID number starting from OSWPPT+1 (OSWPPT is a parameter specified by

PARAM Bulk Data entry). The user may request a positive PRTSW parameter (1, 2, 11, or 12) in SWLDPRM Bulk Data entry to view the grid ID of GA or GB in the diagnostic output.

The displacements of GA and GB are requested using the CONNECTOR keyword of the DISPLACEMENT Case Control command. Outputs

The output is integrated with the displacements of the general grid points.

Basic Theory and Methods

For CWELD elements with GRIDID or ELEMID format, the displacements of the projected grids GA and GB in basic coordinate are computed by the following equations:

$$d_A = \sum_I A_I u_I$$

$$d_B = \sum_J B_J u_J$$

where d_A and d_B are displacements at GA and GB. A_I and B_J are constraint matrices. u_I and u_J are displacements of the connected shell grids.

For CWELD elements with ELPAT or PARTPAT format, the displacements of the auxiliary points in basic coordinate are calculated first.

$$u_I = \sum_K G_{IK} u_K$$

$$u_J = \sum_L G_{JL} u_L$$

where u_I and u_J are displacements of the auxiliary points. G_{IK} and G_{JL} are RBE3 type constraint matrices. u_K and u_L are displacements of the connected shell grids.

Then the displacements of the projected grids GA and GB in basic coordinate are computed by the same constraint equations used for GRIDID and ELEMID options.

$$d_A = \sum_I A_I u_I$$

$$d_B = \sum_J B_J u_J$$

where d_A and d_B are displacements at GA and GB. A_I and B_J are constraint matrices. u_I and u_J are displacements of the connected auxiliary points.

Example

This example demonstrates the various displacement output requests and their results for a small model with two CWELD elements.

The input file follows:

```
nastran mesh
SOL 101
TIME 60
CEND
TITLE= two elements, identical location for GA, GB, GS
OLOAD= ALL
FORCE = ALL
SUBCASE 1
   SUBTITLE= shear the weld
   SPC= 1
   LOAD= 1
   DISP= ALL
SUBCASE 2
   SUBTITLE= in plane twist
   set 21 = 1002,1003,2011,thru,2014
   spc= 1
   LOAD= 2
   DISP(CONN=ALL)=21
SUBCASE 3
   SUBTITLE= upper bending
   set 32 = 4
   set 33 = 1012,1013,2001,thru,2004
   spc= 1
   LOAD= 3
   DISP(CONN=32)=33
BEGIN BULK
$
$ Grids of lower shell
grid, 1001, , 0., 0., 0.
grid, 1002, , 20., 0., 0.
:
grid, 1013, , 20., 10., 5.
grid, 1014, , 0., 10., 5.
$ Grids of upper shell
grid, 2001, , 0., 0., 0.0
grid, 2002, , 20., 0., 0.0
:
grid, 2013, , 20., 10., 6.0
grid, 2014, , 0., 10., 6.0
$ spot weld grid
grid, 3001, , 10.0, 5.0, 0.0
grid, 3011, , 10.0, 5.0, 10.0
$ quad4s
cquad4, 4001, 10, 1001, 1002, 1003, 1004
cquad4, 5001, 10, 2001, 2002, 2003, 2004
cquad4, 4011, 10, 1011, 1012, 1013, 1014
cquad4, 5011, 10, 2011, 2012, 2013, 2014
:
$ property and material
pshell, 10, 10, 1.0, 10
mat1, 10, 2.e+5 , , 0.3, 0.785e-8
$
$ spot welds
$
cweld, 4, 4, 3001, elemid, , , , , +cw4
```

```
+cw4, 4001, 5001
cweld, 5, 4, 3011, elemid, , , , , +cw5
+cw5, 4011, 5011
pweld, 4, 10, 5.0
$
enddata
```

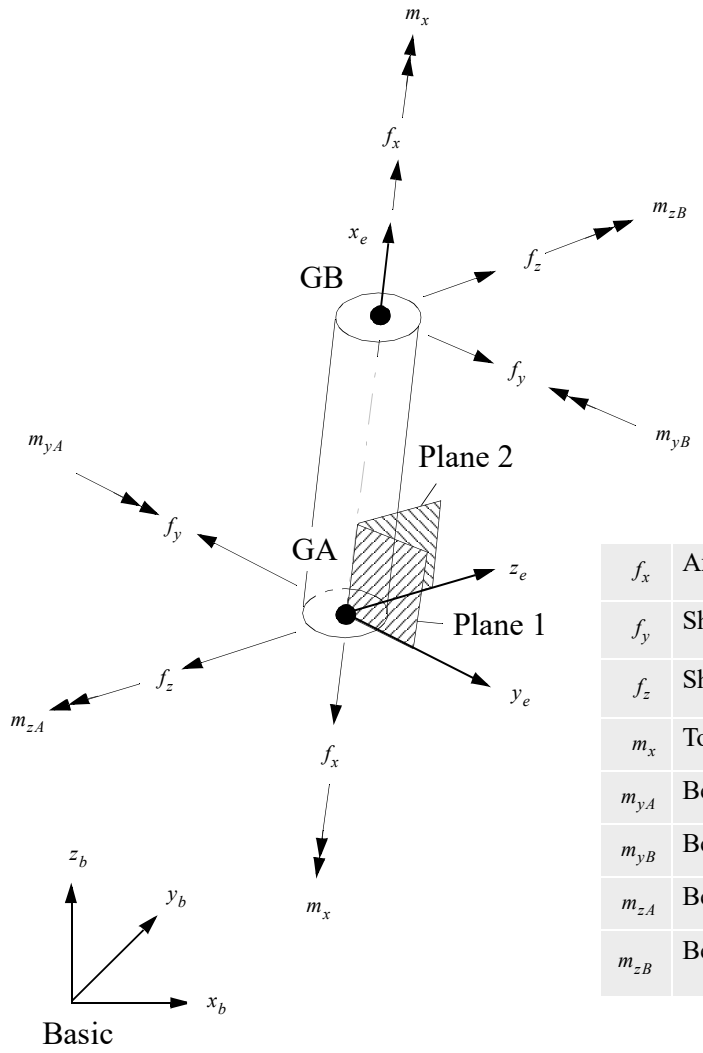
The displacement results are shown as follows:

```
0                                                                                                    SUBCASE 1

                                    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
     1001   G   0.0             0.0             0.0             0.0             0.0             0.0
     1002   G   4.887498E-03    5.624979E-04    0.0             0.0             0.0             1.406245E-05
     1003   G   4.887498E-03   -5.624979E-04    0.0             0.0             0.0            -1.406245E-05
     1004   G   0.0             0.0             0.0             0.0             0.0             0.0
     1011   G   0.0             0.0             0.0             0.0             0.0             0.0
     1012   G   4.887498E-03    5.624979E-04   -7.685000E-01   -6.750000E-03    5.864999E-02    1.406245E-05
     1013   G   4.887498E-03   -5.624979E-04   -7.685000E-01    6.750000E-03    5.864999E-02   -1.406245E-05
     1014   G   0.0             0.0             0.0             0.0             0.0             0.0
     2001   G   6.058339E-04    3.750000E-04    0.0             0.0             0.0             7.733938E-19
     2002   G   5.605834E-03    3.750000E-04    0.0             0.0             0.0             1.084202E-1
     2003   G   5.605834E-03   -3.750000E-04    0.0             0.0             0.0             1.191827E-18
     2004   G   6.058339E-04   -3.750000E-04    0.0             0.0             0.0             7.623363E-19
     2011   G   3.907157E-02    3.750000E-04    8.148733E-04   -2.562901E-16    3.850649E-02    2.836059E-17
     2012   G   4.407157E-02    3.750000E-04   -7.693148E-01   -5.232793E-15    3.850649E-02    2.851452E-17
     2013   G   4.407157E-02   -3.750000E-04   -7.693148E-01   -5.356609E-15    3.850649E-02    2.956961E-17
     2014   G   3.907157E-02   -3.750000E-04    8.148733E-04   -2.636780E-16    3.850649E-02    2.803595E-17
     3001   G   0.0             0.0             0.0             0.0             0.0             0.0
     3011   G   0.0             0.0             0.0             0.0             0.0             0.0
101000001   G   2.443749E-03    8.917563E-18    0.0             0.0             0.0             8.402567E-19
101000002   G   3.105834E-03    8.944668E-18    0.0             0.0             0.0             9.774760E-19
101000003   G   2.443749E-03    2.517788E-16   -3.842500E-01   -2.775558E-15    3.842500E-02    2.303930E-17
101000004   G   4.157158E-02    3.029749E-15   -3.842500E-01   -2.789435E-15    3.850649E-02    2.786400E-17


0                                                                                                    SUBCASE 2

                                    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
     1002   G   1.499616E-02    3.324066E-02    0.0             0.0             0.0             2.330633E-03
     1003   G  -1.499616E-02    3.324066E-02    0.0             0.0             0.0             2.330633E-03
     2011   G   6.107051E-03   -8.968769E-03   -1.542189E-16    3.303767E-17    1.479144E-16    1.890160E-03
     2012   G   2.110705E-02    4.220943E-02   -3.132043E-15    5.117434E-17    1.498367E-16    3.390160E-03
     2013   G  -2.110705E-02    4.220943E-02   -2.615099E-15    5.202362E-17    1.383761E-16    3.390160E-03
     2014   G  -6.107051E-03   -8.968769E-03    1.645747E-16    3.118436E-17    1.398878E-16    1.890160E-03
101000001   G   6.505213E-18    1.662033E-02    0.0             0.0             0.0             1.580825E-03
101000002   G   6.722053E-18    1.662033E-02    0.0             0.0             0.0             2.640160E-03
101000003   G   3.035766E-18    1.662033E-02   -1.434168E-15    4.168935E-17    1.434168E-16    1.580825E-03
101000004   G   1.461505E-16    1.662033E-02   -1.434197E-15    4.178690E-17    1.439374E-16    2.640160E-03


0                                                                                                    SUBCASE 3

                                    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
     1012   G   3.460202E-16    9.473882E-16   -1.537000E+00   -1.350000E-02    1.173000E-01    6.294140E-17
     1013   G  -4.478185E-16    9.592263E-16   -1.537000E+00    1.350000E-02    1.173000E-01    6.130147E-17
     2001   G   0.0             0.0             8.021409E-04   -1.965454E-14    7.693148E-02    0.0
     2002   G   0.0             0.0            -1.537827E+00    6.397660E-15    7.693148E-02    0.0
     2003   G   0.0             0.0            -1.537827E+00    6.758049E-15    7.693148E-02    0.0
     2004   G   0.0             0.0             8.021409E-04   -2.021495E-14    7.693148E-02    0.0
101000001   G   0.0             0.0            -7.685000E-01   -6.591949E-15    7.685000E-02    0.0
101000002   G   0.0             0.0            -7.685127E-01   -6.664110E-15    7.693148E-02    0.0
```

Force output is also produced. For internal processing, the following element types are created depending on the user specified formats.

Element Name	Element Number	User Specification
CWELD	200	ALIGN ELEMID, GRIDID with MSET=ON
CWELDC	117	ELEMID, GRIDID with MSET=OFF
CWELDP	118	ELPAT, PARTPAT

The output record of the CWELD forces is identical to the CBAR output record. The output is in the element coordinate system, see the explanations above how the element coordinate system is constructed. The forces are output at the end points GA and GB, see Figure 3-71.



f_x	Axial force
f_y	Shear force, plane 1
f_z	Shear force, plane 2
m_x	Torque
m_{yA}	Bending moment end A, plane 2
m_{yB}	Bending moment end B, plane 2
m_{zA}	Bending moment end A, plane 1
m_{zB}	Bending moment end B, plane 1

Figure 3-71 CWELD Element Forces

CWELD stress computations are made using above forces and standard beam stress type calculations.

Guidelines and Limitations

Finding the connectivity is not straight forward for a complex topology. The user has an option to do a check out run first. In addition, the user may choose to let the program move GS in case a projection can not be found. Furthermore, the user may choose to reduce the diameter so that the spot weld can be placed near edges and corners. All parameters for the

connectivity search are lumped on the SWLDPRM Bulk Data entry. It is recommended to use the defaults first. If there are problems, it is recommended that the check out option be used next and request diagnostic prints.

The CWELD element is not designed to provide accurate stresses at the spot weld location. The stresses and spot weld forces are mesh dependent. The mesh dependency is more pronounced when the mesh size difference between shell A and B is large, for example, when the element connects a coarse mesh on shell A with a fine mesh on shell B.

- The old CFAST/CSEAM/CWELD formulation and the new CSEAM/CWELD formulation only support geometric nonlinear in SOL 400. Connectors are not supported in SOL106 or SOL129 as nonlinear elements.
- The CSEAM/CWELD elements do not currently support nonlinear material in SOL 400.
- The current CFAST/CSEAM/CWELD do not support thermal loading.

The MSET option on the PWELD entry is only available for the formats ELEMID and GRIDID. The default is MSET=OFF, that means the constraints are eliminated on the element level and there are no external m-set constraints generated. For the formats ELPAT and PARTPAT, all constraints are eliminated on the element level, there are no external constraints generated.

Example

A symmetric hat profile with symmetric mesh, loads, and boundary conditions is used to demonstrate the difference between the ELEMID and the ELPAT formats, see Figure 3-72. For the ELEMID format, it may happen that the connectivity is defined in an unsymmetric manner, if the spot weld locator point GS is on a corner of an element, see Figure 3-73. Then the format ELEMID produces unsymmetric results. With the ELPAT format, even with a user defined unsymmetric connectivity, the internal connectivity is symmetric and the results are symmetric, see the following print out.

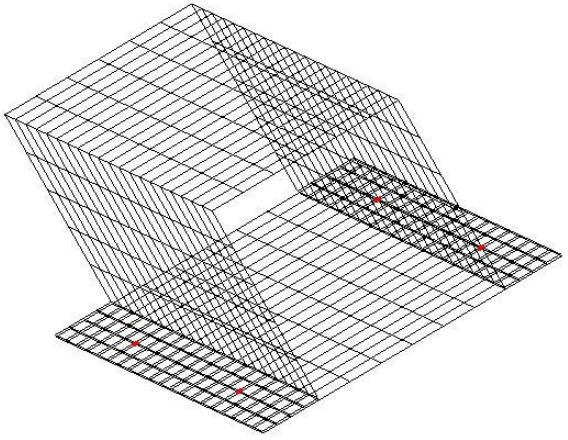


Figure 3-72 Hat Profile

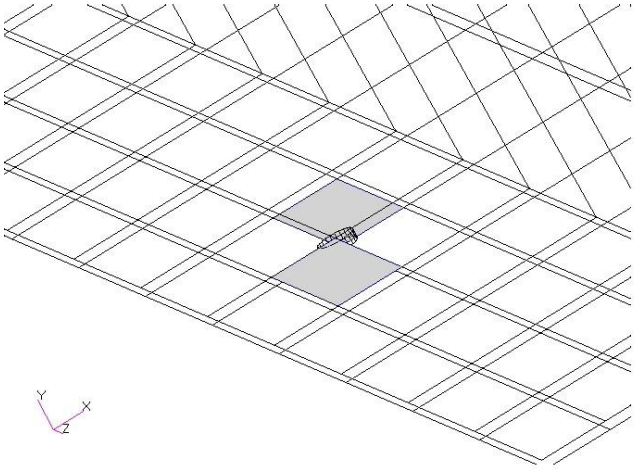


Figure 3-73 Unsymmetric Connectivity for Format ELEMID

The following is the echo of the input for the formats ELEMID and ELPAT:

```
$ Weld definition with format ELEMID
PWELD    10       1        3.
CWELD    761      10       10001    ELEMID
         259      22
CWELD    762      10       10002    ELEMID
         299      62
CWELD    763      10       10003    ELEMID
         639      402
CWELD    764      10       10004    ELEMID
         679      442

$ Weld definition with format ELPAT
PWELD    10       1        3.
CWELD    761      10       10001    ELPAT
         259      22
CWELD    762      10       10002    ELPAT
         299      62
CWELD    763      10       10003    ELPAT
         639      402
CWELD    764      10       10004    ELPAT
         679      442
```

Below is the print out for the element forces. The forces are not symmetric for the format ELEMID, see the first part with the title CWELDC. The forces come out symmetric for the format ELPAT, see the second part with the title CWELDP.

1 HAT SECTION WELDED BEAM, NON-SYM ELEMID OPTION MAY 16, 2003 MSC Nastran 5/15/03 PAGE 16
MZ
0 SUBCASE 3

F O R C E S I N W E L D E L E M E N T S (C W E L D C)

ELEMENT ID	BEND-MOMENT END-A PLANE 1 (MZ)	PLANE 2 (MY)	BEND-MOMENT END-B PLANE 1 (MZ)	PLANE 2 (MY)	- SHEAR - PLANE 1 (FY)	PLANE 2 (FZ)	AXIAL FORCE FX	TORQUE MX
761	-1.422846E-02	1.642372E-01	-8.496036E-03	-3.994170E-01	-5.732428E-03	5.636542E-01	-5.534064E-04	4.985318E-02
762	-1.146957E-02	1.216688E-01	-1.872424E-02	-4.326421E-01	7.254668E-03	5.543109E-01	8.758141E-05	4.411347E-02
763	-2.167158E-02	-1.589983E-01	-1.698017E-02	4.059372E-01	-4.691408E-03	-5.649355E-01	5.010240E-04	4.672464E-02
764	-2.205327E-02	-1.208619E-01	-3.047966E-02	4.344850E-01	8.426391E-03	-5.553468E-01	-1.337890E-04	4.600000E-02

1 HAT SECTION WELDED BEAM, NON-SYM ELPAT OPTION MAY 16, 2003 MSC Nastran 5/15/03 PAGE 16
MZ
0 SUBCASE 3

F O R C E S I N W E L D E L E M E N T S (C W E L D P)

ELEMENT ID	BEND-MOMENT END-A PLANE 1 (MZ)	PLANE 2 (MY)	BEND-MOMENT END-B PLANE 1 (MZ)	PLANE 2 (MY)	- SHEAR - PLANE 1 (FY)	PLANE 2 (FZ)	AXIAL FORCE FX	TORQUE MX
761	-9.314521E-04	1.277454E-01	4.733397E-03	-4.993157E-01	-5.664849E-03	6.270611E-01	-2.275192E-04	5.223577E-02
762	9.314521E-04	1.277454E-01	-4.733397E-03	-4.993157E-01	5.664849E-03	6.270611E-01	2.275192E-04	5.223577E-02
763	-9.314521E-04	-1.277454E-01	4.733397E-03	4.993157E-01	-5.664849E-03	-6.270611E-01	2.275192E-04	5.223577E-02
764	9.314521E-04	-1.277454E-01	-4.733397E-03	4.993157E-01	5.664849E-03	-6.270611E-01	-2.275192E-04	5.223577E-02

Weld stiffening example

The figures below represent a "box" structure of overall physical dimensions of 100cm x 100cm x 100cm. One model is a coarse mesh model and the other a fine mesh model. The vertical center section has the same mesh in both models. Both Models have the same number of CWELD elements (Cluster of blue dots in fringe plots) each of D=5.0 cm at the same spatial locations.

For the coarse mesh the outer horizontal top and bottom panels are 50 cm x 50 cm. The inner horizontal top and bottom panels are 16 cm x 16 cm. The center vertical partition mesh is 20 cm x 20 cm. The outer vertical side panels have mesh 50 cm x 50 cm. The inner vertical side panels have a mesh of 20 cm x 20 cm. The mesh of the horizontal offset panel on the upper right side is 10 cm x 10 cm. For the fine mesh (shown in Figure 3-71c) the inner horizontal top and bottom panels remain a mesh of 16 cm x 16 cm. The outer horizontal top and bottom panels are meshed using CQUAD8 elements at 4 cm x 4 cm. The center vertical partition mesh is 20 cm x 20 cm. The outer vertical side panels have mesh 3.4 cm x 3.4 cm. The inner vertical side panels remain at 20 cm x 20 cm. The mesh of the horizontal offset panel on the upper right side is 10 cm x 10 cm. The mesh of the horizontal offset panel on the upper right side is ~2.6 cm x ~2.6 cm. As shown in Figure 3-62a all the elements contained within the green outlined circle are included in a typical patch of this panel.

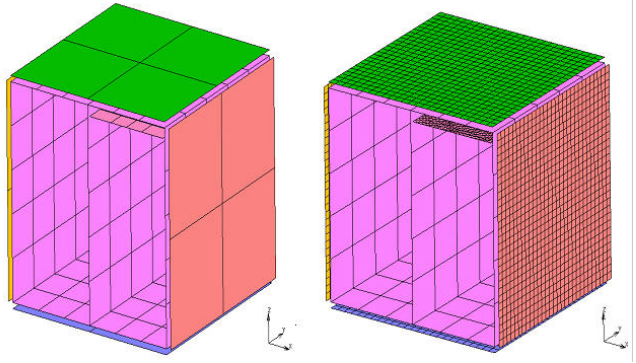


Figure 3-74 (a) Coarse and fine mesh model (b)shows the detail of the fine mesh of the horizontal offset panel.

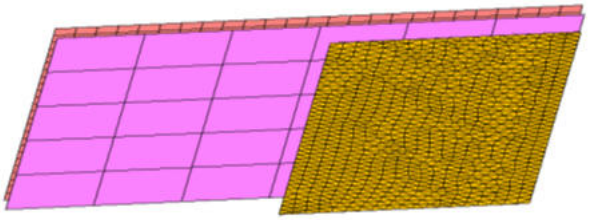


Figure 3-75 Fine mesh horizontal offset panel

As the model mesh is refined there is a well-known tendency for welded models to exhibit a loss of stiffness due to model discretization. This is shown if the next two figures. The coarse mesh model has a maximum displacement of 5.795 cm where the center vertical partition meets the bottom plate. The fine mesh model has a maximum displacement at that location of 13.75 cm at the same edge location.

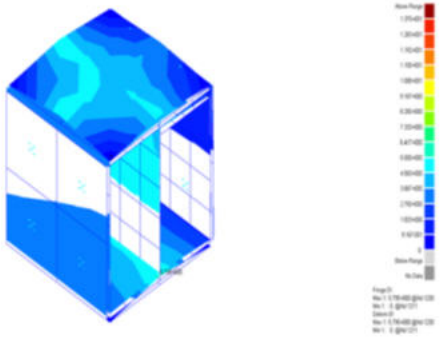


Figure 3-76 Coarse mesh model displacement

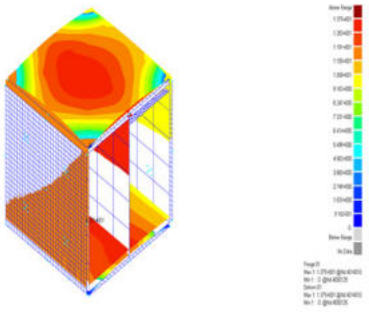
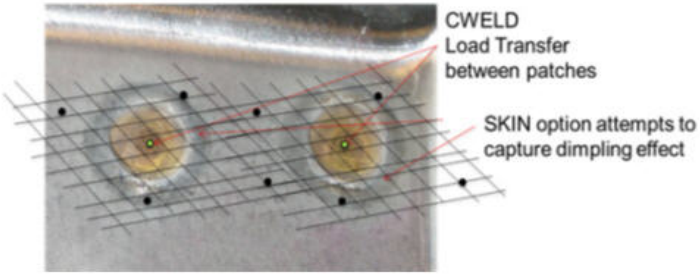


Figure 3-77 Fine mesh model displacement

For design purposes it is desired that the displacement at that location be brought to within 2% of the coarse mesh model. To achieve this the SWLDPRM entry is modified to use a combination of two stiffening techniques available with the CWELD element. This is achieved by modifying the entry SWLDPRM to include the DRATIO and SCLSKIN parameters. The DRATIO factor of 5.95 increases the radius of influence of the CWELDs Dpatch = 5.95 Dweld = 29.75 cm. For patch to patch connection, the user supplied PWELD value of Dweld=5.0 is still used to compute the "beam area and moment of area properties.

"SWLDPRM, PRTSW, 2, NREDIA, 4, DRATIO, 5.95, SCLSKIN, 0.255"

The SKIN parameter initiates a process which measures the discretizing of each CWELD relative to its patch and computes an estimate of the local increase in patch bending resistance which is then applied to the shell element local to the patch. For this model, the associated SWLDPRM parameter SCLSKIN is set to a value of 0.255 and is used to measure the discretization of each patch and compute a "WELD" bending moment of inertia to be applied for the specific patch.



The figure below shows the displacement pattern for this case. It is observed that the displacement of the fine mesh model has a maximum displacement of 5.898 cm where the center vertical partition meets the bottom plate which is within the required 2%.

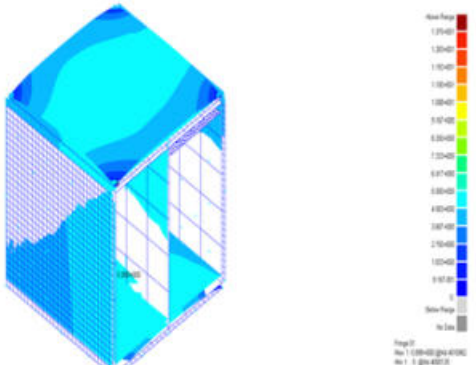


Figure 3-78 Fine mesh model with DRATIO, 5.95 and SKIN, 1 with SCLSKIN, 0.255

For these models, the von Mises stress, in the area of the vertical partition stayed about the same. In the coarse mesh model this was the area of maximum stress. In both fine mesh models the area of maximum stress occurred near the support point of the upper plate furthest from the viewer.

CFAST Element

The CFAST element extends existing weld element capabilities by adding a flexible, user-defined connection between either two surface patches or two shell elements. The PFAST entry gives the option of specifying longitudinal and rotational stiffness, a lumped mass, and damping along a defined orientation.

When used with the new CFDIAGP and CFRANDEL parameters, you can elect to randomly remove a percentage of elements and look at potential failures for the connection. See CFAST Element in the *MSC Nastran Linear Static User's Guide*. The CFAST element has the ability to define the properties of the connector itself. Introducing connector flexibility, mass, and damping extends the uses of the CWELD. The remark section of the bulk data entry PFAST contains many details about this element and the bulk data entry SWLDPRM shows how to modify the elements parameters.

CSEAM Element

The CWELD /CFAST elements have been changed so that there is now a consistent formulation between linear and SOL 400 nonlinear analysis. The details are described above in the Section CWELD Element.

The CFAST element which has been extended to include frequency dependency and material nonlinearity. The PFASTT entry extends to the CFAST element nonlinear load deflection capabilities and limit load fusing ability of the current CBUSH element in SOL400.

PFASTT entry

The CFAST element with property entry PFAST has been extended to have a related PFASTT entry which requires the same PID as the PFAST. The options specified are similar to the existing options on the PBUSHT entry with the exception that the CFAST allows only a single GE value and no B values. In nonlinear analysis the PFASTT with "KN" options support various material nonlinear features similar to the PBUSHT including the "FUSE" option.

Limitations and Guidelines

When using the "FUSE" option in conjunction with geometric nonlinear (LGDISP=1) the fuse stiffness retention "FSR" may need adjustment as fusing to quickly may cause instant divergence while setting to large a FSR value may cause a different convergence path than expected.

The CSEAM element connects surface patches. The seam weld is defined by selecting two surface patches by their property IDs and specifying the width and the thickness of the seam. Extensive error checking and user diagnostics include detection of missing connections for CSEAMs, defined across corners and cutouts. See CSEAM Element in the *MSC Nastran Linear Static User's Guide*.

Mathematical Model to Construct the CSEAM Auxiliary Points

The four auxiliary vertex points of the cross section at start point GS are constructed by the following equations (see Figure 3-79), where t_1^s and t_2^s are tangent vectors of the element coordinate system at start point GS, W is the width of the seam, and T is the thickness of the seam. For a continuous seam, t_1^s and t_2^s vectors are adjusted to a common face for the two consecutive seam elements. Therefore, the four auxiliary grids at the start point S of the ith seam should be coincident with the four auxiliary grids at the end point E of the (i-1)th seam, if both elements have same width and thickness.

$$x_{SA1} = x_S - \frac{W}{2}t_1^S - \frac{T}{2}t_2^S$$

$$x_{SA2} = x_S + \frac{W}{2}t_1^S - \frac{T}{2}t_2^S$$

$$x_{SB1} = x_S - \frac{W}{2}t_1^S + \frac{T}{2}t_2^S$$

$$x_{SB2} = x_S + \frac{W}{2}t_1^S + \frac{T}{2}t_2^S$$

The tangent vector t_2^S is calculated based on the coordinates of the projection points from GS to patch A and patch B, $x_{SA}$and x_{SB}. If these two patches are not too close, then

$$t_2^S = \frac{x_{SB} - x_{SA}}{\|s_{SB} - x_{SA}\|}$$

Otherwise, t_2^S must be determined from the shell normal vectors of patch A and patch B, i.e. n_{SA} and n_{SB}.

If $n_{SA}.n_{SB} > 0$, then

$$t_2^S = \frac{n_{SB} + n_{SA}}{\|n_{SB} + n_{SA}\|}$$

If $n_{SA}.n_{SB} < 0$, then

$$t_2^S = \frac{n_{SB} - n_{SA}}{\|n_{SB} - n_{SA}\|}$$

For t_1^S, if $|n_{SA} \cdot t_2^S| < 0.9$ and $|n_{SB} \cdot t_2^S| < 0.9$ then,

$$t_1^S = \frac{n_{SA} - (t_2^S \cdot n_A)t_2^S}{\|n_{SA} - (t_2^S \cdot n_A)t_2^S\|}$$

$$t_1^S = \frac{e_1 \times n_{SA}}{\|e_1 \times n_{SA}\|}$$

where,

$$e_1 = \frac{x_{SC} - x_{EC}}{\|x_{SC} - x_{EC}\|}$$

$$x_{SC} = \frac{x_{SA} + x_{SB}}{2}$$

$$x_{EC} = \frac{x_{EA} + x_{EB}}{2}$$

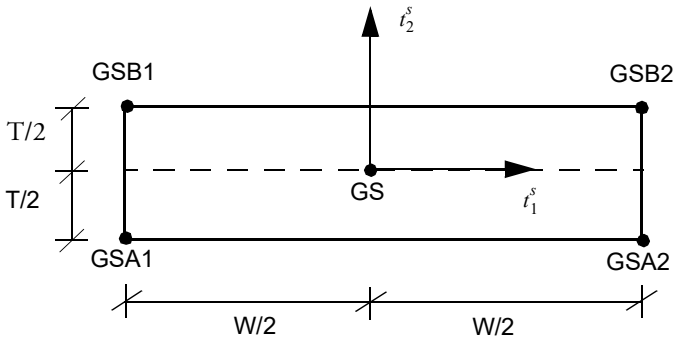


Figure 3-79 Seam Weld Cross Section at Start Point S

The four auxiliary vertex points of the cross section at end point GE are calculated in the same way as that for the start point GS. These eight auxiliary points form an auxiliary HEXA element with the following vertex points.

1	2	3	4	5	6	7	8
GSA1	GSA2	GSB2	GSB1	GEA1	GEA2	GEB2	GEB1

CINTC Element

The CINTC Element is used to connect dissimilar meshes along the edges of finite element mesh subdomains. These subdomains have boundaries usually associated with either two-dimensional shell elements or one dimensional beam elements. This feature is particularly useful for global-local modeling, where a local fine mesh is required in a particular region due to high stress gradients. A set of MPC (Multipoint Constraint) equations are internally generated with the interface boundary grids to enforce the compatibility of displacements and rotations across the interface. See CINTC Element in the *MSC Nastran Linear Static User's Guide.*

Applications of Line Interface Element

Line Interface Element

A line interface element is a computational entity which consists of two or more interface boundaries along a virtual common interface curve. An interface boundary coincides with an edge of a finite element mesh subdomain. Primarily, an interface boundary is defined by a set of sequentially listed grid points, as specified by Bulk Data entry GMBNDC. On each boundary, there are at least two grids for a linear interpolation scheme and three for a quadratic one.

There is a dependency (or master-slave) relationship among the interface boundaries. This dependency relation is used to internally generate MPC equations between the independent and dependent degrees of freedom. At each grid point on a boundary, there are six degrees of freedom, three translational and three rotational. An interface boundary is called the dependent if all the degrees of freedom related to its grids are put in the m-set. An independent boundary is the one that all the degrees of freedom related to its grids are assigned to the n-set. There is only one independent boundary in an interface element.

In terms of element characteristics, an interface element is similar to a rigid or constraint element, such as RBE1 and RSPLINE. Unlike rigid elements, however, its processing is not selectable by Case Control command RIGID. In other words, only the linear elimination method is applied to this element. The degrees of freedom associated to the grids on a dependent interface boundary may not be simultaneously assigned dependent by other rigid, constraint or connection elements, a multipoint constraint or another interface element. On the other hand, the degrees of freedom with the grids on an independent interface boundary can be made dependent by another element or a multipoint constraint. This recursive dependent-independent relationship should be carefully examined to avoid potential conflicts when an interface element interacts with a multipoint constraint, a rigid, constraint or connection element, or another interface element. In addition, the dependent set (m-set) specified by the dependent interface boundary may not be specified on other entries that define mutually exclusive set (See "Degree-of-Freedom Sets (p. 1101) in the *MSC Nastran Quick Reference Guide*. AUTOMSET can be used with the interface element. It is designed to handle potentially ill-conditioned matrix $[R_{mg}]$ by reassigning dependent set (m-set) from the existing MPC equations, based on its rank deficiency. It needs further verifications to see how well the interface element works with AUTOMSET.

Line Interface Boundary

An interface element has two or more boundaries. Each boundary is defined primarily by a set of grid points listed sequentially from the initial grid through the final one, as specified by Bulk Data entry GMBNDC with Field ENTITY=”GRID”. All the boundaries in a line interface element must have their initial grids at the starting end and final grids at the terminal end of the interface.

One of the interface boundaries is specified as the independent by Bulk Data entry CINTC. If there are three or more boundaries in an interface element, the boundaries are internally grouped in pairs by the dependent-independent relationship. There are *N-1* pairs of boundaries where *N* is the total number of boundaries with the interface element. MSC Nastran generates the MPC equations between the dependent and independent grids on the pair of boundaries.

A boundary can be either open, i.e., the initial and final grids are different grids, or closed, i.e., the initial and final grids are the same grid, as shown in Figure 3-80. For a closed boundary, we have GRIDI=GRIDF in GMBNDC. There is a requirement for the minimum number of grids on a boundary. There must be at least two grids for the linear and three for the quadratic interpolation scheme.

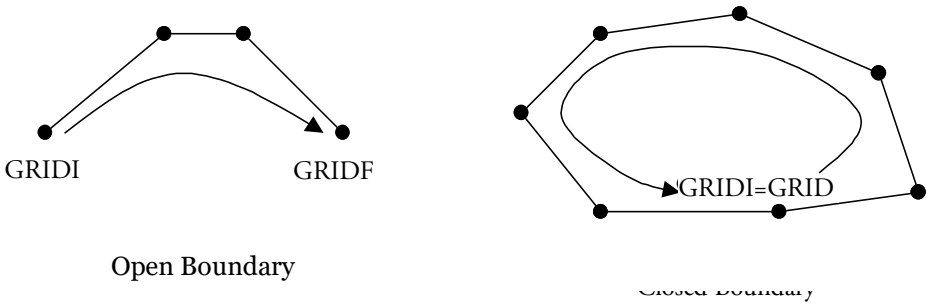


Figure 3-80 Open and Closed Boundaries

Both dependent and independent open boundaries may share either initial or final grids or both of them. By assigning the same end grid to a pair of boundaries, the you have more flexibility in applying the interface element when one interface element connects or intersects with another interface element. This rule, however, may not apply to the closed boundaries.

A pair of interface boundaries can either be matching or non-matching, as shown in Figure 3-81. The non-matching boundaries have at least one dangling end grid. The interface line integration is taken along the common arc-length shared by both boundaries. Redundant dangling grids, as shown in Figure 3-82, are excluded from the processing of interface element.

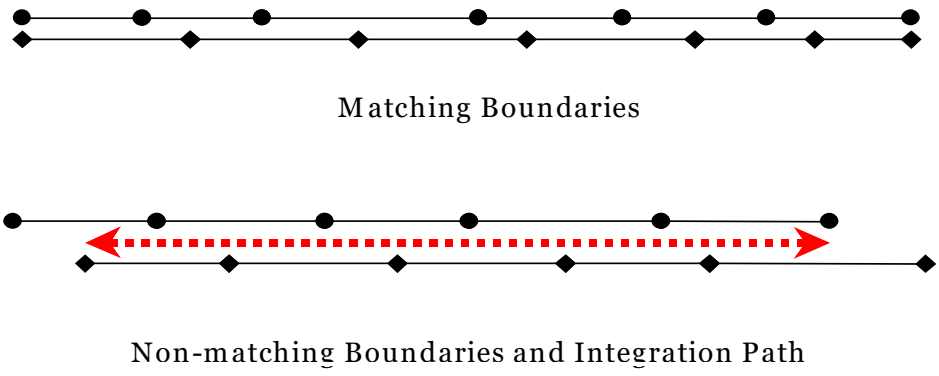


Figure 3-81 Matching and Non-matching Boundaries

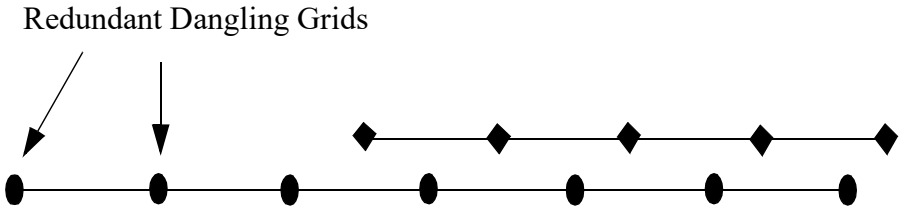


Figure 3-82 Redundant Dangling Grids on Boundary

For an interface element with closed boundaries, special caution must be taken when selecting the initial grids. Since the initial grid can be any one on a closed boundary, it is required that the two initial grids on the dependent-independent closed boundaries be as closed as possible, as shown in Figure 3-83. In other words, a selection of an initial grid being a redundant grid in terms of the other initial grid is not supported. When one of the boundaries is open, this rule does not apply.

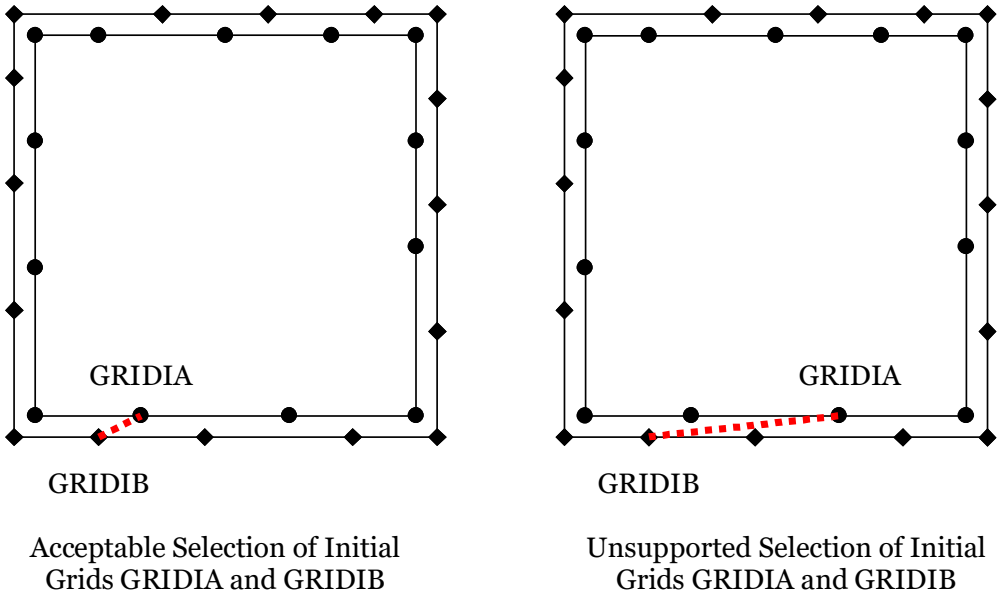


Figure 3-83 Selection of Initial Grids on Closed Boundaries

When the interface consists of curved boundaries, as shown in Figure 3-84, the interface boundary grids should be aligned along the actual line of structural geometry. In this release we have not implemented a capability that enables MSC Nastran to check the integrity of interface boundary geometries. The existence of awry grids in the boundary grid list will lead to some interface malfunction without notice.

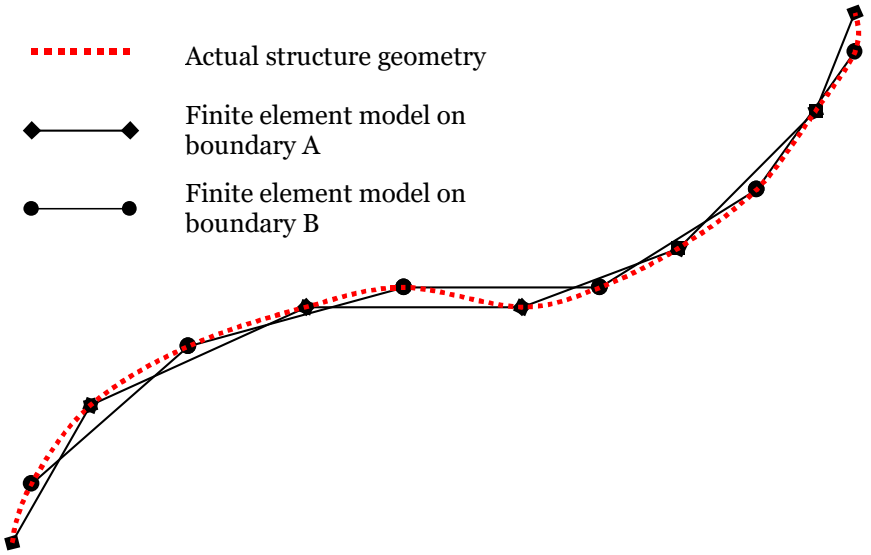


Figure 3-84 Curved Boundaries

Element Geometry Checks

Mathematical models are usually composed of several different types of finite elements depending upon the simulation being performed and the type of structure being investigated. For each of these finite element formulations, thorough evaluations of the geometry presented are performed prior to and during element matrix generation operations. These geometry tests are typically used to determine whether (1) the element geometry is adequate for finite element matrix generation at all, and if so, (2) how close the geometry could be to producing poor formulations. Taking a beam element for example, a fatal geometry condition would be locations of the two end points that result in an element with zero length. An example of the second type of geometry check for a beam would be an excessive ratio of the length of the beam with offset vectors to the length of the beam without them. In the first case, the geometry check is fatal because the finite element cannot be formulated from the given geometry and the simulation cannot be done. In the second case, the geometry check is informational and the simulation analysis can proceed. It is the user's responsibility to inspect these informational messages and determine if the results are acceptable.

Several different basic geometry tests are performed; for example, duplicate grid point location tests, mid-side node location tests, warped element tests, and aspect ratio tests. One or more of these types of tests is performed using the geometry supplied for each finite element. Obviously, not all tests apply to every element type. Each one of the geometry tests is accompanied by a message that can generate up to four lines of informational output for an element if the test tolerance is violated. For models that employ a small number of elements, this amount of output is not overwhelming. However, the size and complexity of mathematical simulation models have continued to grow over time. It is no longer surprising to see models that generate hundreds of thousands of lines of output from these tests alone. In an effort to reduce the amount of output generated by geometry checks, an alternative output for reporting the results of the element geometry tests is available. The results for all tests for a single element are reported on a single line together with a visual indication of which

tests have exceeded their tolerances and the severity of the failure. Default values for all test tolerances are defined such that geometry tests result in the same type of pass/fail decision as previous releases. In a few cases, additional tests are performed. At the same time, the GEOMCHECK Executive statement is available to give users more control over the test tolerances, their severity levels and the number of messages output.

GEOMCHECK Statement

The GEOMCHECK statement provides users with a way to override the default values supplied for all tests. (Refer to the Executive Statement, GEOMCHECK (p. 133) in the *MSC Nastran Quick Reference Guide*.) Except for a very few isolated instances, bad geometry that does not allow the formulation of finite element matrix data is always fatal and is not under user control. In these cases, the informational message formats have not been modified. Every test that fails will produce a message for every element. For the cases where geometry does not prevent finite element matrix generation, additional tests that evaluate various geometry parameters are performed and the results reported to the user. For these tests, the user can control both the test tolerance and the severity level of test failure. Using the CBAR element as an example, there is a test performed that evaluates the ratio of the CBAR's length with offset effects to the length without offset effects. The tolerance for this test is 15%, meaning that if the length with offset is different from the length without the offset by more than fifteen percent, an informational message is issued and element processing continues. The user can change the tolerance if desired. The severity of the test failure can also be changed from informational to fatal if desired. In that case, any CBAR element that exceeds the offset ratio test tolerance will cause the job to stop after element matrix generation has been attempted for all elements.

The GEOMCHECK statement can only be used to modify geometry test activities for the CQUAD4, CQUADR, CTRIA3, CTRIAR, CBAR, CBEAM, CHEXA, CPENTA, and CTETRA finite elements at this time. As stated previously, only informational tests can be affected using the GEOMCHECK statement. Results of geometry generated by MSC Nastran that prevent the generation of a finite element matrix cannot be modified using the GEOMCHECK statement.

Keywords are available that allow control over the number of messages to be generated (MSGLIMIT) as well as the severity of the message (MSGTYPE) and its associated effect on the job. Only a severity of FATAL will cause the job to abort after the element matrix generation module has executed. In this case, the identifier associated with failed tests is **FAIL** and the run will also terminate.

Geometry Test Descriptions

Several different types of geometry tests are performed. For each finite element type, the geometry must be tested to ensure that it is adequate to allow generation of element matrices. No attempt is made here to describe those tests. Rather, tests that optionally evaluate certain characteristics of the geometry are described. These tests fall into several categories depending upon the element type. Thus, not all tests are applicable to every element type. The discussion that follows summarizes the optional geometry tests that are applicable to various element types.

CBAR and CBEAM

The only optional test available for these element types is the offset test. The length of the element offset is compared to the original length of the element. If the ratio of these lengths is greater than the tolerance, an informational message is issued identifying the element and its length with and without the effects of offset.

CQUAD4 and CQUADR

Four optional tests are performed on the quadrilateral shaped elements. These tests are:

1. Interior Angle test: This test evaluates the interior angles measured at each of the four corner grid points. If any one of the four angles exceeds minimum or maximum tolerance levels, an informational message is produced.
2. Taper test: Taper ratio for the quadrilateral element is defined to be the ratio of the area of the triangle formed at each corner grid point to one half the area of the quadrilateral. The largest of the four ratios is compared against the tolerance value. It may be noted that as the ratio approaches 1.0, the shape approaches a rectangle.
3. Skew test: This test evaluates the distortion or "parallelogram shape effect" by measuring the angle between lines that join the midpoints of opposite sides of the element. If the angle exceeds the tolerance, an informational message is produced.
4. Warp test: This test evaluates how far out of plane the four corner grid points are by measuring the distance of each point from a "mean" plane passing through the locations of the four points. The corner points are alternately H units above and H units below this mean plane. If the lengths of the diagonals of the element are denoted by D1 and D2, the warping coefficient is obtained from the equation WC = H / 2(D1+D2). If this value exceeds the tolerance, an informational message is produced.

CTRIA3 and CTRIAR

Only one optional test is performed on the triangular shaped planar elements. It is the same as the interior angle test discussed previously under the CQUAD4 and CQUADR test description.

CHEXA, CPENTA and CTETRA

Five optional tests are performed on the solid element family. Not all tests are applicable to every element type. These tests are:

1. Aspect Ratio test: This test evaluates the ratio of the longest length (edge or height) to the shortest length (edge or height) encountered in the element. If the ratio exceeds the tolerance, an informational message is produced.
2. Edge Point Length Ratio test: This test evaluates the location of the "mid-side" nodes if any are present. The node should be located on the line connecting the two adjacent corner grid points at approximately the mid-way point. This determination is made in two ways. First, the distance of the node from the two corner nodes is measured and if the ratio of the distance from one node to the other exceeds the tolerance, an informational message is issued. Next, the angles between the lines connecting the mid-side node to its two adjacent corner nodes, and the line connecting the corner nodes themselves, is evaluated. If either of the angles exceeds 30 degrees, an informational message is issued.
3. Integration Point Jacobian Determinant test: This test evaluates the determinant of the Jacobian at each integration point. If it is zero, or changes sign from integration point to integration point, an informational message is issued. Note that detection of a zero value should always be fatal because element matrices calculated from such geometry rarely produce satisfactory analysis results.

4. Warped Face test: This test is applicable to CHEXA and CPENTA elements. It evaluates the warping coefficient of the quadrilateral faces of such elements in a manner similar to that discussed previously under the CQUAD4 element warping test description. If the warping coefficient exceeds the tolerance, an informational message is issued.
5. Grid Point Jacobian Determinant test: This test is applicable to CTETRA elements only. The test is the same as that done for the integration point test discussed previously except it uses the location of the corner grid points to perform the test. If any determinant is zero or changes sign, an informational message is produced.

Sample Output:

The following output from a small example provides a sample of the new message formats that are produced for geometry tests that can be controlled using the GEOMCHECK statement. For this output, most of the default test tolerances were modified so that the effect could be observed in the tolerance information lines of the messages that were produced. Also note that a small table is generated at the end of all messages that summarizes the number of tests that actually exceeded the tolerance value for each element type as well as a list of the elements that produced the worst violations. The summary table itself can be produced by using the keyword SUMMARY on the GEOMCHECK statement.

```
*** USER INFORMATION MESSAGE 7555 (GMTSTD)
     FINITE ELEMENT GEOMETRY CHECK RESULTS EXCEED TOLERANCE LEVELS FOR THE FOLLOWING ELEMENTS.
     User Action: Use the GEOMCHECK (Executive Control Statement) keyword=value to change tolerance values if desired.
     A MINIMUM OF        100 OFFSET LENGTH RATIO  TOLERANCE LIMIT VIOLATIONS WILL BE IDENTIFIED AND INDICATED BY xxxx.


TOLERANCE LIMIT IS:   BAR_/BEAM_OFF(SET) =      .15 (xxxx = LIMIT VIOLATED)
 ELEMENT TYPE    ID       LENGTH W/O OFFSET  LENGTH W/ OFFSET   OFFSET LENGTH RATIO
    BAR          7101        1.00000E+00        3.66367E+00         2.66 xxxx


 *** USER INFORMATION MESSAGE 7555 (GMTSTD)
     FINITE ELEMENT GEOMETRY CHECK RESULTS EXCEED TOLERANCE LEVELS FOR THE FOLLOWING ELEMENTS.
     A MINIMUM OF          4 OFFSET LENGTH RATIO  TOLERANCE LIMIT VIOLATIONS WILL BE IDENTIFIED AND INDICATED BY WARN.


     TOLERANCE LIMIT IS:   BAR_/BEAM_OFF(SET) =     -.02 (WARN = LIMIT VIOLATED)
 ELEMENT TYPE    ID       LENGTH W/O OFFSET  LENGTH W/ OFFSET   OFFSET LENGTH RATIO
    BEAM         8101        1.00000E+00        1.10372E+01        10.04 WARN
    BEAM         8102        1.00000E+00        1.00000E+00          .00 WARN

 *** USER INFORMATION MESSAGE 7555 (EHEXGD)
     FINITE ELEMENT GEOMETRY CHECK RESULTS EXCEED TOLERANCE LEVELS FOR THE FOLLOWING ELEMENTS.
     A MINIMUM OF        100 EDGE LENGTH ASPECT RATIO (AR)    TOLERANCE LIMIT VIOLATIONS WILL BE IDENTIFIED AND INDICATED BY xxxx.
     A MINIMUM OF        100 EDGE NODE POINT LENGTH RATIO (ER)TOLERANCE LIMIT VIOLATIONS WILL BE IDENTIFIED AND INDICATED BY xxxx.
                             EDGE ANGLE VALUES ARE INDICATED BY AN * AFTER THE VALUE.  THE TOLERANCE LIMIT IS 30 DEGREES.
     ALL NEG/ZERO DET(JACOBIAN) VALUE (DJ)TOLERANCE LIMIT VIOLATIONS WILL BE IDENTIFIED AND INDICATED BY FAIL.
     A MINIMUM OF        100 WARPED FACE COEFFICIENT          TOLERANCE LIMIT VIOLATIONS WILL BE IDENTIFIED AND INDICATED BY xxxx.

     TOLERANCE LIMITS ARE:  HEX_AR =   300.00, HEX_EPLR =    .40, HEX_DETJ =    .01, HEX_WARP =    .65 (xxxx = LIMIT VIOLATED)
                          LONGEST       SHORTEST         ASPECT           EDGE POINT      MIN. JACOBIAN       FACE WARP
 ELEMENT TYPE    ID        EDGE          EDGE             RATIO         LENGTH RATIO       DETERMINANT       COEFFICIENT
    HEXA         8602       4.00          1.00             4.00       N/A- ONLY 8 NODE          .92              .52 xxxx
    HEXA         8620       4.00          1.00             4.00               .60xxxx          .50             1.00

*** USER INFORMATION MESSAGE 7555 (GMTSTD)
     FINITE ELEMENT GEOMETRY CHECK RESULTS EXCEED TOLERANCE LEVELS FOR THE FOLLOWING ELEMENTS.
     A MINIMUM OF        100 SKEW ANGLE (SA)      TOLERANCE LIMIT VIOLATIONS WILL BE IDENTIFIED AND INDICATED BY xxxx.
     A MINIMUM OF        100 MIN INT. ANGLE (IA)  TOLERANCE LIMIT VIOLATIONS WILL BE IDENTIFIED AND INDICATED BY xxxx.
     A MINIMUM OF        100 MAX INT. ANGLE (IA)  TOLERANCE LIMIT VIOLATIONS WILL BE IDENTIFIED AND INDICATED BY xxxx.
     A MINIMUM OF        100 WARPING FACTOR (WF)  TOLERANCE LIMIT VIOLATIONS WILL BE IDENTIFIED AND INDICATED BY xxxx.
     A MINIMUM OF        100 TAPER RATIO (TR)     TOLERANCE LIMIT VIOLATIONS WILL BE IDENTIFIED AND INDICATED BY xxxx.

TOLERANCE LIMITS ARE:  SA =  30.00, IA(MIN) =  30.00, IA(MAX) = 150.00, WF =    .05, TR =    .50 (xxxx = LIMIT VIOLATED)
 ELEMENT TYPE    ID          SKEW ANGLE      MIN INT. ANGLE    MAX INT. ANGLE    WARPING FACTOR      TAPER RATIO
    QUAD4         105        29.74 xxxx        29.05 xxxx       150.95 xxxx          .00               .05
    QUAD4         106        28.39 xxxx        27.76 xxxx       152.24 xxxx          .00               .05
    QUAD4         107        27.15 xxxx        26.57 xxxx       153.43 xxxx          .00               .05
    QUAD4         108        26.00 xxxx        25.46 xxxx       154.54 xxxx          .00               .05
    QUAD4         109        24.94 xxxx        24.44 xxxx       155.56 xxxx          .00               .05
    QUAD4        2105        29.74 xxxx        29.05 xxxx       150.95 xxxx          .00               .05
    QUAD4        2106        28.39 xxxx        27.76 xxxx       152.24 xxxx          .00               .05
    QUAD4        2107        27.15 xxxx        26.57 xxxx       153.43 xxxx          .00               .05
    QUAD4        2108        26.00 xxxx        25.46 xxxx       154.54 xxxx          .00               .05
    QUAD4        2109        24.94 xxxx        24.44 xxxx       155.56 xxxx          .00               .05


*** USER INFORMATION MESSAGE 7555 (GMTSTD)
     FINITE ELEMENT GEOMETRY CHECK RESULTS EXCEED TOLERANCE LEVELS FOR THE FOLLOWING ELEMENTS.
     A MINIMUM OF        100 SKEW ANGLE (SA)      TOLERANCE LIMIT VIOLATIONS WILL BE IDENTIFIED AND INDICATED BY xxxx.
     A MINIMUM OF        100 MIN INT. ANGLE (IA)  TOLERANCE LIMIT VIOLATIONS WILL BE IDENTIFIED AND INDICATED BY xxxx.
     A MINIMUM OF        100 MAX INT. ANGLE (IA)  TOLERANCE LIMIT VIOLATIONS WILL BE IDENTIFIED AND INDICATED BY xxxx.
     A MINIMUM OF        100 WARPING FACTOR (WF)  TOLERANCE LIMIT VIOLATIONS WILL BE IDENTIFIED AND INDICATED BY xxxx.
     A MINIMUM OF        100 TAPER RATIO (TR)     TOLERANCE LIMIT VIOLATIONS WILL BE IDENTIFIED AND INDICATED BY xxxx.
```

```
TOLERANCE LIMITS ARE:  SA =  30.00, IA(MIN) =  30.00, IA(MAX) = 150.00, WF =    .05, TR =    .50 (xxxx = LIMIT VIOLATED)
 ELEMENT TYPE    ID          SKEW ANGLE      MIN INT. ANGLE    MAX INT. ANGLE    WARPING FACTOR      TAPER RATIO
    QUADR        1105        29.74 xxxx        29.05 xxxx       150.95 xxxx          .00               .05
    QUADR        1106        28.39 xxxx        27.76 xxxx       152.24 xxxx          .00               .05
    QUADR        1107        27.15 xxxx        26.57 xxxx       153.43 xxxx          .00               .05
    QUADR        1108        26.00 xxxx        25.46 xxxx       154.54 xxxx          .00               .05
    QUADR        1109        24.94 xxxx        24.44 xxxx       155.56 xxxx          .00               .05


 *** USER INFORMATION MESSAGE 7555 (ETETGD)
     FINITE ELEMENT GEOMETRY CHECK RESULTS EXCEED TOLERANCE LEVELS FOR THE FOLLOWING ELEMENTS.
     A MINIMUM OF          4 EDGE LENGTH TO HEIGHT  RATIO (AR)TOLERANCE LIMIT VIOLATIONS WILL BE IDENTIFIED AND INDICATED BY WARN.
     A MINIMUM OF          4 EDGE NODE POINT LENGTH RATIO (ER)TOLERANCE LIMIT VIOLATIONS WILL BE IDENTIFIED AND INDICATED BY WARN.
                             EDGE ANGLE VALUES ARE INDICATED BY AN * AFTER THE VALUE.  THE TOLERANCE LIMIT IS 30 DEGREES.
     A MINIMUM OF          4 NEG/ZERO DET(JACOBIAN) VALUE (DJ)TOLERANCE LIMIT VIOLATIONS WILL BE IDENTIFIED AND INDICATED BY WARN.
     A MINIMUM OF          4 NEG DET(JAC) AT VERTEX NODE  (DG)TOLERANCE LIMIT VIOLATIONS WILL BE IDENTIFIED AND INDICATED BY WARN.


     TOLERANCE LIMITS ARE:  TET_AR =   351.00, TET_EPLR =      .45, TET_DETJ =   -.01, TET_DETG =   -.02 (xxxx = LIMIT VIOLATED)
                          LONGEST       SHORTEST          ASPECT           EDGE POINT      MIN. JACOBIAN     MIN. DET(JAC)
 ELEMENT TYPE    ID        EDGE          HEIGHT            RATIO          LENGTH RATIO       DETERMINANT       AT VERTEX
    TETRA        6601       1.41           .58             2.45               .25 WARN          .14              -.20 WARN


E L E M E N T   G E O M E T R Y   T E S T   R E S U L T S   S U M M A R Y
                                    TOTAL NUMBER OF TIMES TOLERANCES WERE EXCEEDED
                                ASPECT/       MINIMUM       MAXIMUM     SURFACE/FACE                    EDGE POINT     JACOBIAN
     ELEMENT TYPE  SKEW ANGLE  TAPER RATIO  INTER. ANGLE  INTER. ANGLE   WARP FACTOR  OFFSET RATIO  LENGTH RATIO  DETERMINANT
         BAR           N/A          N/A          N/A          N/A          N/A            0            N/A           N/A
         BEAM          N/A          N/A          N/A          N/A          N/A            0            N/A           N/A
         HEXA          N/A            0          N/A          N/A            1          N/A              1             0
         PENTA         N/A            0          N/A          N/A            0          N/A              1             0
         QUAD4          10            0           10           10            0          N/A            N/A           N/A
         QUADR           5            0            5            5            0          N/A            N/A           N/A
         TETRA         N/A            0          N/A          N/A          N/A          N/A              1             1
         TRIA3           0          N/A          N/A            0          N/A          N/A            N/A           N/A
         TRIAR           0          N/A          N/A            0          N/A          N/A            N/A           N/A

         N/A IN THE ABOVE TABLE INDICATES TESTS THAT ARE NOT APPLICABLE TO THE ELEMENT TYPE AND WERE NOT PERFORMED.
         FOR ALL ELEMENTS WHERE GEOMETRY TEST RESULTS HAVE EXCEEDED TOLERANCES,

         HEXA     ELEMENT ID     8620 PRODUCED SMALLEST EDGE COS(ANGLE)  OF        .60 (TOLERANCE =         .87).
         HEXA     ELEMENT ID     8602 PRODUCED SMALLEST FACE WARP FACTOR OF        .52 (TOLERANCE =         .65).
         PENTA    ELEMENT ID     3915 PRODUCED SMALLEST EDGE COS(ANGLE)  OF        .78 (TOLERANCE =         .87).
         QUAD4    ELEMENT ID      109 PRODUCED SMALLEST SKEW ANGLE       OF      24.94 (TOLERANCE =       30.00).
         QUAD4    ELEMENT ID      109 PRODUCED SMALLEST INTERIOR ANGLE   OF      24.44 (TOLERANCE =       30.00).
         QUAD4    ELEMENT ID      109 PRODUCED LARGEST  INTERIOR ANGLE   OF     155.56 (TOLERANCE =      150.00).
         QUADR    ELEMENT ID     1109 PRODUCED SMALLEST SKEW ANGLE       OF      24.94 (TOLERANCE =       30.00).
         QUADR    ELEMENT ID     1109 PRODUCED SMALLEST INTERIOR ANGLE   OF      24.44 (TOLERANCE =       30.00).
         QUADR    ELEMENT ID     1109 PRODUCED LARGEST  INTERIOR ANGLE   OF     155.56 (TOLERANCE =      150.00).
         TETRA    ELEMENT ID     6601 PRODUCED LARGEST  EDGE POINT LR    OF        .25 (TOLERANCE =         .45).
         TETRA    ELEMENT ID     6601 PRODUCED SMALLEST VERTEX DET(JAC) -UWM 6828- OF       -.20 (TOLERANCE =        -.02).
```

Introduction

A long-standing key feature of MSC Nastran is the ability of the PSHELL(, PCOMP), PBAR(, PBARL), PBEAM(, PBEAML, PBCOMP), PROD(, CONROD), PBEND, PSHEAR, PTUBE, PCONEAX, and PRAC2D property entries to include nonstructural mass. The five Bulk Data entries -- NSM, NSM1, NSML, NSML1, and NSMADD -- distribute nonstructural mass by element lists or specific property lists associated with the above fourteen property entries.

Theory

NSML and NSML1 Entries

The NSML and NSML1 entries compute a nonstructural mass coefficient value for "area" elements identified by an element list or property list or both by the relationship:

$$\text{NSM_value} = \frac{\text{Lumped_non_structural_mass_value}}{\Sigma_{\text{elements}} AREA}$$

The NSML and NSML1 entries compute a nonstructural mass coefficient value for "line" elements identified by an element list or property list or both by the relationship:

$$\text{NSM_value} = \frac{\text{Lumped_non_structural_mass_value}}{\Sigma_{\text{elements}} LENGTH}$$

In the above two expressions, AREA corresponds to the area of each individual area element (CQUAD4 for example) and LENGTH corresponds to the length of each individual length element (CBAR for example).

The NSML and NSML1 entries are then converted internally to NSM and NSM1 entries.

Inputs

Case Control Commands

NSM Case Control command allows for the selection of different NSM sets for residual and superelements.

Bulk Data Entries

NSM and its alternate form NSM1 allows the user to allocate an NSM_value to selected sets of elements.

NSML and its alternate form NSML1 allows the user to allocate and smear a lumped nonstructural mass value to selected sets of elements.

NSMADD allows the user to form combinations of NSM, NSM1, NSML, and NSML1 sets and sum their results to selected sets of elements.

Guidelines and Limitations

The following requirements are associated with the new capability:

- The NSML and NSML1 entries cannot mix "area" and "line" elements on the same entry.
- Area elements are: CQUAD4, CQUAD8, CQUADR, CTRIA3, CTRIA6, CTRIAR, CSHEAR, and CRAC2D. (The CCONEAX is a stand alone element.)
- Line elements are: CBAR, CBEAM, CBEND, CTUBE, CROD, and CONROD.
- The CCONEAX is excluded from the NSML and NSML1 entries.

Example

The following example partial input demonstrates the form of the new entries and the expected result:

Partial Input File

```
SOL108 $
```

```
CEND
  DISPL = ALL
     .
     .
     .
  ELSUM = ALL
  NSM = 1222
BEGIN BULK
$
     .
     .
     .
$
$        2        3        4        5        6        7        8        9        0
NSML     9        PBEAM    211      .026     311      .028     411      .030
NSML     9        PBEND    21       .4869    31       .4869
NSML     10       PRAC2D   1        .132
NSML1    10       PBCOMP   .126     21       THRU     41
NSML1    11       PBEAML   .138     212      312      412
NSML     11       PROD     12       .17
NSML     12       CONROD   12       .19
NSML     12       PSHELL   51       4.5
NSML1    13       PSHELL   132.     ALL
NSML1    13       PSHELL   107.8    50       THRU     52
NSML1    14       PSHELL   7.5      57
NSML1    14       ELEMENT  .594     20       30       40       201      202
         301      302      401      402
NSML1    15       PSHEAR   .264     25       THRU     45       BY       10
NSML     15       PBAR     22       .22
NSML1    15       PBARL    .52      ALL
NSML     16       ELEMENT  101      .077
NSML     16       ELEMENT  27       1.021    13       .67
NSML     17       ELEMENT  14       .41
NSML     17       PTUBE    14       .43
NSML1    18       ELEMENT  16.2     41       81234
NSML1    18       PSHEAR   1.476    ALL
NSML     18       PSHEAR   1        1.476
NSML1    19       ELEMENT  .112     201      402
NSML1    19       ELEMENT  4.2      3123     6134     134
$
NSMADD   1222     9        11       12       14       13       15
         16     1718       19     2010       12
$
$           FINAL                                PROPERTY
$         DISTRIBUTED                             VALUE
$  ELM       NSM     ID      PID       PROPERTY     NSM
$ ----       ---     --      ---       -------     ----
$ CBAR       .061    25       22       - PBAR      .013
$ CBAR       .039    26       26       - PBARL     .013
$ CBEAM      .065    20       21       - PBCOMP    .011
$            .065
$ CBEAM      .065    30       31       - PBCOMP    .011
$            .065
$ CBEAM      .065    40       41       - PBCOMP    .011
$            .065
$ CBEAM      .085    201      211      - PBEAM     .011,.012,.014
$            .086 is1
```

```
$            .088 is2 all others zero
$            .088 end B
$ CBEAM      .058   301     311     - PBEAM    .011,.012,.014
$            .059 is1
$            .061 is2 all others zero
$            .061 end B
$ CBEAM      .059   401     411     - PBEAM    .011,.012,.014
$            .060 is1
$            .062 is2 all others zero
$            .062 end B
$ CBEAM      .0671  202     212     - PBEAML   .0111,.0112,.0113,.0114,.0115
$            .0672 is1
$            .0673 is2
$            .0674 is3 all others zero
$            .0675 is4
$            .0675 end B
$ CBEAM      .0671  302     312     - PBEAML   .0111,.0112,.0113,.0114,.0115
$            .0672 is1
$            .0673 is2
$            .0674 is3 all others zero
$            .0675 is4
$            .0675 end B
$ CBEAM      .0951  402     412     - PBEAML   .0111,.0112,.0113,.0114,.0115
$            .0952 is1
$            .0953 is2
$            .0954 is3 all others zero
$            .0955 is4
$            .0955 end B
$ CBEND      .109   27      21      - PBEND    .013
$ CBEND      .044   37      31      - PBEND    .013
$ CBEND      .013   47      41      - PBEND    .013
$ CONROD     .032   12      1       - CONROD   .013
$ CQUAD4     .257   41      57      - PCOMP    .013
$ CQUAD4     .178   41234   50      - PSHELL   .013
$ CQUAD4     .178   4123456 52      - PSHELL   .013
$ CQUAD4     .193   4123456751      - PSHELL   .013
$ CQUAD8     .178   81      50      - PSHELL   .013
$ CQUAD8     .259   81234   50      - PSHELL   .013
$ CQUAD8     .178   8123456 52      - PSHELL   .013
$ CQUAD8     .193   8123456751      - PSHELL   .013
$ CQUADR     .178   1       50      - PSHELL   .013
$ CQUADR     .178   1234    50      - PSHELL   .013
$ CQUADR     .178   123456  52      - PSHELL   .013
$ CQUADR     .193   1234567 51      - PSHELL   .013
$ CROD       .097   13      12      - PROD     .013
$ CSHEAR     .158   25      1       - PSHEAR   .013
$ CSHEAR     .158   35      1       - PSHEAR   .013
$ CSHEAR     .158   45      1       - PSHEAR   .013
$ CTRIA3     .206   3123    50      - PSHELL   .013
$ CTRIA3     .178   3134    50      - PSHELL   .013
$ CTRIA6     .178   6123    50      - PSHELL   .013
$ CTRIA6     .206   6134    50      - PSHELL   .013
$ CTRIAR     .178   123     50      - PSHELL   .013
$ CTRIAR     .206   134     50      - PSHELL   .013
$ CTUBE      .097   14      14      - PTUBE    .013
$
$
```

```
      .
      .
      .
$
ENDDATA
```

Typical Element mass summary is given as:

```
                                    E L E M E N T   P R O P E R T Y   S U M M A R Y
ELEMENT TYPE = BAR
      ID          MID          LENGTH          AREA          VOLUME       STRUCT.MASS    NON-STR.MASS     TOTAL MASS       TM*WTMASS
      25            1    1.000000E+01  2.000000E-02  2.000000E-01  0.000000E+00  6.100000E-01  6.100000E-01  6.100000E-01
      26            1    1.000000E+01  3.141593E-02  3.141593E-01  0.000000E+00  3.900000E-01  3.900000E-01  3.900000E-01
SUBTOTAL MASS =                                                    0.000000E+00  1.000000E+00  1.000000E+00  1.000000E+00
ELEMENT TYPE = BEAM
      ID          MID          LENGTH          AREA          VOLUME       STRUCT.MASS    NON-STR.MASS     TOTAL MASS       TM*WTMASS
      20            1    2.000000E+00  1.200800E-02  2.401600E-02  0.000000E+00  1.300000E-01  1.300000E-01  1.300000E-01
      30            1    2.000000E+00  1.200800E-02  2.401600E-02  0.000000E+00  1.300000E-01  1.300000E-01  1.300000E-01
      40            1    2.000000E+00  1.200800E-02  2.401600E-02  0.000000E+00  1.300000E-01  1.300000E-01  1.300000E-01
     201            1    2.000000E+00  2.000000E-02  4.000000E-02  0.000000E+00  1.700000E-01  1.700000E-01  1.700000E-01
     202            1    2.000000E+00  1.256637E+00  2.513275E+00  0.000000E+00  1.342000E-01  1.342000E-01  1.342000E-01
     301            1    2.000000E+00  2.000000E-02  4.000000E-02  0.000000E+00  1.160000E-01  1.160000E-01  1.160000E-01
     302            1    2.000000E+00  1.256637E+00  2.513275E+00  0.000000E+00  1.342000E-01  1.342000E-01  1.342000E-01
     401            1    2.000000E+00  2.000000E-02  4.000000E-02  0.000000E+00  1.180000E-01  1.180000E-01  1.180000E-01
     402            1    2.000000E+00  1.256637E+00  2.513275E+00  0.000000E+00  1.902000E-01  1.902000E-01  1.902000E-01
SUBTOTAL MASS =                                                    0.000000E+00  1.252600E+00  1.252600E+00  1.252600E+00
ELEMENT TYPE = BEND
      ID          MID          LENGTH          AREA          VOLUME       STRUCT.MASS    NON-STR.MASS     TOTAL MASS       TM*WTMASS
      27            1    1.570796E+01  1.000000E+00  1.570796E+01  0.000000E+00  1.712104E+00  1.712104E+00  1.712104E+00
      37            1    1.570796E+01  1.000000E+00  1.570796E+01  0.000000E+00  6.911035E-01  6.911035E-01  6.911035E-01
      47            1    1.570796E+01  1.000000E+00  1.570796E+01  0.000000E+00  2.042035E-01  2.042035E-01  2.042035E-01
SUBTOTAL MASS =                                                    0.000000E+00  2.607411E+00  2.607411E+00  2.607411E+00
ELEMENT TYPE = HEXA
      ID          MID                                VOLUME       STRUCT.MASS    NON-STR.MASS     TOTAL MASS       TM*WTMASS
    1001         1003                          8.000000E+03  6.400000E+01  0.000000E+00  6.400000E+01  6.400000E+01
SUBTOTAL MASS =                                                    6.400000E+01  0.000000E+00  6.400000E+01  6.400000E+01
ELEMENT TYPE = CONROD
      ID          MID          LENGTH          AREA          VOLUME       STRUCT.MASS    NON-STR.MASS     TOTAL MASS       TM*WTMASS
      12           51    1.000000E+01  1.100000E+00  1.100000E+01  6.600000E-01  3.200000E-01  9.800000E-01  9.800000E-01
SUBTOTAL MASS =                                                    6.600000E-01  3.200000E-01  9.800000E-01  9.800000E-01
ELEMENT TYPE = QUAD4
      ID          MID      THICKNESS          AREA          VOLUME       STRUCT.MASS    NON-STR.MASS     TOTAL MASS       TM*WTMASS
      41    100000057    2.000000E-01  1.000000E+02  2.000000E+01  5.040089E+01  2.570000E+01  7.610089E+01  7.610089E+01
   41234           51    1.000000E+00  1.000000E+02  1.000000E+02  6.000000E+00  1.780000E+01  2.380000E+01  2.380000E+01
  4123456          51    1.000000E+00  1.000000E+02  1.000000E+02  6.000000E+00  1.780000E+01  2.380000E+01  2.380000E+01
  41234567         51    1.000000E+00  1.000000E+02  1.000000E+02  6.000000E+00  1.930000E+01  2.530000E+01  2.530000E+01
SUBTOTAL MASS =                                                    6.840089E+01  8.060000E+01  1.490009E+02  1.490009E+02
ELEMENT TYPE = QUAD8
      ID          MID      THICKNESS          AREA          VOLUME       STRUCT.MASS    NON-STR.MASS     TOTAL MASS       TM*WTMASS
      81           51    1.000000E+00  1.000000E+02  1.000000E+02  6.000000E+00  1.780000E+01  2.380000E+01  2.380000E+01
   81234           51    1.000000E+00  1.000000E+02  1.000000E+02  6.000000E+00  2.590000E+01  3.190000E+01  3.190000E+01
  8123456          51    1.000000E+00  1.000000E+02  1.000000E+02  6.000000E+00  1.780000E+01  2.380000E+01  2.380000E+01
  81234567         51    1.000000E+00  1.000000E+02  1.000000E+02  6.000000E+00  1.930000E+01  2.530000E+01  2.530000E+01
SUBTOTAL MASS =                                                    2.400000E+01  8.080000E+01  1.048000E+02  1.048000E+02
ELEMENT TYPE = QUADR
      ID          MID      THICKNESS          AREA          VOLUME       STRUCT.MASS    NON-STR.MASS     TOTAL MASS       TM*WTMASS
       1           51    1.000000E+00  1.000000E+02  1.000000E+02  6.000000E+00  1.780000E+01  2.380000E+01  2.380000E+01
    1234           51    1.000000E+00  1.000000E+02  1.000000E+02  6.000000E+00  1.780000E+01  2.380000E+01  2.380000E+01
  123456           51    1.000000E+00  1.000000E+02  1.000000E+02  6.000000E+00  1.780000E+01  2.380000E+01  2.380000E+01
 1234567           51    1.000000E+00  1.000000E+02  1.000000E+02  6.000000E+00  1.930000E+01  2.530000E+01  2.530000E+01
SUBTOTAL MASS =                                                    2.400000E+01  7.270000E+01  9.670000E+01  9.670000E+01
ELEMENT TYPE = ROD
      ID          MID          LENGTH          AREA          VOLUME       STRUCT.MASS    NON-STR.MASS     TOTAL MASS       TM*WTMASS
      13           51    1.000000E+01  1.100000E+00  1.100000E+01  6.600000E-01  9.700000E-01  1.630000E+00  1.630000E+00
SUBTOTAL MASS =                                                    6.600000E-01  9.700000E-01  1.630000E+00  1.630000E+00
ELEMENT TYPE = SHEAR
      ID          MID      THICKNESS          AREA          VOLUME       STRUCT.MASS    NON-STR.MASS     TOTAL MASS       TM*WTMASS
      25            1    1.000000E+00  4.000000E+00  4.000000E+00  0.000000E+00  3.860000E-01  3.860000E-01  3.860000E-01
      35            1    1.000000E+00  4.000000E+00  4.000000E+00  0.000000E+00  3.860000E-01  3.860000E-01  3.860000E-01
      45            1    1.000000E+00  4.000000E+00  4.000000E+00  0.000000E+00  3.860000E-01  3.860000E-01  3.860000E-01
     125            1    1.000000E+00  4.000000E+00  4.000000E+00  0.000000E+00  7.900000E-01  7.900000E-01  7.900000E-01
     135            1    1.000000E+00  4.000000E+00  4.000000E+00  0.000000E+00  7.900000E-01  7.900000E-01  7.900000E-01
     145            1    1.000000E+00  4.000000E+00  4.000000E+00  0.000000E+00  7.900000E-01  7.900000E-01  7.900000E-01
SUBTOTAL MASS =                                                    0.000000E+00  3.528000E+00  3.528000E+00  3.528000E+00
ELEMENT TYPE = TRIA3
      ID          MID      THICKNESS          AREA          VOLUME       STRUCT.MASS    NON-STR.MASS     TOTAL MASS       TM*WTMASS
     3123           51    1.000000E+00  5.000000E+01  5.000000E+01  3.000000E+00  1.030000E+01  1.330000E+01  1.330000E+01
     3134           51    1.000000E+00  5.000000E+01  5.000000E+01  3.000000E+00  8.900000E+00  1.190000E+01  1.190000E+01
SUBTOTAL MASS =                                                    6.000000E+00  1.920000E+01  2.520000E+01  2.520000E+01
ELEMENT TYPE = TRIA6
      ID          MID      THICKNESS          AREA          VOLUME       STRUCT.MASS    NON-STR.MASS     TOTAL MASS       TM*WTMASS
     6123           51    1.000000E+00  5.000000E+01  5.000000E+01  3.000000E+00  8.900000E+00  1.190000E+01  1.190000E+01
     6134           51    1.000000E+00  5.000000E+01  5.000000E+01  3.000000E+00  1.030000E+01  1.330000E+01  1.330000E+01
SUBTOTAL MASS =                                                    6.000000E+00  1.920000E+01  2.520000E+01  2.520000E+01
ELEMENT TYPE = TRIAR
      ID          MID      THICKNESS          AREA          VOLUME       STRUCT.MASS    NON-STR.MASS     TOTAL MASS       TM*WTMASS
     123           51    1.000000E+00  5.000000E+01  5.000000E+01  3.000000E+00  8.900000E+00  1.190000E+01  1.190000E+01
     134           51    1.000000E+00  5.000000E+01  5.000000E+01  3.000000E+00  1.030000E+01  1.330000E+01  1.330000E+01
SUBTOTAL MASS =                                                    6.000000E+00  1.920000E+01  2.520000E+01  2.520000E+01
ELEMENT TYPE = TUBE
      ID          MID          LENGTH          AREA          VOLUME       STRUCT.MASS    NON-STR.MASS     TOTAL MASS       TM*WTMASS
      14           51    1.000000E+01  3.141593E-01  3.141593E+00  1.884956E-01  9.700000E-01  1.158496E+00  1.158496E+00
SUBTOTAL MASS =                                                    1.884956E-01  9.700000E-01  1.158496E+00  1.158496E+00
TOTAL    MASS =                                                    1.999094E+02  3.023480E+02  5.022575E+02  5.022575E+02
```

ELSUM Case Control Command

Introduction

The ELSUM Case Control command controls the generation of a printed table of informational properties for the various element types present in the input data file. The information produced includes element measures (e.g. length, thickness, area, volume) and mass property data. The information is grouped according to the finite element type. Several new options have been added to the ELSUM command giving users more control over the amount of output produced and another view of the data grouped by element property.

Method and Theory

For each element present in the model, physical metrics appropriate to the element type are computed. These metrics may include length, thickness, area, and volume. Mass properties are computed for supported element types. For the CONM1 element, the mass is taken to be the average of the M11, M22 and M33 values on the CONM1 Bulk Data entry. In addition, references to material properties and element properties are gathered for each element. For bar, beam and shell elements, an attempt is made to determine whether the property reference is to a basic property entry, or has been derived from one of the available alternate input entries (PBARL, PBEAML, PBCOMP, PCOMP). If it is determined that the property is derived, a character (L for PBARL or PBEAML, P for PBCOMP or PCOMP) is appended to the property identification number on output. For the case of the element type grouping, each element type is summarized in turn. For each element type, information for each element is displayed. Subtotals are generated for mass property information. For the case of the property type grouping, each property type referenced by an element type is summarized in turn. All of the property ids referenced within the type are summarized in ascending property id order. Within this summary, information for each element referencing the property id is displayed. Various subtotals are accumulated and displayed.

Inputs

The ELSUM Case Control command has been enhanced to give the user new controls over the amount and ordering of the element summary information produced. The general format of the command is:

$$\text{ELSUM}([\text{EID, PID, BOTH, PIDSUM, EIDSUM}]) = \left\{ \begin{matrix} \text{ALL} \\ \text{n} \\ \text{NONE} \end{matrix} \right\}$$

See ELSUM (case) (p. 322) in the *MSC Nastran Quick Reference Guide* for a complete description of the ELSUM command. The EID and PID keywords are used to select the particular grouping desired for the element summary output. The EID keyword requests grouping by element type while the PID keyword requests grouping by element property type. The BOTH keyword requests both groupings. The PIDSUM keyword requests that only mass property totals be output for the PID grouping. The EIDSUM keyword requests that only mass property totals be output for the EID grouping.

Outputs

The ELSUM Case Control command causes printed output to be generated. Figure 3-85 is an example of the EID grouping output. Figure 3-86 is an example of the PID grouping output. If PIDSUM or EIDSUM were used, then only the subtotals and totals information would be present in the figures.

```
                                        E L E M E N T   P R O P E R T Y   S U M M A R Y     (BY ELEMENT TYPE / ID)

 ELEMENT TYPE = BAR
   ELEM ID   PROP ID     MATL ID     LENGTH         AREA         VOLUME      STRUCT.MASS NON-STR.MASS     TOTAL MASS      TM*WTMASS
      3401      3401           1  1.00000E+00  1.00000E+00  1.00000E+00  5.00000E-02  5.00000E-01    5.50000E-01    5.50000E-01
      3403      3403           1  3.00000E+00  1.00000E+00  3.00000E+00  1.50000E-01  1.50000E+00    1.65000E+00    1.65000E+00
      3410      3403           1  2.00000E+00  1.00000E+00  2.00000E+00  1.00000E-01  1.00000E+00    1.10000E+00    1.10000E+00
     17101     17103 L         1  1.00000E+00  1.01034E+00  1.01034E+00  5.05168E-02  0.00000E+00    5.05168E-02    5.05168E-02
     17103     17103 L         1  3.00000E+00  1.01034E+00  3.03101E+00  1.51550E-01  0.00000E+00    1.51550E-01    1.51550E-01
     17110     17103 L         1  2.00000E+00  1.01034E+00  2.02067E+00  1.01034E-01  0.00000E+00    1.01034E-01    1.01034E-01
                                                                          -----------  -----------    -----------    -----------
  SUBTOTAL MASS FOR ALL BAR                                               6.03101E-01  3.00000E+00    3.60310E+00    3.60310E+00

      .
      .
      .

 ELEMENT TYPE = BEAM
   ELEM ID   PROP ID     MATL ID     LENGTH         AREA         VOLUME      STRUCT.MASS NON-STR.MASS     TOTAL MASS      TM*WTMASS
       200       200           1  1.00000E+00  1.00000E+02  1.00000E+02  5.00000E+00  0.00000E+00    5.00000E+00    5.00000E+00
     17000     17000 L         1  1.00000E+00  9.99987E+01  9.99987E+01  4.99994E+00  0.00000E+00    4.99994E+00    4.99994E+00
                                                                          -----------  -----------    -----------    -----------
  SUBTOTAL MASS FOR ALL BEAM                                              9.99994E+00  0.00000E+00    9.99994E+00    9.99994E+00

      .
      .
      .

 ELEMENT TYPE = PENTA
   ELEM ID   PROP ID     MATL ID                              VOLUME      STRUCT.MASS NON-STR.MASS     TOTAL MASS      TM*WTMASS
      6801      6801           2                          4.00000E-02  2.00000E-03  0.00000E+00    2.00000E-03    2.00000E-03
                                                                          -----------  -----------    -----------    -----------
  SUBTOTAL MASS FOR ALL PENTA                                             2.00000E-03  0.00000E+00    2.00000E-03    2.00000E-03

      .
      .
      .

 ELEMENT TYPE = QUAD4
   ELEM ID   PROP ID     MATL ID    THICKNESS       AREA         VOLUME      STRUCT.MASS NON-STR.MASS     TOTAL MASS      TM*WTMASS
      3301      3301           1  1.00000E+00  1.00000E+00  1.00000E+00  5.00000E-02  5.00000E-01    5.50000E-01    5.50000E-01
      3313      3313           1  6.12009E-01  6.36864E+00  3.89766E+00  1.94883E-01  1.91715E+00    2.11203E+00    2.11203E+00
      3321      3321 P      3321  1.00000E+00  1.00000E+02  1.00000E+02  1.00000E+02  0.00000E+00    1.00000E+02    1.00000E+02
     17503     17503           1  1.00000E+00  1.00000E+00  1.00000E+00  5.00000E-02  0.00000E+00    5.00000E-02    5.00000E-02
                                                                          -----------  -----------    -----------    -----------
  SUBTOTAL MASS FOR ALL QUAD4                                             1.00295E+02  2.41715E+00    1.02712E+02    1.02712E+02

      .
      .
      .

 ELEMENT TYPE = ROD
   ELEM ID   PROP ID     MATL ID     LENGTH         AREA         VOLUME      STRUCT.MASS NON-STR.MASS     TOTAL MASS      TM*WTMASS
       101       101           1  1.00000E+00  1.00000E+00  1.00000E+00  5.00000E-02  5.00000E-01    5.50000E-01    5.50000E-01
       107       101           1  1.00000E+00  1.00000E+00  1.00000E+00  5.00000E-02  5.00000E-01    5.50000E-01    5.50000E-01
                                                                          -----------  -----------    -----------    -----------
  SUBTOTAL MASS FOR ALL ROD                                               1.00000E-01  1.00000E+00    1.10000E+00    1.10000E+00

      .
      .
      .

 ELEMENT TYPE = TETRA
   ELEM ID   PROP ID     MATL ID                              VOLUME      STRUCT.MASS NON-STR.MASS     TOTAL MASS      TM*WTMASS
      3901      3901           1                          5.77350E-01  2.88675E-02  0.00000E+00    2.88675E-02    2.88675E-02
                                                                          -----------  -----------    -----------    -----------
  SUBTOTAL MASS FOR ALL TETRA                                             2.88675E-02  0.00000E+00    2.88675E-02    2.88675E-02

      .
      .
      .

 TOTAL MASS FOR ALL SUPPORTED ELEMENT TYPES                               1.38647E+02  1.53344E+01    1.53981E+02    1.53981E+02
```

Figure 3-85 Sample Element Summary by Element Type Output

```
                                                E L E M E N T   P R O P E R T Y   S U M M A R Y     (BY PROPERTY TYPE / ID)


 PROPERTY TYPE = PBAR,    ID =     3401
*****************************************************************************************
   ELEM ID     ELEM TYPE    MATL ID     LENGTH          AREA          VOLUME     STRUCT.MASS  NON-STR.MASS     TOTAL MASS      TM*WTMASS
      3401      BAR             1  1.00000E+00  1.00000E+00  1.00000E+00  5.00000E-02  5.00000E-01    5.50000E-01    5.50000E-01
                                                                            -----------  -----------    -----------    -----------
 SUBTOTAL MASS FOR ALL BAR      ELEMENTS    FOR PBAR,    ID =      3401      5.00000E-02  5.00000E-01    5.50000E-01    5.50000E-01

      .
      .
      .

PROPERTY TYPE = PBEAM,   ID =      200     *****************************************************************************************
   ELEM ID     ELEM TYPE    MATL ID     LENGTH          AREA          VOLUME     STRUCT.MASS  NON-STR.MASS     TOTAL MASS      TM*WTMASS
      200       BEAM            1  1.00000E+00  1.00000E+02  1.00000E+02  5.00000E+00  0.00000E+00    5.00000E+00    5.00000E+00
                                                                            -----------  -----------    -----------    -----------
 SUBTOTAL MASS FOR ALL BEAM     ELEMENTS    FOR PBEAM,   ID =       200      5.00000E+00  0.00000E+00    5.00000E+00    5.00000E+00

      .
      .
      .

PROPERTY TYPE = PSOLID, ID =     3901     *****************************************************************************************
   ELEM ID     ELEM TYPE    MATL ID                          VOLUME     STRUCT.MASS  NON-STR.MASS     TOTAL MASS      TM*WTMASS
      3901      TETRA           1                         5.77350E-01  2.88675E-02  0.00000E+00    2.88675E-02    2.88675E-02
                                                                        -----------  -----------    -----------    -----------
 SUBTOTAL MASS FOR ALL SOLID    ELEMENTS    FOR PSOLID, ID =      3901      2.88675E-02  0.00000E+00    2.88675E-02    2.88675E-02

      .
      .
      .

PROPERTY TYPE = (NONE)                   *****************************************************************************************
   ELEM ID     ELEM TYPE    MATL ID                                       STRUCT.MASS  NON-STR.MASS     TOTAL MASS      TM*WTMASS
      2901      CONM1           0                                      1.00000E+00  0.00000E+00    1.00000E+00    1.00000E+00
      2904      CONM1           0                                      2.20000E+00  0.00000E+00    2.20000E+00    2.20000E+00
                                                                       -----------  -----------    -----------    -----------
 SUBTOTAL MASS FOR ALL CONM1    ELEMENTS                               3.20000E+00  0.00000E+00    3.20000E+00    3.20000E+00

      .
      .
      .

PROPERTY TYPE = PSHELL, ID =     3301     *****************************************************************************************
   ELEM ID     ELEM TYPE    MATL ID    THICKNESS       AREA          VOLUME     STRUCT.MASS  NON-STR.MASS     TOTAL MASS      TM*WTMASS
      3301      QUAD4           1  1.00000E+00  1.00000E+00  1.00000E+00  5.00000E-02  5.00000E-01    5.50000E-01    5.50000E-01
                                                                            -----------  -----------    -----------    -----------
 SUBTOTAL MASS FOR ALL SHELL    ELEMENTS    FOR PSHELL, ID =      3301      5.00000E-02  5.00000E-01    5.50000E-01    5.50000E-01

      .
      .
      .

                                                                             ===========  ===========    ===========
===========
 TOTAL MASS FOR ALL SUPPORTED ELEMENT TYPES                                 1.38647E+02  1.53344E+01    1.53981E+02    1.53981E+02
                                                                            ===========  ===========    ===========    ===========
```

Figure 3-86 Sample Element Summary by Element Property Type Output

Guidelines and Limitations

- The ELSUM command keywords can be used in any combination. However, if both EID and EIDSUM are present, EIDSUM takes precedence over EID and only the mass property totals summary will be generated for the grouping by element type. This is also true for the PID and PIDSUM keywords. PIDSUM takes precedence over PID and only the mass property totals will be generated for the grouping by element property type.

- Only the following element types produce mass property information: CBAR, CBEAM, CBEND, CHEXA, CMASSi, CONM1, CONM2, CONROD, CPENTA, CQUAD4, CQUAD8, CQUADR, CRAC2D, CRAC3D, CROD, CSHEAR, CTETRA, CTRIA3, CTRIA6, CTRIAR, CTRIAX6 and CTUBE.
- The ELSUM command is ignored in heat transfer solution sequences.

Example

No new example is presented for this enhanced ELSUM Case Control command since any existing input data file available can be used to demonstrate the new features. One simply inserts an ELSUM command with the desired keywords into the Case Control Section of the input file. Element summary information will be generated and placed on the standard .f06 output file.

4 Material Properties

- Isotropic Material 262
- Orthotropic Material 264
- Anisotropic Material 265
- Nonuniform Material and Composite Properties for Shells 267
- Hyperelastic Material 296

The material properties used in structural analysis are described in this section. Material properties for heat transfer are described in the *MSC Nastran Thermal Analysis User's Guide*. The material definitions describe the stiffness or constitutive relationships, strength, density, and thermal expansion characteristics of the material. Material Bulk Data entries can be classified by type as isotropic (MAT1), orthotropic (MAT3, MAT8), or anisotropic (MAT2, MAT9).

The materials input data types MAT1, MAT2, MAT8 and MAT9 entries can be used to describe temperature-dependent material when used in conjunction with MATT1, MATT2, MATT8 and MATT9 entries, respectively. In addition, the MATS1 can be used with the MAT1entry to describe nonlinear material behavior.

Also included in this section is a description of composite materials (PBCOMP, PCOMP) which are used in the formulation of equivalent material constitutive relations. The method used to model composites is also used in the analysis of materials with nonhomogeneous cross sections. Composite material properties are used in the analysis of the composite (nonuniform) beams and shells.

The advanced materials used in nonlinear solution, SOL 400, are described in the *MSC Nastran Nonlinear User's Guide*, Chapter 10: Materials. The advanced materials used in explicit nonlinear solution, SOL 700, are described in the *MSC Nastran Explicit Nonlinear (SOL 700) User's Guide*, Materials (Ch. 7).

Table 4-1 summarizes the relationships between elements and material input data types.

Table 4-1 Correspondence Between Element and Material Type

Element	Material Type					Hyper-elastic	Stress Dependent	Temperature Dependent	Composite
Type	MAT1	MAT2	MAT3	MAT8	MAT9	MATHP	MATS1	MATTi	
CBAR	X							X	
CBEAM	X						X	X	X
CBEAM3	X						X	X	
CBEND	X							X	
CCONEAX	X	X						X	
CFAST	X								
CRAC2D	X							X	
CRAC3D	X							X	
CHEXA	X				X	X	X	X	
CPENTA	X				X	X	X	X	
CQUAD						X			
CQUAD4	X	X		X		X	X	X	X
CQUAD8	X	X		X		X		X	X
CQUADR	X	X		X				X	
CQUADX						X			

Table 4-1 Correspondence Between Element and Material Type (continued)

Element	Material Type					Hyper-elastic	Stress Dependent	Temperature Dependent	Composite
Type	MAT1	MAT2	MAT3	MAT8	MAT9	MATHP	MATS1	MATTi	
CROD	X						X	X	
CSEAM	X								
CSHEAR	X							X	
CTETRA	X				X	X	X	X	
CTRIA3	X	X		X		X	X	X	X
CTRIA6	X	X		X		X		X	X
CTRIAR	X	X		X				X	
CTRIAX						X			
CTRIAX6	X		X						
CTUBE	X						X	X	
CWELD	X								

Isotropic Material

The MAT1 entry is used to describe the isotropic material properties for general, plain stress, plain strain, and axisymmetric isotropic material behavior. This section describes the MAT1 Bulk Data entries used to define isotropic materials and how that definition relates to the constitutive relations. See Table 4-1 for a summary of elements that use MAT1 entries.

The general three-dimensional isotropic material stress-strain constitutive relationship is:

$$\begin{Bmatrix}\sigma_x\\ \sigma_y\\ \sigma_z\\ \tau_{xy}\\ \tau_{yz}\\ \tau_{zx}\end{Bmatrix} = \begin{bmatrix} \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} & \frac{\nu E}{(1+\nu)(1-2\nu)} & \frac{\nu E}{(1+\nu)(1-2\nu)} & 0 & 0 & 0 \\ & \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} & \frac{\nu E}{(1+\nu)(1-2\nu)} & 0 & 0 & 0 \\ & & \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} & 0 & 0 & 0 \\ & & & \frac{E}{2(1+\nu)} & 0 & 0 \\ & \text{symmetric} & & & \frac{E}{2(1+\nu)} & 0 \\ & & & & & \frac{E}{2(1+\nu)} \end{bmatrix}$$

$$\left\{\begin{Bmatrix}\varepsilon_x\\ \varepsilon_y\\ \varepsilon_z\\ \gamma_{xy}\\ \gamma_{yz}\\ \gamma_{zx}\end{Bmatrix} - (T - T_{ref})\begin{Bmatrix}A\\ A\\ A\\ 0\\ 0\\ 0\end{Bmatrix}\right\}$$

where $\{\sigma\}$, $\{\varepsilon\}$, and $\{A\}$ are the stress, strain, and thermal expansions coefficients, respectively. E, ν, G, and $(T - T_{ref})$ are Young's modulus, Poisson's ratio, shear modulus, and the change in element temperature used to calculate initial element thermal expansions.

For MAT1, $G = E/21 + \nu$ is the relationship used to determine the isotropic elastic material constants when only two of the three are input on the MAT1 entry. When all three are given, G will not be required to satisfy this relationship (see Table 4-2 for the use of the elastic constants with respect to the finite elements usage).

Isotropic material constitutive relationship for plane stress behavior is:

$$\begin{Bmatrix}\sigma_x\\ \sigma_y\\ \tau_{xy}\end{Bmatrix} = \begin{bmatrix} \frac{E}{1-\nu^2} & \frac{\nu E}{1-\nu^2} & 0 \\ \frac{\nu E}{1-\nu^2} & \frac{\nu E}{1-\nu^2} & 0 \\ 0 & 0 & \frac{E}{2(1+\nu)} \end{bmatrix} \left\{\begin{Bmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}\end{Bmatrix} - (T - T_{ref})\begin{Bmatrix}A\\ A\\ 0\end{Bmatrix}\right\}$$

Isotropic material constitutive relationship for plane strain behavior is:

$$\begin{Bmatrix}\sigma_x\\ \sigma_y\\ \tau_{xy}\end{Bmatrix} = \begin{bmatrix}\frac{E(1-\nu)}{(1+\nu)(1-2\nu)} & \frac{\nu E}{(1+\nu)(1-2\nu)} & 0\\ \frac{\nu E}{(1+\nu)(1-2\nu)} & \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} & 0\\ 0 & 0 & \frac{E}{2(1+\nu)}\end{bmatrix}\left\{\begin{Bmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}\end{Bmatrix} - (T-T_{ref})(1+\nu)\begin{Bmatrix}A\\ A\\ 0\end{Bmatrix}\right\}$$

When MAT1 is used for transverse shear

$$\begin{pmatrix}\sigma_{zx}\\ \sigma_{zy}\end{pmatrix} = \begin{bmatrix}G & 0\\ 0 & G\end{bmatrix}\begin{pmatrix}\gamma_{zx}\\ \gamma_{zy}\end{pmatrix}$$

Table 4-2 MAT1 – Material Property Usage Versus Element Types

Element Entry	E	NU	G
CROD CBEAM CBAR	Extension and Bending	Not Used	Torsion Transverse Shear
CQUADi CTRIAi CCONEAX	Membrane and Bending		Transverse Shear
CSHEAR	Not Used		Shear
CRAC2D	All Terms		Not Used
CHEXA CPENTA CTETRA CRAC3D	All Terms		Not Used
CTRIAX6	Radial, Axial, Circumferential	All Coupled Ratios	Shear

Orthotropic Material

This section describes the MAT3 and MAT8 Bulk Data entries which define orthotropic materials and their corresponding constitutive relations. The MAT3 and MAT8 entries are used to describe the orthotropic material properties for two-dimensional and three-dimensional element orthotropic material behavior. See Table 4-1 for a summary of elements that use MAT3 and MAT8 entries.

The MAT3 entry is used to define orthotropic material matrix for axisymmetric solid elements in the stress-strain relationship as given by:

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_\theta \\ \varepsilon_z \\ \gamma_{zx} \end{Bmatrix} = \begin{bmatrix} \frac{1}{E_x} & \frac{-\nu_{\theta x}}{E_\theta} & \frac{-\nu_{zx}}{E_z} & 0 \\ \frac{-\nu_{x\theta}}{E_x} & \frac{1}{E_\theta} & \frac{-\nu_{z\theta}}{E_z} & 0 \\ \frac{-\nu_{xz}}{E_x} & \frac{-\nu_{\theta z}}{E_\theta} & \frac{1}{E_z} & 0 \\ 0 & 0 & 0 & \frac{1}{G_{zx}} \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_\theta \\ \sigma_z \\ \tau_{zx} \end{Bmatrix} + (T - T_{ref}) \cdot \begin{Bmatrix} A_x \\ A_\theta \\ A_z \\ 0 \end{Bmatrix}$$

where $\{\sigma\}$, $\{\varepsilon\}$, and $\{A\}$ are the stress, strain, and thermal expansions coefficients, respectively. E, ν, G, and $(T - T_{ref})$ are Young's modulus, Poisson's ratio, shear modulus, and the change in element temperature used to calculate initial element thermal expansions.

Where $\nu_{x\theta}/E_x = \nu_{\theta z}/E_\theta$, $\nu_{zx}/E_z = \nu_{xz}/E_x$, and $\nu_{\theta z}/E_\theta = \nu_{z\theta}/E_z$ must be satisfied to preserve symmetry. Only the $\nu_{\theta x}$, ν_{xz}, and $\nu_{z\theta}$ terms are used in the input.

The MAT8 entry is used to define orthotropic material matrix for shell elements in the stress-strain relationship as given by:

$$\begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \\ \gamma_{1z} \\ \gamma_{2z} \end{Bmatrix} = \begin{bmatrix} \frac{1}{E_1} & \frac{-\nu_{21}}{E_2} & 0 & 0 & 0 \\ \frac{-\nu_{12}}{E_1} & \frac{1}{E_2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{G_{12}} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{1z}} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{2z}} \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \\ \tau_{1z} \\ \tau 2z \end{Bmatrix} + (T - T_{ref}) \begin{Bmatrix} A_1 \\ A_2 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

where $\{\sigma\}$, $\{\varepsilon\}$, and $\{A\}$ are the stress, strain, and thermal expansions coefficients, respectively. E, ν, G, and $(T - T_{ref})$ are Young's modulus, Poisson's ratio, shear modulus, and the change in element temperature used to calculate initial element thermal expansions.

Where $\nu_{21}/E_2 = \nu_{12}/E_1$ must be satisfied to preserve symmetry. Only the ν_{12} term is used in the input.

When either G_{1z} or G_{2z} is input as zero, it implies that the corresponding shear modulus is infinite. When this is the case, the transverse shear flexibility effects are neglected in the shell formulation.

Anisotropic Material

This section describes the MAT2 and MAT9 entries that define anisotropic materials and their corresponding constitutive relations. The MAT2 and MAT9 entries are used to describe the anisotropic material properties for two-dimensional and three-dimensional element anisotropic material behavior. See Table 4-1 for a summary of elements that use MAT2 and MAT9 entries.

The MAT2 entry used to define two-dimensional anisotropic constitutive stress-strain relationship using the anisotropic material matrix, [G]. The stress-strain relationship is given by:

$$\begin{Bmatrix}\sigma_1\\ \sigma_2\\ \tau_{12}\end{Bmatrix} = \begin{bmatrix}G_{11} & G_{12} & G_{13}\\ G_{12} & G_{22} & G_{23}\\ G_{13} & G_{23} & G_{33}\end{bmatrix}\left\{\begin{Bmatrix}\varepsilon_1\\ \varepsilon_2\\ \gamma_{12}\end{Bmatrix} - (T - T_{ref})\begin{Bmatrix}A_1\\ A_2\\ A_{12}\end{Bmatrix}\right\}$$

where $\{\sigma\}$, $\{\varepsilon\}$, and $\{A\}$ are the stress, strain, and thermal expansions coefficients, respectively. E, ν, G, and $(T - T_{ref})$ are Young's modulus, Poisson's ratio, shear modulus, and the change in element temperature used to calculate initial element thermal expansions.

The MAT2 entry is also used to define two-dimensional orthotropic constitutive stress-strain relation using the anisotropic material matrix, $[G]$. The stress-strain relationship is given by:

$$\begin{Bmatrix}\sigma_1\\ \sigma_2\\ \tau_{12}\end{Bmatrix} = \begin{bmatrix}G_{11} & G_{12} & 0\\ G_{12} & G_{22} & 0\\ 0 & 0 & G_{33}\end{bmatrix}\left\{\begin{Bmatrix}\varepsilon_1\\ \varepsilon_2\\ \gamma_{12}\end{Bmatrix} - (T - T_{ref})\begin{Bmatrix}A_1\\ A_2\\ 0\end{Bmatrix}\right\}$$

where the elastic coefficients are given by engineering constants used to describe orthotropic material properties as:

$$G_{11} = \frac{E_1}{(1 - \nu_{12}\nu_{21})}$$

$$G_{22} = \frac{E_2}{(1 - \nu_{12}\nu_{21})}$$

$$G_{12} = \frac{\nu_{12}E_2}{(1 - \nu_{12}\nu_{21})} = \frac{\nu_{21}E_1}{(1 - \nu_{12}\nu_{21})}$$

When MAT2 is used for transverse shear

$$\begin{pmatrix}\sigma_{zx}\\ \sigma_{zy}\end{pmatrix} = \begin{bmatrix}G_{11} & G_{12}\\ G_{12} & G_{22}\end{bmatrix}\begin{pmatrix}\gamma_{zx}\\ \gamma_{zy}\end{pmatrix}$$

G_{33} must be set to 0.

The MAT9 entry is used to define general three-dimensional anisotropic constitutive stress-strain relationship using the anisotropic material matrix, $[G]$, the stress-strain constitutive relationship is:

$$\begin{Bmatrix}\sigma_x\\ \sigma_y\\ \sigma_z\\ \tau_{xy}\\ \tau_{yz}\\ \tau_{zx}\end{Bmatrix} = \begin{bmatrix} G_{11} & G_{12} & G_{13} & G_{14} & G_{15} & G_{16} \\ & G_{22} & G_{23} & G_{24} & G_{25} & G_{26} \\ & & G_{33} & G_{34} & G_{35} & G_{36} \\ & & & G_{44} & G_{45} & G_{46} \\ & & & & G_{55} & G_{56} \\ \text{symmetric} & & & & & G_{66} \end{bmatrix} \left\{ \begin{Bmatrix}\varepsilon_x\\ \varepsilon_y\\ \varepsilon_z\\ \gamma_{xy}\\ \gamma_{yz}\\ \gamma_{zx}\end{Bmatrix} - (T - T_{ref}) \begin{Bmatrix} A_1\\ A_2\\ A_3\\ A_4\\ A_5\\ A_6 \end{Bmatrix} \right\}$$

where $\{\sigma\}$, ε, and $\{A\}$ are the stress, strain and thermal expansions coefficients, respectively, and E, ν, G, and $(T - T_{ref})$ are Young's modulus, Poisson's ratio, shear modulus, and the change in element temperature used to calculate initial element thermal expansions.

The MAT9 entry is also used to define three-dimensional orthotropic constitutive stress-strain relation using the anisotropic material matrix $[G]$. The stress-strain relationship is given by:

$$\begin{Bmatrix}\sigma_x\\ \sigma_y\\ \sigma_z\\ \tau_{xy}\\ \tau_{yz}\\ \tau_{zx}\end{Bmatrix} = \begin{bmatrix} G_{11} & G_{12} & G_{13} & 0 & 0 & 0 \\ & G_{22} & G_{23} & 0 & 0 & 0 \\ & & G_{33} & 0 & 0 & 0 \\ & & & G_{44} & 0 & 0 \\ & & & & G_{55} & 0 \\ \text{symmetric} & & & & & G_{66} \end{bmatrix} \left\{ \begin{Bmatrix}\varepsilon_x\\ \varepsilon_y\\ \varepsilon_z\\ \gamma_{xy}\\ \gamma_{yz}\\ \gamma_{zx}\end{Bmatrix} - (T - T_{ref}) \begin{Bmatrix} A_1\\ A_2\\ A_3\\ 0\\ 0\\ 0 \end{Bmatrix} \right\}$$

Nonuniform Material and Composite Properties for Shells

The CQUAD4, CQUAD8, CTRIA3, and CTRIA6 surface elements can be used to model laminated composites in finite element models. Laminated composites may be conceptually viewed as a "stack" of laminae with different orientations of the principal material directions in the individual lamina. An exploded view of three cross-ply laminated plates is illustrated in Figure 4-1. The n-laminae (n = 1,2,3,4) of each of the three configurations are normal to the z-axis of the indicated coordinate system and the 1- and 2-axes appended to the individual lamina denote principal material axis directions. The directions of the principal material axes of each lamina alternate as implied by the use of the word "cross-ply" to describe the configuration. The xy-plane of the coordinate axes is defined in the geometric middle plane of the laminae.

An entire stack of laminae may be modeled with a single plate or shell element because the geometry of the plate/shell is thin and often the material properties of the "stack" are completely reflected in the matrices of elastic moduli for the element. These matrices are automatically calculated from user-supplied definitions of the thickness, the material properties, and the relative orientation of these properties for the individual lamina. Once these matrices of elastic moduli are calculated, the analysis proceeds in a standard manner as described in Chapter 7: Solution Sequences. This capability for the automatic representation of laminated composites is available in linear static analysis, real and complex eigenvalue analysis, buckling analysis, and dynamic analysis.

In static analysis and in static analysis with cyclic symmetry, the analyst has the option to request that the program evaluate stresses or strains and appropriate failure indices in individual laminae. Interlaminar shear stresses and strains and a failure index for bonding may also be requested.

If any of the plies have material nonlinear behavior including progressing failures, SOL 400 should be used. See the MSC Nastran Nonlinear User's Guide.

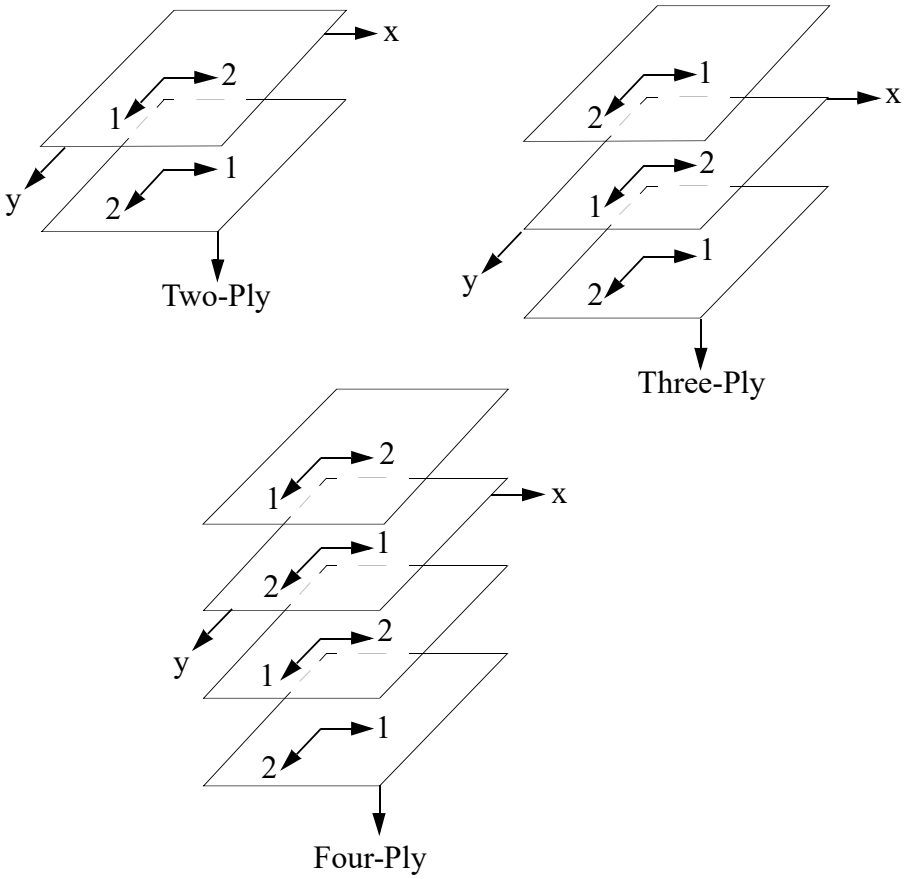


Figure 4-1 Exploded View of Three Cross-Ply Laminated Plates

Layered Composite Material Analysis

Two Bulk Data entries, MAT8, PCOMP, and PCOMPG are used in the modeling of composite structures. The MAT8 Bulk Data entry (defined in Nonuniform Material and Composite Properties for Shells, 267) defines the material property for an orthotropic material. The material properties of the individual lamina along the longitudinal direction (also called the fiber direction or 1-direction) and the transverse direction (also called the matrix direction or 2-direction) are supplied using this entry. The PCOMP Bulk Data entry defines the properties of the n-ply composite material laminate. The thickness, orientation, and the material identification number of each of the individual lamina is provided using this entry. PCOMPG is similar to PCOMP, but you can define unique ply layer IDs. For this section, they can be considered and only PCOMP will be identified.

From the data supplied by the PCOMP and MAT8 Bulk Data entries, the program automatically calculates the membrane, bending, transverse shear, and coupled membrane-bending material properties of the laminate as a whole. These calculated properties are output in the form of PSHELL and MAT2 entries when a sorted echo of the Bulk Data Section is requested.

The coefficients of thermal expansion derived for membrane-bending coupling, which appear in the A1, A2, and A12 fields of MAT2 entry and correspond to the MID4 Field on the PSHELL, require special interpretation. They are given by:

$$\{\alpha_{MAT2}\} = [G_{ij_{MAT2}}]\{\alpha_{ACTUAL}\} \tag{4-1}$$

In order to obtain the actual values of A1, A2, and A12 $\{\alpha_{MAT2}\}$, the user must solve equation (4-1).

If the user specifies ECHO = PUNCH in the Case Control Section, the generated PSHELL and MAT2 entries are routed to the PUNCH file, where they are written after the sorted bulk data echo. Since the MIDi fields use more than eight digits, a large field format is used for the PSHELL and MAT2 entries. The PUNCH file thus contains the sorted Bulk Data echo and the generated PSHELL and MAT2 entry images. After the MAT8 and PCOMP entry images are deleted, the entry images contained in the PUNCH file may be included in the Bulk Data to be submitted in a subsequent run. The existence of composite elements is recognized by a large value (more than eight digits) on the MIDi fields of the PSHELL entry, which may be generated by PCOMP entries. If stress output is desired, ensure that PARAM,NOCOMPS is set to -1 for any subsequent runs because composite stress recovery is not possible without the PCOMP entries.

Certain postprocessing features are available for Solution Sequences 101, 103, 105, 106, 114, 115, 153, 200, 400, and 600 which include all the statics, normal modes, steady nonlinear heat transfer, and buckling analyses. PARAM,NOCOMPS may be used to modify these post processing options in superelement solution sequences. These features are (1) stresses in individual laminae including interlaminar shear stresses and (2) a failure index for each lamina.

It is to be noted that two-dimensional plate theory used in the CQUAD4, CQUAD8, CTRIA3, or CTRIA6 elements does not allow for the exact calculation of interlaminar stresses. An approximate procedure is therefore used to determine the interlaminar shear stresses. The approximation consists of neglecting the effects of twisting moments in the computation of interlaminar shear stresses. For the majority of composite material structural analyses, this approximation is expected to yield satisfactory results for all elements in the model except those at the edges of the structure.

Voluminous output will result if the stresses for all elements for all laminae are requested. To help the analyst to review this information, provisions are made to calculate a failure index for each ply. This failure index is obtained by considering the failure criteria for unidirectional fiber composites as in the commonly used failure theories. The user has the option to choose any one of the four commonly used failure criteria for composites. These are: (1) Hill's Theory, (2) Hoffman's Theory, (3) Tsai-Wu Theory, and (4) Maximum Strain. The user must supply allowable stresses/strains for the laminae if a failure index is to be calculated. If more modern failure criteria are desired (such as, Puck Hashin or Digimat), SOL 400 should be used. See the MSC Nastran Nonlinear User's Guide.

Classical Lamination Theory

Classical lamination theory makes the following assumption regarding the behavior of the laminae:

- The laminae are perfectly bonded together.
- The bonds are infinitesimally thin and no lamina can slip relative to another.

- Linear variation of strain through the laminate thickness is assumed.

Deformation in the X-Y plan of the plate at any point C at a distance z in the normal direction to plate middle surface is

$$U = U_0 + z\theta_y$$

$$V = V_0 + z\theta_x$$

where U, V, and W are the displacements along the X, Y, and Z directions in the element coordinate system, and θ_x, θ_y are the rotations.

The strain-displacement-middle surface strain and curvatures relationship is given by:

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial U_0}{\partial x} \\ \dfrac{\partial V_0}{\partial y} \\ \dfrac{\partial U_0}{\partial y} + \dfrac{\partial V_0}{\partial x} \end{Bmatrix} + z \begin{Bmatrix} \dfrac{\partial \theta_y}{\partial x} \\ -\dfrac{\partial \theta_x}{\partial y} \\ \dfrac{\partial \theta_y}{\partial y} - \dfrac{\partial \theta_x}{\partial x} \end{Bmatrix} = \begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_y^0 \\ \gamma_{xy}^0 \end{Bmatrix} - z \begin{Bmatrix} \chi_x \\ \chi_y \\ \chi_{xy} \end{Bmatrix}$$

where the ε^0's and χ's are the middle surface strains and curvatures, respectively.

The stress resultants for an N-layer laminate are obtained by integration of the stresses in each lamina through the laminate thickness as:

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix} = \int_{-\frac{t}{2}}^{\frac{t}{2}} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} dz = \sum_{k=1}^{N} \int_{z_{k-1}}^{z_k} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}_k dz$$

$$\begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = -\int_{-\frac{t}{2}}^{\frac{t}{2}} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} z\,dz = -\sum_{k=1}^{N} \int_{z_{k-1}}^{z_k} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}_k z\,dz$$

The stress resultant to strain relationship is:

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix} = \sum_{k=1}^{N} [G]_k \left\{ \int_{z_{k-1}}^{z_k} \begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_y^0 \\ \gamma_{xy}^0 \end{Bmatrix}_k dz - \int_{z_{k-1}}^{z_k} \begin{Bmatrix} \chi_x \\ \chi_y \\ \chi_{xy} \end{Bmatrix}_k z\,dz \right\}$$

$$\begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \sum_{k=1}^{N} [G]_k \left\{ -\int_{z_{k-1}}^{z_k} \begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_y^0 \\ \gamma_{xy}^0 \end{Bmatrix}_k z\,dz + \int_{z_{k-1}}^{z_k} \begin{Bmatrix} \chi_x \\ \chi_y \\ \chi_{xy} \end{Bmatrix}_k z^2\,dz \right\}$$

where $[G]_k$ is the material matrix transformed from the laminate coordinate system into the lamina coordinate system.

These relations can be written in the following form used to describe composite elements:

$$\begin{Bmatrix} F \\ M \end{Bmatrix} = \begin{bmatrix} A & B \\ B & D \end{bmatrix} \begin{Bmatrix} \varepsilon^0 \\ \chi \end{Bmatrix}$$

where:

$$[A] = \sum_{k=1}^{N} [G]_k (z_k - z_{k-1})$$

$$[B] = \frac{1}{2} \sum_{k=1}^{N} [G]_k (z_k^2 - z_{k-1}^2)$$

$$[D] = \frac{1}{3} \sum_{k=1}^{N} [G]_k (z_k^3 - z_{k-1}^3)$$

are named in composite element literature as the membrane, membrane-coupling, and bending matrices, respectively. In the shell element formulation in MSC Nastran, these relationships take the following form:

$$\begin{Bmatrix} F \\ M \\ Q \end{Bmatrix} = \begin{bmatrix} TG_1 & T^2G_4 & 0 \\ T^2G_4 & \frac{T^3}{12}G_2 & 0 \\ 0 & 0 & T_sG_3 \end{bmatrix} \begin{Bmatrix} \varepsilon^0 \\ \chi \\ \gamma \end{Bmatrix}$$

where:

$$[A] = TG_1$$

$$[B] = -T^2G_4$$

$$[D] = \frac{T^3}{12}G_2$$

$$\{Q\} = \begin{Bmatrix} Q_x \\ Q_y \end{Bmatrix} = \text{transverse shear resultants}$$

$$\{\gamma\} = \begin{Bmatrix} \gamma_x \\ \gamma_y \end{Bmatrix} = \text{transverse shear strains}$$

$$T = \text{nominal plate thickness}$$

T_s	=	effective transverse shear material thickness
G_3	=	effective transverse shear material matrix

MSC Nastran allows G_1, G_2, G_4, T, G_3, and T_s to be input directly in PSHELL or to have the composite equivalent material matrices calculated internally from the PCOMP data.

Equivalent thermal properties are determined as follows:

$$\begin{Bmatrix} \alpha_{\varepsilon_0} \\ \alpha_{\chi} \\ \alpha_{\chi\varepsilon_0} \end{Bmatrix} = \begin{Bmatrix} T\alpha_1 \\ \frac{T^3}{12}\alpha_2 \\ T^2\alpha_3 \end{Bmatrix} = \begin{bmatrix} A & B & 0 \\ B & D & 0 \\ 0 & 0 & T_s G_3 \end{bmatrix}^{-1} \begin{Bmatrix} G\alpha_{\varepsilon_0} \\ G\alpha_{\chi} \\ G\alpha_{\chi\varepsilon_0} \end{Bmatrix}$$

The following coefficients are used to determine equivalent thermal properties:

$$\{G\alpha_{\varepsilon_0}\} = \sum_{k=1}^{N} [G]_k \{\alpha\}_k (z_k - z_{k-1})$$

$$\{G\alpha_{\chi}\} = \frac{1}{3}\sum_{k=1}^{N} [G]_k \{\alpha\}_k (z_k^3 - z_{k-1}^3) \quad \{G\alpha_{\chi\varepsilon_0}\} = \frac{1}{2}\sum_{k=1}^{N} [G]_k \{\alpha\}_k (z_k^2 - z_{k-1}^2)$$

These coefficients are used to calculate the equivalent thermal properties as follows:

$$\{\alpha_{\varepsilon_0}\} = [A]^{-1}\{G\alpha_{\varepsilon_0}\}$$

and

$$\{\alpha_x\} = [D]^{-1}\{G\alpha_x\}$$

where $\{\alpha_{\varepsilon_0}\}$ and $\{\alpha_x\}$ are the membrane and bending equivalent thermal properties.

Note that $\{\alpha_{\varepsilon_0}\}$ is not directly calculated, but is determined from $\{G\alpha_{\varepsilon_0}\}$ in MSC Nastran when the PCOMP input is used when the MID4 field on the PSHELL is > 400,000,000. Also note that $\{G_{x\xi_0}\}$ cannot be input directly using PSHELL and $\{\alpha_{\varepsilon_0}\}$ can be input only if $[B]$ is invertible (which is generally not true).

The thermal expansion relationships in the shell element formulation take the following form:

$$\alpha_{\varepsilon_0} = T\alpha_1$$

$$\alpha_{\chi} = \frac{T^3}{12}\alpha_2$$

$$\alpha_{\chi\varepsilon_0} = T^2\alpha_3$$

where α_1, α_2, and α_3 are the thermal expansion inputs on the materials referenced by the MID1, MID2, and MID4 fields on the PSHELL entry. If PCOMP is used, these relationships are automatically calculated.

The terms G_1, G_2, and G_4 are defined by the following integrals:

$$G_1 = \frac{1}{T}\int [G_e]dz$$

$$G_2 = \frac{1}{I}\int z^2 [G_e]dz$$

$$G_4 = \frac{1}{T^2}\int (-z)[G_e]dz$$

The limits on the integration are from the bottom surface to the top surface of the laminated composite. The matrix of material moduli, $[G_e]$, has the following form for isotropic materials:

$$[G_e]_I = \begin{bmatrix} \frac{E}{1-\nu^2} & \frac{\nu E}{1-\nu^2} & 0 \\ \frac{\nu E}{1-\nu^2} & \frac{E}{1-\nu^2} & 0 \\ 0 & 0 & \frac{E}{2(1+\nu)} \end{bmatrix}$$

For orthotropic materials, the matrix, $[G_e]$, is written as follows:

$$[G_e]_0 = \begin{bmatrix} \frac{E_1}{1-\nu_1\nu_2} & \frac{\nu_1 E_2}{1-\nu_1\nu_2} & 0 \\ \frac{\nu_2 E_1}{1-\nu_1\nu_2} & \frac{E_2}{1-\nu_1\nu_2} & 0 \\ 0 & 0 & G_{12} \end{bmatrix}$$

Here, $\nu_1 E_2 = \nu_2 E_1$ in order to satisfy the requirement that the matrix of elastic moduli be symmetric. In general, the analyst may supply element properties with respect to a particular orientation which does not necessarily correspond to the principal material axes. In this case, the analyst must also supply the value of the angle, θ or material coordinate system that orients the element material axis relative to the side G1-G2 of the element. The material elastic modulus matrix is then transformed by the program into the element modulus matrix through the relation

$$[G_e] = [U]^t [G_m][U]$$

where:

$$[U] = \begin{bmatrix} \cos^2\theta & \sin^2\theta & \cos\theta\sin\theta \\ \sin^2\theta & \cos^2\theta & -\cos\theta\sin\theta \\ -2\cos\theta\sin\theta & 2\cos\theta\sin\theta & \cos^2\theta - \sin^2\theta \end{bmatrix}$$

The finite element model for a structure composed of composite materials requires the evaluation of the matrix of elastic moduli for each plate element of the model. The characteristics of the composite media are totally contained in these matrices.

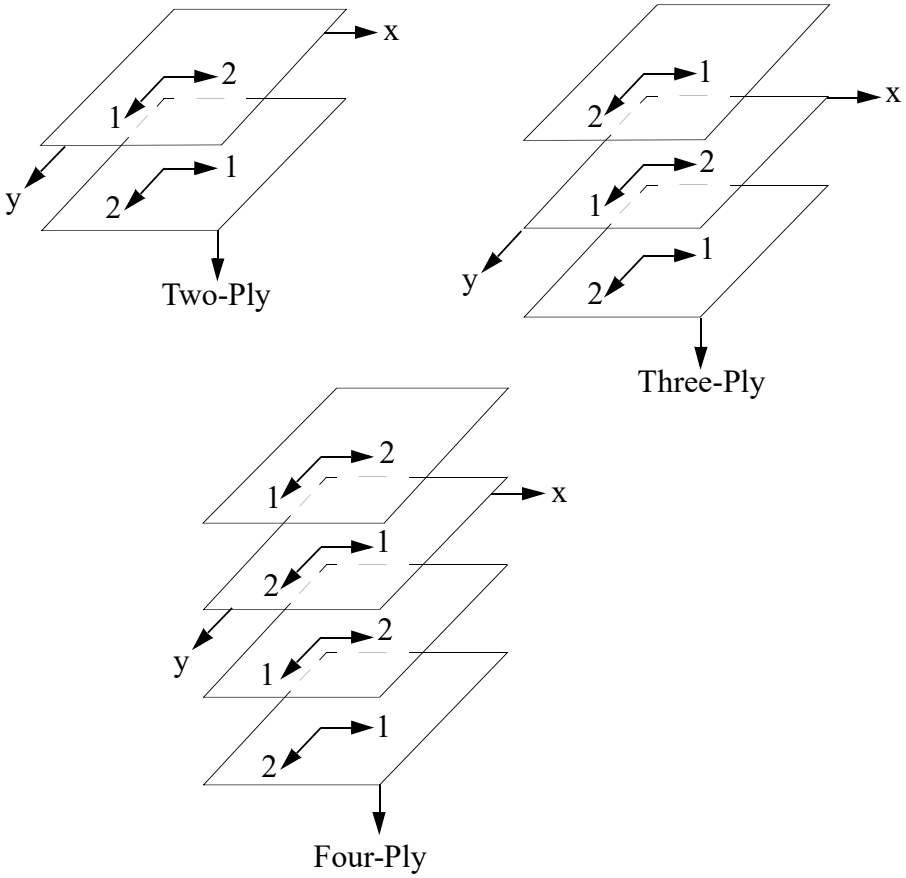


Figure 4-2 Exploded View of Three Cross-Ply Laminated Plates

To illustrate evaluation of these matrices, consider the cross-ply laminates of Figure 4-2. In this portion of the discussion, the three configurations shown in Figure 4-2 are each assumed to be represented by a single quadrilateral plate element and the coordinate axes shown are coincident with the element coordinate axes. Then, if it is assumed that each lamina of the n-ply laminates is of thickness T/n, where T is the total thickness of each of three configurations, the matrices of elastic moduli may be evaluated from the following relations:

$$[G_1] = \frac{1}{T}\left\{\int_{-\frac{T}{2}}^{-\frac{T}{2}+\frac{T}{n}} [G_e]_1 dz + \int_{-\frac{T}{2}+\frac{T}{n}}^{-\frac{T}{2}+\frac{2T}{n}} [G_e]_2 dz + \ldots + \int_{-\frac{T}{2}+\frac{(n-1)T}{n}}^{\frac{T}{2}} [G_e]_n dz\right\}$$

$$[G_2] = \frac{1}{I}\left\{\int_{-\frac{T}{2}}^{-\frac{T}{2}+\frac{T}{n}} [G_e]_1 z^2 dz + \int_{-\frac{T}{2}+\frac{T}{n}}^{-\frac{T}{2}+\frac{2T}{n}} [G_e]_2 z^2 dz + \ldots + \int_{-\frac{T}{2}+\frac{(n-1)T}{n}}^{\frac{T}{2}} [G_e]_n z^2 dz\right\}$$

$$[G_4] = \frac{1}{T^2}\left\{\int_{-\frac{T}{2}}^{-\frac{T}{2}+\frac{T}{n}} [G_e]_1 (-z) dz + \int_{-\frac{T}{2}+\frac{T}{n}}^{-\frac{T}{2}+\frac{2T}{n}} [G_e]_2 (-z) dz + \ldots + \int_{\frac{T}{2}+\frac{(n-1)T}{n}}^{\frac{T}{2}} [G_e]_n (-z) dz\right\}$$

These relations reflect the assumption that the xy-plane of the element coordinate system is coincident with the geometric middle plane of the laminate. The xy-plane of the element coordinate system is defined in the mean plane of the element so that any offset between the mean plane of the connected grid points and the geometric middle plane of the laminate would be reflected in the integration limits of the preceding relations.

Note that $I = T^3/12$ in the evaluation of $[G_2]$, i.e., the value $(12I)/T^2$, will be assigned the default value of 1.0 on the plate element property entry.

Transverse Shear Theory

The matrix of elastic moduli for transverse shear, $[G_3]_m$ is defined as a two-by-two matrix of the form

$$[G_3]_m = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}$$

and the corresponding matrix transformed into an element coordinate system is given by

$$[G_3]_e = [W]^T [G_3]_m [W]$$

where

$$[W] = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

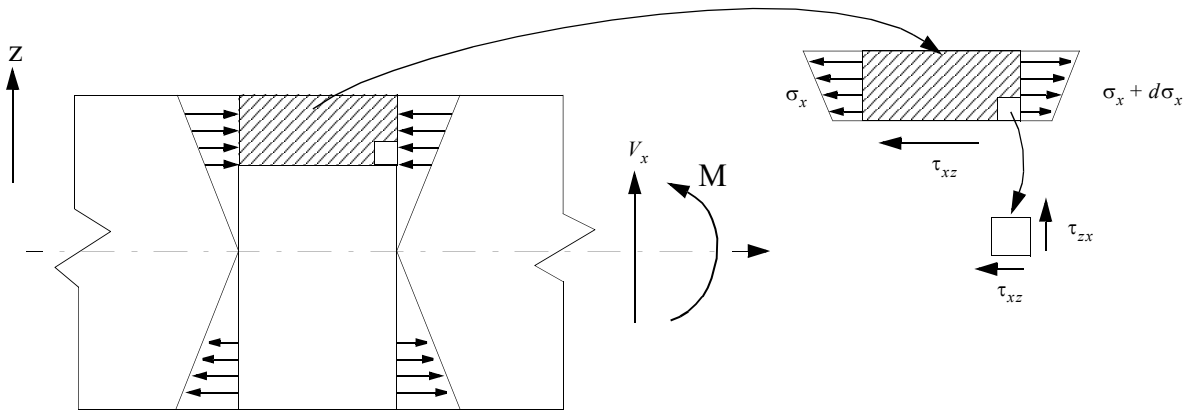
The mean value of the transverse shear modulus $\bar{G}$ for the laminated composite is defined in terms of the transverse shear strain energy, U, through the depth

$$U = \frac{1}{2}\frac{V^2}{\bar{G}T} = \frac{1}{2}\int\frac{(\tau(z))^2}{G(z)}dz \tag{4-2}$$

A unique mean value of the transverse shear strain is assumed to exist for both the x and y components of the element coordinate system, but for ease of discussion, only the evaluation of an uncoupled x component of the shear moduli will be illustrated here. From equation (4-2), the mean value of transverse shear modulus may be written in the following form

$$\frac{1}{G_x} = \frac{T}{V_x^2}\sum_{i=1}^{N}\int_{z_{i-1}}^{z_i}\frac{(\tau_{zx}(z))^2}{(G_x)_i}dz \tag{4-3}$$

where G is an "average" transverse shear coefficient used by the element code and $(G_x)_i$ is the local shear coefficient for layer i. To evaluate equation (4-3), it is necessary to obtain an expression for $(\tau_{zx}(z))$. This can be accomplished by assuming that the x- and y-components of stress are decoupled from one another. This assumption allows the desired equation to be deduced through an examination of a beam unit cross-sectional width.



The equilibrium conditions in the horizontal direction and for total moment are

$$\frac{\partial\tau_{xz}}{\partial z} + \frac{\partial\sigma_x}{\partial x} = 0 \tag{4-4}$$

$$V_x + \frac{\partial M_x}{\partial_x} = 0 \tag{4-5}$$

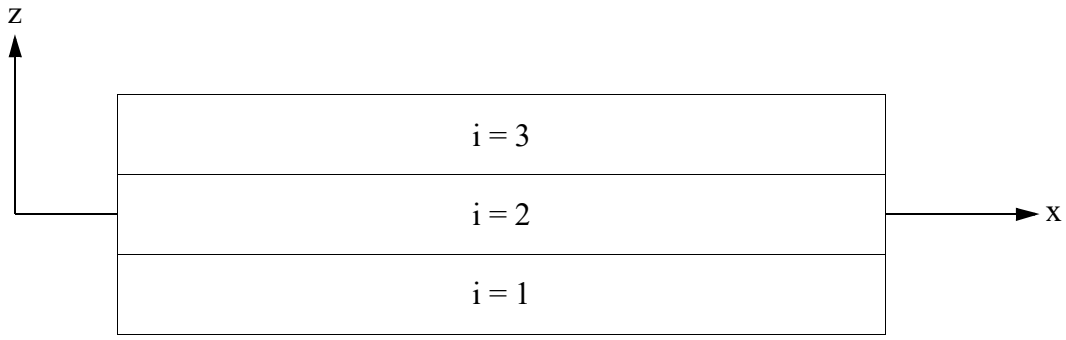
Now, if the location of the neutral surface is denoted by $\bar{z}_x$ and ρ is the radius of curvature of the beam, the axial stress E_x may be expressed in the form

$$\sigma_x + \frac{E_x(\bar{z}_x - z)}{(\overline{EI})_x} = 0$$

Relation 20 may be differentiated with respect to x combined with equations (4-4) and (4-5). In a region of constant E_x the result may be integrated to yield the following expression

$$\tau_{xz} = C_i + \frac{V_x}{(\overline{EI})_x}\left(\bar{z}_x z - \frac{z^2}{2}\right)E_{xi} \qquad z_{i-1} < z < z_i \tag{4-6}$$

Equation (4-7) is particularly convenient to use in the analysis of n-ply laminates because sufficient conditions exist to determine the constants Ci (i = 1,2,...,n) and the "directional bending center" $\bar{z}_x$. For example, consider the following laminated configuration:



At the bottom surface ($i = 1$, $z = z_0$, and $\tau_{xz} = 0$)

$$C_1 = \frac{-V_x}{(\overline{EI})_x}\left(\bar{z}_x z_0 - \frac{z_0^2}{2}\right)E_{x1}$$

and for the first ply at the interface between plies $i = 1$ and $i = 2$ ($z = z_1$)

$$(\tau_{xz})_1 = +\frac{V_x}{(\overline{EI})_x}\left[\bar{z}_x(z_1 - z_0)\frac{1}{2}[z_1^2 - z_0^2]\right]E_{x1}$$

At this interface between plies $i = 1$ and $i = 2$,

$$(\tau_{xz})_2 = C_2 + \frac{V_x}{(\overline{EI})_x}\left(\bar{z}_x z_1 - \frac{z_1^2}{2}\right)E_{x2}$$

and as $(\tau_{xz})_2 = (\tau_{xz})_1$ at $z = z_1$,

$$C_2 = (\tau_{xz})_1 - \frac{V_x E_{x2}}{(\overline{EI})_x}\left[\bar{z}_x z_1 - \frac{1}{2}z_1^2\right] \tag{4-7}$$

Then, in the ply $z_1 < z < z_2$ the shear is

$$\tau_{xz}(z) = (\tau_{xz})_1 \frac{V_x E_{x2}}{(\overline{EI})_x}\left[\bar{z}_x(z - z_1) - \frac{1}{2}(z^2 - {z_1}^2)\right]$$

In general, for any ply, $Z_{i-1} < z < z_i$, the shear is

$$\tau_{xz}(z) = (\tau_{xz})_{i-1} \frac{V_x E_{xi}}{(\overline{EI})_x}\left[\bar{z}_x(z - z_{i-1}) - \frac{1}{2}(z^2 - (z_{i-1})^2)\right] \tag{4-8}$$

At any ply interface, z_i, the shear is therefore

$$(\tau_{xz})_i = \frac{V_x}{(\overline{EI})_x} \sum_{j=1}^{i} E_{xj} T_j \left[\bar{z} - \frac{1}{2}(z_j + z_{j-1})\right]$$

where $T_j = z_j - (z_j - 1)$.

Note that the shear at the top face, $(\tau_{xz})_n$, is zero and therefore

$$(\tau_{xz})_n = \frac{V_x}{(\overline{EI})}\left[\bar{z}_x \sum_{j=1}^{n} E_{xj} T_j - \sum_{j=1}^{n} E_{xj} T_j (z_j + z_{j-1})/2\right] = 0 \tag{4-9}$$

Equation (4-9) proves that if $\bar{z}_x$ is the bending center, the shear at the top surface must be zero.

Equation (4-8) could be substituted into (4-4) and integrated. A better form of equation (4-9), for this purpose is

$$(\tau_{xz}(z))_i = \frac{V_x E_{xi}}{(\overline{EI})_x}\left[f_{xi} + \bar{z}_x(z - z_{i-1})\frac{1}{2}(z^2 - (z_{i-1})^2)\right] \tag{4-10}$$

where

$$f_{xi} = \frac{1}{E_{xi}} \sum_{j=1}^{i-1} E_{xj} T_j \left[\bar{z}_x - \frac{1}{2}(z_j + z_{j-1})\right] \tag{4-11}$$

Substituting equation (4-10) into equation (4-6) and after a considerable effort of integrating the results, we obtain

$$\frac{1}{G_x} = \frac{T}{(\overline{EI})_x^2} \sum_{i=1}^{N} \frac{1}{G_{xi}} R_{xi} \tag{4-12}$$

where

$$R_{xi} = (E_{xi})^2 T_i \left[\left\{f_{xi} + (\bar{z}_x - z_{i-1})T_i - \frac{1}{3}{T_i}^2\right\} f_{xi}\right.$$
$$\left. + \left\{\frac{1}{3}(\bar{z}_x - 2\, z_{i-1}) - \frac{1}{4}T_i\right\}\bar{z}_x {T_i}^2 + \left\{\frac{1}{3}z_{i-1}^2 + \frac{1}{4}z_{i-1}T_i + \frac{1}{20}{T_i}^2\right\}{T_i}^2\right]$$

This expression for the inverse shear modulus for the x-direction may be generalized to provide for the calculation of each term in the two-by-two matrix of shear moduli.

$$[\bar{G}_{kl}] = \left[\frac{T}{(\bar{E}I)_{kk}^2}\sum_{i=1}^{n}[G_{kl}^i]^{-1}R_{ki}\right]^{-1} \tag{4-13}$$

where:

$k = 1,2$

$l = 1,2$

Note that if no shear is given, $[G^i]^{-1} = 0$, and also that in equation (4-8)

where $[G_2]^*$ is calculated in the same manner as $[G_2]$ except that Poisson's Ratio is set to zero.

The moduli for individual plies are provided through user input because, in general, $G_{12} \neq G_{21}$ will be used for the coupling terms. Finally,

$$[G_3] = \begin{bmatrix} \bar{G}_{11} & (\bar{G}_{12})_{avg} \\ (\bar{G}_{12})_{avg} & \bar{G}_{22} \end{bmatrix}$$

As an example, consider a single layer element. For this case let $z_{i-1} = -T/2$, $\bar{z} = 0$, $f_o = 0$, and $EI = ET^3/12$. Evaluating equation (4-13), we obtain

$$R_i = E^2T^5\left[\frac{1}{12} - \frac{1}{8} + \frac{1}{20}\right] = \frac{E^2T^5}{120}$$

and

$$\frac{1}{G} = \left(\frac{(12)^2T}{E^2T^6}\right)\cdot\left(\frac{E^2T^5}{120\ G_1}\right) = \frac{6}{5G_1}$$

which provides the correct factor for a nonuniform shear distribution in a plate.

Data Recovery

As the material properties of the laminated composite plate are completely contained in the matrices of elastic moduli, standard data recovery methods (see Nonuniform Material and Composite Properties for Shells, 267) may be used to calculate stresses in individual laminae and the forces sustained by the laminate.

An additional aid to the analyst is provided through optional output of a failure index for individual laminae. Failure indices assume a value of one on the periphery of a failure surface in stress space. If the failure index is less than one, the lamina stress is interior to the periphery of the failure surface and the lamina is assumed safe. Conversely, if the failure is greater

than one, the lamina stress is exterior to the periphery of the failure surface and the lamina is assumed to have failed. These failure indices represent a phenomenological failure criterion in that only an occurrence of a failure is indicated and not the mode of failure.

In the present context, concern is with the analytical definition of a failure surface in stress or strain space for use with laminae under biaxial loading. Four commonly used definitions of the failure surface are provided: Hill's Theory, Hoffman's Theory, Tsai-Wu Theory, and maximum strain theory.

The discussion of data recovery activities concludes with a description of the approximate procedure provided to obtain interlaminar shear stresses.

Failure Indices

In the analysis of isotropic materials, strength is independent of the orientation of the body under load and one may compare the largest computed principal stress with an allowable stress to establish the integrity of the structure. Laminated composites, on the other hand, are orthotropic materials and may exhibit unequal properties in tension and compression. Thus, the strength of these orthotropic laminae is a function of body orientation relative to the imposed stresses.

As the evaluation of the matrices of material moduli for laminated composites provides sufficient information to determine the actual stress field sustained by the material, the determination of structural integrity will depend on the definition of an allowable stress field. The basic ingredient of this definition is the establishment of a set of allowable stresses or strengths in the principal material directions.

X_t = Allowable tensile stress in the principal x (or 1) direction of the material.

X_c = Allowable compressive stress in the principal x (or 1) direction of the material.

Y_t = Allowable tensile stress in the principal y (or 2) direction of the material.

Y_c = Allowable compressive stress in the principal y (or 2) direction of the material.

S = Allowable shear stress in the principal material system.

Expressions for the three available failure theories are shown in Table 4-3.

Table 4-3 Failure Theories Provided in MSC Nastran

Theory	Failure Index	Remarks
Hill	$\frac{\sigma_1^2}{x^2} - \frac{\sigma_1\sigma_2}{x^2} + \frac{\sigma_2^2}{y^2} + \frac{\tau_{12}^2}{s^2} = \text{FI}$	Orthotropic materials with equal strengths in tension and compression, i.e., $x_t = x_c$ and $y_t = y_c$.
Hoffman	$\left(\frac{1}{X_t} - \frac{1}{X_c}\right)\sigma_1 + \left(\frac{1}{y_t} - \frac{1}{y_c}\right)\sigma_2 + \frac{\sigma_1^2}{x_t x_c} + \frac{\sigma_2^2}{y_t y_c} + \frac{\sigma_{12}^2}{s^2} - \frac{\sigma_1\sigma_2}{x_t x_c} = \text{FI}$	Orthotropic materials under a general state of plane stress with unequal tensile and compressive strengths.
Tsai-Wu	$\left(\frac{1}{X_t} - \frac{1}{X_c}\right)\sigma_1 + \left(\frac{1}{y_t} - \frac{1}{y_c}\right)\sigma_2 + \frac{\sigma_1^2}{x_t x_c} + \frac{\sigma_2^2}{y_t y_c} + \frac{\sigma_{12}^2}{s^2} + 2F_{12}\sigma_1\sigma_2 = \text{FI}$	Orthotropic materials under a general state of plane stress with unequal tensile and compressive strengths.
MAX STRAIN	$\left[\left(\frac{\varepsilon_1}{X_t}\right), \left(\frac{\varepsilon_2}{Y_t}\right), \left(\frac{\lvert\gamma_{12}\rvert}{S}\right)\right]$	

The following remarks apply to Table 4-3:

1. The failure index of Hill represents an analytical expression for an ellipse in the $\sigma_1\sigma_2$ plane. This original expression for failure was subsequentially modified to that shown on the left-hand side of the expression in Table 4-3 by Tsai and hence this failure theory is sometimes known as the Tsai-Hill Theory.
2. The Tsai-Wu tensor polynomial theory for orthotropic materials is specialized to the expression given in Table 4-3 for orthotropic laminae under a condition of plane stress. The coefficient F_{12} is experimentally determined from test specimens under biaxial loading. This inconvenience along with the constraint that F_{12} satisfy a stability criterion of the form

   $$\left(\frac{1}{x_t x_c}\right)\left(\frac{1}{y_t y_c}\right) - F_{12}^2 > 0$$

   creates some complication in the use of this theory. Narayanaswami and Adelman (see Reference 1) have suggested that F_{12} be set to zero and the use of Hoffman's Theory or the Tsai-Wu theory with $F_{12} = 0$ are preferred alternatives. The analyst may provide a value of f_{12} to be used in the Tsai-Wu failure index if he wishes.

Hill's Theory (Failure Theory ID = "HILL")

$$\text{Failure Index} = \frac{\sigma_1^2}{X^2} - \frac{\sigma_1\sigma_2}{X^2} + \frac{\sigma_2^2}{Y^2} + \frac{\tau_{12}^2}{S^2}$$

where:

X =is allowable stress in 1-direction

Y =is allowable stress in 2-direction

S =is allowable stress in shear

and $X = X_t$ if σ_1 is positive or $X = X_c$ if σ_1 is negative and similarly for Y and σ_2. For the interaction term $\sigma_1\sigma_2/X^2$, $X = X_t$ if $\sigma_1\sigma_2$ is positive or $X = X_c$ if $\sigma_1\sigma_2$ is negative.

Note that Hill's theory is used for materials that have the same strength in tension and compression.

A plot of the above equation obtained by setting the failure index to 1 on the σ_1-σ_2 plane yields an ellipse and is the anisotropic yield criterion of Hill (modified later by Tsai, and hence also sometimes known as the Tsai-Hill theory). Therefore, if the failure index so calculated is less than 1, the ply stresses are inside the yield ellipse and the ply is said to be "safe"; conversely, if the failure index is greater than 1, the ply stresses are outside the yield ellipse and the ply has failed.

Hoffman's Theory (Failure Theory ID = "HOFF")

The Hoffman's theory for an orthotropic lamina in a general state of plane stress is given by

$$\left(\frac{1}{X_t}-\frac{1}{X_c}\right)\sigma_1+\left(\frac{1}{Y_t}-\frac{1}{Y_c}\right)\sigma_2+\frac{\sigma_1^2}{X_tX_c}+\frac{\sigma_2^2}{Y_tY_c}+\frac{\sigma_{12}^2}{S^2}-\frac{\sigma_1\sigma_2}{X_tX_c}=1$$

The failure index is obtained by evaluating the left-hand side of the above equation.

Note that this theory takes into account the difference in tensile and compressive allowable stresses by using linear terms in the equation.

Tensor Polynomial Theory of Tsai-Wu (Failure Theory ID = "TSAI")

The theory of strength for anisotropic materials proposed by Tsai and Wu specialized to the case of an orthotropic lamina in a general state of plane stress is

$$F_1\sigma_1+F_2\sigma_2+F_{11}\sigma_1^2+F_{22}\sigma_2^2+2F_{12}\sigma_1\sigma_2+F_{66}\sigma_{12}^2=1$$

where:

$$F_1=\frac{1}{X_t}-\frac{1}{X_c}$$

$$F_2=\frac{1}{Y_t}-\frac{1}{Y_c}$$

$$F_{11} = +\frac{1}{X_t X_c}$$

$$F_{22} = +\frac{1}{Y_t Y_c}$$

$$F_{66} = +\frac{1}{S^2}$$

and F_{12} is to be evaluated experimentally.

The magnitude of F_{12} is, however, constrained by the following inequality called a "stability criterion":

$$F_{11}F_{22} - F_{12}^2 > 0$$

The necessity of satisfying the stability criterion, together with the requirement that F_{12} be determined experimentally from a combined stress state, poses difficulties in the use of this theory. Narayanaswami and Adelman (see Reference 1) have suggested that F_{12} be set to zero and that use of Hoffman's theory or Tensor Polynomial theory with $F_{12} = 0$ is a preferred alternative to the experimental determination of F_{12}. If the user has a value for use with F_{12} in the theory, the same may be input in the MAT8 Bulk Data entry; otherwise, F_{12} in the above theory will be set to 0.0. The left-hand side of the above equation will be evaluated as the failure index by this theory.

The failure index of bonding material will be calculated as the maximum interlaminar shear stress divided by the allowable bonding stress.

The failure indices for all the plies will be written into OEFIT (Output Element Failure Index Table) and output if stresses are requested. The failure index for the element is the largest value of the failure indices for all plies of the element.

Maximum Strain Theory (Failure Theory ID = "STRN")

The midplane strains and curvatures are available in the element coordinate system. From these, the stresses and strains in each individual lamina along the fiber direction and transverse direction can be calculated quite easily. The STRAIN Case Control command is used to request the output of lamina strains.

The maximum strain criteria has no strain interaction terms. The strain allowables specified on the MAT8 entry for each lamina include

X_t , X_c	Allowable strains in tension and compression, respectively, in the longitudinal direction.
Y_t , Y_c	Allowable strains in tension and compression, respectively, in the transverse direction.
S	Allowable strain for inplane shear.

The Failure Index is calculated using

$\frac{\varepsilon_1}{X_t}$ (or $\frac{\varepsilon_1}{X_c}$ if ε_1 compressive)

$\frac{\varepsilon_2}{X_t}$ (or $\frac{\varepsilon_2}{Y_c}$ if ε_2 compressive)

and

$\left|\frac{\gamma_{12}}{S}\right|$

i.e., the Failure Index = MAX $\left[\left(\frac{\varepsilon_1}{X}\right), \left(\frac{\varepsilon_2}{Y}\right), \left(\frac{|\gamma_{12}|}{S}\right)\right]$.

In addition, the user needs to be informed which mode of Failure Index is critical; i.e., longitudinal (1), transverse (2), or shear (12). Thus in the output for Failure Indices, the mnemonic 1, 2 or 12 is printed alongside the FP value to indicate the critical direction. There is no change in the way the Failure Index is calculated for interlaminar shear stresses.

It is quite possible that for the maximum strain theory, the user may wish to specify lamina stress allowables instead of strain allowables on the MAT8. This can be done by leaving the STRN field blank on the MAT8.

For this case, the failure indices are calculated using

$\frac{\varepsilon_1 \cdot E_{11}}{X_t}$ (or $\frac{\varepsilon_1 \cdot E_{11}}{X_c}$ if ε_1 compressive)

$\frac{\varepsilon_2 \cdot E_{22}}{Y_c}$ (or $\frac{\varepsilon_2 \cdot E_{22}}{Y_c}$ if ε_2 compressive)

and

$\frac{|\gamma_{12}| \cdot G_{12}}{S}$

i.e., the Failure Index = MAX $\left[\left(\frac{\varepsilon_1 \cdot E_{11}}{X}\right), \left(\frac{\varepsilon_2 \cdot E_{22}}{Y}\right), \left(\frac{|\gamma_{12}| \cdot G_{12}}{S}\right)\right]$

Strength Ratios

For laminated composites, Strength Ratio (SR) is a direct failure indicator compared to Failure Index (FI) which indicates only if failure had occurred. Generally Strength Ratio is defined as:

Strength Ratio (SR) = Allowable Stress / Calculated Stress

For example a SR = 1.2 indicates that the applied loads can be increased by 20% before failure occurs. A FI = 0.8 indicates that failure has not occurred and does not indicate 20% safety margin. Therefore the SR is a much more practical design indicator for both analysis and strength-criteria based design.

This new result quantity is available for all solutions sequences that support composites data recovery.

A new parameter -- *'PARAM,SRCOMPS,YES/NO'* requests Strength Ratio (SR) output. SR Output requires specification of a failure theory and allowable stress/strain values. Note that SRs will be computed for plies with an output request (PCOMP-SOUTi field).

For Maximum Stress/Strain and Transverse Stress theories, the Strength Ratio (SR) is simply the inverse of the FI. For the quadratic failure theories such as Hill, Hoffman and Tsai-Wu the SR is calculated by solving the quadratic equation with the FI set to unity and replacing the applied stress with the $SR \cdot applied$ stress.

Below the SR expression for each Failure Criteria is defined.

HILL Failure Criteria

The HILL failure criteria is defined as:

$$FI = \left(\frac{\sigma_1^2}{X^2}\right) + \left(\frac{\sigma_2^2}{Y^2}\right) - \left(\frac{\sigma_1\sigma_2}{X^2}\right) + \left(\frac{\sigma_{12}^2}{S^2}\right)$$

where:

$X = X_T$	if σ_1	=	Tensile
$X = X_C$	if σ_1	=	Compressive
$Y = Y_T$	if σ_2	=	Tensiles
$Y = Y_C$	if σ_2	=	Compressive
σ_1		=	Ply longitudinal stress
σ_2		=	Ply transverse stress
σ_{12}		=	Ply shear stress

For the product term:

$$\frac{\sigma_1, \sigma_2}{X^2}$$

where:

X	=	X_T	if σ_1 and σ_2 are of the same sign.
X	=	X_C	if σ_1 and σ_2 are of different signs.

The FI expression can be redefined in terms of a strength ratio. The roots of the resulting quadratic expression are the reserve factors, the first one for the stresses as applied and the second for when the sign of all the components of stress are reversed.

The FI expression is re-arranged with the applied stress replaced by a ratio SR times the applied stress and equated to a FI of unity, the SR can now be determined.

$$1.0 = [(SR^2\sigma_1^2)\cdot(1/X^2)] + [(SR^2\sigma_2^2)\cdot(1/Y^2)] - \left[\frac{(SR^2\sigma_1\sigma_2)}{(X^2)}\right] + \left[\frac{(SR^2\sigma_{12}^2)}{(Y^2)}\right]$$

$$[(\sigma_1^2/X^2) + (\sigma_2^2/Y^2) - (\sigma_1\sigma_2/X^2) + (\sigma_{12}^2/Y^2)]SR^2 - 1 = 0$$

$$aSR^2 + bSR + c = 0$$

from the general solution of a quadratic equation,

$$SR_1 = \frac{-b+\sqrt{b^2-4ac}}{2a} \text{ and } SR_2 = \frac{-b-\sqrt{b^2-4ac}}{2a}$$

For the HILL criteria,

$$a = (\sigma_1^2/X^2) + (\sigma_2^2/Y^2) - (\sigma_1\sigma_2)/X^2 + (\sigma_{12}^2/Y^2)$$

$$b = 0.0$$

$$c = -1$$

$$SR_1 = \frac{-b+\sqrt{b^2-4ac}}{2a}$$

$$SR_1 = \frac{\sqrt{-4\cdot a\cdot 1.0}}{2\cdot a} = \frac{2\cdot\sqrt{a}}{2\cdot a}$$

$$SR_1 = \frac{1}{\sqrt{a}}$$

$$SR_2 = -\frac{1}{\sqrt{a}}$$

where

$$a = (\sigma_1^2/X^2) + (\sigma_2^2/Y^2) - (\sigma_1\sigma_2/X^2) + (\sigma_{12}^2/Y^2) = FI$$

Therefore for the HILL quadratic failure criteria the SR can be directly derived from the FI:

$$SR = \frac{1}{\sqrt{FI}}$$

HOFFMAN Failure Criteria

Unlike the Hill criteria the Hoffman failure criteria takes into account the difference in tensile and compressive strengths in the longitudinal and transverse directions of the ply.

$$\text{FI} = [1/(X_T \cdot X_C)]\sigma_1^2 + [1/(Y_T \cdot Y_C)]\sigma_2^2 + [1/(X_T \cdot X_C)](\sigma_1\sigma_2) + (\sigma_{12}^2/S^2) + [(1/X_T) - (1/X_C)]\sigma_1 + [(1/Y_T) - (1/Y_C)]\sigma_2$$

defining:

$$\begin{aligned} F_1 &= (1/X_T) - (1/X_C) \\ F_2 &= (1/Y_T) - (1/Y_C) \\ F_{11} &= 1/(X_T \cdot X_C) \\ F_{22} &= 1/(Y_T \cdot Y_C) \\ F_{66} &= 1/(S^2) \\ FI &= F_{11}\sigma_1^2 + F_{22}\sigma_2^2 - F_{11}(\sigma_1\sigma_2) + F_{66}\sigma_{12}^2 + F_1\sigma_1 + F_2\sigma_2 \end{aligned}$$

Now introduce applied stress as a ratio R times applied stress equated to a FI of unity.

$$1.0 = F_{11}\text{SR}^2\sigma_1^2 + F_{22}\text{SR}^2\sigma_2^2 - F_{11}\text{SR}^2(\sigma_1\sigma_2) + F_{66}\text{SR}^2\sigma_{12}^2 + F_1\text{SR}\sigma_1 + F_2\text{SR}\sigma_2$$

Arrange in quadratic form

$$[F_{11}\sigma_1^2 + F_{22}\sigma_2^2 - F_{11}(\sigma_1\sigma_2) + F_{66}\sigma_{12}^2]R^2 + [F_1\sigma_1 + F_2\sigma_2]R - 1 = 0$$

$$\text{SR}_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \text{ and } \text{SR}_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

where:

$$\begin{aligned} a &= [F_{11}\sigma_1^2 + F_{22}\sigma_2^2 - F_{11}(\sigma_1\sigma_2) + F_{66}\sigma_{12}^2] \\ b &= [F_1\sigma_1 + F_2\sigma_2] \\ c &= -1.0 \end{aligned}$$

This is now a quadratic that does not simplify in the same way as the Hill criteria so there is no direct relationship between FI and SR. The roots of the equation have to be determined to obtain the SR.

TSAI-Wu Failure Criteria

The Tsai-Wu failure criterion is very similar to the Hoffman criteria but has a different value for the coefficient associated with the interaction of the direct stresses. In the Hoffman criterion it is the same as the coefficient associated with the direct longitudinal stresses.

For the Tsai-Wu criterion:

$$\text{FI} = F_{11}\sigma_1^2 + F_{22}\sigma_2^2 + 2F_{12}\sigma_1\sigma_2 + F_{66}\sigma_{12}^2 + F_1\sigma_1 + F_2\sigma_2$$

where:

$F_1 = (1/X_T) - (1/X_C)$

$F_2 = (1/Y_T) - (1/Y_C)$

$F_{11} = 1/(X_T \cdot X_C)$

$F_{22} = 1/(Y_T \cdot Y_C)$

$F_{66} = 1/(S^2)$

F_{12} = See (p. 282) Tensor Polynomial Theory of Tsai-Wu for a discussion of F_{12} and for the default settings.

Now introduce applied stress as a ratio R times applied stress equated to a FI of unity.

$$1.0 = F_{11}SR^2\sigma_1^2 + F_{22}SR\sigma_2^2 + 2F_{12}SR^2\sigma_1\sigma_2 + F_{66}SR^2\sigma_{12}^2 + F_1SR\sigma_1 + F_2SR\sigma_2$$

Arrange in quadratic form

$$[F_{11}\sigma_1^2 + F_{22}\sigma_2^2 + 2F_{12}\sigma_1\sigma_2 + F_{66}\sigma_{12}^2]SR^2 + [F_1\sigma_1 + F_2\sigma_2]SR - 1 = 0$$

$$SR_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \text{ and } SR_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

where:

$a = [F_{11}\sigma_1^2 + F_{22}\sigma_2^2 + 2F_{12}\sigma_1\sigma_2 + F_{66}\sigma_{12}^2]$

$b = [F_1\sigma_1 + F_2\sigma_2]$

$c = -1.0$

This is a quadratic equation that does not simplify in the same way as the Hill criteria so there is no direct relationship between FI and SR. The roots of the equation have to be determined to obtain the SR.

Maximum Stress (Strain)

Failure Index FI = Calculated Stress / Allowable Stress

By substituting the Allowable stress by RF*Allowable Stress and setting FI to unity

$$SR = 1/(FI)$$

Transverse Shear Stress

For this case the Strength Ratio is simply the inverse of the failure index.

"LAM" Options

The PCOMP Bulk Data entry is used to provide input data for the properties of an n-ply composite material laminate. As such, it is widely used in aerospace applications where a composite structure is built up from a number of discrete sheets of material, such as a carbon-epoxy fiber.

1. The SMEAR options to the LAM field smears the ply effects while ignoring stacking sequence.
2. The SMCORE option to the LAM field enables the modeling of a skin/core combination where the ply effects are smeared in the skin, the core inertial terms are treated exactly and the core stiffness terms are ignored.

MSC Nastran develops mass and stiffness data from PCOMP input in a two-step process. First, the PCOMP input data are considered together with the material data referenced by MIDi entries to produce PSHELL/MAT2 combinations that will lead to the required stiffness results and then this spawned data are used in the actual stiffness and mass calculations. The spawned PSHELL has four MIDis, identifying the MAT2s to be used for membrane, bending, transverse shear and membrane-bending coupling. The SMEAR and SMCORE options use these MIDis in the following manner:

SMEAR

The spawned PSHELL has MID1=MID2 with MID3, MID4, 12I/T**3, and the TS/T fields set as blanks. This results in a bending term given as

$$[D] = \frac{T^2[A]}{12}$$

SMCORE

Computation of the membrane and bending stiffness matrices is performed using the following derivation. Note that membrane-bending coupling is ignored.

Definitions:

$t_{face} = T_1 + T_2 + \ldots + T_{N-1}$ (total thickness of both face sheets)

$t_{core} = T_N$ (total core thickness)

Membrane Stiffness Matrix:

$[A]$ is computed using method utilized by LAM=BLANK assuming that core stiffness (layer N) is zero.

Face Sheet Properties:

$$\mu_{xy} = \frac{-[A]_{12}^{-1}}{[A]_{11}^{-1}}$$

$$\mu_{yx} = \frac{-[A]_{12}^{-1}}{[A]_{22}^{-1}}$$

$$E_x = \frac{A_{11}(1.0 - \mu_{xy}\mu_{yx})}{t_{face}}$$

$$E_y = \frac{A_{22}(1.0 - \mu_{xy}\mu_{yx})}{t_{face}}$$

$$G_{xy} = \frac{A_{33}}{t_{face}}$$

Moment of inertia:

$$I_{xx} = I_{yy} = \frac{t_{face}\left(t_{core} + \frac{t_{face}}{2}\right)^2}{4} + \frac{t_{face}^3}{48}$$

Note, this collapses to $t_{face}^3/12$ if t_{core} is zero.

Bending Stiffness Matrix:

$$D_{11} = \frac{E_x I_{xx}}{(1.0 - \mu_{xy}\mu_{yx})}$$

$$D_{22} = \frac{E_y I_{yy}}{(1. - \mu_{xy}\mu_{yx})}$$

$$D_{12} = D_{21} = \frac{D_{11}\mu_{xy}E_y}{E_x}$$

$$D_{33} = G_{xy}I_{xx}$$

$$D_{13} = D_{31} = D_{23} = D_{32} = 0$$

Input

The format of the PCOMP entry is unchanged, but the description of the LAM field is expanded and an additional Remark is added.

Field	Contents
LAM	Laminate Options (Character or Blank, default=Blank). See Remarks 3. and 4.
“Blank”	All plies must be specified and all stiffness terms are developed
“SYM”	Only plies on one side of the element centerline are specified. The plies are numbered starting with 1 for the bottom layer. If an odd number of plies are desired, the center ply thickness (T1 or TN) should be half the actual thickness.
“MEM”	All plies must be specified, but only membrane terms (MID1 on the derived PSHELL entry) are computed.

Field	Contents
"BEND"	All plies must be specified, but only bending terms (MID2 on the derived PSHELL entry) are computed.
"SMEAR"	All plies must be specified, stacking sequence is ignored MID1=MID2 on the derived PSHELL entry and MID3, MID4 and TS/T and 12I/T**3 terms are set to zero).
"SMCORE"	All plies must be specified, with the last ply specifying core properties and the previous plies specifying face sheet properties. The stiffness matrix is computed by placing half the face sheet thicknesses above the core and the other half below with the result that the laminate is symmetric about the mid-plane of the core. Stacking sequence is ignored in calculating the face sheet stiffness.

Remarks

3. The SYM option for the LAM option computes the complete stiffness properties while specifying half the plies. The MEM, BEND, SMEAR and SMCORE options provide special purpose stiffness calculations. SMEAR ignores stacking sequence and is intended for cases where this sequence is not yet known, stiffness properties are smeared. SMCORE allows simplified modeling of a sandwich panel with equal face sheets and a central core.
4. Element output for the SMEAR and SMCORE options is produced using the PARAM NOCOMPS –1 methodology that suppresses ply stress/strain results and prints results for the equivalent homogeneous element.

Output

Standard element response data are recovered for elements that employ the new LAM options with the key provision that the SMEAR and SMCORE options results in element results that correspond to the equivalent homogenous element and not the ply-by-ply results. This is equivalent to using PARAM NOCOMPS –1 and only the homogeneous results are produced regardless of the user input specification of the NOCOMPS parameter.

Guidelines for Design Optimization

Special procedures must be employed when using the PCOMP with the SMEAR and SMCORE options in an automated design task in SOL 200. The FSD option in SOL 200 does not support the SMEAR or SMCORE options since the resizing of a ply layer in FSD is driven by a response in that layer. The SMEAR/SMCORE options do not produce layer responses, therefore, FSD resizing is precluded. For the standard mathematical programming of SOL 200, composite stress, strain and force results are not available for the PCOMP with SMEAR or SMCORE. Instead, the homogeneous element results must be invoked. This is done by using the standard STRESS, STRAIN, FORCE response types (as opposed to the CSTRESS, CSTRAIN, CFAILURE response types) and the PTYPE can be set to either PCOMP or PSHELL. The PID for the PSHELL is identical to that of the PCOMP. The rules for the design of properties are not changed by these new options in that the user can invoke a lamina thickness or orientation on the PCOMP entry from a DVPREL1 or DVPREL2. Invoking the thickness of the equivalent PSHELL that is derived from a PCOMP is not allowed and produces a fatal error.

Temperature-Dependent Properties

Theory

Temperature Distribution

The temperature through the shell thickness is assumed to be linear:

$$T(\xi, \eta, \zeta) = T(\xi, \eta) + \zeta \cdot T' , \quad -\frac{t}{2} \le \zeta \le \frac{t}{2}$$

where $T(\xi, \eta)$ is the membrane temperature defined by the weighted average of the grid temperatures from the TEMP and/or TEMPD Bulk Data entries, or the element temperature from the TEMPP1 Bulk Data entry; and T' is the temperature gradient from the TEMPP1 Bulk Data entry.

Currently the smeared laminate properties are calculated using the reference temperature from the PCOMP Bulk Data entry, which has no gradient defined through the thickness. This implies that all the plies are at the same temperature, which is not a valid assumption in certain cases. For a more accurate representation of thermal strains and temperature-dependent properties, use SOL 400.

The temperature at the midplane of the *i*-th ply will be calculated as:

$$T_i = \overline{T} + \left(z_0 + \sum_{j=1}^{i-1} t_j + \frac{1}{2} t_i \right) \cdot T'$$

where $\overline{T}$ is the reference plane temperature; z_0 is the bottom surface offset from the reference plane; and t_i are the ply thicknesses defined on the PCOMP Bulk Data entry. This temperature will be used to calculate the temperature-dependent material properties of each ply.

Coefficient of Thermal Expansion

Secant Approach

The coefficient of thermal expansion (in one dimension) is defined as the normalized change in length for a given change in temperature:

$$\alpha = \frac{1}{L} \frac{\Delta u}{\Delta T}$$

such that the thermal strain is given by:

$$\varepsilon = \alpha \Delta T$$

which is the secant approach. This can be computed in several ways, depending on the case control and material dependence.

For linear analysis using neither TEMP(MAT) nor TEMP(INIT):

$$\varepsilon = \alpha (T_{load} - T_{ref})$$

For linear analysis using TEMP(INIT) with no MATTi:

$$\varepsilon = \alpha(T_{load} - T_{init})$$

For linear analysis using TEMP(MAT) with MATTi:

$$\varepsilon = \alpha_{mat}(T_{load} - T_{ref})$$

For linear analysis using TEMP(INIT) with MATTi:

$$\varepsilon = \alpha_{init}(T_{load} - T_{init})$$

For nonlinear analysis using TEMP(INIT) with MATTi (TEMP(INIT) is required and TEMP(MAT) is not allowed), the expression is more complex, in an attempt to better approximate the temperature-dependent properties:

$$\varepsilon = \alpha_{load}(T_{load} - T_{ref}) - \alpha_{init}(T_{init} - T_{ref})$$

Integral Approach

This secant approximation is adequate for small changes in temperature; however, for temperature-dependent properties, a derivative approximation is more accurate:

$$\alpha(T) = \frac{1}{L}\frac{\partial u}{\partial T}$$

such that the thermal strain is given by:

$$\varepsilon = \int_{T_{init}}^{T_{load}} \alpha(T)dT$$

which is the integral approach.

Temperature dependence for each material property, such as the coefficient of thermal expansion, may be defined as a linear function or a power series, and may include offsets and scale factors. These are input on the TABLEMi Bulk Data entries, and may be integrated analytically for each interval of the table.

Input

For the temperature-dependent composites, two parameters have been added. The COMPMATT parameter controls whether the smeared laminate properties are updated with the temperature-dependent material properties of the individual plies. It has a default value of NO and must be changed to YES to invoke the more accurate integral approach. The EPSILONT parameter controls whether the thermal strain is computed using an integral or a secant method. It has a default value of SECANT and must be changed to INTEGRAL to invoke the enhancement.

The MATT8 Bulk Data entry is used for temperature-dependent shell orthotropic material properties, and thermal strains may be input directly by using a negative TABLEMi ID for the coefficients of thermal expansion on the MATTi Bulk Data entry. All of the other bulk data entries and case control remain the same.

Output

For the strains, the thermal strains will be calculated using the smeared coefficients of thermal expansion at the current load temperature, in addition to using the appropriate secant or integral method. For the stresses and forces, the smeared material properties will be calculated at the proper temperature. For the ply strains and stresses, the thermal strain will be calculated using the coefficients of thermal expansion at the current load temperature and appropriate method for the specific ply, and the ply properties will be calculated at the proper temperature, rather than using the reference temperature on the PCOMP Bulk Data entry.

In addition, if the parameter COMPMATT is set, the composite elements will be included in the nonlinear stress/strain output.

Guidelines and Limitations

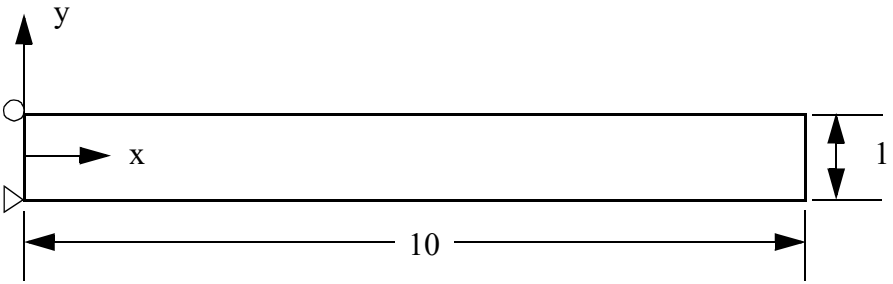
Composite capabilities are currently available with the QUAD4, QUAD8, TRIA3, and TRIA6 h-elements. Therefore, only the QUAD4 and TRIA3 element types will be supported. (Note that although smeared laminate properties may be applied to other two-dimensional elements as well, ply-by-ply postprocessing is not performed on those elements.)

These capabilities are not implemented for SOL 101, Linear Statics, nor SOL 129, Nonlinear Transient Response; use SOL 400.

Examples

(TDCRN_*.DAT)

In order to illustrate the new enhancements, a very simple problem will be solved for each permutation of the parameters. The problem consists of a simply-supported cantilever beam, ten units long by one unit wide, undergoing a uniform free expansion from $T_{init} = 100$ to $T_{load} = 200$ with $T_{ref} = 50$, as shown in the following figure:



Two independent versions of the model will comprise each test file: one will reference a PSHELL, and the other will reference a PCOMP with a single ply of the same thickness and orientation as the shell. The material is isotropic, although a MAT8/MATT8 will be used to demonstrate the new MATT8.

The coefficient of thermal expansion is temperature-dependent, as shown in the following table:

	Temperature (T)	CTE (α)
T_{ref}	50	0.0005
T_{init}	100	0.001
T_{load}	200	0.005

Most of the models will be run in SOL 106, Nonlinear Statics, though one will be run in SOL 101, Linear Statics, for comparison purposes. (Note again that the enhancements are not implemented in SOL 101.)

The six example problems are listed in the following table, along with the thermal strains computed using the equations given above. For the final problem, the thermal strains, marked with an asterisk, are input directly on the TABLEM4 entry as the quadratic integral of the linear property variation of the coefficient of thermal expansion.

	Solution Sequence	Parameters		Thermal Strain (ε_t)	
		COMPMATT	EPSILONT	PSHELL	PCOMP
TDCRN_L	101	n/a	n/a	0.10	0.05
TDCRN_NS	106	no	secant	0.70	0.05
TDCRN_YS	106	yes	secant	0.70	0.70
TDCRN_NI	106	no	integral	0.30	0.05
TDCRN_YI	106	yes	integral	0.30	0.30
TDCRN_YIE	106	yes	integral	0.30*	0.30*

Results from the six example problems are listed in the following table, along with the axial displacement at the final load step.

	Solution Sequence	Parameters		Axial Displacement (u_x)	
		COMPMATT	EPSILONT	PSHELL	PCOMP
TDCRN_L	101	n/a	n/a	1.0	0.5
TDCRN_NS	106	no	secant	7.0	0.5
TDCRN_YS	106	yes	secant	7.0	7.0
TDCRN_NI	106	no	integral	3.0	0.5
TDCRN_YI	106	yes	integral	3.0	3.0
TDCRN_YIE	106	yes	integral	3.0*	3.0*

The axial displacements consist of the thermal strains multiplied by the length, as expected; the strains are those given above; and the stresses are numerically zero.

Hyperelastic Material

This section describes the MATHP Bulk Data entry, which defines hyperelastic analysis of rubber-like material used in SOL 106 and SOL 129. For more complex material modeling of elastomers, use SOL 400.

The generalized Mooney-Rivlin strain energy function may be expressed as follows:

$$U(J, \bar{I}_1, \bar{I}_2) = \sum_{\substack{i+j=1 \\ \text{A00} = 0}}^{\text{NA}} \text{Aij}\,(\bar{I}_1 - 3)^i (\bar{I}_2 - 3)^j + \sum_{i=1}^{\text{ND}} \text{Di}\,(J - 1 - \text{AV}(T - T_0))^{2i}$$

where $\bar{I}_1$ and $\bar{I}_2$ are the first and second distortional strain invariants, respectively; $J = \det F$ is the determinant of the deformation gradient; and $2\text{D1} = K$ and $2(\text{A10} + \text{A01}) = G$ at small strains, in which K is the bulk modulus and G is the shear modulus. The model reduces to a Mooney-Rivlin material $\text{NA} = 1$ and to a Neo-Hookean material if $\text{NA} = 1$ and $\text{A01} = 0.0$. For Neo-Hookean or Mooney-Rivlin materials no continuation entry is required. T is the current temperature and T_0 is the initial temperature.

Conventional Mooney-Rivlin and Neo-Hookean materials are fully incompressible. Full incompressibility is not presently available but may be simulated with a large enough value of D1. A value of D1 higher than $10^3 \cdot (\text{A10} + \text{A01})$ is, however, not recommended.

Aij (material constants related to distortional deformation) and Di (material constants related to volumetric deformation) are obtained from least squares fitting of experimental data. One or more of four experiments (TAB1 to TAB4) may be used to obtain Aij. Di may be obtained from pure volumetric compression data (TABD). If all TAB1 through TAB4 are blank, Aij must be specified by the user. Parameter estimation, specified through any of the TABLES1 entries, supersedes the manual input of the parameters.

If $\text{ND} = 10$ and a nonzero value of D1 is provided or is obtained from experimental data in TABD, then the parameter estimation of the material constants Aij takes compressibility into account in the cases of simple tension/compression, equibiaxial tension, and general biaxial deformation. Otherwise, full incompressibility is assumed in estimating the material constants.

Reference

1. Narayanaswami, R., and H. M. Adelman, “Evaluation of the Tensor Polynomial and Hoffman Strength Theories for Composite Materials,” *Journal of Composite Materials*, Vol. II, 1977, p. 366.

5 Applied Loads

- Static Loads 298
- Point Loads 298
- Distributed Loads 298
- Inertia Loads 299
- Thermal and Enforced Element Deformation Loads 299
- Loads Due to Enforced Motion 303
- Frequency-Dependent Loads 304
- Time-Dependent Loads 305
- Simplified Static Loading Data in Dynamic Analysis 306
- Combining of Loads 308
- p-Element Loads and Constraints 309
- Summary of Nonlinear Static Loads 312
- Summary of Nonlinear Transient Loads 317

Static Loads

In MSC Nastran, static loads are applied to geometric and scalar points in a variety of ways, including:

- Loads applied directly to grid points
- Pressure loads on surfaces
- Distributed and concentrated loads on CBAR, CBEAM, and CBEND elements
- Gravity loads
- Centrifugal loads due to steady rotation
- Tangential loads due to angular acceleration
- Loads resulting from thermal expansion
- Loads resulting from enforced deformations of a structural element
- Loads resulting from enforced displacements at a grid point

Additional information on static loads is given in Section 3.6 of *The NASTRAN Theoretical Manual*. Any number of load sets can be defined in the Bulk Data Section. However, only those sets selected in the Case Control Section, as described in Case Control Commands, 191, will be used in the problem solution. The manner of selecting each type of load is specified on the associated Bulk Data entry description in the Bulk Data Entries (p. 1119) in the *MSC Nastran Quick Reference Guide*.

Point Loads

The FORCE entry is used to define a static load applied to a geometric grid point in terms of components defined by a local coordinate system. The orientation of the load components depends on the type of local coordinate system used to define the load. The directions of the load components are the same as those indicated on Figure 2-2 of Grid Point and Coordinate System Definition for displacement components. The FORCE1 entry is used if the direction is determined by a vector connecting two grid points, and a FORCE2 entry is used if the direction is specified by the cross product of two such vectors. The MOMENT, MOMENT1 and MOMENT2 entries are used in a similar fashion to define the application of a concentrated moment at a geometric grid point. The SLOAD entry is used to define a load at a scalar point. In this case, only the magnitude is specified since only one component of motion exists at a scalar point.

The FORCEAX and MOMAX entries are used to define the loading of specified harmonics on rings of conical shell elements. FORCE and MOMENT entries may be used to apply concentrated loads or moments to conical shell elements, providing that such points have been defined with a POINTAX entry.

Distributed Loads

The PLOAD1 entry is used to define either distributed or concentrated loads on CBAR and CBEAM elements. This entry is only used to define loads at intermediate points on CBAR and CBEAM elements. Applied loads at end points are defined with either FORCE or MOMENT entries. The CBARAO entry is used to define intermediate points on CBAR elements for which stress and/or force output is desired. The PLOAD1 entry is also used to define only distributed loads for the CBEND element. The distributed load is always applied along the entire length of the element and does not define loads

at intermediate points on the CBEND. The PLOAD1 entry cannot be used to define concentrated loads at intermediate stations on the CBEND element.

Pressure loads on three-node triangular and four-node quadrilateral elements are defined with a PLOAD2 entry. The positive direction of the loading is determined by the order of the grid points on the element connection entry, using the right hand rule. The magnitude and direction of the load is automatically computed from the value of the pressure and the coordinates of the connected grid points. The load is applied to the connected grid points. The PLOAD entry is used in a similar fashion to define the loading of any three or four grid points regardless of whether they are connected with two-dimensional elements. The PLOAD4 entry may be used with CQUAD4, CQUAD8, CQUADR, CTRIA3, CTRIA6, and CTRIAR elements. Since the surface hyperelastic elements CQUAD4, CQUAD8, CQUAD, CTRIA3, and CTRIA6 are plane strain elements, no pressure load may be applied on these elements.

The PRESAX entry is used to define pressure loading on the conical shell element. The PLOADX1 entry is used to define pressure loading on the CTRIAX6 element and on the hyperelastic CTRIAX and CQUADX elements.

Pressure loads on the CHEXA, CPENTA, and CTETRA elements are defined with the PLOAD4 entry. The magnitude and direction of the equivalent grid point forces are automatically computed using the isoparametric shape functions of the element to which the load has been applied.

Inertia Loads

The GRAV entry is used to specify a gravity load (or any other type of acceleration) by providing the components of the gravity vector in any defined coordinate system. The gravity load is obtained from the gravity vector and the mass matrix. The gravitational acceleration is not calculated at scalar points. The user is required to introduce gravity loads at scalar points directly.

The RFORCE entry is used to define a static loading condition due to angular velocity and/or acceleration. These loads are specified by the designation of a grid point that lies on the axis of rotation and by the components of rotational velocity or angular acceleration in any defined coordinate system. There are two methods to compute these loads. In the first method, the mass matrix is regarded as pertaining to a set of distinct rigid bodies connected to grid points. Deviations from this viewpoint, such as the use of scalar points or the use of mass coupling between grid points, can result in errors. The second method allows mass coupling, but does not allow offsets. The second method is the only method used for angular acceleration.

Thermal and Enforced Element Deformation Loads

Temperatures may be specified for selected elements. The temperatures for CROD, CBAR, CBEAM, CBEND, CONROD, or TUBE elements are specified on the TEMPRB Bulk Data entry. This entry specifies the average temperature on both ends and, in the case of the CBAR, CBEAM and CBEND elements, is used to define temperature gradients over the cross section. Temperatures for two-dimensional plate and membrane elements are specified on a TEMPP1, TEMPP2, or TEMPP3 Bulk Data entry. The user-defined average temperature over the volume is used to produce in-plane loads and stresses. Thermal gradients over the depth of the bending elements, or the resulting moments, may be used to produce bending loads and stresses.

If no thermal element data is given for an element, the temperatures of the connected grid points given on the TEMP, TEMPD, or TEMPAX entries are simply averaged to produce an average temperature for the element. The thermal expansion coefficients are defined on the material definition entries. Regardless of the type of thermal data, if the material coefficients for an element are temperature dependent by use of the MATTi entry, they are always calculated from the "average" temperature of the element. The presence of a thermal field does not imply the application of a thermal load. A thermal load will not be applied unless the user makes a specific request in the Case Control Section.

Enforced axial deformations can be applied to CROD, CBAR, and CBEAM elements. Enforced deformations, for example, are useful in the simulation of misfit or misalignment in engineering structures. As in the case of thermal expansion, the equivalent loads are calculated by separate subroutines for each type of structural element and are applied to the connected grid points. The magnitude of the axial deformation is specified on a DEFORM entry.

Temperature Loads

If temperature loads are present in one or more of the subcases, the thermal strain must be taken into account in the calculation of element stresses and forces. That is, a body subjected to a temperature field will deform in a stress free manner in the absence of constraints. However, if the deformation is resisted through boundary conditions and/or the presence of applied loads, the total actual strain, at a point must be reduced by the thermal strain, $\alpha(T - T_o)$, in order to obtain the stress at the point. For example, the stress-strain relation for a rod that undergoes thermal expansion and also sustains an axial force is written in the form

$$\varepsilon - \alpha(T - T_o) = \frac{\sigma}{E}$$

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, the user must supply a definition of the temperature field whenever element stresses and forces are required.

Consider the earlier example, with the modification that a temperature load is specified in SUBCASE 2.

```
        .
        .
        .

SUBCASE 1
  LOAD=2
SUBCASE 2
 TEMP(LOAD)=4
SUBCASE 10
  LOAD=10
SUBCOM 20
  SUBSEQ=.5,1.,1.6
  TEMP(LOAD)=4
        .
        .
        .
```

In this example, the multiplier for SUBCASE 2 is unity so that the temperature distribution required to calculate element forces and stresses for the combined loading case is precisely TEMP(LOAD)=4. If, on the other hand, the SUBSEQ entry is modified as indicated below, an additional TEMP(LOAD) must be defined.

```
        .
        .
        .

SUBCASE 1
  LOAD=1
SUBCASE 2
  TEMP(LOAD)=4
SUBCASE 10
 LOAD=10
SUBCOM 20
SUBSEQ=.5,.75,1.6
 TEMP(LOAD)=40
```

Here, TEMP(LOAD)=40 must reflect a temperature distribution of the form

$$(T_{40} - T_o) = .75(T_4 - T_o)$$

$$T_{40} = T_o + .75(T_4 - T_o)$$

This new TEMP(LOAD) must be supplied by the user through standard Bulk Data entries such as TEMPD, TEMP, TEMPP1, and TEMPP3.

In general, with n temperature loading conditions in the subcases immediately preceding a SUBCOM, the temperature distribution that must be defined under the SUBCOM if element force or stress output is desired is of the form

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

where the coefficients a_i are defined on the SUBSEQ entry under the SUBCOM in question. The reference temperature T_o, defined on MATi Bulk Data entries, can conveniently be defined as zero in order to simplify the calculation of the temperature distribution for the combined loading condition of the SUBCOM.

The same approach is used with combined loading condition defined through a SYMCOM Case Control entry. However, the use of SYM subcases and SYMCOM entries implies the use of the concepts of reflective symmetry, in which case it is strongly recommended that the DIH option of cyclic symmetry be used (Cyclic Symmetry). The appropriate boundary conditions are generated internally in MSC Nastran, and the user simply defines the actual temperature distribution on each portion of the structure without regard to decomposing the temperatures and/or mechanical loads into symmetric and antisymmetric components.

Element Deformation

The DEFORM Bulk Data entry enables the user to define enforced axial deformations for the CBAR, CBEAM, CROD, and CTUBE elements. The nature of this enforced deformation is analogous to a turnbuckle that may be adjusted to provide a desired axial deformation in an element. The problem is analogous to the temperature problem discussed in the previous section and, for example, the appropriate stress-strain relation for a rod element of length that undergoes an enforced axial deformation, δ is given by the following equation:

$$\varepsilon - \frac{\delta}{l} = \frac{\sigma}{E}$$

Consider the final example of the previous section.

```
        .
        .
        .

SUBCASE 1
 LOAD=1
SUBCASE 2
  DEFORM=5
SUBCASE 10
  LOAD=10
SUBCOM 20
  SUBSEQ=.5,.75,1.6
  DEFORM=50
        .
        .
        .
```

Here DEFORM = 50 must reflect a set of enforced deformations of the form

$$\delta_{50} = .75\delta_5$$

For n DEFORM loading conditions in the subcases that immediately precede a SUBCOM, the DEFORM loading condition that must be defined under the SUBCOM, if element force and stress output is desired, is of the form

$$\delta_{SUBCOM} = \sum_{i=1}^{n} a_i \delta_i$$

Loads Due to Enforced Motion

Statics

Enforced zero displacements may be specified on GRID, SPC, or SPC1 entries. Zero displacements that result in nonzero forces of constraint are usually specified on SPC or SPC1 entries. If GRID entries are used, the constraints become part of the structural model and modifications cannot be made at the subcase level.

Nonzero enforced displacements may be specified on SPC or SPCD entries. The SPC entry specifies both the component to be constrained and the magnitude of the enforced displacement. The SPCD entry specifies only the magnitude of the enforced displacement. When an SPCD entry is used, the component to be constrained must be specified on either an SPC or SPC1 entry. The use of the SPCD entry avoids the decomposition of the stiffness matrix when changes are only made in the magnitudes of the enforced displacements.

The equivalent loads resulting from enforced displacements of grid points are calculated by the program and added to the other applied loads. The magnitudes of the enforced displacements are specified on SPC entries (SPCAX in the case of conical shell problems) in the global coordinate system. The application of the load is automatic when the user selects the associated SPC set in the Case Control Section.

The SPCD is selected by a load request in the Case Control Section.

Dynamics

Enforced motion in earlier versions of MSC Nastran used a nonzero value in the TYPE field (field 5) on the TLOAD1 and TLOAD2 entries to specify whether an enforced displacement, velocity, or acceleration were to be applied using the large mass method. Although these integer values are still allowed, the field has now been generalized to permit character input, and a similar TYPE field has been included in field 8 of the RLOAD1 and RLOAD2 entries to support enforced motion in frequency response analysis. The new TYPE field for these dynamic loading/enforced motion entries now reads as follows:

Value	Excitation Function
0, or LOAD	Applied load (Default)
1, or DISP	Enforced displacement using large mass or SPC/SPCD data
2, or VELO	Enforced velocity using large mass or SPC/SPCD data
3, or ACCE	Enforced acceleration using large mass or SPC/SPCD data

The character fields may be shortened to as little as a single character on input, if desired.

In order to allow maximum input flexibility and to support existing input data files that had used the large mass method, the decision regarding whether to use the older, large mass method, or the newer direct enforced motion will be made based on a search for SPC/SPC1/SPCD data as follows: If the TYPE field on the TLOAD1/2 entries indicates an enforced motion, the EXCITEID field will first be assumed to point to SPC-type data. If not present, MSC Nastran will then assume the excitation is enforced motion using large masses and will then look for DAREA and various static loading data, just as

in the case of applied loads. RLOAD1 and RLOAD2 entries will only look for SPC-type data in the case of enforced motion. For all entries (TLOAD1/2, RLOAD1/2), if the TYPE field indicates an applied load, the program will search only for static loading data.

With this feature, enforced motion data is supplied via the SPC/SPC1/SPCD Bulk Data entries. (Note that components specified in SPCD data must be also be referenced on SPC or SPC1 entries.) Enforced motion could be defined solely with SPC data (without the use of SPCD's), but such usage is discouraged in favor of SPC/SPC1 data in connection with SPCD's. In this way, SPCD entries are selected via the EXCITEID field in the dynamic load data, while the SPC/SPC1 sets are used to identify the constrained displacement set in the Case Control Section. With SPC data alone, enforced motion is applied solely via the Case Control Section. The entries referenced by the EXCITEID field on the TLOAD1/2 or RLOAD1/2 entries do not have to exist. This method of specifying enforced motion is less elegant and somewhat more difficult to interpret.

Procedure for Using the Enforced Motion Feature

In summary, the procedure for applying enforced motion in a dynamic analysis:

1. Specifies the appropriate type of enforced motion to be applied via the TYPE field in RLOAD1/RLOAD2 (field 8) or TLOAD1/TLOAD2 (field 5) Bulk Data entry, as appropriate.
2. Defines the desired enforced motion using the SPCD Bulk Data entry. The set IDs of these SPCD data must match the IDs appearing in the EXCITEID fields of the corresponding dynamic load data in step 1.
3. Ensures that the components referenced in the SPCD Bulk Data entry above are also specified in the SPC1 Bulk Data entry and this SPC1 is also selected in the Case Control Section.

Frequency-Dependent Loads

A discussion of frequency response calculations is given in Frequency Response Analysis (p. 123) in the *MSC Nastran Dynamic Analysis User's Guide*. The DLOAD entry is used to define linear combinations of frequency dependent loads that are defined on RLOAD1 or RLOAD2 entries. The RLOAD1 entry defines a frequency dependent load of the form

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

where A is defined on a DAREA entry, $C(f)$ and $D(f)$ are defined on TABLEDi entries, θ is defined on a DPHASE entry, τ is defined on a DELAY entry, and f is defined on a FREQ, FREQ1 or FREQ2 entry. The RLOAD2 entry defines a frequency-dependent load of the form

$$\{P(f)\} = AB(f)e^{i\{\phi(f) + \theta - 2\pi f\tau\}}$$

where A is defined on a DAREA entry, $B(f)$ and $\phi(f)$ are defined on TABLEDi entries, θ is defined on a DPHASE entry, and τ is defined on a DELAY entry. The coefficients on the DAREA, DELAY, and DPHASE entries may be different for each loaded degree of freedom. The loads are applied to the specified components in the global coordinate system.

Static loading data may be used to supply DAREA values. Using a LOADSET Case Control request with LSEQ Bulk Data entries, the user can build DAREA load vectors from any valid static load set. This option is available in the frequency response solutions and in superelement analysis.

A discussion of random response calculations is given in Section 12.2 of *The NASTRAN Theoretical Manual*. The RANDPS entry defines load set power spectral density factors for use in random analysis of the form

$$S_{jk}(f) = (X + iY)G(f)$$

where $G(f)$ is defined on a TABRNDi entry. The subscripts j and k define the subcase numbers of the load definitions. If the applied loads are independent, only the diagonal terms ($j = k$) need be defined. The RANDT1 entry is used to specify the time lag constants for use in the computation of the autocorrelation functions.

Time-Dependent Loads

A discussion of transient response calculations is given inTransient Response Analysis (p. 199) in the *MSC Nastran Dynamic Analysis User's Guide.* The DLOAD entry is used to define linear combinations of time-dependent loads that are defined on TLOAD1 and TLOAD2 entries. The TLOAD1 entry defines a time-dependent load of the form

$$\{P(t)\} = \{AF(t-\tau)\}$$

where A is defined on a DAREA entry, τ is defined on a DELAY entry, and $F(t-\tau)$ is defined on a TABLEDi entry. The TLOAD2 entry defines a time-dependent load of the form

$$\{P(t)\} = \begin{cases} 0 & , \ t < (T_1 + \tau) \text{ or } t > (T_2 + \tau) \\ A\tilde{t}^B e^{C\tilde{t}} \cos(2\pi F\tilde{t} + P) & , \quad (T_1 + \tau) \le t \le (T_2 + \tau) \end{cases}$$

where T_1 and T_2 are time constants, $\tilde{t} = t - T_1 - \tau$, A and τ are the same as on the TLOAD1 entry, B is the growth coefficient, C is the exponential coefficient, F is the frequency, and P is the phase angle.

The coefficients on the DAREA and DELAY entries may be different for each loaded degree of freedom. The loads are applied to the specified components in the global coordinate system.

Two methods are available to automatically create DAREA values from static load data. Simple static loads that supply load values directly to grid points (such as FORCE, MOMENT, SLOAD, and PLOAD Bulk Data entries) will add to the DAREA factors when the load identification matches the requested set.

More complicated loads may be generated using the LOADSET Case Control command. The requested LSEQ Bulk Data entry matches DAREA sets to static loading sets. Any type of static load (e.g., LOAD, GRAV, RFORCE, and SPCD Bulk Data entries) may be generated for either superelement or residual only (non-superelement) models.

Nonlinear effects are treated as an additional applied load vector, for which the components are functions of the degrees of freedom. This additional load vector is added to the right side of the equations of motion and treated along with the applied load vector during numerical integration. It is required that the points to which the nonlinear loads are applied and the degrees of freedom on which they depend be members of the solution set; i.e., that they cannot be degrees of freedom eliminated by constraints. It is further required that, if a modal formulation is used, the points referenced by the nonlinear loads be members of the set of extra scalar points introduced for dynamic analysis (see Section 11.2 of *The NASTRAN Theoretical Manual).*

There are four different types of nonlinear transient forcing functions: arbitrary function generators, multiplies, positive power functions, and negative power functions. For a discussion of nonlinear forcing functions, see Section 11.2 of *The NASTRAN Theoretical Manual.*

The NOLIN1 entry defines a nonlinear load of the form

$$P_i(t) = S_i T(u_j)$$

where P_i is the load applied to u_i, S_i is a scale factor, $T(u_j)$ is a tabulated function defined with a TABLEDi entry, and u_j is any permissible displacement component or velocity component.

The NOLIN2 entry defines a nonlinear load of the form

$$P_i(t) = S_i\, u_j\, u_k$$

where u_j and u_k are any permissible pair of displacement components and/or velocity components. They may be the same.

The NOLIN3 entry defines a nonlinear load of the form

$$P_i(t) = \begin{cases} -S_i(u_j)^A & , u_j > 0 \\ 0 & , u_j \le 0 \end{cases}$$

where A is an exponent and u_j is any permissible displacement component or velocity component.

The NOLIN4 entry defines a nonlinear load of the form

$$P_i(t) = \begin{cases} -S_i(u_j)^A & , u_j < 0 \\ 0 & , u_j \ge 0 \end{cases}$$

where A is an exponent and u_j is any permissible displacement component or velocity component.

Simplified Static Loading Data in Dynamic Analysis

Prior to MSC Nastran 2001, static loading data in dynamic loads had to be included through the use of the LOADSET/LSEQ combination. The LOADSET command, which is selected in the Case Control Section, points to a LSEQ Bulk Data entry. The LSEQ Bulk Data entry, in turn, acts as a "bridge" between the DAREA ID and the static load ID/thermal load ID. If the DAREA ID of the selected dynamic load (RLOAD1/RLOAD2/TLOAD1/TLOAD2/ACSRCE) matches the DAREA ID on the selected LSEQ entry, then all static loads and thermal loads referenced in the selected LSEQ Bulk Data entry will be selected automatically. Note that the DAREA entry does even have to exist in this case.

It is no longer necessary for the user to explicitly specify LOADSET/LSEQ combination in order to employ static loading data in dynamic analysis. Instead, when the user selects a dynamic load, all static loads and thermal loads that have the same ID as the DAREA ID on the dynamic load entry are automatically selected. The usage of this feature is illustrated below.

Pre-2001 Usage

```
.
CEND
.
$ THE FOLLOWING REQUEST SELECTS LSEQ ENTRY 1000
LOADSET = 1000
$ THE FOLLOWING REQUEST SELECTS RLOAD1 ENTRY 150
DLOAD = 150
.
BEGIN BULK
$ THE FOLLOWING ENTRY, WHICH ACTS AS A "BRIDGE"
$ BETWEEN EXCITEID ID 100 AND STATIC LOAD ID 200,
$ CAUSES THE SELECTION OF PLOAD4,200
LSEQ,1000,100,200
PLOAD4,200,…
RLOAD1,150,100,…
.
```

Current Usage

```
.
CEND
.
$ THE FOLLOWING REQUEST SELECTS RLOAD1 ENTRY 150
DLOAD = 150
.
BEGIN BULK
$ THE FOLLOWING PLOAD4 IS AUTOMATICALLY SELECTED
$ BECAUSE ITS ID OF 100 MATCHES THE EXCITEID ID OF THE
$ SELECTED RLOAD1 ENTRY
PLOAD4,100,…
RLOAD1,150,100,…
$ THE ABOVE USAGE IS EQUIVALENT TO THE USER
$ SELECTING AN LSEQ BULK DATA OF THE FORM:-
$ LSEQ,SET_ID,100,100,100
.
```

The DAREA Bulk Data entry is used to specify point loads in dynamic analysis. In the case of grid points, these loads are implicitly assumed to be in the displacement (or local) coordinate systems of those points. Until now, the usage of these data in superelement analysis had been restricted to the residual structure. Prior to MSC Nastran 2001, to specify DAREA-type loads for upstream superelements in dynamic analysis, the user was forced to specify the LOADSET/LSEQ combination in conjunction with the appropriate FORCE, FORCE1, FORCE2, MOMENT, MOMENT1, MOMENT2, or SLOAD Bulk Data entry. An enhancement incorporated into MSC Nastran avoids this problem by automatically converting all DAREA Bulk Data entries for grid and scalar points into the appropriate equivalent FORCE/MOMENT/SLOAD Bulk Data entries.

Two additional, important advantages of the above enhancement follow.

- When performing dynamic analysis using the modal approach, it is very often desirable to employ residual vectors to improve the quality of the solution. The user must specify static loads at those points that are dynamically excited. However, with the above automatic conversion feature, it is not necessary to explicitly specify static loads for the purpose of residual vector calculations. Such loads are automatically generated by the MSC Nastran.
- Prior to MSC Nastran 2001, the usage of the DAREA Bulk Data entry had been restricted to dynamic analysis. The above automatic conversion feature permits the DAREA entry to be used in static analysis as well. This is particularly advantageous when the user wishes to apply loads at grid points in the displacement (or local) coordinate systems of those grid points.

When the above conversion is made, a message is issued at the end of the Preface module informing the user of the conversion. This message also outputs an image not only of each converted DAREA Bulk Data entry, but also of the corresponding FORCE/MOMENT/SLOAD Bulk Data entry into which it has been converted.

Combining of Loads

The LOADCYH Bulk Data entry is used to define the harmonic coefficients of a load in cyclic symmetry analysis. The LOADCYN entry is used to define a physical load in cyclic symmetry analysis, and the LOADCYT entry specifies a tabular load input for axisymmetric cyclic symmetry problems.

The LOAD entry defines a static loading condition that is a linear combination of load sets consisting of loads applied directly to grid points, pressure loads, gravity loads, and centrifugal forces. This entry must be used if gravity loads are to be used in combination with loads applied directly to grid points, pressure loads, or centrifugal forces. The application of the combined loading condition is requested in the Case Control Section by selecting the set number of the LOAD combination.

It should be noted that the equivalent loads (thermal, enforced deformation, and enforced displacement) must have unique set identification numbers and be separately selected in Case Control. For any particular solution, the total static load will be the sum of the applied loads (grid point loading, pressure loading, gravity loading, and centrifugal forces) and the equivalent loads.

The LSEQ entry defines static load sets that may be referenced by dynamic load entries RLOAD1, RLOAD2, TLOAD1, and TLOAD2. The LSEQ entry may be used to select static loading conditions in the superelement static solution sequences and is used to correlate static loading conditions with dynamic loads in dynamic analysis.

The loads actually applied to points may be requested by use of the OLOAD entry.

Temperature Loads and Enforced Deformations in Subcase Combinations (SUBCOMS)

The use of the Case Control Section entries SUBCOM and SYMCOM in static analysis provides the user with a relatively straightforward method to combine the results of two or more immediately preceding subcases or SYM cases respectively. For example, the following portion of a Case Control Section illustrates the use of the SUBCOM.

```
        .
        .
        .

SUBCASE 1
  LOAD=1
  TEMP(LOAD)=7
SUBCASE 2
  LOAD=2
SUBCASE 10
  LOAD=10
  DEFORM=11
SUBCOM 20
  SUBSEQ=.5,1.,1.6
        .
        .
        .
```

Here, SUBCOM 20 provides results for a combined loading condition of the form

$$0.5 \cdot (\text{LOAD} = 1 + \text{TEMP(LOAD)} = 7) + 1.0 \cdot (\text{LOAD} = 2) + 1.6 \cdot (\text{LOAD} = 10 + \text{DEFORM} = 11)$$

In MSC Nastran, the results for this combined loading condition are efficiently obtained by forming the linear combination of the displacement vectors obtained in the previous subcases. Thus,

$$\{U\}_{\text{SUBCOM 20}} = 0.5 \cdot \{U\}_{\text{SUBCASE 1}} + 1.0 \cdot \{U\}_{\text{SUBCASE 2}} + 1.6 \cdot \{U\}_{\text{SUBCASE 10}}$$

and the element forces and stresses are calculated in the usual manner from the displacement vector for the combined loading condition. This technique is quite straightforward when the several loading conditions are composed of forces and moments. However, if any of the previous subcases contain thermal loads and/or enforced element deformations, the user must exercise some care in order to obtain correct element stresses and forces for the combined loading condition defined through the SUBCOM.

p-Element Loads and Constraints

Point Loads

The FORCE entry can be used to define point loads applied to grid points. No point loads can be applied to finite element edges or faces.

Line and Pressure Loads in p-Element Analysis

The GMLOAD entry is used to specify line and pressure loads by:

- Applying directly to the finite elements entities, FEEDGEs and FEFACEs.

- Applying to the geometry entities GMCURV and GMSURF.

Body Loads

Body loads consists of thermal loads and gravity loads.

For thermal loading, the TEMP Bulk Data entry is used for a temperature distribution that is trilinear over the element, whereas the TEMPF entry is used over a set of elements. The initial or reference temperature of a body must be supplied using these entries.

Gravity loads are defined by the GRAV entry.

Boundary Conditions

The SPC and SPCD entries are used for the point constraints that are allowed only on corner GRID points.

The GMSPC entry is used to define zero constraints for FEEDGEs, FEFACEs, GMCURVs, and GMSURFs.

For nonzero constraints, the GMBC entry is used.

In general, for multiple input data for FEEDGE, FEFACE, GMCURV, and GMSURF entities, the hierarchy set to resolve these conflicts is:

- GRIDs
- FEEDGEs
- GMCURVs
- FEFACEs
- GMSURFs

These priorities apply for the constraints and the enforced displacement input data.

The rules for nonzero boundary constraints:

- Displacement values specified for each component of a given GMSURF entry are applied to the same component of all GRID, FEEDGE and FEFACE degrees of freedom that lie within the GMSURF.
- Displacement values specified for each component of different GMSURF entries are averaged and applied to the same component of all GRID, FEEDGE degrees of freedom that are shared by (common to) the multiple GMSURFs.
- Displacement values specified for a given FEFACE entry are applied to all GRID, FEEDGE and FEFACE degrees of freedom that lie within the FEFACE. This data overrides the data that is specified for all the components of the given GRID, FEEDGE and FEFACE degrees of freedom that lie within the FEFACE by using GMSURF entries.
- Displacement values specified for each component of different FEFACE entries are averaged and applied to the same component of all GRID, FEEDGE degrees of freedom that are shared by (common to) the multiple FEFACEs. This data overrides the data that is specified for all the components of the given FEEDGE and GRIDs by using GMSURF entries.

- Displacement values specified for each component of a given GMCURV entry are applied to the same component of all GRID, FEEDGE, degrees of freedom that lie within the GMCURV. This data overrides the data for all components that is specified for the given FEEDGE and GRIDs by using GMSURF and FEFACE entries.
- Displacement values specified for each component of different CURV entries are averaged and applied to the same component of all GRID degrees of freedom that are shared by (common to) the multiple GMCURVs. This data overrides the data for all the components that is specified for the given FEEDGE and GRIDS by using GMSURF and FEFACE entries.
- Displacement values specified for each component of a given FEEDGE entry are applied to the same component of all GRID, FEEDGE, degrees of freedom that lie within the FEEDGE. This data overrides the data for all components that is specified for the given FEEDGE by using GMCURV or FEFACE or GMSURF entries.
- Displacement values specified for each component of different FEEDGE entries are averaged and applied to the same component of all GRID degrees of freedom that are shared by (common to) the multiple FEEDGEs. This data overrides the data for all the components that is specified for the given FEEDGE by using GMCURV or FEFACE or GMSURF entries.
- Grids have the highest priority, i.e., any value/property specified using a GRID entry overrides all other information associated with that GRID. If multiple entries are used for a given GRID, e.g., multiple SPCs, then the existing rules of MSC Nastran govern (SPCs are combined, FORCE fare added, SPCDs for the same component are not allowed).
- It is important to recall that these displacements are assumed to be in the global coordinate system and that the interconsistency of the output coordinate systems of the various GRIDs, FEEDGEs, FEFACEs is not checked.

Resolution of Conflicting Zero Constraints

In general, the hierarchy set to resolve the conflicts arising in the enforced displacement input data is the same as for the constraints:

- GRIDs
- FEEDGEs
- GMCURVs
- FEFACEs
- GMSURFs

This means that:

- The specified component of a given GMSURF entry that is to be constrained is also constrained on all GRID, FEEDGE and FEFACE degrees of freedom that lie within the GMSURF.
- The specified component of different SURF entries are combined and applied to all GRID, FEEDGE degrees of freedom that are shared by (common to) the multiple GMSURFs.
- The specified component of a given FEFACE entry that is to be constrained is also constrained on all GRID, FEEDGE and FEFACE degrees of freedom that lie within the FEFACE. This data overrides the data that is specified for the given FEFACE by using GMSURF entries.

- The specified component of different FEFACE entries are combined and applied to all GRID, FEEDGE degrees of freedom that are shared by (common to) the multiple FEFACE. This data overrides the data that is specified for the given FEEDGE and GRIDs by using GMSURF entries.
- The specified component of a given GMCURV entry that is to be constrained is also constrained on all GRID, FEEDGE degrees of freedom that lie within the GMCURV. This data overrides the data that is specified for the given FEEDGE by using GMSURF or FEFACE entries.
- The specified component of different GMCURV entries are combined and applied to all GRID degrees of freedom that are shared by (common to) the multiple GMCURVs. This data overrides the data that is specified for the given FEEDGE by using GMCURV or FEFACE entries.
- The specified component of a given FEEDGE entry that is to be constrained is also constrained on also constrained on all GRID, FEEDGE degrees of freedom that lie within the FEEDGE. This data overrides the data that is specified for the given FEEDGE by using GMCURV or FEFACE or GMSURF entries.
- The specified component of different FEEDGE entries are combined and applied to all GRID degrees of freedom that are shared by (common to) the multiple FEEDGEs. This data overrides the data that is specified for the given FEEDGE by using GMCURV or FEFACE or GMSURF entries.
- Grids have the highest priority, i.e., any value/property specified using a GRID entry overrides all other information associated with that GRID. If multiple entries are used for a given GRID, e.g., multiple SPCs, then the existing rules of MSC Nastran govern (SPCs are combined, FORCE is added, SPCDs for the same component are not allowed).
- It is important to recall that the interconsistency of the output coordinate systems of the various GRIDs, FEEDGEs, FEFACEs is not checked.

Summary of Nonlinear Static Loads

Loads Overview

The static loads in nonlinear analysis consist of a subset of the total MSC Nastran static load set. Most of the relevant loads data applicable to the linear static analysis are also applicable to nonlinear static analysis except for the Bulk Data entry DEFORM. The enforced displacements can be applied as loads using the Bulk Data entry SPCD, that defines an enforced motion of a grid in a specific fixed direction. Each grid point with an enforced displacement must also appear on an SPC or SPC1 entry.

The loads are selected in each subcase by the LOAD Case Control command. The load for a subcase is subdivided into the number of increments specified for the subcase. The solution strategy in nonlinear is to apply the loads in an incremental fashion until the desired load level is reached. The algorithms “remember” the loads from one subcase to the next. If the load reaches the desired level in a subcase and if the load description is left out in the subsequent subcase inadvertently, the nonlinear solution algorithm will begin in an incremental fashion to remove the load. The loads described below make up the valid subset applicable for nonlinear static analysis.

Concentrated Loads

The concentrated loads consist of two categories: those that are stationary in direction throughout the analysis and those that follow the grid motion. The first category consists of the following:

FORCE
MOMENT
SLOAD

The second category consists of the following:

FORCE1
FORCE2
MOMENT1
MOMENT2

The three FORCE entries differ only in the manner in which the direction of the force is specified. FORCE uses the components of a vector. FORCE1 uses two grid points, not necessarily the same as the loaded grid points. FORCE2 defines the direction of the force as a vector that is the vector product of two other vectors. The distinctions between the three MOMENT entries are similar. The SLOAD entry is used to apply loads to scalar points.

Distributed Loads

The distributed loads consist of two categories: those that are stationary in direction throughout the analysis and those that follow the element motion. The first category consists of the PLOAD1.

The second category consists of the following:

PLOAD
PLOAD2
PLOAD4

Loads may be applied to the interiors of finite elements. The PLOAD1 entry defines concentrated and linearly distributed forces and moments to the CBEAM element. However, the stress calculation for nonlinear CBEAM element ignores the distributed loads, but lump them onto the nodal points. The PLOAD entry is exceptional in that it references three or four arbitrarily located grids rather than an element. The PLOAD2 entry defines a uniform static pressure load applied to the CQUAD4 or CTRIA3 elements. The PLOAD4 defines a pressure load on surfaces of CHEXA, CPENTA, CTETRA, CTRIA3, and CQUAD4 elements that need not be normal to the surface. Since the surface hyperelastic elements CQUAD4, CQUAD8, CQUAD, CTRIA3, and CTRIA6 are plane strain elements, no pressure load may be applied on these elements.

Mass Related Loads (GRAV and RFORCE Entries)

The mass related loads are gravity and centrifugal loads, which require specification of mass data. The gravity loads that can be specified on the GRAV Bulk Data entry are stationary in direction throughout the analysis. The GRAV entry is used to define the direction and magnitude of the gravity vector in some identified coordinate system. The components of the gravity vector are multiplied by the mass matrix to obtain the components of the gravity force at each grid point.

Centrifugal loads are specified on the RFORCE Bulk Data entry. RFORCE defines the components of a spin vector that is used internally to compute centrifugal forces. Each component of the spin vector is multiplied by a scale factor.

Thermal Loads

Thermal loads are selected in each subcase by the Case Control command TEMP(LOAD). Thermal loading that is a follower loading is accomplished through the following entries:

TEMP
TEMPD
TEMPP1
TEMPP3
TEMPRB

The TEMP and TEMPD provide for grid point temperature specification which are then interpolated to points within elements. For the nonlinear CQUAD4 and CTRIA3 elements the TEMPP1 provides for an average temperature and thermal gradient in the thickness direction and the TEMPP3 provides for temperature at points through the thickness. TEMPP1 and TEMPP3 are not available for the plane strain hyperelastic elements CQUAD4, CQUAD8, CQUAD, CTRIA3, and CTRIA6. The TEMPRB provides for average temperature at ends A and B for the CROD and CBEAM element. Average temperature specified directly for an element will take precedence over the temperature interpolated from the elements connected grid points.

The Case Control command TEMP(INIT) is required above subcase level to specify an initial temperature distribution. The use of TEMP(MAT) or TEMP(BOTH) will cause a fatal error. Subcases that do not contain a TEMP(LOAD) will default to TEMP(INIT), which implies no thermal load. Thermal loads are often used to provide initial strains.

Load Combinations

Various load specifications can be combined using the LOAD, CLOAD, and LSEQ entries.

The LOAD provides for the linear combination of static load sets. The static loads (LOAD, FORCE, etc.) applied to the upstream superelements cannot be referenced by the LOAD Case Control command. The CLOAD entry is designed to apply static loads to upstream superelements by combining loads defined in LSEQ entries. The (CLOAD, LSEQ) combination allows the nonlinear algorithms to apply in an incremental fashion on the upstream superelement loads to the boundary of the residual structure.

The CLOAD defines a static load combination for superelement loads acting on residual boundary and the LSEQ defines a static load combination for superelement upstream loads. The LSEQ assigns load vectors to the superelements and numerically labels them via the DAREA field entry value. The LID and TID field entries point to the appropriate load and temperature distribution entries. The CLOAD defines a static load as a linear combination of previously calculated superelement loads defined by the LSEQ. The IDVi fields correspond directly to the numeric labels defined via the DAREA fields of the LSEQ entries.

The LSEQ is selected by a LOADSET Case Control command above any residual subcase. Only one LOADSET may appear in Case Control. The hierarchy of the loads data is shown schematically in Figure 5-1. An example for the Case Control set-up follows:

```
        .
        .
        .

SEALL=ALL
LOADSET = 1000 $ Selects LSEQ 1000 for upstream loads
SUPER = ALL $ Identify SEs to process
        .
        .
        .

DISP = ALL
        .
        .
        .

        .
        .
        .

        .
        .
        .

$ NONLINEAR SOLUTIONS FOR RESIDUAL SUPERELEMENTS
SUBCASE 10
CLOAD = 1001 $ Refers to CLOAD Bulk Data
NLPARM = 12 $ Iteration control
        .
        .
        .

SUBCASE 20
CLOAD = 1002
LOAD = 10 $ Residual SE load
NLPARM = 22
        .
        .
        .
```

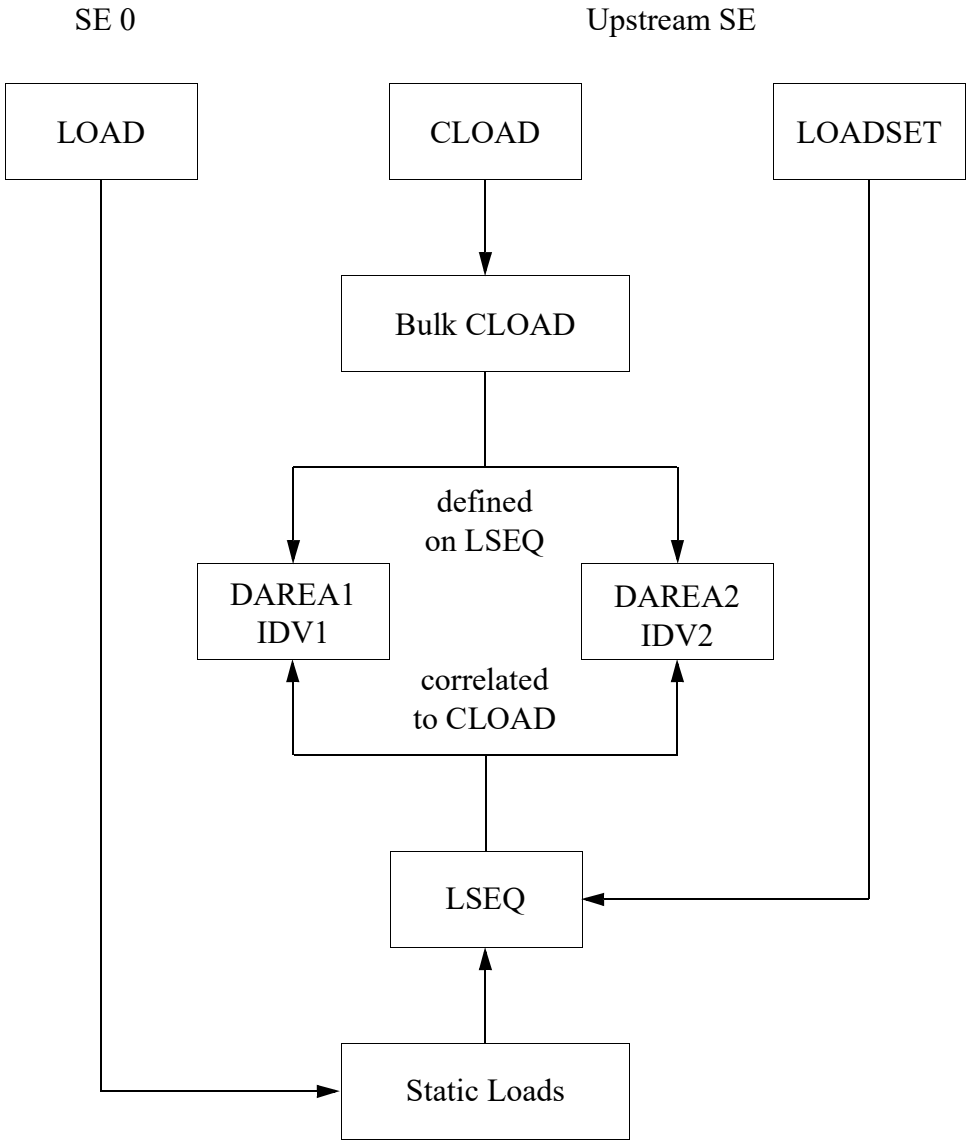


Figure 5-1 Hierarchy of Load Data

Summary of Nonlinear Transient Loads

Transient Loads And Initial Conditions

The methods employed to define loads in nonlinear transient analysis are similar to those used in the linear solutions. A single degree of freedom or a set of grid points may be loaded with force pattern that varies with time. Functions may be tabular such as an earthquake or a booster liftoff, or they may be simple analytic functions such as a sine wave. Simple static load sets generated in both upstream superelements and the default residual may be used to create the dynamic loads. They may be scaled and combined with other loads to simulate complex loading problems.

For special problems involving simple scalar nonlinearities, the traditional nonlinear loads (NOLINi option) may be used in both types of transient solutions. These are useful for simulating nonlinear damping mechanisms and rotational coupling not provided by the CGAP elements.

Transient loads define the loadings as functions of time and the location. They can be a load applied at a particular degree of freedom, pressure over the surface area, or the body force simulating an acceleration. The time history is provided by the TLOADi Bulk Data entries and the static loads may be converted to dynamic loads using LSEQ entry. The transient dynamic loads are selected in the Case Control Section by the following commands:

DLOAD	Selects TLOADi or DLOAD Bulk Data entries. Each stage of the transient solution may be defined with a separate SUBCASE, each with its own DLOAD request. The DLOAD entry is restricted to act only in the time defined in its subcase.
LOADSET	Selects LSEQ entries. Must be placed above the solution subcases and/or in the Superelement Subcases.
NOLINEAR	Selects NOLINi entries. May not be changed between solution subcases.
NLLOAD	Required for NOLINi output request.

Bulk Data entries associated with the transient loads in SOL 129 are briefly described below:

TLOAD1, TLOAD2	Required data defining the coupling between load factors (DAREA entry) and functions of time (TABLEDi, DELAY, or coefficients). Specifies the time functions, $F_i(t)$, for dynamic load i. TLOAD1 defines the load history in TABLEDi format while TLOAD2 defines the load history by an analytic expression. Note that each TLOADi entry must have a unique set identification number.
DAREA	Defines load scale factors, A_{ij}, for each degree of freedom, u_j requested in set i. The DAREA factors may also be derived from static load entries with the LSEQ Bulk Data entry.
LSEQ	Generates transient load history for static loads and controls the assembly of static load vectors to be used in dynamic analysis. Each set is equivalent to a subcase in Case Control. It will cause a static load vector A_j to be generated and will label it with the DAREA identification. This is useful for modeling problems with distributed transient loading such as GRAV loads or pressurized areas.

DLOAD	Combines different TLOADi functions into a single set and scales each by a factor, C_{ki} as shown in . Performs the same function as the static LOAD Bulk Data input. Unlike statics, this is the only method to combine loads in dynamic analysis.	
TABLEDi, i=1,2,3,4	Used with TLOAD1 entries to specify general time varying tabular functions. Since extrapolation is used beyond the range of data the user is cautioned to be careful with the end points.	
DELAY	Specifies the delay time for applying the forcing function defined in TLOADi to each GRID point. The DELAY entry is associated with the degree of freedom and not the forcing function in each TLOADi. This option is useful for defining loads that travel across a structure such as an oblique wave along a dam or a bump in the road exciting the wheels of a traveling vehicle. This saves the analyst the effort of defining multiple TLOADi entries for simple time lags.	
NOLINi, i=1,2,3,4	Define specific nonlinear scalar loads as functions of velocity and/or displacement. Basic analog nonlinear transfer functions are defined. However, these functions do not generate tangent matrix terms and the Newton iteration method may experience difficulties in convergence. Line search and BFGS solution methods are recommended for these models.	
	NOLIN1	Nonlinear transient load as a tabular function.
	NOLIN2	Nonlinear transient load as products of two variables.
	NOLIN3	Nonlinear transient load as a positive variable raised to a power.
	NOLIN4	Nonlinear transient load as a negative variable raised to a power.
NLRGAP	The NLRGAP is an idealization of the contact between any circular shaft enclosed by a circular housing (Figure 5-2). In this idealization, grid point A represents the outer surface of the shaft while grid point B represents the inner surface of the housing. Grid points A and B are coincident and must have a parallel displacement coordinate system. A rudimentary friction capability is included to account for the friction-induced torque and friction-induced lateral loads that occur with a spinning shaft. However, friction can be neglected, in which case the solution is consistent with that for a nonrotating shaft in a frictionless housing.	

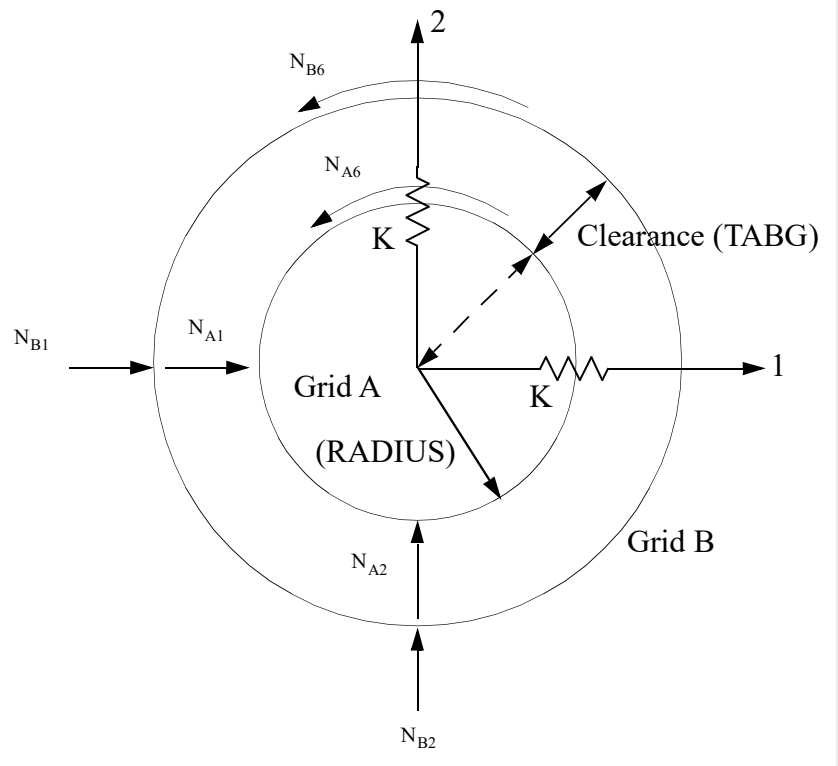


Figure 5-2 Radial Gap Orientations and Nonlinear Load Sign Conventions

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

NLRGAP Underlying Equations

Definitions

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

Equations

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

The element relative displacement in directions 1 and 2 are:

$$\Delta U_1 = U_{A1} - U_{B1}$$

$$\Delta U_2 = U_{A2} - U_{B2}$$

The relative radial displacement is:

$$\Delta r = \sqrt{\Delta U_1^2 + \Delta U_2^2}$$

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

$$P = \Delta r - G(t)$$

The nonlinear gap forces are:

$$S = K(t,u)\left(1 - \frac{G(t)}{\Delta r}\right) \tag{5-1}$$

$$N_{A1} = -\Delta U_1 S + \Delta U_2 S\mu(t) \tag{5-2}$$

$$N_{B1} = \Delta U_1 S - \Delta U_2 S\mu(t) \tag{5-3}$$

$$N_{A2} = -\Delta U_2 S - \Delta U_1 S\mu(t) \tag{5-4}$$

$$N_{B2} = \Delta U_2 S + \Delta U_1 S\mu(t) \tag{5-5}$$

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

$$N_{A6} = -R\Delta r S\mu(t) \tag{5-6}$$

$$N_{B6} = R\Delta r S\mu(t) \tag{5-7}$$

Benefit

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

Input

The Bulk Data entry NLRGAP is used to model the radial gap.

The contact stiffness can be input either as a function of time or as a function of penetration. In both cases, the stiffness curves are input on TABLEDi Bulk Data entries which are referenced by the NLRGAP entry. Inputting stiffness as a function of time allows the user to model special situations where structural changes over time cause the contact stiffness

to vary over time. Inputting stiffness as a function of penetration allows the user to input a nonlinear stiffness curve which might be required, for example, if the shaft or housing were covered by some nonlinear-elastic material.

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

The NLRGAP entry can also approximate the torque on the shaft and housing due to friction. If this effect is desired, the RADIUS field of the NLRGAP must be specified. If not specified, then friction effects will still be included in the lateral loads as shown in equations (5-2) through (5-5). The torque loads, equations (5-6) and (5-7), will not be computed nor applied if RADIUS is input as 0.0 or left blank.

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

Output

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

Guidelines

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

Limitations

- The NLRGAP can only be positioned in the XY, YZ or ZX plane of the displacement coordinates of the referenced grids.

- The DOF in the plane of the NLRGAP must be in the d-set. This means that these DOF cannot be constrained by SPCs or be dependent DOF on MPCs and RBE type elements.
- The NLRGAP can only be used in SOLution sequences 109 and 129.

A diagram of input data linkages is shown in Figure 5-3. The general form of the transient load, P_{kj}, for each degree of freedom, u_j, is:

$$P_{kj} = C_{ki} \cdot A_j \cdot F_i(t)$$

where the factors are defined by various Bulk Data entries as shown in Figure 5-4.

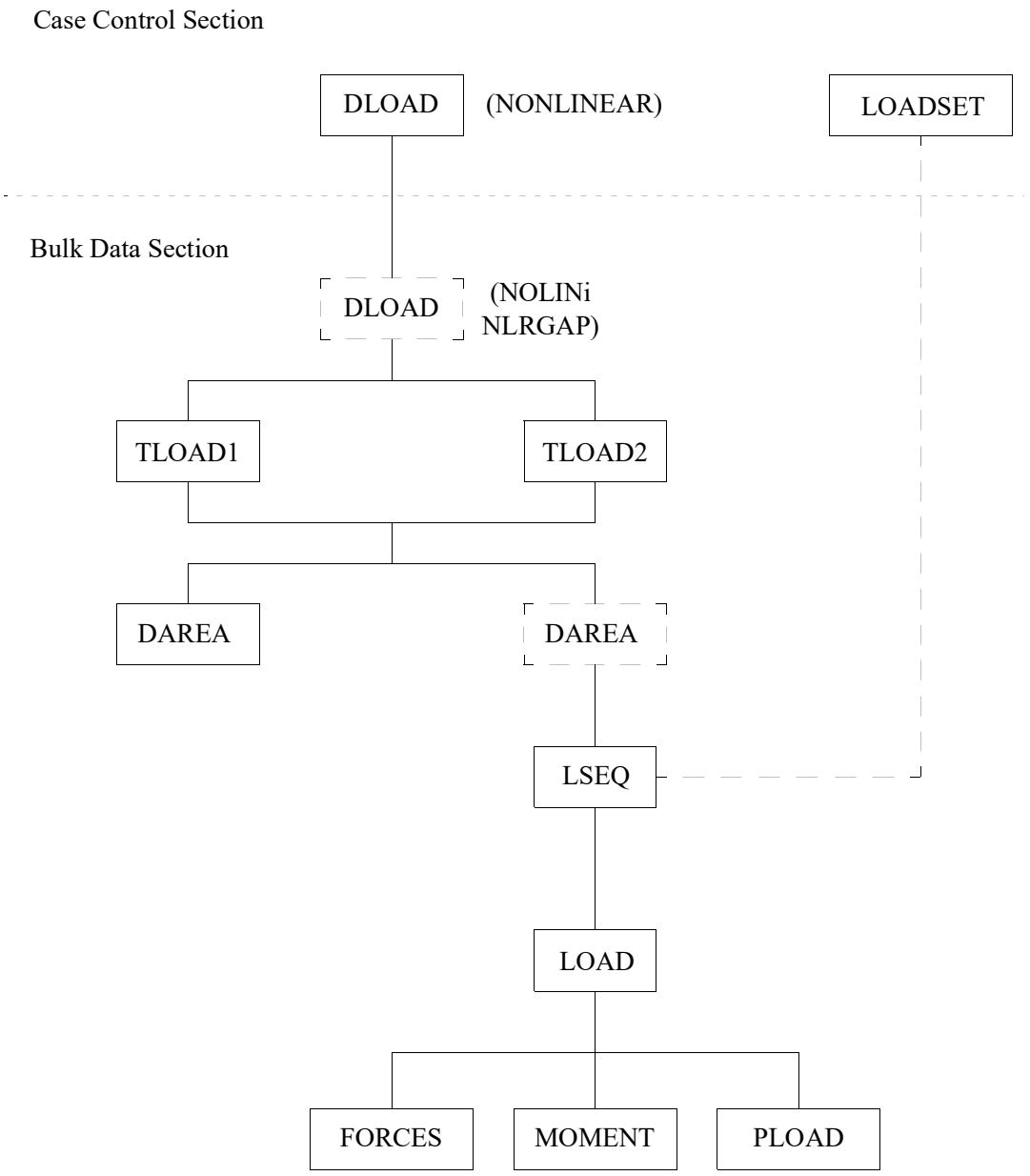
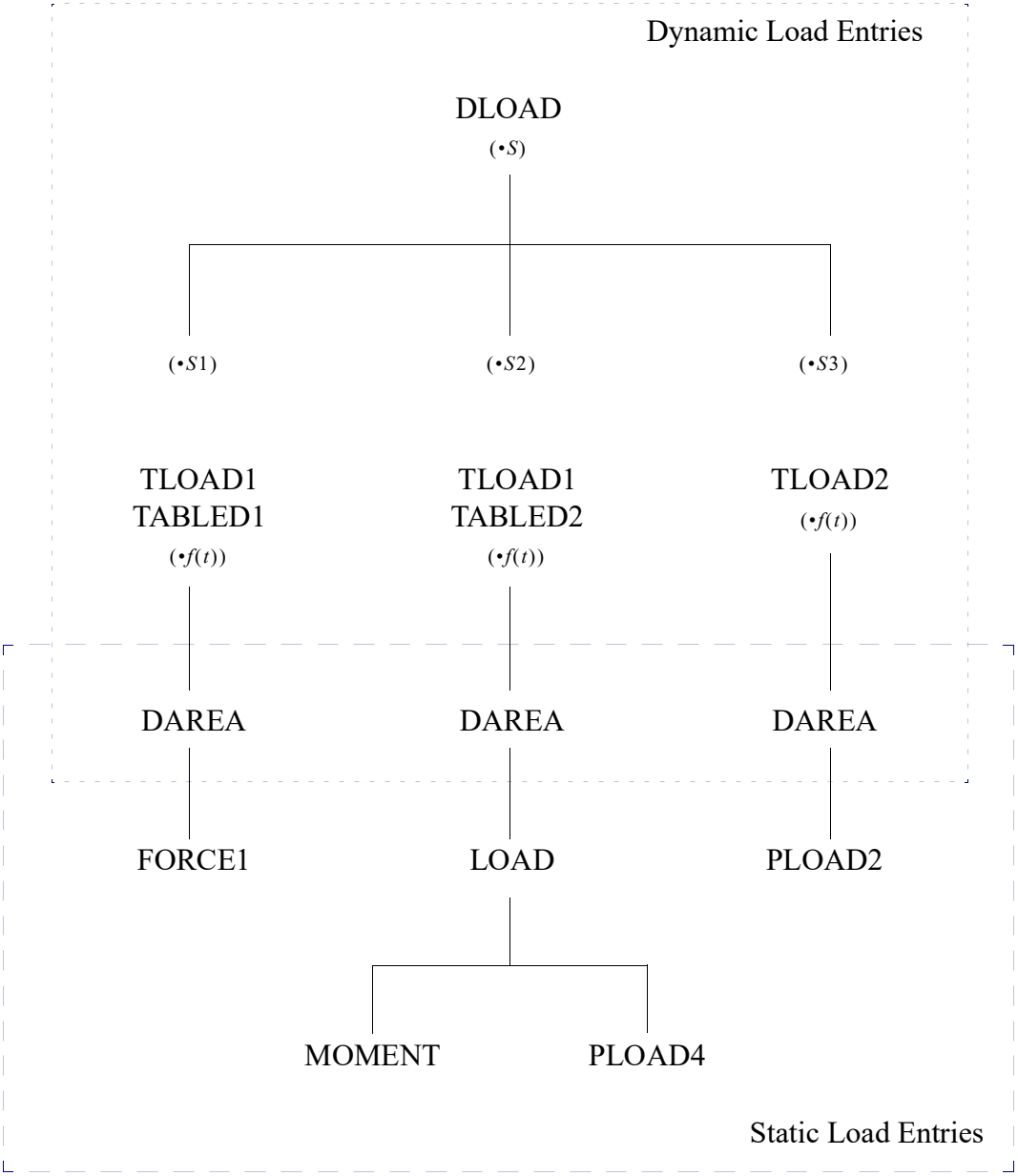


Figure 5-3 Dynamic Loads for Transient Analysis



Note that each load is scaled by $S \cdot Si \cdot f(t)$ at each time step in dynamic loads.

Figure 5-4 Example of Combining Loads

6 Constraint and Set Notation

- Degree of Freedom Sets 326
- Multipoint Constraints 329
- Single Point Constraints 331
- Rigid Body Supports 332
- Sets for Dynamic Reduction 333
- Sets for Aerodynamics 333
- Rules of Sets for Undefined Degrees of Freedom 334
- Output Selection Via Set Specification 335

Degree of Freedom Sets

Structural matrices are initially assembled in terms of all structural points, which excludes only the extra points introduced for dynamic analysis. These matrices are generated with six degrees of freedom for each geometric grid point and a single degree of freedom for each scalar point. These degrees of freedom are partitioned into sets, based on user inputs. The sets are used to successively eliminate variables during the solution process, as described in General Operations, 362.

Degree-of-Freedom Set Definitions

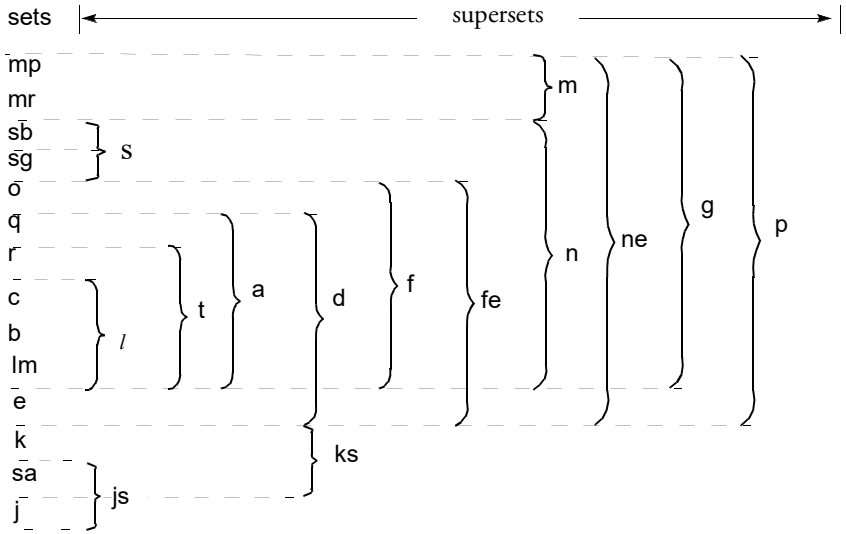
Each degree of freedom is a member of one mutually exclusive set. Set names have the following definitions:

Set Name	Definition
mp	degrees of freedom eliminated by multipoint constraints.
mr	degrees of freedom eliminated by multipoint constraints created by the rigid elements using the LGELIM method on the Case Control command RIGID.
sb*	degrees of freedom eliminated by single-point constraints that are included in boundary condition changes and by the AUTOSPC feature.
sg*	degrees of freedom eliminated by single-point constraints that are specified on the PS field on GRID Bulk Data entries.
o	degrees of freedom omitted by structural matrix partitioning.
q	Generalized degrees of freedom assigned to component modes and residual vectors.
r	reference degrees of freedom used to determine free body motion.
c	degrees of freedom that are free during component mode synthesis or dynamic reduction.
b	degrees of freedom fixed during component mode analysis or dynamic reduction.
lm	Lagrange multiplier degrees of freedom created by the rigid elements using the LAGR method on the Case Control command, RIGID.
e	extra degrees of freedom introduced in dynamic analysis.
sa	Permanently constrained aerodynamic degrees of freedom.
k	Aerodynamic mesh point set for forces and displacements on the aero mesh.
j	Aerodynamic mesh collocation point set (exact physical interpretation is dependent on the aerodynamic theory).

*Strictly speaking, sb and sg are not exclusive with respect to one another. Degrees of freedom may exist in both sets simultaneously. Since these sets are not used explicitly in the solution sequences, this need not concern the user. However, those who use these sets in their own DMAPs should avoid redundant specifications when using these sets for partitioning or merging operations. That is, a degree of freedom should not be specified on both a PS field of a GRID

entry (sg set) and on a selected SPC entry (sb set). Redundant specifications will cause UFM 2120 in the VEC module and behavior listed in *MSC Nastran DMAP Programmer's Guide* *i* for the UPARTN module. These sets are exclusive, however, from the other mutually exclusive sets.

Each degree of freedom is also a member of one or more combined sets called "supersets." Supersets have the following definitions:



Set Name	Meaning (+ indicates union of two sets)
s = sb + sg	all degrees of freedom eliminated by single point constraints
l = b + c + *l*m	the structural degrees of freedom remaining after the reference degrees of freedom are removed (degrees of freedom left over)
t = *l* + r	the total set of physical boundary degrees of freedom for superelements
a = t + q	the set assembled in superelement analysis
d = a + e	the set used in dynamic analysis by the direct method
f = a + o	unconstrained (free) structural degrees of freedom
fe = f + e	free structural degrees of freedom plus extra degrees of freedom
n = f + s	all structural degrees of freedom not constrained by multipoint constraints
ne = n + e	all structural degrees of freedom not constrained by multipoint constraints plus extra degrees of freedom
m = mp + mr	all degrees of freedom eliminated by multiple constraints
g = n + m	all structural (grid) degrees of freedom including scalar degrees of freedom

Set Name	Meaning (+ indicates union of two sets)
p = g + e	all physical degrees of freedom
ks = k + sa	the union of k and the re-used s-set (6 degrees of freedom per grid)
js = j + sa	the union of j and the re-used s-set (6 degrees of freedom per grid)
fr = o + *l*	statically independent set minus the statically determinate supports (fr = f – q – r)
v = o + c + r	the set free to vibrate in dynamic reduction and component mode synthesis

The a-set and o-set are created in the following ways:

1. If only OMITi entries are present, then the o-set consists of degrees of freedom listed explicitly on OMITi entries. The remaining f-set degrees of freedom are placed in the b-set, which is a subset of the a-set.
2. If ASETi or QSETi entries are present, then the a-set consists of all degrees of freedom listed on ASETi entries and any entries listing its subsets, such as QSETi, SUPORTi, CSETi, and BSETi entries. Any OMITi entries are redundant. The remaining f-set degrees of freedom are placed in the o-set.
3. If there are no ASETi, QSETi, or OMITi entries present but there are SUPORTi, BSETi, or CSETi entries present, then the entire f-set is placed in the a-set and the o-set is not created.
4. There must be at least one explicit ASETi, QSETi, or OMITi entry for the o-set to exist, even if the ASETi, QSETi, or OMITi entry is redundant.

In dynamic analysis, additional vector sets are obtained by a modal transformation derived from real eigenvalue analysis of the a-set. These sets are as follows:

ξ_o = rigid body (zero frequency) modal degrees of freedom

ξ_f = finite frequency modal degrees of freedom

ξ_i = $\xi_o + \xi_f$, the set of all modal degrees of freedom

One vector set is defined that combines physical and modal degrees of freedom:

u_h = $\xi_i + u_e$, the set of all modal degrees of freedom

The membership of each degree of freedom can be printed by use of the Bulk Data entries PARAM,USETPRT and PARAM,USETSEL.

Degree-of-Freedom Set Bulk Data Entries

Degrees of freedom are placed in sets as specified by the user on the following Bulk Data entries:

Name	Bulk Data Entry Name
m	MPC, MPCADD, MPCAX, POINTAX, RBAR, RBE1, RBE2, RBE3, RROD, RSPLINE, RTRPLT, GMBC, GMSPC*
sb	SPC, SPC1, SPCADD, SPCAX, FLSYM, GMSPC*, BNDGRID, (PARAM,AUTOSPC,YES)
sg	GRID, GRIDB, GRDSET (PS field)
o	OMIT, OMIT1, OMITAX, GRID (SEID field), SESET
q	QSET, QSET1
r	SUPORT, SUPORT1, SUPAX
c	CSET, CSET1
b	BSET, BSET1
e	EPOINT
sa	CAEROi
k	CAEROi
a	ASET, ASET1, Superelement exterior degrees of freedom, CSUPEXT
*Placed in set only if constraints are not specified in the basic coordinate system.	

In superelement analysis, the appropriate entry names are preceded by the letters SE, and have a field reserved for the superelement identification number. This identification is used because a boundary (exterior) grid point may be in one mutually exclusive set in one superelement and in a different set in the adjoining superelement. The SE-type entries are internally translated to the following types of entry for the referenced superelement:

Entry Type	Equivalent Type
SEQSETi	QSETi
SESUP	SUPORT
SECSETi	CSETi
SEBSETi	BSETi

Multipoint Constraints

Each multipoint constraint is described by a single equation that specifies a linear relationship for two or more degrees of freedom. In static analysis, multiple sets of multipoint constraints can be provided in the Bulk Data Section, with selections made at execution time by using the subcase structure in Case Control as explained in Case Control Section, 15. Multipoint

constraints are also discussed in Executive Control Section, 15 of this manual and Section 5.4 of *The NASTRAN Theoretical Manual*.

Multipoint constraints are defined on MPC, MPCADD, and MPCAX entries. MPC is the basic entry for defining multipoint constraints. The first component specified on the entry is the dependent degree of freedom, i.e., that degree of freedom that is removed from the equations of motion. Dependent degrees of freedom may appear as independent terms in other equations of the set; however, they may appear as dependent terms in only a single equation. The MPCADD entry defines a union of multipoint constraints. The MPCAX defines multipoint constraints in conical shell problems. Some uses of multipoint constraints are:

- To describe rigid elements and mechanisms such as levers, pulleys, and gear trains. In this application, the degrees of freedom associated with the rigid element that are in excess of those needed to describe rigid body motion are eliminated with multipoint constraint equations. Treating very stiff members as rigid elements eliminates the ill-conditioning associated with their treatment as ordinary elastic elements.
- To enforce zero motion in directions other than those corresponding with components of the global coordinate system. In this case, the multipoint constraint will involve only the degrees of freedom at a single grid point. The constraint equation relates the displacement in the direction of zero motion to the displacement components in the global system at the grid point.
- To be used with scalar elements to generate nonstandard structural elements and other special effects.
- To describe parts of a structure by local vibration modes. This application is treated in Section 14.1 of *The NASTRAN Theoretical Manual.* The general idea is that the matrix of local eigenvectors represents a set of constraints relating physical coordinates to modal coordinates.

In general, the user must provide the coefficients in the multipoint constraint equations. However, several rigid elements have been introduced which will generate the MPC equations for some applications:

RROD	A pin-ended rod which is rigid in extension.
RBAR	A rigid bending element with six degrees of freedom at each end.
RTRPLT	A rigid triangular plate with six degrees of freedom at each vertex.
RBE1	A rigid body connected to an arbitrary number of grid points. The independent and dependent degrees of freedom can be arbitrarily selected by the user.
RBE2	A rigid body connected to an arbitrary number of grid points. The independent degrees of freedom are the six components of motion at a single grid point. The dependent degrees of freedom are specified at an arbitrary number of grid points.
RBE3	Defines the motion at a reference grid point as the weighted average of the motions at a set of other grid points.
RSPLINE	Defines multipoint constraints for the interpolation of displacements at grid points.
RSSCON	Defines multipoint constraints to model clamped connections of shell-to-solid elements.

The rigid elements will always meet equilibrium and continuity requirements, whereas this is a user responsibility for MPC equations. (See Rigid Elements and Multipoint Constraints (R-type, MPC), 155 for further information on rigid elements.)

Multipoint forces of constraint may be output with MPCFORCE Case Control command in SOLution Sequences 101 through 200.

Single Point Constraints

A single point constraint (SPC) applies a fixed value to a translational or rotational component at a geometric grid point or to a scalar point. Common uses of single point constraints are to specify the boundary conditions of a structural model by fixing the appropriate degrees of freedom and to eliminate unwanted degrees of freedom with zero stiffness. Multiple sets of single point constraints can be provided in the Bulk Data Section, with selections made at execution time by using the subcase structure in the Case Control Section as explained in Case Control Section, 15. This procedure is particularly useful in the solution of problems having one or more planes of symmetry.

The elements connected to a grid point may not provide resistance to motion in certain directions, causing the stiffness matrix to be singular. Single point constraints are used to remove these degrees of freedom from the stiffness matrix. A typical example is a planar structure composed of membrane and extensional elements. The translations normal to the plane and all three rotational degrees of freedom must be constrained since the corresponding stiffness matrix terms are all zero. If a grid point has a direction of zero stiffness, the single point constraint needs not be exactly in that direction, but only needs to have a component in that direction. This allows the use of single point constraints for the removal of such singularities regardless of the orientation of the global coordinate system. Although the displacements will depend on the direction of the constraint, the internal forces will be unaffected.

One of the tasks performed by the Grid Point Singularity Processor (GPSP) (see Constraint and Mechanism Problem Identification in SubDMAP SEKR, 370) is to examine the stiffness matrix for singularities at the grid point level. Singularities remaining at this level, following the application of the multipoint and single point constraints, are listed in the Grid Point Singularity Table (GPST), which is automatically printed. The GPST lists all singular degrees of freedom, in the global coordinate system, and the ratio of stiffness between the softest and stiffest degree of freedom for the grid point. The user may request that single point constraints be generated for all identified singularities by use of the Bulk Data entry PARAM,AUTOSPC,YES.

Single point constraints are defined on SPC, SPC1, SPCADD and SPCAX entries. The SPC entry is the most general way of specifying single point constraints. The SPC1 entry is a less general entry that is more convenient when several grid points have the same components constrained to a zero displacement. The SPCADD entry defines a union of single point constraint sets specified with SPC or SPC1 entries. The SPCAX entry is used only for specifying single point constraints in problems using conical shell elements.

Single point constraints can also be defined on the GRID entry. In this case, however, the points are constrained for all subcases. The default value for enforced displacement on points constrained on GRID entries is zero. The default value can be overridden at the subcase level with SPC entries.

The printed output for single point forces of constraints can be requested in Case Control. All nonzero forces are printed, whether they originate from SPCi entries, the PS field on GRID entries, or by PARAM,AUTOSPC,YES.

Rigid Body Supports

In the following discussion, a free body is defined as a structure that is capable of motion without internal stress; i.e., it has one or more rigid body degrees of freedom. The stiffness matrix for a free body is singular with the defect equal to the number of stress-free, or rigid body modes. A solid three-dimensional body has up to six rigid body modes. Linkages and mechanisms can have a greater number. In order to permit the analysis of mechanisms, no restriction is placed in the program on the number of stress-free modes.

Free-body supports are defined with a SUPORT or SUPORT1 entry. In the case of problems using conical shell elements, the SUPAX entry is used. Free-body supports must be defined in the global coordinate system. The SUPORT1 entry must be selected by the SUPORT1 Case Control command.

In static analysis by the displacement method, the rigid body modes must be restrained in order to remove the singularity of the stiffness matrix. The required constraints may be supplied with single point constraints, multipoint constraints, or free body supports. If free body supports are used, the rigid body characteristics will be calculated and a check will be made on the sufficiency of the supports. Such a check is obtained by calculating the rigid body error ratio and the strain energy as defined in the Rigid Body Matrix Generator operation in Static Solutions in SubDMAP SEKRRS, 376. This error ratio and the strain energy are automatically printed following the execution of the Rigid Body Matrix Generator. The error ratio and the strain energy should be zero, but may be nonzero for any of the following reasons:

- Round-off error accumulation.
- Insufficient free body supports have been provided.
- Redundant free body supports have been provided.

The redundancy of the supports may be caused by improper use of the free body supports themselves or by the presence of single point or multipoint constraints that constrain the rigid body motions.

Static analysis with inertia relief is necessarily made on a model having at least one rigid body motion. Such rigid body motion must be constrained by the use of free body supports. These supported degrees of freedom define a reference system, and the elastic displacements are calculated relative to the motion of the support points. The element stresses and forces will be independent of any valid set of supports.

Rigid body vibration modes are calculated by a separate procedure provided that a set of free body supports is supplied by the user. This is done to improve efficiency and, in some cases, reliability. The determinant method, for example, has difficulty extracting zero frequency roots of high multiplicity, whereas the alternate procedure of extracting rigid body modes is both efficient and reliable. If the user does not specify free body supports (or he specifies an insufficient number of them), the (remaining) rigid body modes will be calculated by the method selected for the finite frequency modes, provided zero frequency is included in the range of interest. If the user does not provide free body supports, and if zero frequency is not included in the range of interest, the rigid body modes will not be calculated.

Free body supports must be specified if the mode acceleration method of solution improvement is used for dynamic problems having rigid body degrees of freedom (see Rigid-body Modes (p. 155) in the). This solution improvement technique involves a static solution, and although the dynamic solution can be made on a free body, the static solution cannot be performed without removing the singularities in the stiffness matrix associated with the rigid body motions.

Sets for Dynamic Reduction

There are several methods for reducing the size of models in dynamic analysis. The method described here uses shapes derived as generalized functions by static analysis. The same entries may be used in static analysis with the same nomenclature being used internally as in superelement analysis.

The statically independent degrees of freedom remaining after constraint elimination are partitioned into two sets. The partitions are defined by listing the degrees of freedom for one of the partitions on the ASETi entries. These degrees of freedom are referred to as the "analysis set." The remaining degrees of freedom are referred to as the "omitted set." The OMITi entry can be used to place degrees of freedom in the omitted set with the remaining degrees of freedom being placed in the analysis set. This is easier if the analysis set is large. In the case of problems using conical shell elements, the OMITAX entry is used.

One of the more important applications of partitioning is Guyan reduction, described in Dynamic Reduction (p. 667) in the . This technique is a means for reducing the number of degrees of freedom used in dynamic analysis with minimum loss of accuracy. Its basis is that many fewer grid points are needed to describe the inertia of a structure than are needed to describe its elasticity with comparable accuracy. The error in the approximation is small provided that the set of displacements used for dynamic analysis is judiciously chosen. Its members should be uniformly dispersed throughout the structure, and all large mass items should be connected to grid points that are members of the analysis set.

The user is cautioned to consider the fact that the matrix operations associated with this partitioning procedure tend to create nonzero terms and to fill what were previously very sparse matrices. The partitioning option is most effectively used if the members of the omitted set are either a very large fraction or a very small fraction of the total set. In most of the applications, the omitted set is a large fraction of the total and the matrices used for analysis, while small, are usually full. If the analysis set is not a small fraction of the total, a solution using the large, but sparser, matrices may be more efficient.

The a-set is further partitioned into the q-, r-, c-, and b-sets. The q-set is used to store the coefficients for the generalized coordinates determined by dynamic reduction and/or component mode calculations. The r-set is again used to determine rigid body modes. The c-set contains coordinates considered free to move while computing the generalized coordinate functions. The b-set contains coordinates fixed during this process.

The set used to describe the generalized functions for dynamic reduction and component mode synthesis is the v-set, the union of the o-, r-, and c-sets.

The d-set ("dynamic") for direct formulations is formed from the union of the structural degrees of freedom in the a-set and the extra points of the e-set. The d-set can be used to input unsymmetric terms into the equations of motion using DMIG and TF Bulk Data entries. In the modal formulations, the h-set is composed of the union of the c_i (modal) set and the e-set. Again, the DMIG and TF Bulk Data entries are a means to introduce unsymmetric matrix coefficients.

Sets for Aerodynamics

Aerodynamic calculations are made in a Cartesian aerodynamic coordinate system. By convention, the flow is in the positive x direction. The basic coordinate system was not chosen, since it would place an undesirable restriction upon the description of the structural model. Any MSC Nastran Cartesian system may be specified, and flow will be defined in the direction of its positive x-axis. All element and aerodynamic grid point data, computed initially in the basic coordinate system, will be converted to the aerodynamic coordinate system. The global (displacement) coordinate system of the aerodynamic grid

points will have its T1 direction in the flow direction. T3 is normal to the element for boxes, and parallel to the aerodynamic T3 in the case of bodies. Coordinate system data are generated for the aerodynamic grid points.

The grid points are physically located at the centers of the boxes and body elements. Permanent constraints are generated for the unused degrees of freedom. A second set of grid points, used only for undeformed plotting, is located at the element corners. All six degrees of freedom associated with each grid point in this second set are permanently constrained. Grid point numbers are generated based upon the element identification number. For any panel, the external grid point numbers for the boxes start with the panel identification number and increase consecutively.

Aerodynamic degrees of freedom, along with the extra points, are added after the structural matrices and modes have been determined. This introduces the following displacement sets:

k	Aerodynamic box and body degrees of freedom
sa	Permanently constrained degrees of freedom associated with aerodynamic grid points
ks = k + sa	the union of k and the re-used s-set (6 degrees of freedom per grid)
js = j + sa	the union of j and the re-used s-set (6 degrees of freedom per grid)

Rules of Sets for Undefined Degrees of Freedom

Most of the Bulk Data entries used to list set membership specify mutually exclusive sets. This means that degrees of freedom listed on more than one such entry will result in a fatal error. For example, a degree of freedom may not be in the m-set on two RBAR entries, or on a selected SPC entry and a SUPORT entry.

The a-set, by contrast, defines a combined set. Degrees-of-freedom listed on Bulk Data entries that define its subsets, the q-, r-, c-, and b-set, may also be listed on ASETi Bulk Data entries, although this is a redundant specification and is not necessary. However, automatic restarts are more reliable if ASETi entries are present for all degrees of freedom listed on SUPORT entries.

Bulk Data entries used to define mutually exclusive sets automatically place them in their specified set and any combined set to which they belong. Since the ASETi entries define a combined set, the subsets of the a-set, if not defined, must be placed in a mutually exclusive set by an arbitrary convention. The conventions described below are used because they allow the user to specify these sets with the minimum input for most cases. In superelement analysis, exterior (“boundary”) degrees of freedom are automatically determined and placed in the a-set of the superelement. Exterior degrees of freedom for superelements are therefore equivalent to degrees of freedom listed on ASETi entries in the discussion below.

The f-set is the union of the mutually exclusive o-, q-, r-, c-, and b-sets and includes the a-set. The o-set exists under two conditions only:

1. If OMITi entries are present, the o-set consists of degrees of freedom listed explicitly on OMITi entries. The remaining f-set degrees of freedom are placed in the b-set. They are then also members of the a-set.
2. If ASETi or QSETi entries are present, the a-set consists of all degrees of freedom listed on ASETi entries and any entries listing its subsets, such as QSETi, SUPORTi, CSETi, and BSETi entries. The remaining f-set degrees of freedom are placed in the o-set. Note, however, there must be at least one explicit ASETi or QSETi entry type present for the o-set to exist, even if it is a redundant specification. If there are no ASETi or QSETi entries present, and SUPORT, BSETi, or CSETi are present, the entire f-set is placed in the a-set; that is, the o-set is not formed.

If both OMITi and ASETi entries are present, the OMITi entries are ignored, so that condition 2, above, applies. Similarly, if ASETi entries are used in conjunction with entries defining its subsets, undefined degrees of freedom in the a-set are placed in mutually exclusive sets as shown in the following table (0 = no entry types present; 1 = entry type present):

Set Defined by Entry Type	Case			
	1	2	3	4
c	0	0	1	1
b	0	1	0	1
Undefined a-set placed in	b	c	b	b

Output Selection Via Set Specification

The membership of grid points and their components and scalar points in the various sets is stored in the USET table as digits in a binary number. Some modules use input parameters that are translations of these binary numbers as a shorthand notation. Three formats are used:

Set name

(DEFUSET, USET and USET1 Bulk Data entries, MATGPR, UPARTN, VEC, UMERGE, UMERGE1, and PARAML (OP = USET) Modules)

The data block naming convention in the solution sequences for matrices related to the sets is based on the set names. For example, the system stiffness matrix $[K_{gg}]$ has the name KGG. Some sets are used that do not appear in the USET table. The h-set, for example, is the set used for modal formulations. Its size is not known until after the USET table is formed, so it does not appear in the table. The j-set, a set synonymous with the g-set, is used in superelement analysis. There are several other sets used that also do not appear in the USET table.

Some names may be used for other purposes. The name H, for example, may be input to the MATGPR module when one wants to label rows or columns of matrices sequentially, rather than by their external sequence numbers. This feature may be used to number vectors in load matrices or in other instances not related to modal analysis, even though “H” is defined for modal analysis.

7 Solution Sequences

- Introduction 337
- Output Description 340
- Restart Procedures 361
- General Operations 362
- Summary of Solution Sequence Operations 394

Introduction

MSC Nastran is composed of a large number of building blocks called modules. A module is a collection of FORTRAN subroutines designed to perform a specific task—processing model geometry, assembling matrices, applying constraints, solving matrices, calculating output quantities, conversing with the database, printing the solution, and so on. The modules are controlled by an internal language called the Direct Matrix Abstraction Program (DMAP). Each type of analysis available in MSC Nastran is called a solution sequence, and each solution sequence is a prepackaged collection of hundreds or thousands of DMAP commands. Once a solution sequence is chosen, its particular set of DMAP commands sends instructions to the modules that are needed to perform the requested solution. All of this happens automatically with no effort on your part beyond choosing a solution sequence.

The solution sequences contain complete DMAP sequences required to perform certain types of analysis as well as superelement generation (phase I) and data recovery (phase III). The solution sequences are listed in Table 7-1. An automatic restart capability is employed in these sequences. The automatic restart is not only driven by modifications to the Case Control and Bulk Data Sections but also by modifications in upstream superelements. A full description of automatic restarts is given in Restart Procedures, 361.

A solution is a feature selected by a SOL Executive Control command. It is a collection of DMAP statements that performs a specific type of analysis. For example, static analysis is requested by the command SOL 101. There are two types of solutions at present.

The major type is called a Solution Sequence. They are called with numbers ranging from 100 to 200, 400, 600 and 700. There were formerly Structured Solution Sequences and Unstructured Solution Sequences, but the latter class has been removed, so the name Solution Sequence is now used, although you may find references to Structured Solutions in older documentation. All new technology is introduced through Solution Sequences. They also have automatic restart capability. This is accomplished with the NASTRAN Data Definition Language feature, a file that lists the names and attributes of data types saved for restart. The evidence that a Solution Sequence has restart capability can be found by the existence of TYPE statements in its DMAP compile listings. It uses the SubDMAP feature, which allow breaking a large collection of DMAP statements into more manageable data structures similar to subroutines in other programming languages.

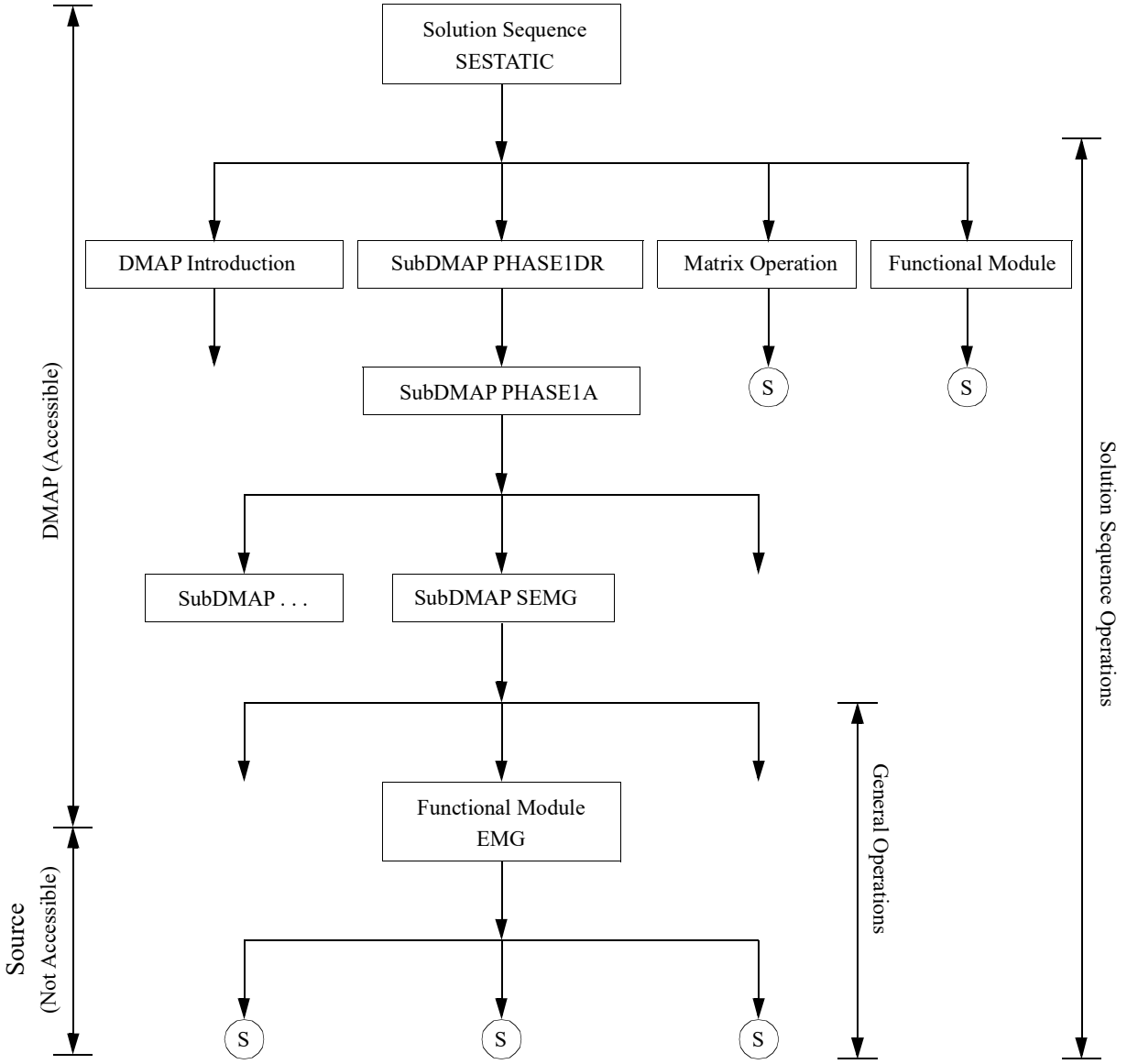


Figure 7-1 Hierarchy in the Structural Solution Sequences

Table 7-1 Structured Solution Sequences

SOL Number	SOL Name	Description	References
101	SESTATIC	Statics with options: Linear Heat Transfer Alternate Reduction Inertia Relief Design Sensitivity -- Statics	page 376 page 418
103	SEMODES	Normal Modes with option: Design Sensitivity -- Modes	page 403
105	SEBUCKL	Buckling with options: Static Analysis Design Sensitivity -- Buckling	page 418
106	NLSTATIC	Nonlinear Statics	page 673
107	SEDCEIG	Direct Complex Eigenvalues	page 416
108	SEDFREQ	Direct Frequency Response	page 417
109	SEDTRAN	Direct Transient Response	page 417
110	SEMCEIG	Modal Complex Eigenvalues	page 416
111	SEMFREQ	Modal Frequency Response	page 417
112	SEMTRAN	Modal Transient Response	page 417
114	CYCSTATX	Cyclic Statics with option: Alternate Reduction	page 703
115	CYCMODE	Cyclic Normal Modes	page 703
118	CYCFREQ	Cyclic Direct Frequency Response	page 703
129	NLTRAN	Nonlinear Transient Response	page 568
144	AESTAT	Static Aeroelastic Response	See the MSC Nastran Aeroelastic Analysis User's Guide
145	SEFLUTTR	Aerodynamic Flutter	See the MSC Nastran Aeroelastic Analysis User's Guide
146	SEAERO	Aeroelastic Response	See the MSC Nastran Aeroelastic Analysis User's Guide
153	NLSCSH	Static Structural and/or Steady State Heat Transfer Analysis	page 432
159	NLTCSH	Transient Structural and/or Transient Heat Transfer Analysis	page 432

Table 7-1 Structured Solution Sequences (continued)

SOL Number	SOL Name	Description	References
200	DESOPT	Design Optimization	See the MSC.Nastran Design Sensitivity and Optimization User's Guide
400	NONLIN	Nonlinear Static and Transient Analysis	See the MSC Nastran Nonlinear User's Guide
600,ID	SESTATIC	MSC Nastran, Marc in MSC Nastran	See the MSC Nastran Implicit Nonlinear User's Guide
700	NLTRAN	Explicit nonlinear analysis	See the MSC Nastran Explicit Nonlinear (SOL 700) User's Guide

The DMAP listings for any of the solutions sequences can be obtained by specifying a COMPILER LIST statement in the Executive Control Section. A description of the data processing and matrix operations for the solution sequences is given in Summary of Solution Sequence Operations, 394. The complete details for each of the functional modules are given in Chapter 4 of the *MSC Nastran Programmer's Manual*. Those DMAP modules which are of general use (matrix operations and utility modules) are described along with examples of their use in General Operations, 362.

If the user wishes to modify a DMAP sequence in some manner, he can use the alter feature described in *MSC Nastran DMAP Programmer's Guide*. Typical uses are to schedule an EXIT prior to completion in order to check intermediate output, schedule the printing of a table or matrix for diagnostic purposes, and to delete or add a functional module to the DMAP sequence. The user should be familiar with the rules for DMAP programming, as described in the *Msc Nastran DMAP Programmer's Guide*, to make alterations to a solution sequence. A library of DMAP alters is delivered with MSC Nastran. These alters are used to request options not available in the solution sequences and for exceptional operations that do not fit in with the normal flow of the solution sequences.

Output Description

Although most of the solution sequence output is optional and requested in the Case Control Section, some of the printer output is automatic or under control of DIAG statements in the Executive Control Section. The printer output is designed for 132 characters per line, with the lines per page controlled by the LINE command in the Case Control Section. The default is LINE = 50 for 11-inch paper. Optional titles are printed at the top of each page from information in the Case Control Section. These titles may be defined at the subcase level. The pages are automatically dated and numbered.

The output from data recovery and plot modules is all optional and its selection is controlled by commands in the Case Control Section. The details of making selections in the Case Control Section are described in Case Control Section, 15 for printer and punch output and in Plotting, 460 for plotter output. The available output for each solution sequence is described in Summary of Solution Sequence Operations, 394. Detailed information on the force and stress output available for each element type is given in Structural Elements, 41.

A few printer output items are under the control of PARAM Bulk Data entries. The use of PARAM entries is described in Use of Parameters, 32. The DIAG statement is used to control the printing of diagnostic output. The available output under the DIAG statement is given in the description of the Executive Control statements in the Executive Control Statements (p. 105) in the *MSC Nastran Quick Reference Guide*.

The first part of the output for an MSC Nastran run is prepared during the execution of the preface, prior to the beginning of the solution sequence. The following output is either automatically or optionally provided during execution:

- MSC Nastran title page — automatic
- File Management Section echo — automatic if statements are present
- Executive Control Section echo — automatic
- Case Control Section echo — automatic
- Unsorted Bulk Data Section echo — optional, selected in the Case Control Section
- Sorted Bulk Data Section echo — automatic, unless suppressed in the Case Control Section

MSC Nastran Output

MSC Nastran generates the following output files.

File	Contents
Execution Summary Table in *.f04	Computer model, operating system, MSC Nastran version, system creation date, problem run date, MSC Nastran file assignments, and a summary of the MSC Nastran execution: time of day, elapsed time, I/O seconds, CPU seconds, module execution messages, matrix trailers (DIAG 8), value of REAL (DIAG 13), MPYAD, SDCOMP and FBS estimates (DIAG 19), and statistical information on module expected/actual CPU item (DIAG 49).
Standard Output in *.f06	MSC Nastran results (i.e., displacements, stresses, etc.). See the *MSC Nastran Linear Static Analysis User's Guide* and the *MSC Nastran Numerical Methods User's Guide* for more information.

An example follows which shows the execution summary (.f04) and the standard output (.f06) for SOL 101.

Execution Summary Contents (see Listing 7-1)

The information in the Execution Summary file can provide the user with CPU time, I/O time, elapsed time and disk space resource usage information on the execution of the problem. To aid in the interpretation of these data, an example is presented on the following pages. Circled integers have been added for ease of reference.

1	The column headed DAY TIME indicates the time of day as the run progresses.
2	The column headed ELAPSED indicates the number of elapsed "wall clock" minutes:seconds at the beginning on the job. This number may vary greatly depending on the system load in a time-sharing environment.

3	The column headed I/O SEC is an internally generated measure of I/O. This measure is a count of the number of BUFFSIZE blocks transferred, divided by the value of SYSTEM(84) (the estimated rate of number of blocks moved per second). This number does not include I/O for loading the executable file or I/O for FORTRAN files. The column headed DEL I/O is the I/O seconds spent in the previous DMAP statement.
4	The column headed CPU SEC shows the amount of CPU seconds that elapsed when this line was printed. It is recommended that the user examine these entries and look for large jumps in this number and try to understand why each one occurred. For example, the cost for "formatting" the output (modules SDR2 and OFP) may be greater than the cost of generating the numbers. In such situations, simple techniques such as reducing output requests may save more money than other more well-known techniques such as sequencing, OMITs, better matrix methods, etc. In short, the user should be aware of the expensive operations on each large run. The column headed DEL CPU is the CPU time spent in the previous DMAP statement.
5	Primary database file assignments.
6	The column headed BUFFSIZE is the buffer size of each DBset. This buffer size may be defined by the INIT command for each DBset (see the File Management Statements (p. 41) in the *MSC Nastran Quick Reference Guide*).
7	The column headed CLUSTER SIZE is the unit of allocation for GINO and executive blocks. This unit may be defined on the INIT command (see the File Management Statements (p. 41) in the *MSC Nastran Quick Reference Guide*).
8	The column headed TIME STAMP is the time the file was created. The format is YYMMDDHHMMSS.
9	The delivery database file assignments. These files contain the Solution Sequences and timing constants.
10	Indicates the FMS statement INIT MASTER (RAM) has been invoked (by default).
11	SUBDMAP name.
12	DMAP module name or subDMAP name.
13	The line REAL CORE IS XXX shows the value of the NASTRAN keyword REAL. This message is generated by using the DIAG 13 command in the Executive Control Section of the MSC Nastran data file. It only appears after module executions which use the value. The open core (i.e., working space) for other modules is set via HICORE (see the MSC Nastran 2020 Installation and Operations Guide).
14	The number that precedes the module name on lines that terminate with BEGN or END is the DMAP statement number. This number corresponds to the statement number in the DMAP listing for the solution sequence being used. The DMAP sequence listing will be printed as a part of the output if DIAG 14 or COMPILER LIST is inserted in the Executive Control Section.

15	BEGN indicates the start of execution of an MSC Nastran module or subDMAP. The functions of these modules are given in Section . Section 4 of the *MSC Nastran Programmer's Manual* provides additional detailed information on MSC Nastran functional modules. END indicates the end of execution of an MSC Nastran module or subDMAP. The resources utilized by each module can be estimated by computing the incremental CPU SECs and I/O SECs required to execute the module. This line of information will not normally appear unless the module required more than the CPU seconds defined in SYSTEM(20). SYSTEM is defined via the NASTRAN statement and has a default of 5. M is the timing constant for the user's computer type, as described in the MSC Nastran 2020 Installation and Operations Guide. The user may force the output of this line by including the keyword SYSTEM(20) = 0 on the NASTRAN statement.
16	The line "*8**" is a typical matrix trailer. This line is output for each MSC Nastran matrix that is generated if DIAG 8 is included in the Executive Control Section. Some parameters may be redefined using DMAP as explained in the *MSC Nastran DMAP Programmer's Guide*. Other terms in the DIAG 8 output are: the number of GINO blocks in the matrix (BLOCKS); the average number of terms per string (STRL); the number of strings in a matrix data block (NBRSTR); the number of bank terms in a matrix data block on the average (BNDL); the number of banks in a matrix data block (NBRBND); the average first nonzero row position (ROWL); the average of full bank width (BNDAVG); the maximum full bank width of all columns of data blocks (BNDMAX); the number of null columns in a data block (NULCOL).
17	User Information Message 4157 describes parameters for decomposition. The format of this message varies slightly if either sparse or parallel methods are selected.
18	The time estimation and method selection data are printed for DCMP, FBS, and MPYAD modules if DIAG 19 is included in the Executive Control Section.
19	A detailed analysis of timing data will be printed in the .f04 file if DIAG 49 is set. The results of this analysis appear at the end of the .f04 file in three separate sections. The first section contains information about each module and submodule. The second section reports statistical information about the submodules (FBS, MPYAD, and DCMP) grouped by specific methods. Instances where the CPU estimate does not pass the following criteria are output: $\frac{\lvert\text{Estimated CPU} - \text{Actual CPU}\rvert}{\text{Actual CPU}} < 0.05$ or $\lvert\text{Estimated CPU} - \text{Actual CPU}\rvert < 5\%$ of the total CPU of the job The third section contains information about the modules listed in alphabetical order. (The numbers shown are for a very small model with execution times less than the integer seconds used in these calculations.)
20	This table provides the CPU usage for each process created and executived by a parallel module.
21	The column headed BLOCKS USED contains the number of GINO and executive blocks used at the end of the job. One GINO Block = BUFFSIZE-1 words.
22	The column headed HIWATER BLOCK contains the maximum number of GINO and executive blocks used during the entire execution.

Standard Output Contents

Listing 7-1 Example Execution Summary File (.f04)

```
MACHINE                          MODEL              OPERATING SYSTEM       MSC/NASTRAN       BUILD DATE         RUN DATE
DIGITAL EQUIPMENT                VAX 6000-410       VMS V5.4-3             VERSION 68        MAR  3, 1994       MAR 11, 1994

                              ===    M S C / N A S T R A N          E X E C U T I O N   S U M M A R Y    ===


DAY TIME  ELAPSED  I/O SEC  DEL_I/O  CPU SEC  DEL_CPU   SUB_DMAP/DMAP_MODULE MESSAGES

10:01:52     0:00     0.0      0.0      0.0      0.0    DBINIT   BGN
     ** CURRENT PROJECT ID = ' "BLANK" '  ** CURRENT VERSION ID =          1


   S U M M A R Y   O F   F I L E   A S S I G N M E N T   F O R   T H E   P R I M A R Y   D A T A B A S E    ( DBSNO   1, V68   )
   (9)

  ASSIGNED PHYSICAL FILE NAME (/ORIGINAL)      LOGICAL NAME     DBSET        STATUS    BUFFSIZE     CLUSTER SIZE      TIME STAMP
   ---------------------------------------     ------------     -----        ------    --------     ------------      ----------
   SCR_KNIGHT:[SCRATCH]N10013850.MASTER        MASTER           MASTER       NEW          1409          1           940311100153
   SCR_KNIGHT:[SCRATCH]N10013850.USROBJ        USROBJ           USROBJ       NEW          1409          1           940311100154
   SCR_KNIGHT:[SCRATCH]N10013850.USRSOU        USRSOU           USRSOU       NEW          1409          1           940311100155
   SCR_KNIGHT:[SCRATCH]N10013850.DBALL         DBALL            DBALL        NEW          1409          1           940311100156
   SCR_KNIGHT:[SCRATCH]N10013850.OBJSCR        OBJSCR           OBJSCR       NEW          1409          1           940311100157
            **** MEM FILE ****                  * N/A *          SCRATCH
   SCR_KNIGHT:[SCRATCH]N10013850.SCRATCH       SCRATCH          SCRATCH      NEW          1409          1           940311100158
   SCR_KNIGHT:[SCRATCH]N10013850.SCR300        SCR300           SCRATCH      NEW          1409          1           940311100159
10:01:57     0:05     4.5      4.5      0.6      0.7    DBINIT   END
10:01:57     0:05     4.7      0.2      0.7      0.0    XCSA     BGN


   S U M M A R Y   O F   F I L E   A S S I G N M E N T   F O R   T H E   D E L I V E R Y   D A T A B A S E   ( DBSNO   2, V68   )


  ASSIGNED PHYSICAL FILE NAME (/ORIGINAL)      LOGICAL NAME     DBSET        STATUS    BUFFSIZE     CLUSTER SIZE      TIME STAMP
   ---------------------------------------     ------------     -----        ------    --------     ------------      ----------
   nast:[nast68.DEL]SSS.MASTERA                MASTERA          MASTER       OLD          1409          1           940303153920
  /NAST:[NAST68.DEL.SOURCE.OUT]SSS.MASTERA
   (10)
  /NAST:[NAST68.DEL.SOURCE.OUT]SSS.MSCOBJ
   nast:[nast68.DEL]SSS.MSCSOU                 MSCSOU           MSCSOU       OLD          1409          1           940303154032
  /NAST:[NAST68.DEL.SOURCE.OUT]SSS.MSCSOU
10:02:57     1:05   115.3    110.7      7.1      6.4    XCSA     END
10:02:57     1:05   115.3      0.0      7.1      0.0    CGPI     BGN
10:02:57     1:05   115.3      0.0      7.1      0.1    CGPI     END
10:02:57     1:05   115.3      0.0      7.1      0.0    LINKER   BGN
10:03:09     1:17   133.6     18.3      9.1      2.0    LINKER   END

   ** MASTER DIRECTORIES ARE LOADED IN MEMORY.                    (11)    (12)

        USER OPENCORE (HICORE)          =     6074952  WORDS
        EXECUTIVE SYSTEM WORK AREA      =       76045  WORDS
        MASTER(RAM)                     =       30000  WORDS
        SCRATCH(MEM) AREA               =      140900  WORDS (     100 BUFFERS)
        BUFFER POOL AREA (GINO/EXEC)    =       52503  WORDS (      37 BUFFERS)
        TOTAL MSC/NASTRAN MEMORY LIMIT =     6374400  WORDS


DAY TIME  ELAPSED  I/O SEC  DEL_I/O  CPU SEC  DEL_CPU   SUB_DMAP/DMAP_MODULE MESSAGES

10:03:11     1:19   135.6      2.0      9.4      0.3    SESTATIC    19  IFPL    BEGN
10:03:11     1:19   135.8      0.1#     9.4      0.0#   SESTATIC    19  IFPL    END
10:03:12     1:20   136.1      0.4      9.5      0.0    IFPL        22  IFP1    BEGN
10:03:12     1:20   136.1      0.0#     9.5      0.0#   IFPL        22  IFP1    END
10:03:12     1:20   136.1      0.0      9.5      0.0    IFPL        24  XEQUIV  BEGN
10:03:12     1:20   136.1      0.0#     9.5      0.0#   IFPL        24  XEQUIV  END
10:03:12     1:20   136.1      0.0      9.5      0.0    IFPL        25  XEQUIV  BEGN
10:03:12     1:20   136.1      0.0#     9.5      0.0#   IFPL        25  XEQUIV  END
10:03:12     1:20   136.1      0.0      9.5      0.0    IFPL        26  XEQUIV  BEGN
10:03:12     1:20   136.1      0.0#     9.5      0.0#   IFPL        26  XEQUIV  END
10:03:12     1:20   136.1      0.0      9.5      0.0    IFPL        27  XEQUIV  BEGN
10:03:12     1:20   136.1      0.0#     9.5      0.0#   IFPL        27  XEQUIV  END
10:03:12     1:20   136.1      0.0      9.6      0.0    IFPL        28  XEQUIV  BEGN
10:03:12     1:20   136.1      0.0#     9.6      0.0#   IFPL        28  XEQUIV  END
10:03:12     1:20   136.1      0.0      9.6      0.0    IFPL        35  XSORT   BEGN
                                                        REAL CORE IS       6074952
10:03:14     1:22   136.1      0.0#    10.8      1.3#   IFPL        35  XSORT   END
10:03:36     1:44   137.1      1.0     14.8      4.0    IFPL        61  IFP     BEGN
10:03:37     1:45   137.1      0.0     14.9      0.1    IFP
10:03:39     1:47   137.1      0.0#    15.1      0.3#   IFPL        61  IFP     END
10:03:39     1:47   137.1      0.0     15.1      0.0    IFPL        62  MODEPT  BEGN
10:03:39     1:47   137.1      0.0#    15.1      0.0#   IFPL        62  MODEPT  END
```

(13)

Listing 7-1 Example Execution Summary File (.f04) (continued)

```
10:03:39    1:47   137.1      0.0     15.2      0.0    IFPL        63  MODGM2  BEGN
10:03:40    1:48   137.1      0.0#    15.2      0.0#   IFPL        63  MODGM2  END
10:03:40    1:48   137.1      0.0     15.2      0.0    IFPL        64  PVT     BEGN
10:03:40    1:48   137.1      0.0#    15.3      0.0#   IFPL        64  PVT     END
10:03:40    1:48   137.6      0.5     15.3      0.1    IFPL        79  PROJVER BEGN
10:03:41    1:49   137.6      0.0#    15.3      0.0#   IFPL        79  PROJVER END
10:03:41    1:49   137.6      0.0     15.3      0.0    IFPL        81  XEQUIV  BEGN
10:03:41    1:49   137.6      0.0#    15.4      0.0#   IFPL        81  XEQUIV  END
10:03:41    1:49   137.6      0.0     15.4      0.0    IFPL        82  XEQUIV  BEGN
10:03:41    1:49   137.6      0.0#    15.4      0.0#   IFPL        82  XEQUIV  END
10:03:41    1:49   137.6      0.0     15.4      0.0    IFPL        83  XEQUIV  BEGN
10:03:41    1:49   137.6      0.0#    15.4      0.0#   IFPL        83  XEQUIV  END
10:03:41    1:49   137.6      0.0     15.4      0.0    IFPL        84  XEQUIV  BEGN
10:03:41    1:49   137.6      0.0#    15.4      0.0#   IFPL        84  XEQUIV  END
10:03:41    1:49   137.6      0.0     15.4      0.0    IFPL        85  XEQUIV  BEGN
10:03:41    1:49   137.6      0.0#    15.4      0.0#   IFPL        85  XEQUIV  END
10:03:41    1:49   137.6      0.0     15.4      0.0    IFPL        86  XEQUIV  BEGN
10:03:41    1:49   137.6      0.0#    15.4      0.0#   IFPL        86  XEQUIV  END
10:03:41    1:49   137.6      0.0     15.4      0.0    IFPL        87  XEQUIV  BEGN
10:03:41    1:49   137.6      0.0#    15.4      0.0#   IFPL        87  XEQUIV  END
10:03:41    1:49   137.6      0.0     15.4      0.0    IFPL        88  XEQUIV  BEGN
10:03:41    1:49   137.6      0.0#    15.4      0.0#   IFPL        88  XEQUIV  END
10:03:41    1:49   137.6      0.0     15.4      0.0    IFPL        89  XEQUIV  BEGN
10:03:41    1:49   137.6      0.0#    15.4      0.0#   IFPL        89  XEQUIV  END
10:03:41    1:49   137.6      0.0     15.4      0.0    IFPL        90  XEQUIV  BEGN
10:03:41    1:49   137.6      0.0#    15.4      0.0#   IFPL        90  XEQUIV  END
10:03:41    1:49   137.6      0.0     15.5      0.0    IFPL        92  XEQUIV  BEGN
10:03:41    1:49   137.6      0.0#    15.5      0.0#   IFPL        92  XEQUIV  END
10:03:41    1:49   137.6      0.0     15.5      0.0    IFPL        93  XEQUIV  BEGN
10:03:41    1:49   137.6      0.0#    15.5      0.0#   IFPL        93  XEQUIV  END
10:03:41    1:49   137.6      0.0     15.5      0.0    IFPL        94  XEQUIV  BEGN
10:03:41    1:49   137.6      0.0#    15.5      0.0#   IFPL        94  XEQUIV  END
10:03:41    1:49   137.6      0.0     15.5      0.0    IFPL        96  XEQUIV  BEGN
10:03:41    1:49   137.6      0.0#    15.5      0.0#   IFPL        96  XEQUIV  END
10:03:41    1:49   137.6      0.0     15.5      0.0    IFPL        97  XEQUIV  BEGN
10:03:41    1:49   137.6      0.0#    15.5      0.0#   IFPL        97  XEQUIV  END
10:03:41    1:49   137.6      0.0     15.5      0.0    IFPL        98  XEQUIV  BEGN
10:03:41    1:49   137.6      0.0#    15.6      0.0#   IFPL        98  XEQUIV  END
10:03:41    1:49   137.6      0.0     15.6      0.0    IFPL        99  XEQUIV  BEGN
10:03:41    1:49   137.6      0.0#    15.6      0.0#   IFPL        99  XEQUIV  END
10:03:41    1:49   137.6      0.0     15.6      0.0    IFPL       100  XEQUIV  BEGN
10:03:41    1:49   137.6      0.0#    15.6      0.0#   IFPL       100  XEQUIV  END
10:03:41    1:49   137.6      0.0     15.6      0.0    IFPL       101  XEQUIV  BEGN
10:03:41    1:49   137.6      0.0#    15.6      0.0#   IFPL       101  XEQUIV  END
10:03:41    1:49   137.6      0.0     15.6      0.0    IFPL       115  PVT     BEGN
10:03:41    1:49   137.6      0.0#    15.6      0.0#   IFPL       115  PVT     END
10:03:41    1:49   137.6      0.0     15.6      0.0    IFPL       128  DELETE  BEGN
10:03:41    1:49   137.6      0.0#    15.6      0.0#   IFPL       128  DELETE  END
10:03:41    1:49   137.8      0.2     15.6      0.0    IFPL       154  PLTSET  BEGN
10:03:41    1:49   137.8      0.0#    15.7      0.0#   IFPL       154  PLTSET  END
10:03:41    1:49   137.8      0.0     15.7      0.0    IFPL       157  GP0     BEGN
10:03:42    1:50   137.8      0.0#    15.7      0.0#   IFPL       157  GP0     END
10:03:42    1:50   137.8      0.0     15.7      0.0    IFPL       176  SEQP    BEGN
10:03:43    1:51   137.8      0.0#    15.9      0.1#   IFPL       176  SEQP    END
10:03:43    1:51   137.8      0.0     15.9      0.0    IFPL       182  GP1     BEGN
10:03:44    1:52   137.8      0.0#    16.1      0.2#   IFPL       182  GP1     END
10:03:44    1:52   137.9      0.1     16.1      0.0    SESTATIC    14  PHASE0  BEGN
10:03:44    1:52   138.1      0.2#    16.1      0.0#   SESTATIC    14  PHASE0  END
10:03:44    1:52   138.1      0.0     16.1      0.0    PHASE0      24  DTIIN   BEGN
10:03:44    1:52   138.1      0.0#    16.2      0.0#   PHASE0      24  DTIIN   END
10:03:44    1:52   138.1      0.0     16.2      0.0    PHASE0      25  GP1     BEGN
10:03:45    1:53   138.1      0.0#    16.4      0.2#   PHASE0      25  GP1     END
10:03:46    1:54   138.1      0.0     16.4      0.0    PHASE0      28  SEP1    BEGN
10:03:46    1:54   138.1      0.0#    16.4      0.0#   PHASE0      28  SEP1    END
10:03:46    1:54   138.1      0.0     16.4      0.0    PHASE0      37  XEQUIV  BEGN
10:03:46    1:54   138.1      0.0#    16.4      0.0#   PHASE0      37  XEQUIV  END
10:03:46    1:54   138.1      0.0     16.5      0.0    PHASE0      40  PVT     BEGN
10:03:48    1:56   138.1      0.0#    16.6      0.2#   PHASE0      40  PVT     END
10:03:48    1:56   138.1      0.0     16.6      0.0    PHASE0      46  TABPRT  BEGN
10:03:48    1:56   138.1      0.0#    16.7      0.0#   PHASE0      46  TABPRT  END
10:03:48    1:56   138.1      0.0     16.7      0.0    PHASE0      47  SEP3    BEGN
10:03:48    1:56   138.1      0.0#    16.7      0.0#   PHASE0      47  SEP3    END
10:03:48    1:56   138.2      0.1     16.7      0.1    PHASE0      88  PVT     BEGN
10:03:49    1:57   138.2      0.0#    16.8      0.1#   PHASE0      88  PVT     END
10:03:49    1:57   138.9      0.7     16.8      0.0    PHASE0      96  XEQUIV  BEGN
10:03:49    1:57   138.9      0.0#    16.8      0.0#   PHASE0      96  XEQUIV  END
10:03:49    1:57   138.9      0.0     16.9      0.0    PHASE0      97  XEQUIV  BEGN
10:03:49    1:57   138.9      0.0#    16.9      0.0#   PHASE0      97  XEQUIV  END
10:03:49    1:57   138.9      0.0     16.9      0.0    PHASE0      98  XEQUIV  BEGN
10:03:49    1:57   138.9      0.0#    16.9      0.0#   PHASE0      98  XEQUIV  END
10:03:49    1:57   138.9      0.0     16.9      0.0    PHASE0      99  XEQUIV  BEGN
10:03:49    1:57   138.9      0.0#    16.9      0.0#   PHASE0      99  XEQUIV  END
10:03:49    1:57   138.9      0.0     16.9      0.0    PHASE0     100  XEQUIV  BEGN
10:03:49    1:57   138.9      0.0#    16.9      0.0#   PHASE0     100  XEQUIV  END
```

(14) (15)

Listing 7-1 Example Execution Summary File (.f04) (continued)

```
10:03:49     1:57   138.9      0.0     16.9      0.0    PHASE0      101 XEQUIV  BEGN
10:03:49     1:57   138.9      0.0#    16.9      0.0#   PHASE0      101 XEQUIV  END
10:03:49     1:57   138.9      0.0     16.9      0.0    PHASE0      102 XEQUIV  BEGN
10:03:49     1:57   138.9      0.0#    16.9      0.0#   PHASE0      102 XEQUIV  END
10:03:49     1:57   138.9      0.0     16.9      0.0    PHASE0      103 XEQUIV  BEGN
10:03:49     1:57   138.9      0.0#    16.9      0.0#   PHASE0      103 XEQUIV  END
10:03:49     1:57   138.9      0.0     16.9      0.0    PHASE0      104 XEQUIV  BEGN
10:03:49     1:57   138.9      0.0#    16.9      0.0#   PHASE0      104 XEQUIV  END
10:03:49     1:57   138.9      0.0     16.9      0.0    PHASE0      105 XEQUIV  BEGN
10:03:49     1:57   138.9      0.0#    17.0      0.0#   PHASE0      105 XEQUIV  END
10:03:50     1:58   140.7      1.8     17.1      0.2    PHASE0      224 PROJVER BEGN
10:03:50     1:58   140.7      0.0#    17.1      0.0#   PHASE0      224 PROJVER END
10:03:50     1:58   140.7      0.0     17.1      0.0    PHASE0      241 SETQ    BEGN
10:03:50     1:58   140.8      0.1#    17.1      0.0#   PHASE0      241 SETQ    END
10:03:50     1:58   140.8      0.0     17.1      0.0    SETQ         10 XEQUIV  BEGN
10:03:50     1:58   140.8      0.0#    17.1      0.0#   SETQ         10 XEQUIV  END
10:03:50     1:58   140.8      0.1     17.3      0.2    PHASE0      307 GP1     BEGN
*8** MODULE  DMAP MATRIX     COLS    ROWS  F  T  NZWDS    DENS BLOCKS STRL  NBRSTR BNDL  NBRBND ROW1 BNDAVG BNDMAX NULCOL
     GP1     307  VGFS          1      12  2  2      0 0.0000      1    0       1    0       0    0       0       0      1 *8**
10:03:51     1:59   140.8      0.0#    17.5      0.2#   PHASE0      307 GP1     END
10:03:51     1:59   140.9      0.1     17.5      0.0    PHASE0      311 XEQUIV  BEGN
10:03:51     1:59   140.9      0.0#    17.5      0.0#   PHASE0      311 XEQUIV  END
10:03:51     1:59   140.9      0.0     17.5      0.0    PHASE0      313 SEPREP2 BEGN
10:03:51     1:59   141.0      0.1#    17.5      0.0#   PHASE0      313 SEPREP2 END
10:03:51     1:59   141.0      0.0     17.5      0.0    SEPREP2       6 GP2     BEGN
10:03:51     1:59   141.0      0.0#    17.5      0.0#   SEPREP2       6 GP2     END
10:03:51     1:59   141.1      0.1     17.6      0.0    SEPREP2      12 XEQUIV  BEGN
10:03:51     1:59   141.1      0.0#    17.6      0.0#   SEPREP2      12 XEQUIV  END
10:03:51     1:59   141.1      0.0     17.6      0.0    SEPREP2      13 XEQUIV  BEGN
10:03:51     1:59   141.1      0.0#    17.6      0.0#   SEPREP2      13 XEQUIV  END
10:03:51     1:59   141.1      0.0     17.6      0.0    SEPREP2      14 XEQUIV  BEGN
10:03:51     1:59   141.1      0.0#    17.6      0.0#   SEPREP2      14 XEQUIV  END
10:03:51     1:59   141.1      0.0     17.6      0.0    SEPREP2      15 XEQUIV  BEGN
10:03:51     1:59   141.1      0.0#    17.6      0.0#   SEPREP2      15 XEQUIV  END
10:03:51     1:59   141.1      0.0     17.6      0.0    PHASE0      332 XEQUIV  BEGN
10:03:51     1:59   141.1      0.0#    17.6      0.0#   PHASE0      332 XEQUIV  END
10:03:51     1:59   141.1      0.0     17.6      0.0    PHASE0      333 XEQUIV  BEGN
10:03:51     1:59   141.1      0.0#    17.6      0.0#   PHASE0      333 XEQUIV  END
10:03:51     1:59   141.1      0.0     17.7      0.0    PHASE0      335 GP3     BEGN
10:03:51     1:59   141.1      0.0#    17.7      0.0#   PHASE0      335 GP3     END
10:03:51     1:59   141.1      0.0     17.7      0.0    PHASE0      336 LCGEN   BEGN
10:03:51     1:59   141.1      0.0#    17.7      0.0#   PHASE0      336 LCGEN   END
10:03:51     1:59   141.1      0.0     17.7      0.0    PHASE0      341 XEQUIV  BEGN
10:03:51     1:59   141.1      0.0#    17.7      0.0#   PHASE0      341 XEQUIV  END
10:03:51     1:59   141.1      0.0     17.8      0.0    PHASE0      352 BCDR    BEGN
10:03:51     1:59   141.1      0.0#    17.8      0.0#   PHASE0      352 BCDR    END
10:03:51     1:59   141.2      0.1     17.8      0.0    PHASE0      367 GP4     BEGN
*8** MODULE  DMAP MATRIX     COLS    ROWS  F  T  NZWDS    DENS BLOCKS STRL  NBRSTR BNDL  NBRBND ROW1 BNDAVG BNDMAX NULCOL
     GP4     367  YSB           1       6  2  2      0 0.0000      1    0       1    0       0    0       0       0      1 *8**
10:03:52     2:00   141.2      0.0#    17.8      0.0#   PHASE0      367 GP4     END
10:03:52     2:00   141.2      0.0     17.9      0.0    PHASE0      373 XPURGE  BEGN
10:03:52     2:00   141.2      0.0#    17.9      0.0#   PHASE0      373 XPURGE  END
10:03:52     2:00   141.2      0.0     17.9      0.0    PHASE0      378 UMERGE1 BEGN
*8** MODULE  DMAP MATRIX     COLS    ROWS  F  T  NZWDS    DENS BLOCKS STRL  NBRSTR BNDL  NBRBND ROW1 BNDAVG BNDMAX NULCOL
     UMERGE1 378  SCRATCH       1      12  2  1      6 0.5000      1    6       1    0       0    0       6       6      0 *8**
10:03:52     2:00   141.2      0.0#    17.9      0.0#   PHASE0      378 UMERGE1 END
10:03:52     2:00   141.3      0.1     18.0      0.1    SESTATIC     39 SETQ    BEGN
10:03:52     2:00   141.3      0.0#    18.0      0.0#   SESTATIC     39 SETQ    END
10:03:52     2:00   141.3      0.0     18.0      0.0    SETQ         10 XEQUIV  BEGN
10:03:52     2:00   141.3      0.0#    18.0      0.0#   SETQ         10 XEQUIV  END
10:03:52     2:00   141.3      0.0     18.2      0.2    SESTATIC     40 DBSTATUSBEGN
10:03:52     2:00   141.3      0.0#    18.2      0.0#   SESTATIC     40 DBSTATUSEND
10:03:52     2:00   141.3      0.0     18.2      0.0    SESTATIC     45 MATGEN  BEGN
*8** MODULE  DMAP MATRIX     COLS    ROWS  F  T  NZWDS    DENS BLOCKS STRL  NBRSTR BNDL  NBRBND ROW1 BNDAVG BNDMAX NULCOL
     MATGEN  45   TEMPALL       2       2  6  2      2 0.5000      1    1       2    0       0    0       1       1      0 *8**
10:03:52     2:00   141.3      0.0#    18.2      0.0#   SESTATIC     45 MATGEN  END
10:03:52     2:00   141.3      0.0     18.2      0.0    SESTATIC     46 RESTART BEGN
       TEMPALL                                HAS CHANGED
10:03:52     2:00   141.9      0.6#    18.3      0.0#   SESTATIC     46 RESTART END
10:03:53     2:01   142.0      0.1     18.3      0.0    SESTATIC     47 DELETE  BEGN
10:03:53     2:01   142.0      0.0#    18.3      0.0#   SESTATIC     47 DELETE  END
10:03:53     2:01   142.1      0.1     18.3      0.0    SESTATIC     49 PHASE1DRBEGN
10:03:53     2:01   142.2      0.1#    18.3      0.0#   SESTATIC     49 PHASE1DREND
10:03:53     2:01   142.4      0.2     18.4      0.1    PHASE1DR     70 PVT     BEGN
10:03:53     2:01   142.4      0.0#    18.4      0.1#   PHASE1DR     70 PVT     END
10:03:53     2:01   142.4      0.0     18.4      0.0    PHASE1DR     71 SETQ    BEGN
10:03:53     2:01   142.4      0.0#    18.5      0.0#   PHASE1DR     71 SETQ    END
10:03:53     2:01   142.4      0.0     18.5      0.0    SETQ         10 XEQUIV  BEGN
10:03:53     2:01   142.4      0.0#    18.5      0.0#   SETQ         10 XEQUIV  END
10:03:53     2:01   142.4      0.0     18.6      0.2    PHASE1DR     74 PHASE1A BEGN
10:03:53     2:01   142.6      0.2#    18.7      0.0#   PHASE1DR     74 PHASE1A END
10:03:53     2:01   142.6      0.0     18.7      0.0    PHASE1A      24 TA1     BEGN
                                                 REAL CORE IS         6073541
```

Listing 7-1 Example Execution Summary File (.f04) (continued)

```
*8** MODULE   DMAP MATRIX     COLS    ROWS  F  T  NZWDS    DENS BLOCKS STRL  NBRSTR BNDL  NBRBND ROW1 BNDAVG BNDMAX NULCOL
     EMA      79   KJJZ         12      12  6  2       8 0.2777      1    0      40    0       0    0       9      11       0 *8**
10:03:54     2:02   142.9       0.0#     19.0       0.0#   SEMG         79  EMA     END
10:03:54     2:02   142.9        0.0     19.0        0.0   SEMG         95  XPURGE  BEGN
10:03:54     2:02   142.9       0.0#     19.0       0.0#   SEMG         95  XPURGE  END
10:03:55     2:03   143.1        0.2     19.0        0.0   SEMG         98  MTRXIN  BEGN
                                                      REAL CORE IS          6074952
10:03:55     2:03   143.1       0.0#     19.0       0.0#   SEMG         98  MTRXIN  END
10:03:55     2:03   143.1        0.0     19.0        0.0   SEMG        103  SEMG1   BEGN
10:03:55     2:03   143.1       0.1#     19.1       0.0#   SEMG        103  SEMG1   END
10:03:55     2:03   143.1        0.0     19.1        0.0   SEMG1        8   XEQUIV  BEGN
10:03:55     2:03   143.1       0.0#     19.1       0.0#   SEMG1        8   XEQUIV  END
10:03:55     2:03   143.1        0.0     19.1        0.0   SEMG1       15   XEQUIV  BEGN
10:03:55     2:03   143.1       0.0#     19.1       0.0#   SEMG1       15   XEQUIV  END
10:03:55     2:03   143.1        0.0     19.1        0.0   SEMG1       17   XEQUIV  BEGN
10:03:55     2:03   143.1       0.0#     19.1       0.0#   SEMG1       17   XEQUIV  END
10:03:55     2:03   143.1        0.0     19.1        0.0   SEMG1       19   XEQUIV  BEGN
10:03:55     2:03   143.1       0.0#     19.1       0.0#   SEMG1       19   XEQUIV  END
10:03:55     2:03   143.1        0.0     19.1        0.0   PHASE1A     31   SESUM   BEGN
10:03:55     2:03   143.1       0.0#     19.1       0.0#   PHASE1A     31   SESUM   END
10:03:55     2:03   143.1        0.0     19.2        0.0   SESUM       22   DELETE  BEGN
10:03:55     2:03   143.1       0.0#     19.2       0.0#   SESUM       22   DELETE  END
10:03:55     2:03   143.4        0.3     19.2        0.0   SESUM       25   XPURGE  BEGN
10:03:55     2:03   143.4       0.0#     19.2       0.0#   SESUM       25   XPURGE  END
10:03:55     2:03   143.6        0.2     19.2        0.0   PHASE1A     35   XEQUIV  BEGN
10:03:55     2:03   143.6       0.0#     19.2       0.0#   PHASE1A     35   XEQUIV  END
10:03:55     2:03   143.6        0.0     19.2        0.0   PHASE1A     60   SELG    BEGN
10:03:55     2:03   143.7       0.1#     19.3       0.0#   PHASE1A     60   SELG    END
10:03:56     2:03   143.7        0.0     19.3        0.0   SELG        11   SSG1    BEGN
*8** MODULE   DMAP MATRIX     COLS    ROWS  F  T  NZWDS    DENS BLOCKS STRL  NBRSTR BNDL  NBRBND ROW1 BNDAVG BNDMAX NULCOL

     SSG1     11   SCRATCH      1      12  2  2       6 0.2500      1    3       1    0       0    0       3       3       0 *8**
     SSG1     11   PJX          1      12  2  2       6 0.2500      1    3       1    0       0    0       3       3       0 *8**
10:03:56     2:04   143.7       0.0#     19.3       0.1#   SELG        11   SSG1    END
10:03:56     2:04   143.7        0.0     19.4        0.0   SELG        22   XEQUIV  BEGN
10:03:56     2:04   143.7       0.0#     19.4       0.0#   SELG        22   XEQUIV  END
10:03:56     2:04   143.7        0.0     19.4        0.0   SELG        60   XEQUIV  BEGN
10:03:56     2:04   143.7       0.0#     19.4       0.0#   SELG        60   XEQUIV  END
10:03:56     2:04   143.7        0.0     19.4        0.0   SELG        62   VECPLOT BEGN
*8** MODULE   DMAP MATRIX     COLS    ROWS  F  T  NZWDS    DENS BLOCKS STRL  NBRSTR BNDL  NBRBND ROW1 BNDAVG BNDMAX NULCOL
     VECPLOT  62   PPJ          1      12  2  1       3 0.2500      1    3       1    0       0    0       3       3       0 *8**
10:03:56     2:04   143.7       0.0#     19.4       0.0#   SELG        62   VECPLOT END
10:03:56     2:04   143.7        0.0     19.5        0.0   PHASE1A     61   SESUM   BEGN
10:03:56     2:04   143.7       0.0#     19.5       0.0#   PHASE1A     61   SESUM   END
10:03:56     2:04   143.7        0.0     19.5        0.0   SESUM       22   DELETE  BEGN
10:03:56     2:04   143.8       0.1#     19.5       0.0#   SESUM       22   DELETE  END
10:03:56     2:04   144.4        0.6     19.6        0.0   SESUM       25   XPURGE  BEGN
10:03:56     2:04   144.4       0.0#     19.6       0.0#   SESUM       25   XPURGE  END
10:03:57     2:05   144.5        0.1     19.6        0.0   PHASE1A     65   SELA1   BEGN
10:03:57     2:05   144.6       0.1#     19.6       0.0#   PHASE1A     65   SELA1   END
10:03:57     2:05   144.6        0.0     19.6        0.0   SELA1        9   XEQUIV  BEGN
10:03:57     2:05   144.6       0.0#     19.6       0.0#   SELA1        9   XEQUIV  END
10:03:57     2:05   144.6        0.0     19.6        0.0   SELA1       11   XEQUIV  BEGN
10:03:57     2:05   144.6       0.0#     19.6       0.0#   SELA1       11   XEQUIV  END
10:03:57     2:05   144.6        0.0     19.7        0.0   PHASE1DR    78   DBSTATUSBEGN
10:03:57     2:05   144.6       0.0#     19.7       0.0#   PHASE1DR    78   DBSTATUSEND
10:03:57     2:05   144.6        0.0     19.7        0.0   PHASE1DR    80   XEQUIV  BEGN
10:03:57     2:05   144.6       0.0#     19.7       0.0#   PHASE1DR    80   XEQUIV  END
10:03:57     2:05   144.6        0.0     19.7        0.0   PHASE1DR    98   BCDR    BEGN
10:03:57     2:05   144.6       0.0#     19.7       0.0#   PHASE1DR    98   BCDR    END
10:03:57     2:05   144.6        0.0     19.7        0.0   PHASE1DR   100   PHASE1B BEGN
10:03:57     2:05   144.7       0.1#     19.8       0.0#   PHASE1DR   100   PHASE1B END
10:03:57     2:05   144.7        0.0     19.8        0.0   PHASE1B     31   DBSTATUSBEGN
10:03:57     2:05   144.7       0.0#     19.8       0.0#   PHASE1B     31   DBSTATUSEND
10:03:57     2:05   144.7        0.0     19.8        0.0   PHASE1B     32   SEKR    BEGN
10:03:57     2:05   144.9       0.2#     19.8       0.0#   PHASE1B     32   SEKR    END
10:03:57     2:05   144.9        0.0     19.9        0.0   SEKR        13   XEQUIV  BEGN
10:03:57     2:05   144.9       0.0#     19.9       0.0#   SEKR        13   XEQUIV  END
10:03:57     2:05   144.9        0.0     19.9        0.0   SEKR        14   XPURGE  BEGN
10:03:57     2:05   144.9       0.0#     19.9       0.0#   SEKR        14   XPURGE  END
10:03:57     2:05   144.9        0.0     19.9        0.0   SEKR        29   GPSP    BEGN
10:03:57     2:05   144.9       0.0#     20.0       0.0#   SEKR        29   GPSP    END
10:03:57     2:05   144.9        0.0     20.0        0.0   SEKR        30   PMLUSET BEGN
10:03:57     2:05   144.9       0.0#     20.0       0.0#   SEKR        30   PMLUSET END
10:03:57     2:05   144.9        0.0     20.0        0.0   SEKR        45   XEQUIV  BEGN
10:03:57     2:05   144.9       0.0#     20.0       0.0#   SEKR        45   XEQUIV  END
10:03:57     2:05   144.9        0.0     20.0        0.0   SEKR        46   XPURGE  BEGN
10:03:57     2:05   144.9       0.0#     20.0       0.0#   SEKR        46   XPURGE  END
10:03:57     2:05   144.9        0.0     20.0        0.0   SEKR        47   UPARTN  BEGN
```

Listing 7-1 Example Execution Summary File (.f04) (continued)

```
*8** MODULE  DMAP MATRIX     COLS    ROWS  F  T  NZWDS   DENS BLOCKS STRL  NBRSTR BNDL  NBRBND ROW1 BNDAVG BNDMAX NULCOL
     UPARTN  47   SCRATCH       1      12  2  1      6 0.5000      1    6       1    0       0    0       6       6       0 *8**
     UPARTN  47   KFF           6       6  6  2      4 0.2777      1    0      10    0       0    0       3       5       0 *8**
     UPARTN  47   KSF           6       6  6  2      4 0.2777      1    0      10    0       0    0       3       5       0 *8**
     UPARTN  47   KFS           6       6  6  2      4 0.2777      1    0      10    0       0    0       3       5       0 *8**
     UPARTN  47   KSS           6       6  6  2      4 0.2777      1    0      10    0       0    0       3       5       0 *8**
10:03:58     2:06   144.9      0.0#     20.1       0.0#   SEKR          47  UPARTN  END
10:03:58     2:06   144.9      0.0      20.1       0.0    SEKR          48  XEQUIV  BEGN
10:03:58     2:06   144.9      0.0#     20.1       0.0#   SEKR          48  XEQUIV  END
10:03:58     2:06   144.9      0.0      20.1       0.0    SEKR          49  XPURGE  BEGN
10:03:58     2:06   144.9      0.0#     20.1       0.0#   SEKR          49  XPURGE  END
10:03:58     2:06   144.9      0.0      20.1       0.0    SEKR          54  XPURGE  BEGN
10:03:58     2:06   144.9      0.0#     20.1       0.0#   SEKR          54  XPURGE  END
10:03:58     2:06   144.9      0.0      20.1       0.0    SEKR         110  SESUM   BEGN
10:03:58     2:06   144.9      0.0#     20.1       0.0#   SEKR         110  SESUM   END
10:03:58     2:06   144.9      0.0      20.2       0.0    SESUM         22  DELETE  BEGN
10:03:58     2:06   144.9      0.0#     20.2       0.0#   SESUM         22  DELETE  END
10:03:58     2:06   145.3      0.4      20.2       0.0    SESUM         25  XPURGE  BEGN
10:03:58     2:06   145.3      0.0#     20.2       0.0#   SESUM         25  XPURGE  END
10:03:58     2:06   145.4      0.1      20.3       0.0    PHASE1B       33  SESUM   BEGN
10:03:58     2:06   145.4      0.0#     20.3       0.0#   PHASE1B       33  SESUM   END
10:03:59     2:07   145.4      0.0      20.3       0.1    SESUM         22  DELETE  BEGN
10:03:59     2:07   145.4      0.0#     20.3       0.0#   SESUM         22  DELETE  END
10:03:59     2:07   145.9      0.4      20.4       0.0    SESUM         25  XPURGE  BEGN
10:03:59     2:07   145.9      0.0#     20.4       0.0#   SESUM         25  XPURGE  END
10:03:59     2:07   146.0      0.1      20.4       0.0    PHASE1B       36  XEQUIV  BEGN
10:03:59     2:07   146.0      0.0#     20.4       0.0#   PHASE1B       36  XEQUIV  END
10:03:59     2:07   146.1      0.1      20.4       0.0    PHASE1B       41  PMLUSET BEGN
10:03:59     2:07   146.1      0.0#     20.4       0.0#   PHASE1B       41  PMLUSET END
10:03:59     2:07   146.5      0.4      20.5       0.1    PHASE1B       76  SEGOA   BEGN
10:03:59     2:07   146.6      0.1#     20.5       0.0#   PHASE1B       76  SEGOA   END
10:03:59     2:07   146.6      0.0      20.5       0.0    PHASE1B       77  DBSTATUSBEGN
10:03:59     2:07   146.6      0.0#     20.5       0.0#   PHASE1B       77  DBSTATUSEND
10:03:59     2:07   146.6      0.0      20.5       0.0    PHASE1B       78  SELR    BEGN
10:03:59     2:07   146.6      0.1#     20.6       0.0#   PHASE1B       78  SELR    END
10:03:59     2:07   146.6      0.0      20.6       0.0    SELR          16  XEQUIV  BEGN
10:03:59     2:07   146.6      0.0#     20.6       0.0#   SELR          16  XEQUIV  END
10:03:59     2:07   146.6      0.0      20.6       0.0    SELR          37  XEQUIV  BEGN
10:03:59     2:07   146.6      0.0#     20.6       0.0#   SELR          37  XEQUIV  END
10:03:59     2:07   146.6      0.0      20.6       0.0    SELR          38  XPURGE  BEGN
10:03:59     2:07   146.6      0.0#     20.6       0.0#   SELR          38  XPURGE  END
10:03:59     2:07   146.6      0.0      20.6       0.0    SELR          40  XPURGE  BEGN
10:03:59     2:07   146.6      0.0#     20.6       0.0#   SELR          40  XPURGE  END
10:03:59     2:07   146.6      0.0      20.6       0.0    SELR          41  UPARTN  BEGN
*8** MODULE  DMAP MATRIX     COLS    ROWS  F  T  NZWDS   DENS BLOCKS STRL  NBRSTR BNDL  NBRBND ROW1 BNDAVG BNDMAX NULCOL
     UPARTN  41   SCRATCH       1      12  2  1      6 0.5000      1    6       1    0       0    0       6       6       0 *8**
10:03:59     2:07   146.6      0.0#     20.6       0.0#   SELR          41  UPARTN  END
10:03:59     2:07   146.6      0.0      20.6       0.0    SELR          52  XEQUIV  BEGN
10:03:59     2:07   146.6      0.0#     20.6       0.0#   SELR          52  XEQUIV  END
10:03:59     2:07   146.6      0.0      20.7       0.0    SELR          64  SSG2    BEGN
*8** MODULE  DMAP MATRIX     COLS    ROWS  F  T  NZWDS   DENS BLOCKS STRL  NBRSTR BNDL  NBRBND ROW1 BNDAVG BNDMAX NULCOL
     SSG2    64   SCRATCH       1      12  2  1      6 0.5000      1    6       1    0       0    0       6       6       0 *8**
     SSG2    64   SCRATCH       1       6  2  2      6 0.5000      1    3       1    0       0    0       3       3       0 *8**
     SSG2    64   PSS           1       6  2  2      0 0.0000      1    0       1    0       0    0       0       0       1 *8**
                                                  REAL CORE IS          6074952
     SSG2    64   PA            1       6  2  2      6 0.5000      1    3       1    0       0    0       3       3       0 *8**
10:04:00     2:08   146.6      0.0#     20.7       0.1#   SELR          64  SSG2    END
10:04:00     2:08   146.6      0.0      20.7       0.0    SELR          65  XPURGE  BEGN
10:04:00     2:08   146.6      0.0#     20.7       0.0#   SELR          65  XPURGE  END
10:04:00     2:08   146.6      0.0      20.7       0.0    SELR          75  XPURGE  BEGN
10:04:00     2:08   146.6      0.0#     20.8       0.0#   SELR          75  XPURGE  END
10:04:00     2:08   146.6      0.0      20.8       0.0    PHASE1B       80  XEQUIV  BEGN
10:04:00     2:08   146.6      0.0#     20.8       0.0#   PHASE1B       80  XEQUIV  END
10:04:00     2:08   146.6      0.0      20.8       0.0    PHASE1B       81  SESUM   BEGN
10:04:00     2:08   146.6      0.0#     20.8       0.0#   PHASE1B       81  SESUM   END
10:04:00     2:08   146.6      0.0      20.8       0.0    SESUM         22  DELETE  BEGN
10:04:00     2:08   146.6      0.0#     20.8       0.0#   SESUM         22  DELETE  END
10:04:00     2:08   147.1      0.4      20.8       0.0    SESUM         25  XPURGE  BEGN
10:04:00     2:08   147.1      0.0#     20.9       0.0#   SESUM         25  XPURGE  END
10:04:00     2:08   147.5      0.4      21.0       0.1    SESTATIC      63  BCDR    BEGN
10:04:00     2:08   147.5      0.0#     21.0       0.0#   SESTATIC      63  BCDR    END
10:04:00     2:08   147.5      0.0      21.0       0.0    SESTATIC      64  DBSTATUSBEGN
10:04:01     2:09   147.5      0.0#     21.0       0.0#   SESTATIC      64  DBSTATUSEND
10:04:01     2:09   147.5      0.0      21.0       0.0    SESTATIC      71  PHASE1C BEGN
10:04:01     2:09   147.6      0.1#     21.0       0.0#   SESTATIC      71  PHASE1C END
10:04:01     2:09   147.6      0.0      21.0       0.0    PHASE1C        9  DBSTATUSBEGN
10:04:01     2:09   147.6      0.0#     21.1       0.0#   PHASE1C        9  DBSTATUSEND
10:04:01     2:09   147.6      0.0      21.1       0.0    PHASE1C       10  SEKRRS  BEGN
10:04:01     2:09   147.6      0.1#     21.1       0.0#   PHASE1C       10  SEKRRS  END
10:04:01     2:09   147.6      0.0      21.1       0.0    SEKRRS        10  XEQUIV  BEGN
10:04:01     2:09   147.6      0.0#     21.1       0.0#   SEKRRS        10  XEQUIV  END
10:04:01     2:09   147.6      0.0      21.1       0.0    SEKRRS        13  XEQUIV  BEGN
10:04:01     2:09   147.6      0.0#     21.1       0.0#   SEKRRS        13  XEQUIV  END
10:04:01     2:09   147.6      0.0      21.1       0.0    SEKRRS        18  DCMP    BEGN
```

Listing 7-1 Example Execution Summary File (.f04) (continued)

```
*** USER INFORMATION MESSAGE 4157 (DFMSA) ---PARAMETERS FOR SPARSE DECOMPOSITION OF DATA BLOCK  KLL     ( TYPE=RDP ) FOLLOW
      (17)
                MATRIX SIZE =       6 ROWS             NUMBER OF NONZEROES =       8 TERMS
         NUMBER OF ZERO COLUMNS =       0        NUMBER OF ZERO DIAGONAL TERMS =       0
              CPU TIME ESTIMATE =       0 SEC                I/O TIME ESTIMATE =       0 SEC
     ESTIMATED MEMORY REQUIREMENT =      15 K WORDS             MEMORY AVAILABLE =    6074 K WORDS
     EST. INTEGER WORDS IN FACTOR =       1 K WORDS           EST. NONZERO TERMS =       1 K TERMS
     ESTIMATED MAXIMUM FRONT SIZE =       2 TERMS
 *** USER INFORMATION MESSAGE 6439 (DFMSA) ---ACTUAL MEMORY AND DISK SPACE REQUIREMENTS FOR SPARSE SYM. DECOMPOSITION
     SPARSE DECOMP MEMORY REQUIRED =      15 K WORDS           MAXIMUM FRONT SIZE =       2 TERMS
           INTEGER WORDS IN FACTOR =       1 K WORDS      NONZERO TERMS IN FACTOR =       1 K TERMS
 *8** MODULE  DMAP MATRIX     COLS   ROWS  F  T  IBLKS  NBLKS NUMFRT FRTMAX
      DCMP    18   LLL           6      6 13  2      1      1      4      2 *8**
 *8** MODULE  DMAP MATRIX     COLS   ROWS  F  T  NZWDS   DENS BLOCKS STRL  NBRSTR BNDL  NBRBND ROW1 BNDAVG BNDMAX NULCOL
      DCMP    18   SCRATCH       1      6  2  2      6 0.5000      1    3       1    0       0    0      6      6      0 *8**
      DCMP    18   SCRATCH       1      6  2  2      6 0.5000      1    3       1    0       0    0      6      6      0 *8**
 10:04:02     2:10   147.6      0.0#    21.2       0.1#    SEKRRS      18  DCMP    END
 10:04:02     2:10   147.6      0.0     21.3       0.0     SEKRRS      26  XPURGE  BEGN
 10:04:02     2:10   147.6      0.0#    21.3       0.0#    SEKRRS      26  XPURGE  END
 10:04:02     2:10   147.6      0.0     21.3       0.0     PHASE1C     11  SESUM   BEGN
 10:04:02     2:10   147.6      0.0#    21.3       0.0#    PHASE1C     11  SESUM   END
 10:04:02     2:10   147.6      0.0     21.3       0.0     SESUM       22  DELETE  BEGN
 10:04:02     2:10   147.6      0.0#    21.3       0.0#    SESUM       22  DELETE  END
 10:04:02     2:10   147.9      0.3     21.3       0.0     SESUM       25  XPURGE  BEGN
 10:04:02     2:10   147.9      0.0#    21.3       0.0#    SESUM       25  XPURGE  END
 10:04:02     2:10   148.1      0.2     21.4       0.0     PHASE1C     12  DBSTATUSBEGN
 10:04:02     2:10   148.1      0.0#    21.4       0.0#    PHASE1C     12  DBSTATUSEND
 10:04:02     2:10   148.1      0.0     21.4       0.0     PHASE1C     18  DBSTATUSBEGN
 10:04:02     2:10   148.1      0.0#    21.4       0.0#    PHASE1C     18  DBSTATUSEND
 10:04:02     2:10   148.1      0.0     21.4       0.0     PHASE1C     21  SELRRS  BEGN
 10:04:02     2:10   148.1      0.1#    21.4       0.0#    PHASE1C     21  SELRRS  END
 10:04:02     2:10   148.1      0.0     21.4       0.0     SELRRS       5  XEQUIV  BEGN
 10:04:02     2:10   148.1      0.0#    21.4       0.0#    SELRRS       5  XEQUIV  END
 10:04:02     2:10   148.1      0.0     21.4       0.0     SELRRS       6  XPURGE  BEGN
 10:04:02     2:10   148.1      0.0#    21.4       0.0#    SELRRS       6  XPURGE  END
 10:04:02     2:10   148.1      0.0     21.5       0.0     PHASE1C     22  SESUM   BEGN
 10:04:02     2:10   148.1      0.0#    21.5       0.0#    PHASE1C     22  SESUM   END
 10:04:02     2:10   148.1      0.0     21.5       0.1     SESUM       22  DELETE  BEGN
 10:04:02     2:10   148.1      0.0#    21.5       0.0#    SESUM       22  DELETE  END
 10:04:03     2:11   148.4      0.3     21.5       0.0     SESUM       25  XPURGE  BEGN
 10:04:03     2:11   148.4      0.0#    21.5       0.0#    SESUM       25  XPURGE  END
 10:04:03     2:11   148.6      0.2     21.6       0.0     SESTATIC    72  DBSTATUSBEGN
 10:04:03     2:11   148.6      0.0#    21.6       0.0#    SESTATIC    72  DBSTATUSEND
 10:04:03     2:11   148.6      0.0     21.6       0.0     SESTATIC    73  STATRS  BEGN
 10:04:03     2:11   148.6      0.1#    21.6       0.0#    SESTATIC    73  STATRS  END
 10:04:03     2:11   148.6      0.0     21.6       0.0     STATRS       6  XPURGE  BEGN
 10:04:03     2:11   148.6      0.0#    21.6       0.0#    STATRS       6  XPURGE  END
 10:04:03     2:11   148.6      0.0     21.6       0.0     STATRS       7  XEQUIV  BEGN
 10:04:03     2:11   148.6      0.0#    21.6       0.0#    STATRS       7  XEQUIV  END
 10:04:03     2:11   148.6      0.0     21.6       0.0     STATRS      27  SSG3    BEGN
                                                        REAL CORE IS          6074952
                                                        REAL CORE IS          6074952
                                                        REAL CORE IS          6074952
                                                        REAL CORE IS          6074952

0*** DIAG 19 --FBS-- FACTOR TRAILER=(     6     6 13 2        2     2222), METHOD S2-PASSES=   1, CPU=     0.0, I/O=     0

                    R.H.S. TRAILER=(     1     6  2 2        6     5000)

*** USER INFORMATION MESSAGE 4153. FBS METHOD S2 TIME ESTIMATE TO FORM UL      - CPU=     0, I/O=     0, TOTAL=     0, PASSES=   1
 10:04:03     2:11   148.6      0.0     21.7       0.0     FBS BEGN
 10:04:03     2:11   148.6      0.0#    21.7       0.0#    FBS END
 *8** MODULE  DMAP MATRIX     COLS   ROWS  F  T  NZWDS   DENS BLOCKS STRL  NBRSTR BNDL  NBRBND ROW1 BNDAVG BNDMAX NULCOL
      SSG3    27   UL            1      6  2  2     10 0.8333      1    2       2    0       0    0      6      6      0 *8**
                                                        REAL CORE IS          6074952
  * M   MATRIX KLL      TRAILER=(     6      6  6 2      4   2777), METHOD 1 -PASSES=   1, CPU=     0.0, I/O=   0.3, TOTAL=     0.3
  * P   MATRIX UL       TRAILER=(     1      6  2 2     10   8333), METHOD 2 -PASSES=   1, CPU=999999.0, I/O=   0.2, TOTAL=999999.0
  * Y   MATRIX PLI      TRAILER=(     1      6  2 2      6   5000), METHOD 3 -PASSES=   1, CPU=999999.0, I/O=   0.3, TOTAL=999999.0
      (18)
  * A   WORKING MEMORY=     6074952      SYSTEM( 66)=          0      METHOD 4 -PASSES=   1, CPU=999999.0, I/O=   0.8,
TOTAL=999999.0
  * D   TRANSPOSE FLAG=          0      SYSTEM(126)=         57      METHOD S -PASSES=   1, CPU=     0.0, I/O=   0.2, TOTAL=     0.2
                                                    METHOD S NT, NBR PASSES=   1, EST. CPU=     0.0, I/O=     0.2, TOTAL=     0.2
 10:04:03     2:11   148.6      0.0     21.7       0.0     MPYAD BGN P=1
 10:04:03     2:11   148.6      0.0#    21.7       0.0#    MPYAD END
```

Listing 7-1 Example Execution Summary File (.f04) (continued)

```
*8** MODULE  DMAP MATRIX     COLS    ROWS  F  T  NZWDS   DENS BLOCKS STRL  NBRSTR BNDL  NBRBND ROW1 BNDAVG BNDMAX NULCOL
     SSG3    27   RUL          1       6  2  2      6 0.5000      1    3       1    0       0    0      3      3      0 *8**
10:04:03     2:11    148.6      0.0#    21.7       0.1#   STATRS       27  SSG3    END
10:04:03     2:11    148.6      0.0     21.8       0.0    SESTATIC     74  DELETE  BEGN
10:04:03     2:11    148.6      0.0#    21.8       0.0#   SESTATIC     74  DELETE  END
10:04:03     2:11    148.6      0.0     21.8       0.0    SESTATIC     75  DBSTATUSBEGN
10:04:03     2:11    148.6      0.0#    21.8       0.0#   SESTATIC     75  DBSTATUSEND
10:04:03     2:11    148.6      0.0     21.8       0.0    SESTATIC     94  SESUM   BEGN
10:04:03     2:11    148.6      0.0#    21.8       0.0#   SESTATIC     94  SESUM   END
10:04:04     2:12    148.6      0.0     21.9       0.0    SESUM        22  DELETE  BEGN
10:04:04     2:12    148.6      0.0#    21.9       0.0#   SESUM        22  DELETE  END
10:04:04     2:12    148.9      0.3     21.9       0.0    SESUM        25  XPURGE  BEGN
10:04:04     2:12    148.9      0.0#    21.9       0.0#   SESUM        25  XPURGE  END
10:04:04     2:12    149.1      0.2     21.9       0.0    SESTATIC     98  XEQUIV  BEGN
10:04:04     2:12    149.1      0.0#    21.9       0.0#   SESTATIC     98  XEQUIV  END
10:04:04     2:12    149.1      0.0     21.9       0.0    SESTATIC    103  SUPER3  BEGN
10:04:04     2:12    149.3      0.2#    22.0       0.1#   SESTATIC    103  SUPER3  END
10:04:04     2:12    149.3      0.0     22.0       0.0    SUPER3       26  XEQUIV  BEGN
10:04:04     2:12    149.3      0.0#    22.0       0.0#   SUPER3       26  XEQUIV  END
10:04:04     2:12    149.3      0.0     22.0       0.0    SUPER3       28  XEQUIV  BEGN
10:04:04     2:12    149.3      0.0#    22.0       0.0#   SUPER3       28  XEQUIV  END
10:04:05     2:13    149.3      0.0     22.2       0.2    SUPER3       61  XEQUIV  BEGN
10:04:05     2:13    149.3      0.0#    22.2       0.0#   SUPER3       61  XEQUIV  END
10:04:05     2:13    149.3      0.1     22.2       0.0    SUPER3       67  SEP4    BEGN
10:04:05     2:13    149.3      0.0#    22.2       0.0#   SUPER3       67  SEP4    END
10:04:05     2:13    149.3      0.0     22.3       0.1    SUPER3       92  SEDRDR  BEGN
10:04:05     2:13    149.3      0.0#    22.3       0.0#   SUPER3       92  SEDRDR  END
10:04:05     2:13    149.3      0.0     22.3       0.0    SUPER3       99  PVT     BEGN
10:04:05     2:13    149.3      0.0#    22.4       0.1#   SUPER3       99  PVT     END
10:04:05     2:13    149.4      0.1     22.4       0.0    SUPER3      106  SEDR    BEGN
10:04:05     2:13    149.4      0.0#    22.5       0.1#   SUPER3      106  SEDR    END
10:04:05     2:13    149.4      0.0     22.5       0.0    SUPER3      112  XEQUIV  BEGN
10:04:05     2:13    149.4      0.0#    22.5       0.0#   SUPER3      112  XEQUIV  END
10:04:05     2:13    149.4      0.0     22.5       0.0    SUPER3      117  SETQ    BEGN
10:04:05     2:13    149.4      0.0#    22.5       0.0#   SUPER3      117  SETQ    END
10:04:05     2:13    149.4      0.0     22.6       0.0    SETQ         10  XEQUIV  BEGN
10:04:05     2:13    149.4      0.0#    22.6       0.0#   SETQ         10  XEQUIV  END
10:04:05     2:13    149.4      0.0     22.7       0.2    SUPER3      135  DBSTATUSBEGN
10:04:05     2:13    149.4      0.0#    22.7       0.0#   SUPER3      135  DBSTATUSEND
10:04:05     2:13    149.4      0.0     22.7       0.0    SUPER3      140  LCGEN   BEGN
10:04:05     2:13    149.4      0.0#    22.7       0.0#   SUPER3      140  LCGEN   END
10:04:05     2:13    149.4      0.1     22.8       0.1    SUPER3      160  XEQUIV  BEGN
10:04:05     2:13    149.4      0.0#    22.8       0.0#   SUPER3      160  XEQUIV  END
10:04:05     2:13    149.4      0.0     22.8       0.0    SUPER3      168  SEDISP  BEGN
10:04:05     2:13    149.6      0.2#    22.9       0.1#   SUPER3      168  SEDISP  END
10:04:05     2:13    149.6      0.0     22.9       0.0    SEDISP       46  XEQUIV  BEGN
10:04:05     2:13    149.6      0.0#    22.9       0.0#   SEDISP       46  XEQUIV  END
10:04:05     2:13    149.6      0.0     22.9       0.0    SEDISP       57  XEQUIV  BEGN
10:04:05     2:13    149.6      0.0#    23.0       0.0#   SEDISP       57  XEQUIV  END
10:04:06     2:14    149.6      0.0     23.0       0.0    SEDISP       65  BCDR    BEGN
10:04:06     2:14    149.6      0.0#    23.0       0.0#   SEDISP       65  BCDR    END
10:04:06     2:14    149.6      0.0     23.0       0.0    SEDISP       68  XEQUIV  BEGN
10:04:06     2:14    149.6      0.0#    23.0       0.0#   SEDISP       68  XEQUIV  END
10:04:06     2:14    149.7      0.1     23.0       0.0    SEDISP       79  SEGOA   BEGN
10:04:06     2:14    149.8      0.1#    23.1       0.0#   SEDISP       79  SEGOA   END
10:04:06     2:14    149.8      0.0     23.1       0.0    SEDISP      105  XEQUIV  BEGN
10:04:06     2:14    149.8      0.0#    23.1       0.0#   SEDISP      105  XEQUIV  END
10:04:06     2:14    149.8      0.0     23.1       0.0    SEDISP      106  XEQUIV  BEGN
10:04:06     2:14    149.8      0.0#    23.1       0.0#   SEDISP      106  XEQUIV  END
10:04:06     2:14    149.8      0.0     23.1       0.0    SEDISP      107  XEQUIV  BEGN
10:04:06     2:14    149.8      0.0#    23.1       0.0#   SEDISP      107  XEQUIV  END
10:04:06     2:14    149.8      0.0     23.1       0.0    SEDISP      108  XEQUIV  BEGN
10:04:06     2:14    149.8      0.0#    23.1       0.0#   SEDISP      108  XEQUIV  END
10:04:06     2:14    149.8      0.0     23.1       0.0    SEDISP      110  SDR1    BEGN
*8** MODULE  DMAP MATRIX     COLS    ROWS  F  T  NZWDS   DENS BLOCKS STRL  NBRSTR BNDL  NBRBND ROW1 BNDAVG BNDMAX NULCOL
     SDR1    110  SCRATCH      1      12  2  1      6 0.5000      1    6       1    0       0    0      6      6      0 *8**
     SDR1    110  SCRATCH      1      12  2  2     10 0.4166      1    2       2    0       0    0      6      6      0 *8**
     SDR1    110  UGI          1      12  2  2     10 0.4166      1    2       2    0       0    0      6      6      0 *8**
10:04:06     2:14    149.8      0.0#    23.2       0.0#   SEDISP      110  SDR1    END
10:04:06     2:14    149.8      0.0     23.2       0.0    SEDISP      135  COPY    BEGN
10:04:06     2:14    149.8      0.0#    23.2       0.0#   SEDISP      135  COPY    END
10:04:06     2:14    149.8      0.0     23.2       0.0    SEDISP      139  COPY    BEGN
10:04:06     2:14    149.8      0.0#    23.2       0.0#   SEDISP      139  COPY    END
10:04:06     2:14    149.8      0.0     23.2       0.0    SEDISP      141  COPY    BEGN
10:04:06     2:14    149.8      0.0#    23.2       0.0#   SEDISP      141  COPY    END
10:04:06     2:14    149.8      0.0     23.2       0.0    SEDISP      213  SESUM   BEGN
10:04:06     2:14    149.8      0.0#    23.2       0.0#   SEDISP      213  SESUM   END
10:04:06     2:14    149.8      0.0     23.3       0.0    SESUM        22  DELETE  BEGN
10:04:06     2:14    149.8      0.0#    23.3       0.0#   SESUM        22  DELETE  END
10:04:06     2:14    150.1      0.3     23.3       0.0    SESUM        25  XPURGE  BEGN
10:04:06     2:14    150.1      0.0#    23.3       0.0#   SESUM        25  XPURGE  END
10:04:06     2:14    150.2      0.1     23.3       0.0    SUPER3      172  XEQUIV  BEGN
10:04:06     2:14    150.2      0.0#    23.3       0.0#   SUPER3      172  XEQUIV  END
10:04:06     2:14    150.3      0.1     23.4       0.0    SUPER3      181  XEQUIV  BEGN
10:04:06     2:14    150.3      0.0#    23.4       0.0#   SUPER3      181  XEQUIV  END
10:04:06     2:14    150.3      0.0     23.4       0.0    SUPER3      190  XEQUIV  BEGN
10:04:06     2:14    150.3      0.0#    23.4       0.0#   SUPER3      190  XEQUIV  END
```

Listing 7-1 Example Execution Summary File (.f04) (continued)

```
 10:04:07     2:15   150.3      0.0     23.4      0.0    SUPER3      203 SEDRCVR BEGN
 10:04:07     2:15   150.4      0.2#    23.5      0.1#   SUPER3      203 SEDRCVR END
 10:04:07     2:15   150.4      0.0     23.5      0.0    SEDRCVR     31  XEQUIV  BEGN
 10:04:07     2:15   150.4      0.0#    23.5      0.0#   SEDRCVR     31  XEQUIV  END
 10:04:07     2:15   150.4      0.0     23.5      0.0    SEDRCVR     32  XEQUIV  BEGN
 10:04:07     2:15   150.4      0.0#    23.5      0.0#   SEDRCVR     32  XEQUIV  END
 10:04:07     2:15   150.4      0.0     23.5      0.0    SEDRCVR     36  VECPLOT BEGN
 10:04:07     2:15   150.4      0.0#    23.6      0.0#   SEDRCVR     36  VECPLOT END
 10:04:07     2:15   150.4      0.0     23.6      0.0    SEDRCVR     37  VECPLOT BEGN
 10:04:07     2:15   150.4      0.0#    23.6      0.0#   SEDRCVR     37  VECPLOT END
 10:04:07     2:15   150.4      0.0     23.6      0.0    SEDRCVR     38  VECPLOT BEGN
 *8** MODULE  DMAP MATRIX     COLS    ROWS  F  T  NZWDS    DENS BLOCKS STRL  NBRSTR BNDL  NBRBND ROW1 BNDAVG BNDMAX NULCOL
      VECPLOT 38   UGVB          1      12  2  1       5 0.4166      1    2       2    0       0    0       6       6       0 *8**
 10:04:07     2:15   150.4      0.0#    23.6      0.0#   SEDRCVR     38  VECPLOT END
 10:04:07     2:15   150.4      0.0     23.6      0.0    SEDRCVR     39  VECPLOT BEGN
 *8** MODULE  DMAP MATRIX     COLS    ROWS  F  T  NZWDS    DENS BLOCKS STRL  NBRSTR BNDL  NBRBND ROW1 BNDAVG BNDMAX NULCOL
      VECPLOT 39   PJB           1      12  2  1       3 0.2500      1    3       1    0       0    0       3       3       0 *8**
 10:04:07     2:15   150.4      0.0#    23.6      0.0#   SEDRCVR     39  VECPLOT END
 10:04:07     2:15   150.4      0.0     23.7      0.0    SEDRCVR     48  SDR2    BEGN
 10:04:08     2:16   150.4      0.0#    23.8      0.1#   SEDRCVR     48  SDR2    END
 10:04:08     2:16   150.5      0.1     23.8      0.0    SEDRCVR     49  SDR2    BEGN
 10:04:08     2:16   150.5      0.0#    23.8      0.0#   SEDRCVR     49  SDR2    END
 10:04:08     2:16   150.5      0.0     23.8      0.0    SEDRCVR     58  XEQUIV  BEGN
 10:04:08     2:16   150.5      0.0#    23.8      0.0#   SEDRCVR     58  XEQUIV  END
 10:04:08     2:16   150.5      0.0     23.8      0.0    SEDRCVR     59  XEQUIV  BEGN
 10:04:08     2:16   150.5      0.0#    23.8      0.0#   SEDRCVR     59  XEQUIV  END
 10:04:08     2:16   150.5      0.0     23.8      0.0    SEDRCVR     60  XEQUIV  BEGN
 10:04:08     2:16   150.5      0.0#    23.8      0.0#   SEDRCVR     60  XEQUIV  END
 10:04:08     2:16   150.5      0.0     23.8      0.0    SEDRCVR     61  XEQUIV  BEGN
 10:04:08     2:16   150.5      0.0#    23.8      0.0#   SEDRCVR     61  XEQUIV  END
 10:04:08     2:16   150.6      0.1     23.9      0.1    SEDRCVR     125 SDRX    BEGN
 10:04:09     2:17   150.6      0.0#    23.9      0.0#   SEDRCVR     125 SDRX    END
 10:04:09     2:17   150.6      0.0     23.9      0.0    SEDRCVR     129 XEQUIV  BEGN
 10:04:09     2:17   150.6      0.0#    23.9      0.0#   SEDRCVR     129 XEQUIV  END
 10:04:09     2:17   150.6      0.0     23.9      0.0    SEDRCVR     130 XEQUIV  BEGN
 10:04:09     2:17   150.6      0.0#    23.9      0.0#   SEDRCVR     130 XEQUIV  END
 10:04:09     2:17   150.6      0.0     24.0      0.0    SEDRCVR     132 XEQUIV  BEGN
 10:04:09     2:17   150.6      0.0#    24.0      0.0#   SEDRCVR     132 XEQUIV  END
 10:04:09     2:17   150.6      0.0     24.0      0.0    SEDRCVR     133 XEQUIV  BEGN
 10:04:09     2:17   150.6      0.0#    24.0      0.0#   SEDRCVR     133 XEQUIV  END
 10:04:09     2:17   150.7      0.1     24.0      0.0    SEDRCVR     205 OFP     BEGN
 10:04:10     2:18   150.7      0.0#    24.1      0.1#   SEDRCVR     205 OFP     END
 10:04:10     2:18   150.7      0.0     24.1      0.0    SEDRCVR     206 OFP     BEGN
 10:04:10     2:18   150.7      0.0#    24.1      0.0#   SEDRCVR     206 OFP     END
 10:04:10     2:18   150.7      0.0     24.1      0.0    SEDRCVR     208 OFP     BEGN
 10:04:10     2:18   150.7      0.0#    24.1      0.0#   SEDRCVR     208 OFP     END
 10:04:11     2:19   150.7      0.0     24.1      0.0    SEDRCVR     211 SDR2    BEGN
 10:04:11     2:19   150.7      0.0#    24.2      0.0#   SEDRCVR     211 SDR2    END
 10:04:11     2:19   150.7      0.0     24.2      0.0    SEDRCVR     212 SDRCOMP BEGN
 10:04:11     2:19   150.7      0.0#    24.2      0.0#   SEDRCVR     212 SDRCOMP END
 10:04:11     2:19   150.7      0.0     24.2      0.0    SEDRCVR     213 STRSORT BEGN
 10:04:11     2:19   150.7      0.0#    24.2      0.0#   SEDRCVR     213 STRSORT END
 10:04:13     2:21   150.8      0.1     24.2      0.0    SEDRCVR     214 STRSORT BEGN
 10:04:13     2:21   150.8      0.0#    24.2      0.0#   SEDRCVR     214 STRSORT END
 10:04:13     2:21   150.8      0.0     24.3      0.0    SEDRCVR     215 OFP     BEGN
 10:04:13     2:21   150.8      0.0#    24.3      0.0#   SEDRCVR     215 OFP     END
 10:04:13     2:21   150.8      0.0     24.3      0.0    SEDRCVR     222 XEQUIV  BEGN
 10:04:14     2:22   150.8      0.0#    24.3      0.0#   SEDRCVR     222 XEQUIV  END
 10:04:14     2:22   150.8      0.0     24.3      0.0    SEDRCVR     224 OFP     BEGN
 10:04:14     2:22   150.8      0.0#    24.3      0.0#   SEDRCVR     224 OFP     END
 10:04:15     2:23   150.9      0.1     24.5      0.2    SESTATIC    132 PRTSUM  BEGN
 10:04:15     2:23   150.9      0.0#    24.5      0.0#   SESTATIC    132 PRTSUM  END
 10:04:15     2:23   150.9      0.0     24.5      0.0    PRTSUM      4   PROJVER BEGN
 10:04:15     2:23   150.9      0.0#    24.5      0.0#   PRTSUM      4   PROJVER END
 10:04:15     2:23   150.9      0.0     24.5      0.0    PRTSUM      5   DBDICT  BEGN
 10:04:15     2:23   150.9      0.0#    24.6      0.1#   PRTSUM      5   DBDICT  END
 10:04:15     2:23   150.9      0.0     24.7      0.0    PRTSUM      6   PRTPARM BEGN
 10:04:15     2:23   150.9      0.0#    24.7      0.0#   PRTSUM      6   PRTPARM END
 10:04:15     2:23   150.9      0.0     24.7      0.0    SESTATIC    133 EXIT    BEGN
```

Listing 7-1 Example Execution Summary File (.f04) (continued)

```
1*** USER INFORMATION MESSAGE 1009 (F04SUM)
     STATISTICAL SUMMARY OF ABOVE LOG FILE FOLLOWS.


    STATISTICS FOR INDIVIDUAL EVENTS

 LINE   DMAP   EVNT   METH    EST CPU    ACT CPU    DEL CPU      % DIF   <25%  <50%  >50%                    % TOT
   65   35     XSOR                          1.                                                               5.
   67   61     IFP                           0.                                                               1.
  120   154    PLTS                          0.                                                               0.
  124   176    SEQP                          0.                                                               1.
  126   182    GP1                           0.                                                               1.
  132   25     GP1                           0.                                                               1.
  134   28     SEP1                          0.                                                               0.
  140   46     TABP                          0.                                                               0.
  142   47     SEP3                          0.                                                               0.
  172   307    GP1                           0.                                                               1.
  180   6      GP2                           0.                                                               0.
  194   335    GP3                           0.                                                               0.
  196   336    LCGE                          0.                                                               0.
  202   367    GP4                           0.                                                               0.
  208   378    UMER                          0.                                                               0.
  218   45     MATG                          0.                                                               0.
  237   24     TA1                           0.                                                               0.
  241   16     ELTP                          0.                                                               0.
  243   22     EMG                           0.                                                               0.
  249   79     EMA                           0.                                                               0.
  278   11     SSG1                          0.                                                               0.
  287   62     VECP                          0.                                                               0.
  319   29     GPSP                          0.                                                               0.
  327   47     UPAR                          0.                                                               0.
  371   41     UPAR                          0.                                                               0.
  377   64     SSG2                          0.                                                               0.
  410   18     DCMP                          0.                                                               0.
  459   27     FBS    S 2          0.         0.         0.         0.      1     0     0                     0.
  459   27     MPYA   S NT         0.         0.         0.         0.      1     0     0                     0.
  459   27     SSG3                          0.                                                               0.
  498   67     SEP4                          0.                                                               0.
  504   106    SEDR                          0.                                                               0.
  514   140    LCGE                          0.                                                               0.
  538   110    SDR1                          0.                                                               0.
  568   36     VECP                          0.                                                               0.
  570   37     VECP                          0.                                                               0.
  572   38     VECP                          0.                                                               0.
  576   39     VECP                          0.                                                               0.
  580   48     SDR2                          0.                                                               0.
  582   49     SDR2                          0.                                                               0.
  592   125    SDRX                          0.                                                               0.
  602   205    OFP                           0.                                                               0.
  604   206    OFP                           0.                                                               0.
  606   208    OFP                           0.                                                               0.
  608   211    SDR2                          0.                                                               0.
  610   212    SDRC                          0.                                                               0.
  612   213    STRS                          0.                                                               0.
  614   214    STRS                          0.                                                               0.
  616   215    OFP                           0.                                                               0.
  620   224    OFP                           0.                                                               0.

  STATISTICS FOR EVENTS BY METHOD

              COUNT           EST CPU    ACT CPU    DEL CPU              <25%  <50%  >50%     A.M. %     S.D. %      % TOT

MPYAD S NT       1               0.         0.         0.                 1     0     0         0.         0.         0.
 MPYAD N TOTALS  1               0.         0.         0.                                                           0.
  FBS S 2        1               0.         0.         0.                 1     0     0         0.         0.         0.
 FBS     TOTALS  1               0.         0.         0.                                                           0.

METHOD SUMMARY:
TOTAL OF EVENTS RECORDED  =         0.
TOTAL CPU                 =        25.
PERCENT RECORDED          =         0.
```

Listing 7-1 Example Execution Summary File (.f04) (continued)

```
 STATISTICS FOR EVENTS BY MODULE

             COUNT                       ACT CPU                                                                  % TOT
ELTPRT           1                            0.                                                                     0.
EMA              1                            0.                                                                     0.
EMG              1                            0.                                                                     0.
GPSP             1                            0.                                                                     0.
GP1              3                            1.                                                                     2.
GP2              1                            0.                                                                     0.
GP3              1                            0.                                                                     0.
GP4              1                            0.                                                                     0.
IFP              1                            0.                                                                     1.
LCGEN            2                            0.                                                                     0.
MATGEN           1                            0.                                                                     0.
OFP              5                            0.                                                                     0.
PLTSET           1                            0.                                                                     0.
SDRCOMP          1                            0.                                                                     0.
SDRX             1                            0.                                                                     0.
SDR1             1                            0.                                                                     0.
SDR2             3                            0.                                                                     1.
SEDR             1                            0.                                                                     0.
SEP1             1                            0.                                                                     0.
SEP3             1                            0.                                                                     0.
SEP4             1                            0.                                                                     0.
SEQP             1                            0.                                                                     1.
SSG1             1                            0.                                                                     0.
SSG2             1                            0.                                                                     0.
SSG3             1                            0.                                                                     0.
STRSORT          2                            0.                                                                     0.
TABPRT           1                            0.                                                                     0.
TA1              1                            0.                                                                     0.
UMERGE1          1                            0.                                                                     0.
UPARTN           2                            0.                                                                     0.
VECPLOT          5                            0.                                                                     0.
XSORT            1                            1.                                                                     5.
DCMP             1                            0.                                                                     0.

MODULE SUMMARY:
TOTAL OF EVENTS RECORDED  =         4.
TOTAL CPU                 =        25.
PERCENT RECORDED          =        15.


*** TOTAL MEMORY USAGE STATISTICS ***

+------SPARSE DECOMPOSITION  MODULES ---------+

  HIWATER              SUB_DMAP         DMAP
   WORDS    DAY_TIME      NAME          MODULE
   314448    10:04:01    SEKRRS     18  DCMP


*** DATABASE USAGE STATISTICS ***

+------------------------- LOGICAL FILES -------------------------+  +----------------- PHYSICAL FILES ----------------+
  DBSET          BLOCKS      BLOCKS       %      BLOCKSIZE  BLOCKS PER  PHYS FILE     BLOCKS    HIWATER     FILE SIZE     MAX %
                ALLOCATED      USED      USED     (WORDS)     CLUSTER               ALLOCATED    BLOCK       (WORDS)       USED
  MASTER          5000         185      3.70       1408           1    MASTER          5000       185        260480       3.70
  USROBJ          5000           7      0.14       1408           1    USROBJ          5000         7          9856       0.14
  USRSOU          5000           7      0.14       1408           1    USRSOU          5000         7          9856       0.14
  DBALL          25000           7      0.03       1408           1    DBALL          25000         7          9856       0.03

  OBJSCR          5000         124      2.48       1408           1    OBJSCR          5000       124        174592       2.48
  SCRATCH       350100          23      0.01       1408           1    MEMFILE          100        58         81664      58.00
                                                                       SCRATCH       175000         1          1408       0.00
                                                                       SCR300        175000         1          1408       0.00

*** BUFFER POOL AND SCRATCH 300 USAGE STATISTICS ***

+----------------- BUFFER POOL -----------------+      +--------------------------- SCRATCH 300 ---------------------------+
  OPTION        BLOCKS       BLOCKS       BLOCKS          OPTION    HIWATER                SUB_DMAP         DMAP      OPN/CLS
 SELECTED     ALLOCATED     REUSED      RELEASED         SELECTED    BLOCK     DAY_TIME      NAME          MODULE    COUNTER
 GINO,EXEC          37         910            67             2          0     10:01:52    PREFACE     0  PREFACE        0
```

The Standard Output contains the MSC Nastran analysis results. This file also contains information about the version of MSC Nastran being run.

Listing 7-2 MSC Nastran Standard Output

```
     Warning: This computer program is protected by copyright law and international treaties.
   Unauthorized use, reproduction or distribution of this computer program, or any portion of it, may
                           result in severe civil and criminal penalties.
         Copyright (C) 2016 MSC Software Corporation and its licensors. All rights reserved.

                                * * * * * * * * * * * * * * * * * * * *
                                * * * * * * * * * * * * * * * * * * * *
                                * *                                 * *
                                * *                                 * *
                                * *           MSC Software          * *
                                * *               CORP              * *
                                * *                                 * *
                                * *      M S C  N A S T R A N       * *
                                * *                                 * *
                                * *     VERSION 2016.0.0-CLXXXXXX   * *
                                * *                                 * *
                                * *          MAR 30, 2016           * *
                                * *                                 * *
                                * *                                 * *
                                * *       x86-64 Based System       * *
                                * *         MODEL Intel x8664       * *
                                * *   Linux 2.6.32-279.el6.x86_64   * *
                                * *Compiled for x8664 (SINGLE Mode) * *
                                * *                                 * *
                                * *                                 * *
                                * * * * * * * * * * * * * * * * * * * *
                                * * * * * * * * * * * * * * * * * * * *
```

Listing 7-2 MSC Nastran Standard Output (continued)

```
                                                                                    MARCH  11, 1994  MSC/NASTRAN   3/ 3/94   PAGE     2

        N A S T R A N   E X E C U T I V E   C O N T R O L   D E C K   E C H O


      ID MSC, AM761B $ JJG 5MAR91
      DIAG 8,13,19,49 $
      TIME 2
      SOL 101 $ STRUCTURED STATICS
      CEND

     EXAMPLE: ONE ELEMENT PROBLEM                          AM761B            MARCH  11, 1994  MSC/NASTRAN   3/ 3/94   PAGE     3

                                  C A S E   C O N T R O L   D E C K   E C H O

             CARD
             COUNT
                1      TITLE = EXAMPLE: ONE ELEMENT PROBLEM                          AM761B
                2      LOAD = 11
                3      DISPLACEMENT = ALL
                4      ELFORCE = ALL
                5      BEGIN BULK
                           INPUT BULK DATA CARD COUNT =       7

     EXAMPLE: ONE ELEMENT PROBLEM                          AM761B            MARCH  11, 1994  MSC/NASTRAN   3/ 3/94   PAGE     4

                                       S O R T E D   B U L K   D A T A   E C H O

              CARD
              COUNT        .   1   .   2   ..  3   ..  4   ..  5   ..  6   ..  7   ..  8   ..  9   ..  10  .
                 1-        CBEAM   1       1       10      20      0.      1.      0.
                 2-        FORCE   11      20              100.    1.      .8      1.
                 3-        GRID    10              0.      0.      0.              123456
                 4-        GRID    20              10.     0.      0.
                 5-        MAT1    100     1.+7            .3
                 6-        PBEAM   1       100     1.      .08     .064            .1
                           ENDDATA
                    TOTAL COUNT=         7

     EXAMPLE: ONE ELEMENT PROBLEM                          AM761B            MARCH  11, 1994  MSC/NASTRAN   3/ 3/94   PAGE     5

                                   SEQUENCE PROCESSOR OUTPUT


     THERE ARE        2 POINTS DIVIDED INTO    1 GROUP(S).
                     CONNECTION DATA
     ELEMENT TYPE       NUMBER    ASSEMBLY TIME(SEC)

        BEAM                 1           0.02
     -------------------------------------------------

     TOTAL MATRIX ASSEMBLY TIME FOR         1 ELEMENTS IS     0.02 SECONDS.
     ORIGINAL PERFORMANCE DATA
     SUPER(GROUP) ID   NO. GRIDS   AV. CONNECTIVITY   C-AVERAGE   C-RMS   C-MAXIMUM   P-GROUPS   P-AVERAGE   DECOMP TIME(SEC)
                                                                                                    (6.0 DOF/GRID )
               0            2           2.00           1.50      1.58        2          0        0.00          0.000

     RESEQUENCED PERFORMANCE DATA
     SUPER(GROUP) ID   NO. GRIDS   AV. CONNECTIVITY   C-AVERAGE   C-RMS   C-MAXIMUM   P-GROUPS   P-AVERAGE   DECOMP TIME(SEC)  METHOD
                                                                                                    (6.0 DOF/GRID )
               0            2           2.00           1.50      1.58        2          0        0.00          0.000        ACTIVE

     EXAMPLE: ONE ELEMENT PROBLEM                          AM761B            MARCH  11, 1994  MSC/NASTRAN   3/ 3/94   PAGE     6

                                          OLOAD    RESULTANT
                    T1              T2              T3              R1              R2              R3
               1   1.0000000E+02  8.0000000E+01  1.0000000E+02  0.0000000E+00 -1.0000000E+03  8.0000000E+02
     *** USER INFORMATION MESSAGE 5293 FOR DATA BLOCK KLL
         LOAD SEQ. NO.            EPSILON              EXTERNAL WORK      EPSILONS LARGER THAN 0.001 ARE FLAGGED WITH ASTERISKS
                    1        -4.6914561E-17          3.9638200E+00
     EXAMPLE: ONE ELEMENT PROBLEM                          AM761B            MARCH  11, 1994  MSC/NASTRAN   3/ 3/94   PAGE     7


                                          MAXIMUM  DISPLACEMENTS
                    T1              T2              T3              R1              R2              R3
               1   9.9999997E-05  3.3541333E-02  5.2343331E-02  0.0000000E+00  7.8124995E-03  4.9999999E-03
     EXAMPLE: ONE ELEMENT PROBLEM                          AM761B            MARCH  11, 1994  MSC/NASTRAN   3/ 3/94   PAGE     8


                                          MAXIMUM  APPLIED LOADS
                    T1              T2              T3              R1              R2              R3
               1   1.0000000E+02  8.0000000E+01  1.0000000E+02  0.0000000E+00  0.0000000E+00  0.0000000E+00
```

Main Index

Listing 7-2 MSC Nastran Standard Output (continued)

```
    EXAMPLE: ONE ELEMENT PROBLEM                          AM761B           MARCH  11, 1994  MSC/NASTRAN   3/ 3/94   PAGE     9


                                          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
            10      G      0.0            0.0            0.0            0.0            0.0            0.0
            20      G      1.000000E-04   3.354133E-02   5.234333E-02   0.0           -7.812500E-03   5.000000E-03
    EXAMPLE: ONE ELEMENT PROBLEM                          AM761B           MARCH  11, 1994  MSC/NASTRAN   3/ 3/94   PAGE    10


                                F O R C E S   I N   B E A M   E L E M E N T S        ( C B E A M )
                    STAT DIST/   - BENDING MOMENTS -          - WEB  SHEARS -          AXIAL          TOTAL          WARPING
  ELEMENT-ID  GRID    LENGTH     PLANE 1       PLANE 2       PLANE 1       PLANE 2      FORCE          TORQUE         TORQUE
        1
               10   0.000    8.000000E+02  1.000000E+03   8.000000E+01  1.000000E+02   1.000000E+02   0.0            0.0
               20   1.000    0.0           0.0            8.000000E+01  1.000000E+02   1.000000E+02   0.0            0.0
    EXAMPLE: ONE ELEMENT PROBLEM                          AM761B           MARCH  11, 1994  MSC/NASTRAN   3/ 3/94   PAGE    11


    EXAMPLE: ONE ELEMENT PROBLEM                          AM761B           MARCH  11, 1994  MSC/NASTRAN   3/ 3/94   PAGE    12

* * * *  D B D I C T   P R I N T  * * * *      SUBDMAP = PRTSUM  ,  DMAP STATEMENT NO.      5

                                  * * * *  A N A L Y S I S   S U M M A R Y   T A B L E  * * * *
 SEID  PEID PROJ VERS APRCH      SEMG SEMR SEKR SELG SELR MODES DYNRED SOLLIN PVALID SOLNL LOOPID DESIGN CYCLE SENSITIVITY
 ----------------------------------------------------------------------------------------------------------------------------
    0     0    1    1 '        '    T    F    T    T    T     F      F      T      0     F     -1            0           F
 0SEID = SUPERELEMENT ID.
  PEID = PRIMARY SUPERELEMENT ID OF IMAGE SUPERELEMENT.
  PROJ = PROJECT ID NUMBER.
  VERS = VERSION ID.
  APRCH = BLANK FOR STRUCTURAL ANALYSIS.  HEAT FOR HEAT TRANSFER ANALYSIS.
  SEMG = STIFFNESS AND MASS MATRIX GENERATION STEP.
  SEMR = MASS MATRIX REDUCTION STEP (INCLUDES EIGENVALUE SOLUTION FOR MODES).
  SEKR = STIFFNESS MATRIX REDUCION STEP.
  SELG = LOAD MATRIX GENERATION STEP.
  SELR = LOAD MATRIX REDUCTION STEP.
  MODES = T (TRUE) IF NORMAL MODES OR BUCKLING MODES CALCULATED.
  DYNRED = T (TRUE) MEANS GENERALIZED DYNAMIC AND/OR COMPONENT MODE REDUCTION PERFORMED.
  SOLLIN = T (TRUE) IF LINEAR SOLUTION EXISTS IN DATABASE.
  PVALID = P-DISTRIBUTION ID OF P-VALUE FOR P-ELEMENTS
  LOOPID = THE LAST LOOPID VALUE USED IN THE NONLINEAR ANALYSIS.  USEFUL FOR RESTARTS.
  SOLNL = T (TRUE) IF NONLINEAR SOLUTION EXISTS IN DATABASE.
  DESIGN CYCLE = THE LAST DESIGN CYCLE (ONLY VALID IN OPTIMIZATION).
  SENSITIVITY = SENSITIVITY MATRIX GENERATION FLAG.
                                        * * * END OF JOB * * *
```

The Grid Point Singularity Table (GPST) is automatically output following the execution of the Grid Point Singularity Processor (GPSP) if singularities remain in the stiffness matrix at the grid point level. The table lists singular degrees of freedom in the global coordinate system. These singular degrees of freedom may be automatically constrained, at the user's option. Further information on this topic is given in Constraint and Mechanism Problem Identification in SubDMAP SEKR, 370.

Any of the matrices or tables that are prepared by the functional modules can be printed by using selected utility modules described in the *MSC Nastran DMAP Programmer's Guide*. These utility modules can be scheduled at any point in a solution sequence by using the alter feature. They should be scheduled immediately after the functional module that generates the table or matrix to be printed. However, the user is cautioned to check the calling sequence for the utility module, in order to be certain that all required inputs have been generated prior to this point.

The following descriptions relate to automatic output that is associated with particular types of solution sequences.

Real Eigenvalue Analysis

Each eigenvalue is identified with a mode number determined by sorting the eigenvalues by their magnitude. The following summary is also automatically printed for all methods:

- Mode number
- Extraction order (this is irrelevant under the Lanczos method)
- Eigenvalue
- Frequency (radians)
- Frequency (Hz)
- Generalized mass
- Generalized stiffness

For all methods except the Inverse Power Method (see below), if the orthogonality criterion is failed, then User Warning Message 3034 is issued:

The following summary is also automatically printed when using the Lanczos method:

- Block size used
- Number of decompositions
- Number of roots found
- Number of (block) solves required
- One of the following termination messages:
- Required number of eigenvalues found
- Not all the roots found within the range
- All eigenvalues found within the range
- Insufficient time to find more eigenvalues

The following summary is automatically printed when using the Inverse Power Method:

- Number of eigenvalues extracted
- Number of starting points used
- Number of starting point moves
- Number of triangular decompositions
- Number of vector iterations
- The reason for termination is indicated by one of the following numbers:

Reason Number	Description
1	Two consecutive singularities encountered while performing triangular decomposition
2	Four shift points while tracking a single root
3	All eigenvalues found in the frequency range specified
4	Three times the number of roots estimated in the frequency range have been extracted
5	All eigenvalues that exist in the problem have been found
6	The number of roots desired have been found
7	One or more eigenvalues have been found outside the frequency range specified
8	Insufficient time to find another root
9	Unable to converge

- Largest off-diagonal modal mass term identified by mode pair
- Number of off-diagonal mass terms failing orthogonality criterion

Complex Eigenvalue Analysis

Each complex eigenvalue is identified with a root number determined by sorting the complex eigenvalues by magnitude. The following summary of the complex eigenvalues extracted is automatically printed for all methods:

- Root number
- Extraction order (this is irrelevant under the Lanczos method)
- Real and imaginary parts of the eigenvalue
- The coefficients f (frequency) and g (damping coefficient) in the following representation of the eigenvalue ($p = \alpha + i\omega$):

  $p = 2\pi f(i - 1/2g)$

  Therefore,

  $f = \left|\frac{\omega}{2\pi}\right|$

  $g = \frac{-2\alpha}{|\omega|} = 2\frac{C}{C_{cr}}$

When using the complex Lanczos method, User Information Messages 5444, 5445, and 5453 are also issued. The following summary is also automatically printed when the Determinant Method is used:

- Number of eigenvalues extracted
- Number of passes through starting points.
- Number of criteria changes
- Number of starting point moves

- Number of triangular decompositions
- Number of failures to iterate to a root
- Number of predictions outside region
- The reason for termination is indicated by one of the following integers:

Reason Number	Description
1	The number of roots desired have been found
2	All predictions for eigenvalues are outside the regions specified
3	Insufficient time to find another root
4	Matrix is singular at the first three starting points

- Swept determinant functions for each starting point

The following summary is automatically printed for each region specified when the Inverse Power Method is used:

- Number of eigenvalues extracted
- Number of starting points used
- Number of starting point or shift point moves
- Number of triangular decompositions
- Number of vector iterations
- The reason for termination is indicated by one of the following integers:

Reason Number	Description
1	Two consecutive singularities encountered while performing triangular decomposition
2	Four starting point moves while tracking a single root
3	All eigenvalues found in the region specified
4	Three times the number of roots estimated in the region have been extracted
5	All eigenvalues that exist in the problem have been found
6	The number of roots desired have been found
7	One or more eigenvalues have been found outside the region specified
8	Insufficient time to find another root
9	Unable to converge

Superelement Analysis

In addition to the previously mentioned automatic output, the superelements solution sequences also print the following:

- Superelement mapping information that lists the membership of grid points and elements in the superelements, the order in which the superelements will be processed, and timing and space estimates for each superelement
- A page header that identifies the superelement for which the printed information on the page pertains
- Matrices wherein mechanisms are identified
- Resultants of external loads
- Maximum applied loads
- Maximum displacements
- Maximum SPC forces

For SOL 101, if PARAM,INREL,-1 or PARAM,INREL,-2, the following data is printed automatically:

INTERMEDIATE MATRIX...QRR

This is the rigid body mass matrix of the total structure, including superelements. It is presented as a 6×6 matrix, measured at the reference grid point, in its global coordinates. Masses on scalar points are ignored.

INTERMEDIATE MATRIX...QRL

This is the resultant of the applied loads, measured at the reference point. There is one column for each loading condition.

INTERMEDIATE MATRIX...URACCEL

This is the user-supplied rigid body acceleration, input on the DMIG,UACCEL entry. This output does not appear if the entry is not present.

INTERMEDIATE MATRIX...URA

This is the rigid body acceleration matrix, computed from the applied loads.

Restart Procedures

Scheduled exits can be requested at any point in a structured solution sequence format by means of the SEMG, SEKR, SELG, SELR, SEMR, and SEALL Case Control commands. If none of these commands are specified, then SEALL = ALL is the default.

Unscheduled exits are usually caused by errors on input entries or errors in the structural model resulting from missing or inconsistent input data. When such errors are detected, an unscheduled exit is performed accompanied with the output of the applicable user error messages. Following the correction of the input data errors, a modified restart can be performed.

Unscheduled exits may also occur because of machine failure or insufficient time allowance. In these cases, an unmodified restart is usually made. In some cases, where a portion of the problem has been completed, including the output for the completed portion, a modified restart must be made following an unscheduled exit due to insufficient time allowance. The situations are discussed under case control requirements in the sections dealing with the individual solution sequences.

The initial execution of any problem must be made with complete MSC Nastran input data, including all of the bulk data. However, all or part of the bulk data may be assembled from alternate input sources.

For restart, the Bulk Data Section consists only of delete "/" entries (see Bulk Data Section, 32) and new entries that the user wishes to add. The previous Bulk Data is read from the database. All other parts of the MSC Nastran Input Data (including the File Management statements, the Executive Control statements, the Case Control commands, the BEGIN BULK command and the ENDDATA entry) must be resubmitted even though no changes are made in the case control and no new bulk data is added. When changing solution sequences, the solution number (SOL) must be changed to the number or name of the new solution sequence.

Any changes in the Case Control Section associated with Bulk Data entry selection or subcase definition, or changes in the Bulk Data Section.

Restarts always start at the beginning of a specified DMAP or SOL sequence and queries the database in two phases.

Phase 1:	Conceptually this phase marks all appropriate existing database data blocks and parameters as existing for the current restart run.
Phase 2:	This phase then conceptually checks the current input against the version from which the restart is starting from and deletes from the Phase 1 determined database data blocks and parameters any data blocks that were modified because of input. This phase is performed by the RESTART module(s) contained within the solution's DMAP.

Restart only executes DMAP Modules for which some or all of the output does not exist on the current version database. The S-type parameter is considered as an output data block for restart purposes, hence if it is not listed in the NDDL, then the module that contains it will be re-executed. All DMAP modules are executed until the first RESTART module after which output checking is performed. Forced execution of modules after this point may be manually controlled by SYSTEM (109) flag.

If no such changes are made, the executive system performs an unmodified restart. If changes have been made only in the output requests, the restart is considered unmodified. However, some modules may have to be re-executed in order to prepare the output. Output requests that were satisfied, plots made, etc., should be removed if they are not to be output again.

For modified restarts, a number of previously executed DMAP instructions may have to be re-executed, depending on the nature of the modifications made by the user. The DMAP instructions that need to be executed in a modified restart are automatically determined within the program by comparing all changes made in Case Control commands and Bulk Data entries.

General Operations

This section describes the data processing and matrix operations that are performed by the functional modules in the solution sequences (SOLs 101 through 200) unless otherwise indicated. Detailed descriptions of the functional modules are given in Chapter 4 or the *MSC Nastran Programmers Manual*. Additional information regarding the use of DMAP modules for matrix operations is given in the *MSC Nastran DMAP Programmer's Guide*.

The following sections will refer to the structural analysis matrices such as stiffness, damping, and mass, but the discussion also applies to heat transfer analysis for conductance and capacitance. There is no analogy for mass in heat transfer analysis.

Geometry Processing in SubDMAP PHASE0

The total model is given its initial processing in PHASE0. Data is taken from the input files and stored in data structures that feed more expensive processes in later SubDMAPS. Flags are set to control branching for special features. All superelement inputs are processed. Tables are indexed for faster processing in post-processors. Parameters are read from Case Control and Bulk Data files by module PVT. The shell element normals at each grid point are computed in module TASNP2, and output in table GPSNT.

Tables for internal program use are generated by the geometry processors from the grid point entries, coordinate system definition entries, sequence entries, connection entries, static load entries, temperature definition entries, and constraint entries. The computer time for all geometry processing is usually only a small part of the time required for problem solution.

The basic geometry processing is performed by GP1 (Geometry Processor — Phase 1). A list of all grid points and scalar points is assembled in internal order for use in relating internal and external identification numbers. Matrices for transformation from basic to all defined global coordinate systems are computed, and all grid point locations are transformed to the basic coordinate system.

The element connection tables (ECT) are generated by GP2 (Geometry Processor — Phase 2) from the connection entries. The external grid point numbers are converted to the internal indices defined in GP1.

The temperature data and static loads are processed by GP3 (Geometry Processor — Phase 3). The temperature data is collected by sets and the external grid point numbers and element numbers are converted to the internal indices defined in GP1 and GP2. In the case of static analysis, the static loads are also collected by sets and the external identification numbers are converted to internal indices.

The constraint data is processed by GP4 (Geometry Processor — Phase 4). The displacement set definition table (USET) is generated from the constraint definition entries. USET contains one coded word for each degree of freedom in the model to identify the set or sets of coordinates to which the degree of freedom belongs (see Constraint and Set Notation, 325 for the definition of degree of freedom sets). The multipoint constraint matrix $[R_{mg}]$ is formed from the MPC and rigid element Bulk Data entries. In static analysis, the enforced displacement matrix $[Y_s]$ is formed for use in the generation of equivalent static loads.

The BNDSPC module moves SPCs on boundary grid points to downstream superelements.

Model Checkout

A checkout procedure is provided for the development of new models. It includes the bookkeeping and control blocks, and part of the generation block of Figure 7-3 in Superelement Analysis, 419. It is requested by PARAM,CHECKOUT,YES.

SEP1 Module

Generates the SEMAP table, which lists superelement membership of grid points and elements. Provides estimates for computation costs and storage requirements. This module will identify any structural partitioning errors.

There are two paths through the superelement capability. If the bulk data entries have a BEGIN SUPER and/or BEGIN SUPER = [SEID] entries the data flow goes through the partitioned superelement capability modules, SEP1X and SEP2X. If only a BEGIN BULK entry is present the grid point list capability modules SEP1 and SEP2 are used. The SEP1X path also allows grid point list superelements, with the exception of fluid structure analysis, SOL 200 (Design Optimization),

and the RELEASE entry. Models using these features must use the grid point list path. More details are given in Superelement Analysis, 419.

SEP2, SEP2CT Modules

Partitions out the data for any superelement selected by SE-type Case Control commands. These modules identify load and rigid element partitioning errors, and errors in the Case Control Section.

Constraint Modules

The GP1 and GP4 modules process all selected superelements. They check for the presence of requested coordinate systems; for illegal duplicate specifications of mutually exclusive sets, such as SPCi and ASETi entries referencing the same point; and for correct specifications on SPCi, xSETi, and MPCi entries, and rigid elements.

The user set table may be printed by use of PARAM,USETPRT, when PARAM,CHECKOUT,YES is present. If rigid elements or selected MPC entries exist, the constraint matrix $[R_{mg}]$ (see Multipoint Constraint Operations in SubDMAPs SEKR and SEMR2 and SEMRB, 366) is given three tests:

1. A matrix of rigid body vectors $[u^o_{gh}]$ is calculated by the VECPLOT module. This matrix includes zeroes for the rows corresponding to scalar points. The product

   $$[E_{mh}] \leftarrow [R_{mg}][u^o_{gh}]$$

   is calculated. Terms of $[E_{mh}]$ larger than PARAM,TINY are printed. If the constraint matrix contains internal constraints, nonzero terms will exist in $[E_{mh}]$. If an MPC equation includes scalar points, nonzero terms may exist in the corresponding row of $[E_{mh}]$ although the equation may be in equilibrium.

2. The product

   $$[R^g_{mm}] \leftarrow [R_{mg}][R_{mg}]^T$$

   is calculated. The product matrix is decomposed by the DCMP module. Diagonal terms of the factor matrix that are larger than PARAM,MAXRATIO are printed. These terms indicate that the printed row is linearly dependent on prior rows. Unless the constraint matrix is changed, it is likely that the model will encounter fatal errors due to singularity in the MCE1 module, or will have a poorly-conditioned constraint elimination process.

3. The product

   $$[R^m_{mm}] \leftarrow [R_{mm}][R_{mm}]^T$$

   is calculated, where $[R_{mm}]$ is the partition of $[R_{mg}]$. This product is also processed by the DCMP module. If a row appears here that also appears in the messages from the prior step, it confirms that the constraint matrix is inherently ill-conditioned. If a row appears here that did not appear in the prior step, it indicates that a change in the m-set exists that can also make the $[R_{mm}]$ matrix well conditioned.

The DBC module stores data for use in postprocessors. It runs at many points in the Solution Sequences. In this SubDMAP it stores basic structural geometric data. The AEROE SubDMAP generates the geometry of the aerodynamic model. The Automated Component Mode Synthesis (ACMS) capability controls branches at many places in the Solution Sequences.

The RESTART module controls the modules that will be run on restart, based on changes in the input files. The run exits after these checks are performed.

Matrix Assembly Operations in SubDMAP SEMG

The stiffness, mass, damping, differential stiffness, and heat conduction matrices are formed during the matrix assembly operations in subDMAP SEMG.

Stiffness is generated from the following sources:

- Structural elements defined on connection entries (e.g., CBAR and CROD).
- General elements defined on GENEL entries.
- Scalar springs defined on CELASi entries.

Mass is generated from the following sources:

- A 6×6 matrix of mass coefficients at a grid point defined on a CONM1 entry.
- A concentrated mass element defined on a CONM2 entry in terms of its mass and moments of inertia about its center of gravity.
- Structural mass for all elements, except plate elements without membrane stiffness, using the mass density on the material definition entry.
- Nonstructural mass for all elements specifying a value on the property entry.
- Scalar masses defined on CMASSi entries.

Damping is generated from the following sources:

- Viscous rod elements defined on CVISC entries.
- Scalar viscous dampers defined on CDAMPi entries.
- Element structural damping by multiplying the stiffness matrix of an individual structural element by a damping factor obtained from the material properties (MATi) entry for the element.

These matrices are formed from the connection and geometric tables prepared by the geometry processors and the material and element property tables prepared in the Phase 0. The matrix assembly operations will usually represent a significant portion of the total solution time. The computer time for matrix assembly operations is linear with the number and type of elements. Unit times for stiffness matrix assembly are given in the MSC Nastran 2020 Installation and Operations Guide. Assembly times for coupled mass matrices, differential stiffness matrices, and heat conduction matrices are about the same as those for stiffness matrices. Assembly times for lumped mass matrices and damping matrices are usually relatively small.

The matrix assembly process begins with the generation of various tables in TA1 (Table Assembler) from the geometric data, connection data and property data. These tables are used in data recovery operations as well as in the matrix assembly operations.

The matrix assembly operations are performed by EMG (Element Matrix Generator) and EMA (Element Matrix Assembler) modules. All of the element matrices (KELM, KDICT, MELM, MDICT, BELM, and BDICT) are generated

by EMG. The stiffness matrix exclusive of general elements, $[K^x_{gg}]$, as well as the structural damping matrix, $[K^4_{gg}]$, the mass matrix, $[M_{gg}]$, and the viscous damping matrix, $[B_{gg}]$, are assembled in separate executions of EMA.

The parameter COUPMASS is used to select either coupled mass or lumped mass generation in EMG. If the parameter WTMASS is present, the mass matrices generated in EMG are multiplied term by term by the value in field 3 of the PARAM entry. This operation allows the user to specify mass data in units of weight.

The structural damping matrix $[K^4_{gg}]$ is assembled from elements with structural damping. Structural damping is specified in the GE field of the MATi entry.

$$[K^4_{gg}] = \sum_{e=1}^{NE} \mathrm{GE}_e [K_e]$$

where:

NE	=	number of elements
GE_e	=	element structural damping coefficient from MATi entries
$[K_e]$	=	element stiffness matrix

The direct matrix inputs requested by the K2GG, M2GG, and B2GG Case Control commands defined the stiffness matrix $[K^2_{gg}]$, mass matrix $[M^2_{gg}]$, and damping matrix $[B^2_{gg}]$, and are assembled by the MTRXIN module. The outputs from EMA and MTRXIN are combined by the ADD module.

The stiffness contributions from the general elements (GENEL Bulk Data entry) are assembled in SMA3 (Structural Matrix Assembler -- Phase 3). The final stiffness matrix $[K_{gg}]$ is formed by combining $[K^x_{gg}]$ prepared by the EMA module with the contributions from general elements generated by SMA3.

Weight and balance information is calculated from the mass matrix $[M_{gg}]$ by GPWG (Grid Point Weight Generator). The execution of GPWG is controlled by the parameter GRDPNT. The details of the output are described in GRDPNT (p. 833) in the *MSC Nastran Quick Reference Guide*.

Multipoint Constraint Operations in SubDMAPs SEKR and SEMR2 and SEMRB

The multipoint constraint operations apply constraints to the stiffness matrices using the linear relationships among the displacements specified by the user on MPC, MPCAX, and rigid element Bulk Data entries.

The multipoint constraint equations are initially expressed in the form,

$$[R_g]\{u_g\} = 0 \ , \tag{7-1}$$

where the coefficients are supplied by the user. The user also specifies the degree of freedom that is made dependent by each equation of constraint, so that the $\{u_g\}$ matrix may immediately be partitioned into two subsets,

$$\{u_g\} = \begin{Bmatrix} u_n \\ u_m \end{Bmatrix} ,$$

where the set, u_m, is the set of dependent degrees of freedom. The matrix of constraint coefficients is similarly partitioned

$$[R_g] = [R_n \mid R_m] , \tag{7-2}$$

so that equation (7-1) becomes

$$[R_n]\{u_n\} + [R_m]\{u_m\} = 0 .$$

$[R_m]$ is a nonsingular matrix. We can, therefore, form the multipoint constraint matrix,

$$[G_m] = -[R_m]^{-1}[R_n] , \tag{7-3}$$

so that equation (7-2) may be stated as

$$\{u_m\} = [G_m]\{u_n\} . \tag{7-4}$$

Prior to the imposition of constraints, the structural problem may be written as

$$[K_{gg}]\{u_g\} = \{P_g\} , \tag{7-5}$$

or, partitioning in terms of the coordinate sets, u_n and u_m

$$\begin{bmatrix} \bar{K}_{nn} & K_{nm} \\ K_{nm}^T & K_{mm} \end{bmatrix} \begin{Bmatrix} u_n \\ u_m \end{Bmatrix} = \begin{Bmatrix} \bar{P}_n \\ P_m \end{Bmatrix}$$

Bars over symbols are used to designate arrays that are replaced in the reduction process.

The addition of constraints to the structure requires that the forces of constraint be added to the equilibrium equations. Thus, writing the equilibrium and constraint equations together in partitioned form,

$$\begin{bmatrix} \bar{K}_{nn} & K_{nm} & G_m^T \\ K_{nm}^T & K_{mm} & -I \\ G_m & -I & 0 \end{bmatrix} \begin{Bmatrix} u_n \\ u_m \\ q_m \end{Bmatrix} = \begin{Bmatrix} \bar{P}_n \\ P_m \\ 0 \end{Bmatrix} ,$$

where $\{q_m\}$ is the vector of constraint forces on $\{u_m\}$. Straightforward elimination of u_m and q_m gives

$$[\bar{K}_{nn} + K_{nm}G_m + G_m^T K_{mm}^T + G_m^T K_{mm} G_m]\{u_n\} = \{\bar{P}_n\} + [G_m^T]\{P_m\} ,$$

or

$$[K_{nn}]\{u_n\} = \{P_n\} ,$$

where

$$K_{nn} = \bar{K}_{nn} + K_{nm}G_m + G_m^T K_{nm}^T + G_m^T K_{mm} G_m \ , \tag{7-6}$$

and

$$P_n = \bar{P}_n + G_m^T P_m \ . \tag{7-7}$$

The initial partition of K_{gg} and the operations indicated by (7-3), (7-6) and (7-7) are performed by appropriate modules of the program. The multipoint constraint matrix, G_m, is used in structural matrix reduction ((7-6)), load vector reduction, ((7-7)) and data recovery ((7-4)).

The multipoint constraint equations, R_{mg}, formed in the GP4 module, are partitioned by the MCE1 module (Multipoint Constraint Eliminator -- Phase 1) as follows:

$$[R_{mg}] = \left[R_{mm} \mid R_{mn}\right]$$

MCE1 also solves the equation

$$[R_{mm}][G_{mn}] = -[R_{mn}]$$

for the transformation matrix $[G_{mn}]$. An alternate path is available for module MCE1 in SubDMAP XMCE1, requested by PARAM,MCE1,NEW. It contains a partitioned solution for $R_{mm}^{-1} \cdot R_{mn}$, where the part of R_{mm} that is diagonal is partitioned from the coupled part. It was developed at a time when the partitioned solution was faster than the single-pass solution used in MCE1. The single pass solution was improved so that it is now faster than the partitioned solution. The partitioned solution is maintained to provide an alternate solution when the single-pass solution is implausible, or does not complete in a normal manner.

The MCE2 module (Multipoint Constraint Eliminator — Phase 2) partitions the stiffness matrix

$$[K_{gg}] = \left[\begin{array}{c|c} \bar{K}_{nn} & K_{nm} \\ \hline K_{mn} & K_{mm} \end{array}\right] \ -$$

and performs matrix reduction

$$[K_{nn}] = [G_{mn}]^T[K_{mm}G_{mn} + K_{mn}] + [K_{mn}^T G_{mn} + \bar{K}_{nn}]$$

The other structural matrices, $[K_{gg}^4]$, $[B_{gg}]$ and $[M_{gg}]$, are transformed by formulas that are identical in form to equation (7-6), in subDMAPs SEMR2 and SEMRB.

The MCE2 module is also used in subDMAP SEMR2 to obtain $[M_{nn}]$.

The UPARTN module partitions $[K_{gg}]$ to $[K_{mg}]$ form which is equivalent to

$$[K_{mg}] = [K_{mn} \mid K_{mm}]$$

and is used to obtain multipoint forces of constraint. See Data Recovery Operations in SubDMAP SEDISP, 388.

Single Point Constraint Operations in SubDMAPs SEKR, SEMR2 and SEMRB

The single point constraint operations apply constraints to the structural matrices using the single point constraints specified by the user on SPCi Bulk Data entries, the PS field on GRID entries, or constrained by the AUTOSPC feature described in Constraint and Mechanism Problem Identification in SubDMAP SEKR, 370.

Single point constraints are applied to the set, u_s, in the form

$$\{u_s\} = \{Y_s\} ,$$

where $\{Y_s\}$ is a vector of enforced deformations, any or all of whose elements may be zero. The set, u_n, is partitioned into u_s and u_f (the free or unconstrained set)

$$\{u_n\} = \begin{Bmatrix} u_f \\ u_s \end{Bmatrix}$$

The stiffness matrix, K_{nn}, is similarly partitioned

$$[K_{nn}] = \begin{bmatrix} K_{ff} & K_{fs} \\ K_{fs}^T & K_{ss} \end{bmatrix}$$

The complete structural equations including the single point forces of constraint, q_s, may be written in partitioned matrix form as

$$\begin{bmatrix} K_{ff} & K_{fs} & 0 \\ K_{fs}^T & K_{ss} & -I \\ 0 & I & 0 \end{bmatrix} \begin{Bmatrix} u_f \\ u_s \\ q_s \end{Bmatrix} = \begin{Bmatrix} \bar{P}_f \\ P_s \\ Y_s \end{Bmatrix} \tag{7-8}$$

Straightforward elimination gives

$$[K_{ff}]\{u_f\} = \{\bar{P}_f\} - [K_{fs}]\{Y_s\} = \{P_f\} \tag{7-9}$$

The forces of constraint are recovered by means of the middle row of equation (7-8), i.e.,

$$\{q_s\} = -\{P_s\} + [K_{fs}^T]\{u_f\} + [K_{ss}]\{u_s\}$$

The distinct partitions K_{fs} and K_{ss}, are needed in subsequent calculations of SPC forces in static analysis. For the other structural matrices (K_{nn}^4, B_{nn}, and M_{nn}) the (sf) and (ss) partitions are saved, for dynamic SPC forces calculation.

The partially constrained stiffness matrix $[K_{nn}]$ is partitioned by UPARTN module as follows:

$$[K_{nn}] = \begin{bmatrix} K_{ff} & K_{fs} \\ K_{sf} & K_{ss} \end{bmatrix}$$

Similar partitioning operations are performed on $[M_{nn}]$ in subDMAP SEMR2 to obtain $[M_{ff}]$, $[M_{sf}]$, and M_{ss}. In subDMAP SEMRB, the structural and viscous damping matrices

$[K^4_{sf}]$, $[K^4_{ss}]$, $[B_{sf}]$ and $[B_{ss}]$

are obtained from the MATREDU module. All (sf) and (ss) matrices are required for the recovery of single point constraint reaction forces described in Data Recovery Operations in SubDMAP SEDISP, 388.

Constraint and Mechanism Problem Identification in SubDMAP SEKR

A singularity in a linear equation solution leads to a condition where a unique solution is not possible. For example, if $[K]\{u\} = \{F\}$, and $[K]$ is zero, then if $\{F\}$ is not zero, no value of $\{u\}$ will satisfy the equation. However, if $\{F\}$ is zero, any value of $\{u\}$ will satisfy the equation. The equation can always be satisfied by prescribing $\{u\}$, rather than $\{F\}$. This is equivalent to constraining u with a single-point constraint.

There are two types of singularity:

1. The singularity can be identified by considering the stiffness terms of only one grid point. This is called a grid point singularity.
2. The singularity requires consideration of stiffness terms of more than one grid point. This is called a mechanism type of singularity.

Grid Point Singularity

Grid point singularities are automatically identified by the GPSP module. They may be automatically constrained, under user control. First, multipoint constraints are eliminated, leading to the matrix $[K_{nn}]$ (see Multipoint Constraint Operations in SubDMAPs SEKR and SEMR2 and SEMRB, 366). Singularities are identified by grid point and component number of the failed direction. The ratio of stiffness at the degree of freedom to the largest stiffness at that grid point is listed. If automatic constraint is selected (see AUTOSPC (p. 797) in the *MSC Nastran Quick Reference Guide*), the set membership of the singular degrees of freedom is listed before the automatic constraint and after the automatic constraint. Both the mutually exclusive set and the highest combined set identical to the mutually exclusive set are listed for each identified singularity.

The output will include a Grid Point Singularity Table with the following format:

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 ID	TYPE	FAILED DIRECTION	STIFFNESS RATIO	OLD USET	NEW USET	
3801	G	1	0.00E+00	L	S	*
3801	G	2	0.00E+00	L	S	*
3801	G	3	0.00E+00	L	S	*
3801	G	4	0.00E+00	L	S	*
3801	G	5	0.00E+00	L	S	*
3801	G	6	0.00E+00	L	S	*

In this table,

1. G (under the heading TYPE) stands for grid point (S for scalar point is also output).
2. FAILED DIRECTION is the degree of freedom component in the displacement coordinate system and closest to the weakest direction. See Grid Point and Coordinate System Definition, 36 for definitions of component numbers.
3. The STIFFNESS RATIO is the ratio of the stiffness in the weakest direction to the stiffness in the strongest direction at the grid point (considering all possible directions rather than just the coordinate directions).

   Translational and rotational stiffnesses are considered separately. The default value for the failure criterion is 10^{-8}, which may be changed by the PARAM,EPZERO entry.
4. Note that, for the example shown, the failed degrees of freedom were originally in the u_l set and were placed in the u_s set. This may not always be the case, for reasons noted earlier, or because the user has not requested the AUTOSPC feature.
5. The user should carefully inspect the Grid Point Singularity Table because a singularity is very often the result of a modeling error, which should be corrected.

User Options

The identification of singularities is an automatic feature; however, the user may exercise the following options through PARAM Bulk Data entries described in Parameters (p. 785) in the *MSC Nastran Quick Reference Guide*.

Parameter Name	Description
AUTOSPC	Controls automatic constraint of a singularity.
EPPRT	Specifies the maximum value of a singularity to print.
EPZERO	Specifies the maximum value of a singularity.
PRGPST	Controls printout of singularity table.
SPCGEN	Writes SPC entries to the PUNCH file based on the singularity table.

The singular degrees of freedom are shifted to other sets by the following logic if PARAM,AUTOSPC,YES is specified:

o-set Condition	Set Before Constraint	Set After Constraint
If o-set exists	o c	s b
If no o-set	c b	s s

Other mutually exclusive sets not listed above will not be constrained. In addition, singular degrees of freedom on CYAX, CYJOIN, and SPCOFFi entries are identified but not constrained. The most usual cause of grid point singularity is degrees of freedom not constrained by elements because they are not of interest to the modeler. An example is out-of-plane rotations on a component made of plates lying in a plane. The automatic constraint provides the correct action for this type of singularity. Another cause is elements left out by oversight. Here, one typically observes that many or all degrees of freedom at a grid point are singular. The automatic constraint may mask a modeling error in this instance. For this reason, all singularity messages should be carefully inspected during the modeling checkout phase to ensure that modeling errors are not being masked.

When using the K2PP, M2PP, B2PP, or TFL Case Control command, PARAM,AUTOSPC,NO is recommended.

Mechanism-Type Singularities

A more complicated type of instability arises when two or more grid points participate in a singularity in a linear equation solution. This is known as a mechanism type of instability. Again, for the simple example shown in Figure 7-2, the equations of equilibrium can not be satisfied at all, or may have non unique solutions for special loading conditions. Note, however, that the structure of this example will pass the grid point singularity checks.

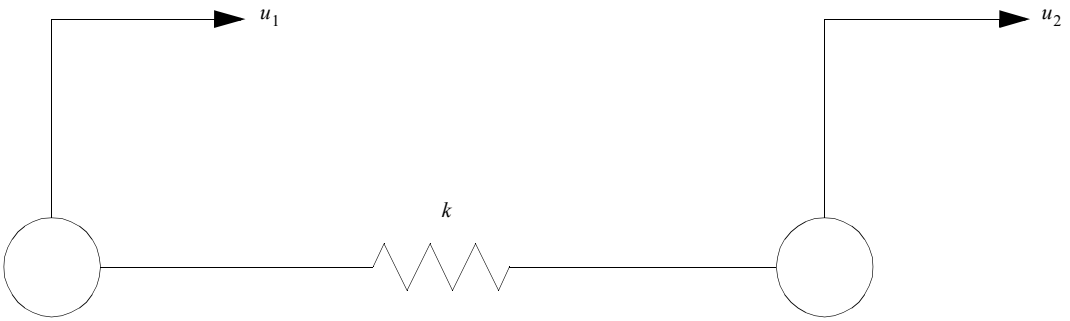


Figure 7-2 A Mechanism Type of Instability

The static equilibrium equation for the example in Figure 7-2 is

$$-\begin{bmatrix} k & -k \\ -k & k \end{bmatrix}\begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = \begin{Bmatrix} P_1 \\ P_2 \end{Bmatrix} -$$

If $P_1 = -P_2$, many solutions are possible, where $(u_1 - u_2) = P_1/k$, but u_1 may have any value. If P_1 and P_2 are arbitrary, the equation cannot be satisfied.

This condition is detected during the matrix decomposition phase of the linear equation solution process by the DCMP module. The stiffness matrix is decomposed into a lower triangular matrix, $[L]$, and a diagonal matrix, $[D]$. As discussed in User Interface (p. 76) in the a diagonal term of $[D]$ ("d_i") will approach zero for every mode of instability of the structure. (In this example there is one instability.) These instabilities are identified by dividing all d_i's into their corresponding

stiffness term, then listing all ratios larger than a large number. See MAXRATIO (p. 895) in the *MSC Nastran Quick Reference Guide* for a description of several other parameters used to control this operation.

Let $[K] = [L][D][L]^T$

Then $[D] = \begin{bmatrix} k & 0 \\ 0 & (k-k) \end{bmatrix}$, $[L] = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$

If $(k-k) = 0.0$, replace d_i with 10^{-10}

$Ratio_i = k_{ii}/d_i$

Identify Grid Component for All Ratios > PARAM,MAXRATIO.

A corrective action for this example would be to constrain one of the grid points or to add springs connecting this spring to other parts of the structure (not shown). Since, in general, there may be many grid points active in a singularity, there is little reason for picking one over another for connection. The usual cause of this type of instability is elements left out by oversight. While this may be the user's intention, it cannot be detected by the program. For these reasons, there is no automatic method to constrain mechanisms. The discussion of MAXRATIO (p. 895) in the *MSC Nastran Quick Reference Guide* describes the various actions and user options that are available in the several contexts where singularities may be detected. Diagnosis techniques and corrective actions are described in the *MSC Nastran Numerical Methods User's Guide 3*.

Static Condensation in SubDMAPs SEKR and SEMR2

Structural partitioning is used to refer structural matrices to the boundary points in substructuring operations and to perform a static condensation in dynamic analyses.

At user option the set of free coordinates, u_f, may be partitioned into two sets, u_o and u_a, such that the u_o set is eliminated first. Thus

$$\{u_f\} = \begin{Bmatrix} u_a \\ u_o \end{Bmatrix} \tag{7-10}$$

The equilibrium equations after the elimination of constraints equation ((7-9)) may be written in partitioned form as

$$\begin{bmatrix} K_{aa} & K_{ao} \\ K_{ao}^T & K_{oo} \end{bmatrix} \begin{Bmatrix} u_a \\ u_o \end{Bmatrix} = \begin{Bmatrix} P_a \\ P_o \end{Bmatrix} \tag{7-11}$$

Rearrange the bottom half of equation (7-11):

$$[K_{oo}]\{u_o\} = \{P_o\} - [K_{ao}]^T\{u_a\} \ , \tag{7-12}$$

and solve for $\{u_o\}$:

$$\{u_o\} = [K_{oo}]^{-1}\{P_o\} - [K_{oo}]^{-1}[K_{ao}]^T\{u_a\} \tag{7-13}$$

Substitute for u_o into the top half of equation (7-11):

$$[\overline{K}_{aa} - K_{ao}K_{oo}^{-1}K_{ao}^T]\{u_a\} = \{\overline{P}_a\} - [K_{ao}][K_{oo}]^{-1}\{P_o\} \tag{7-14}$$

It is convenient to define the matrix

$$[G_o] = -[K_{oo}]^{-1}[K_{ao}]^T ,$$

so the equation (7-14) becomes

$$[\overline{K}_{aa} + K_{ao}G_o]\{u_a\} = \{\overline{P}_a\} + [G_o]^T\{P_o\} ,$$

where advantage is taken of the symmetry of $[K_{oo}]$.

Following the practice of condensation established in preceding subsections,

$$[K_{aa}]\{u_a\} = \{P_a\} ,$$

where

$$[K_{aa}] = [\overline{K}_{aa}] + [K_{ao}][G_o] ,$$

$$\{P_a\} = \{\overline{P}_a\} + [G_o]^T\{P_o\}$$

The $[G_o]$ matrix defined in is obtained practically from the solution of

$$[K_{oo}][G_o] = -[K_{ao}]^T ,$$

where

$$[K_{ao}]^T$$

is treated as a set of load vectors. Each such vector produces a column of $[G_o]$.

Once $\{u_a\}$ is obtained the set of omitted coordinates, $\{u_o\}$, is obtained as follows. Define the set

$$\{u_o^o\}$$

as the solution of

$$[K_{oo}]\{u_o^o\} = \{P_o\} \tag{7-15}$$

Then, using equation (7-15) in equation (7-12),

$$\{u_o\} = \{u_o^o\} + [G_o]\{u_a\} .$$

In subDMAP SEKR, the constrained stiffness matrix, $[K_{ff}]$, is partitioned by UPARTN as follows:

$$[K_{ff}] = \begin{bmatrix} \overline{K}_{aa} & K_{ao} \\ \overline{K}_{oa} & K_{oo} \end{bmatrix} -$$

DCMP and FBS are used to solve the equation

$$[K_{oo}][G_{oa}^{t}] = -[K_{oa}]$$

for the static transformation matrix, $[G_{oa}^{t}]$, and MPYAD performs the matrix reduction

$$[K_{aa}] = [\overline{K}_{aa}] + [K_{ao}][G_{oa}^{t}]$$

If PARAM,ALTRED,YES is specified then the $[K_{ff}]$ matrix is partitioned and reassembled to form $[K_{ff}^{x}]$ which has the a-set degrees of freedom sequenced last. $[K_{ff}^{x}]$ is partially decomposed

$$[K_{ff}^{x}] \rightarrow [L_{fo}],[K_{aa}]$$

where $[L_{fo}]$ is a trapezoidal matrix and $[K_{aa}]$ is the boundary stiffness matrix.

The $[L_{fo}]$ matrix is partitioned

$$[L_{fo}] = \begin{bmatrix} L_{oo} \\ \overline{L}_{ao} \end{bmatrix} -$$

While $[L_{ao}]^{T}$ is of the same dimension as the $[G_{oa}^{t}]$ matrix, it is on the average half as dense, thereby reducing the storage requirements.

In subDMAP SEMR2, UPARTN partitions the constrained mass matrix $[M_{ff}]$ as follows:

$$[M_{ff}] = \begin{bmatrix} \overline{M}_{aa} & M_{ao} \\ \overline{M}_{oa} & M_{oo} \end{bmatrix} -$$

and successive executions of MPYAD and SMPYAD modules perform the matrix reduction

$$[M_{aa}] = [G_{oa}^{t}]^{T}[M_{oo} \quad G_{oa}^{t} + M_{oa}] + [M_{oa}^{T} \quad G_{oa}^{t} + \overline{M}_{aa}]$$

In subDMAP SEMRB, the structural and viscous damping matrices $[B_{ff}]$ and $[K_{ff}^{4}]$ are condensed by the MATREDU module.

The partitioned solution for statics is an exact solution. It merely changes the order of operations of the unpartitioned solution. The static reduction for dynamics is exact for o-set degrees of freedom that are massless, but represents an approximate relumping of o-set masses to the a-set. The user must select enough a-set degrees of freedom to adequately define the eigenvectors of the system. The dynamic reduction technique discussed in Dynamic Reduction and Component Mode

Synthesis in SubDMAP SEMR3, 382 provides a way of improving the approximation by computing the normal modes for the o-set degrees of freedom.

Static Solutions in SubDMAP SEKRRS

There are two major branches through this subDMAP, standard static analysis and static analysis with inertia relief effects.

Inertia Relief Not Selected

When inertia relief is not selected and there are no SUPORTi entries are present the a-set equations are solved directly

$$K_{aa} \cdot u_a = P_a$$

K_{aa} is decomposed into its factor matrices L_{aa} and D_a. These factors are passed on to SubDMAP STATRS for solution of u_a.

When inertia relief is not selected but there are SUPORTi entries are present the a-set equations are partitioned into the r-set, defined on the SUPORTi entries, and the l-set, what is "left over" after the r-set is removed.

$$\begin{bmatrix} K_{ll} & K_{lr} \\ K_{rl} & K_{rr} \end{bmatrix} \begin{bmatrix} u_l \\ u_r \end{bmatrix} \geq \begin{bmatrix} P1 \\ P_r \end{bmatrix}$$

In this context, the r-set is treated as an alternate form of single-point constraints. The partitions of K involving r are discarded. u_r is set to zero. u_l is found from the first row of this equation. Loads in P_r go directly to ground, and are output with the SPC forces, along with the product $-K_{rl} \cdot u_l$.

Inertia Relief Selected

The basic assumption of inertia relief is that the structure is in a state of steady acceleration, and that all transients due to dynamic effects are negligible. The solution is obtained in a moving reference frame from which all displacements are measured. For inertia relief analysis inertial loads are computed automatically that exactly balance the applied loads. A support system is applied to make the system numerically stable. The inertial loads assume rigid body accelerations when computing the loads.

The derivations given here assume that the system mass matrix M_{gg} has been reduced to M_{aa} and is available, as described elsewhere. In the actual DMAP used in MSC Nastran 2001 equivalent methods that never generate M_{aa} are used instead, in the interests of efficiency. M_{aa} can be a larger size data block, and costly to compute. It can be shown that the results provided by the two derivations are identical. For example, the rigid body mass matrix M_{rr}, defined below as

$$M_{rr} = D_{ar} \cdot M_{aa} \cdot D_{ar}$$

can just as easily be formed from the g-set equations

$$M_{rr} = D_{gr} \cdot M_{gg} \cdot D_{gr}$$

Similar comments can be made about computing the inertial loads that balance the applied loads. The derivatives are presented here in the form of a-set equations that are more readily understood, if less efficient in practice, than the g-set equations.

There are at present two forms of inertia relief: automatic selection of a support system ("auto-support"); and manual selection of the support system, using SUPORTi entries. Both react to PARAM,GRDPNT, the parameter used to locate a reference point for grid point weight summaries. The more automatic method is described first.

Auto-Support (PARAM,INREL,-2)

Vectors of rigid body motion D_{gr} for six unit motions of a reference point are computed from grid point geometry in module VECPLOT. The D_{ar} component is partitioned from this matrix. D_{ar} is used to compute the rigid body mass M_{rr} about the reference point.

$$MD_{ar} = M_{aa} \cdot D_{ar}$$

$$M_{rr} = D_{ar} \cdot MD_{ar}$$

M_{rr} is printed out with the matrix name of QRR.

The acceleration of the reference point ar due to the applied loads is computed,

$$M_{rr} \cdot ar \geq D_{ar} \cdot P_a$$

The accelerations ar are printed with the matrix name of URA.

The resulting inertial loads P_{ai} and total loads P_a are

$$P_{ai} = M_{aa} \cdot ar$$

$$P_{a1} = P_a - P_{ai}$$

A constraint equation is written that states that the average weighted motion of the system is zero,

$$MD_{ar} \cdot u_{ar} = 0$$

This equation is appended to the a-set to define the 1-set equations in their augmented form,

$$K_{ll} \cdot u_l = P_l$$

$$\begin{bmatrix} K_{aa} & MD_{ar} \\ MD_{ar} & 0_{rr} \end{bmatrix} \begin{bmatrix} u_a \\ -q_r \end{bmatrix} = \begin{bmatrix} P_{a1} \\ 0_r \end{bmatrix}$$

q_r are the unknown forces required on the system to produce equilibrium. They should be computational zeros, and are output with the SPC forces. The equation involving K_{ll} is solved directly, with u_a and q_r partitioned from u_l.

Manual Support (PARAM, INREL, -1 and SUPORTi)

The user prescribes a set of DOFs that constrain the structure in a statically-determinate manner on SUPORTi entries. These entries define the r-set. There must be six or less DOFs listed, and they must constrain all rigid body modes that are not constrained by SPCs or other modeling devices. The rigid body modes are computed from the statics equation, where there are no loads applied, and u_{rr} is the identity matrix. Given

$$\begin{bmatrix} K_{ll} & K_{lr} \\ K_{le} & K_{rr} \end{bmatrix} \begin{bmatrix} D_{lr} \\ I_{rr} \end{bmatrix} = \begin{bmatrix} 0_{lr} \\ 0_{rr} \end{bmatrix}$$

D_{lr} is found by solving the first row of this equation, then merged with I_{rr} to form D_{ar},

$$K_{ll} \cdot D_{lr} \geq -K_{lr} \cdot I_{rr}$$

The rigid body mass, reference point accelerations, and inertial loads are formed with the same equations used for auto-support. The loads are partitioned into the l- and r-sets, and the partitioned form of K_{ll} is used for solution for displacements,

$$K_{ll} \cdot u_l = P_{ll}$$

u_a is merged from u_l and a null partitioned for u_r.

A Comparison of Auto- and Manual-Support Solution

If one analyzes the same model with both methods, you should find that stresses, constraint forces, and other internal quantities are identical except for truncation errors, but the displacements may differ appreciably. This is because they use a different reference frame. For the many supports, the reference frame is set up so that the motion at the r-set DOFs are identically zero. One can think of a reference frame drawn through these points, and all displacements measured relative to it. This reference frame is moving through space at a constant acceleration rate, so the displacements are relative, rather than inertial displacements.

For the auto-support option, one can visualize the reference frame as a set of axes connected to every point with mass with a very complicated set of levers that allows the structure to “breathe”, as loads are applied to it, but keep the weighted average motion to a zero value. If the model happens to have a grid point at the reference point, you will observe very small motion there, computational zeros but not the binary zeros of the manual support method. This is because this motion is computed for the auto-support method, but assumed to be zero for the manual method.

The choice of an r-set as defined by SUPORTi DOFs requires meeting both a geometric requirement and a stiffness requirement. The geometric requirement is that the set should form a statically determinate tie to ground. This can be difficult to do with complicated geometries. The stiffness requirement is that the DOFs should be stiff enough to provide a stable solution. A method that has been used by some modelers is to pick points stiff enough that they could be used to lift the structure without damaging it. Sometimes such points do not exist. The difficulties in picking a stable, reliable support system was one of the main motivations for development of the more modern auto-support systems.

PARAM, GRDPNT, GID is optional for both methods when superelements are not present, but required by the manual method when superelements are present. The auto-method does not support superelement analysis. The default for

GRDPNT is GID = -1, which is a flag to use the basic origin as the reference point when computing rigid body modes. Use of a GID near the c.g. of the structure produces more esthetically pleasing results. The rigid body accelerations produced by the solution are then measured near the center of the structure, rather than at a point that may be outside of the structure.

Static and Dynamic Load Generation

Applied Static Loads in SubDMAPs SELG, SELR, and SELRRS

The static load vectors $\{P_g\}$ are generated by SSG1 (Static Solution Generator — Phase 1) in subDMAP SELG from the user input of applied loads, temperature fields, and enforced deformations.

The direct matrix input requested by the P2G Case Control command define the load matrix $[P_g^2]$ which is assembled by the MTRXIN module and is added to the output from SSG1.

If PARAM,INREL,-1 is specified, then

- The rigid body transformation matrix $[D_{gr}]$ is generated with the VECPLOT module. Its columns contain the motion of all degrees of freedom due to unit motion of the corresponding reference point degree of freedom. (See PARAM,INREL in Use of Parameters, 32 for a discussion of the reference point ("GIC").)
- Loads due to unit rigid body accelerations are computed,

  $$\{P_j^i\} = -[M_{jj}][D_{gr}]$$

- The loads above are appended to the applied loads $\{P_j\}$,

  $$\{P_j\} \leftarrow \{P_j \quad P_{ij}^i\}$$

  The SELA module assembles $\{P_j\}$ into $\{P_g\}$.

  The constraints are applied to the static load vectors by SSG2 (Static Solution Generator — Phase 2) in subDMAP SELR as follows:

$$\{P_g\} = \begin{Bmatrix} \bar{P}_n \\ \underline{P_m} \end{Bmatrix} - \tag{7-16}$$

$$\{P_n\} = \{\bar{P}_n\} + [G_{mn}]^T\{P_m\} \tag{7-17}$$

$$\{P_n\} = \begin{Bmatrix} \bar{P}_f \\ \underline{P_s} \end{Bmatrix} - \tag{7-18}$$

$$\{P_f\} = \{\bar{P}_f\} - [K_{fs}]\{Y_s\} \tag{7-19}$$

$$\{P_f\} = \begin{Bmatrix} \bar{P}_a \\ \underline{P_o} \end{Bmatrix} - \tag{7-20}$$

$$\{P_a\} = \{\bar{P}_a\} + [G_{oa}]^T\{P_o\} \tag{7-21}$$

If PARAM,ALTRED,YES is specified, the diagonal extracted from $[L_{oo}] \rightarrow [D_{oo}]$

and the reduced load matrix is computed from

$$\{P_a\} = \{\bar{P}_a\} - [L_{ao}][D_{oo}][u_o^x]$$

where $\{u_o^x\}$ is obtained on a forward pass from the equation

$$[L_{oo}\ D_{oo}]\{u_o^x\} = \{P_o\}$$

for PARAM,ALTRED,NO, the SSG3 module (Static Solution Generation — Phase 3) solves the equation

$$[K_{oo}]\{u_o^o\} = \{P_o\}$$

for $\{u_o^o\}$, the displacements of the omitted coordinates.

SSG3 also calculates the residual vector, δP_o, and the residual vector error ratio, ε_o, for the omitted coordinates

$$\{\delta P_o\} = [K_{oo}]\{u_o^o\} - \{P_o\}$$

$$\varepsilon_o = \frac{\{u_o^o\}^T\{\delta P_o\}}{\{P_o\}^T\{u_o^o\}}$$

Except for round-off error, the error ratio ε_o should be zero. Large values of these error ratios usually indicate singularities in the stiffness matrix. The residual load vector, RUOV, may be output by use of PARAM,IRES,1.

The quantity

$$1/2\{P_o\}^T\{u_o\}$$

is calculated by the SSG3 module and printed under the heading “External Work.” This component of strain energy includes effects of thermal loads, element deformations, and enforced displacements that may be subtracted later in the solution process. For example, an enforced displacement that causes zero strain will result in external work.

If a r-set is present, subDMAP SELRRS calculates determinate forces of reaction

$$[q_r] = -[D_{ar}]^T\{P_a\}$$

where:

$$[D_{ar}] = \begin{bmatrix} D_{er} \\ I_{rr} \end{bmatrix}$$

Applied Dynamic Loads

Dynamic loads are produced in many subDMAPs. They are generally of the form:

$$P(x) = \sum_i PO_i \cdot F_i(x)$$

The PO_i vectors are spatial functions that describe what DOFs are loaded and their scale factors. The F_i functions are temporal functions that define how the loads change with time in transient analysis $x = t$, or with excitation frequency in frequency response analysis $x = f$. All spatial loads except those on extra points are calculated in the static analysis load generator module, SSG1, to obtain the P_g vector described in the prior paragraphs.

Enforced motions are produced similarly.

$$Y(x) = \sum_i YO_i \cdot F_i(x)$$

The YO_i vectors are again the spatial distribution, and the $F_i(x)$ temporal functions. The YO_i vectors are produced by the GP4 module to produce the Y_g vector. It is the same vector described for static analysis in (7-19), except that it is merged to g-set size with null terms from *Y*s.

The static load and enforced motion vectors are converted into dynamic loading tables in DPD module, then passed on to the dynamic load generators, FRLG and TRLG. These modules produce the applied loads and enforced motions at the p-set level, $Pp(x)$ and $Yp(x)$, as a function of excitation frequency or time. For transient analysis velocity and acceleration vectors are also provided, $Y\dot{p}(x)$ and $Y\ddot{p}(x)$.

A reduction matrix Rpx is made in the FDRMGEN module, where the subscript "x" implies the d-set, for direct analysis, and h, for modal analysis. It combines the operations shown in equations (7-16) through (7-21) previously. In subDMAP CFORCE2 equation (7-19) is modified to add velocity and acceleration effects,

$$P_f = P_f - K_{fs} \cdot Y_s - B_{fs} \cdot \dot{Y}s - M_{fs} \cdot \ddot{Y}s$$

In frequency response the velocity and acceleration effects can be computed from Y_s by multiplying it by $i \cdot \omega$ once or twice, where ω is the excitation frequency.

P_f is reduced to P_d or P_h by the same techniques used in static analysis.

For modal analysis with enforced motion it is essential that RESVEC be used. It causes static solutions to be formed from $K_{fs} \cdot Y_s$ -type terms. When the structure is tied to ground in a statically-determinate manner, these solutions are linear combinations of the rigid body modes that would exist if the ties to ground were removed. If the ties to ground are redundant, a condition named multi-base input, the resulting shapes will have significant stresses associated with them. They represent the effect of low-frequency excitation on the structure when RESVEC is not present, for the statically determinate case the element stresses will be correct but the displacement-type data will be missing the effect of the static shapes. For the multi-base input case, both stresses and displacement-type output will be incorrect.

Dynamic Reduction and Component Mode Synthesis in SubDMAP SEMR3

These capabilities use several subsets of the a-set not used by other solution techniques. The v-set matrix is formed in subDMAP SEKR

$$K_{vv} = \begin{bmatrix} K_{oo} & K_{or} & K_{oc} \\ K_{ro} & K_{rr} & K_{rc} \\ K_{co} & K_{cr} & K_{cc} \end{bmatrix}$$

The v-set mass matrix is formed similarly in subDMAP SEMR3. The v-set degrees of freedom are free to vibrate during dynamic reduction and component mode computation. They consist of the omitted terms (o-set), the free boundary points used for rigid-body mode designation (r-set), and the other free boundary points (c-set). The $[M_v]$ matrix is formed in a similar manner.

If generalized dynamic reduction is requested by the DYNRED Case Control command, the shifted stiffness matrix is formed,

$$[A_{vv}] = [K_{vv}] + \lambda_s[M_{vv}]$$

where λ_s is determined from data on the specified DYNRED Bulk Data entry. The approximate eigenvectors are found from $[A_{vv}]$ and $[M_{vv}]$ using inverse iteration based on solving the equation

$$[A_{vv}][\phi_v]_{i+1} = [M_{vv}][\phi_v]_i$$

for $[\phi_v]_{i+1}$, starting with random vectors for $[\phi_v]_o$. The $[\phi_v]_i$ are stored, then made mass orthogonal with respect to each other to form the matrix $[\phi_{vy}]$, where N_y the number of generalized functions used, may be determined automatically. N_y must be equal to or less than N_q, the upper limit for the number of generalized degrees of freedom specified by the user.

The generalized mass and stiffness matrices are computed,

$$[M_{yy}] = [\phi_{vy}]^T[M_{vv}][\phi_{vy}]$$

$$[K_{yy}] = [\phi_{vy}]^T[K_{vv}][\phi_{vy}]$$

If the METHOD command is specified for superelements, then the component modes are computed. The auto-omit operations described in Real Eigenvalue Analysis in SubDMAPs SEMR3 and MODERS, 403 are carried out resulting in the reduced eigensolution equation

$$[K_{yy} - \lambda M_{yy}][\phi_{yz}] = 0$$

where N_z is the number of eigenvectors requested by the EIGR or EIGRL Bulk Data entries. The back-transformations for the auto-omit-spc are performed, resulting in the expanded eigenvector matrix $[\phi_{yz}]$. This matrix is used to purify the approximate eigenvectors found in dynamic reduction,

$$[\phi_{vz}] = [\phi_{vy}][\phi_{yz}]$$

The uncoupled eigensolution components are assembled from $[\phi_{vz}]$

$$[\phi_{vz}] = \begin{Bmatrix} \phi_{oz} \\ \phi_{rz} \\ \phi_{cz} \end{Bmatrix} \begin{matrix} - \\ \\ - \end{matrix}$$

$$[\phi_{az}] = \begin{Bmatrix} \phi_{oz} \\ \phi_{cz} \\ \phi_{bz} \end{Bmatrix} \begin{matrix} - \\ \\ - \end{matrix}$$

where $[\phi_{bz}] = [0]$

$[\phi_{oz}]$ and $[\phi_{az}]$ are stored for use with the uncoupled solution obtained using PARAM,FIXEDB,-1, as described in Use of Parameters, 32 and Data Recovery Operations in SubDMAP SEDISP, 388.

The residual vector capability is requested with the RESVEC Case Control command. Details on this capability are given in Matrix Assembly Operations in SubDMAP SEMG, 365, where real eigensolutions are discussed. The RESVEC SubDMAP returns auxiliary functions u_{o2}. They are appended to the eigenvectors, and are given all of the operations given to eigenvectors,

$$H_{vz} = \left[\begin{array}{c|c} \phi_{oz} & u_{o2} \\ \phi_{rz} & 0_{r2} \\ \phi_{cz} & 0_{c2} \end{array}\right]$$

The inertia relief mode shapes are calculated by statically applying the rigid-body inertial loads to the structure.

$$\begin{bmatrix} K_{oo} & K_{oa} \\ K_{ao} & K_{aa} \end{bmatrix} \begin{bmatrix} H_o^i \\ H_a^i \end{bmatrix} = \begin{bmatrix} M_{oo} & M_{oa} \\ M_{ao} & M_{aa} \end{bmatrix} \begin{bmatrix} V_o \\ V_a \end{bmatrix}$$

Setting $[H_a^i]$ to zero and solving for $[H_o^i]$ results in,

$$[H_o^i] = [K_{oo}]^{-1}[[M_{oo}][V_o] + [M_{oa}][V_a]]$$

The $[G_{oq}]$ matrix is formed,

$$[G_{oq}] \leftarrow [G_{oz}^1 \quad H_o^i \quad 0]$$

The $[G_{oq}]$ matrix is truncated or padded with null columns to provide Nq columns, where Nq is the q-set size. $[G_{oq}]$ is expanded to a-size,

$$[G_{oq} \quad 0_{ot}] \rightarrow [G_{oa}^q]$$

The total transformation matrix is formed by adding dynamic and static components and is reassembled whenever needed,

$$[G_{oa}] = [G_{oa}^t] + [G_{oa}^q]$$

The generalized stiffness coefficients are formed from the dynamic transformation,

$$[K_{qq}] = [G_{oq}]^T [K_{oo}][G_{oq}]$$

The physical and generalized degrees of freedom are not stiffness coupled. $[K_{qq}]$ is merged to N_a size,

$$[K^l_{aa}] = \begin{bmatrix} K_{qq} & 0 \\ 0 & 0 \end{bmatrix}$$

where superscript l refers to matrices formed from dynamic reduction. $[K_{qq}]$ is stored in the database. When the total stiffness boundary matrix is needed, it is formed from its partitions,

$$[K^m_{aa}] = \begin{bmatrix} K_{qq} & 0 \\ 0 & K_{tt} \end{bmatrix}$$

or

$$[K_{aa}] = [K^m_{aa}] + [K^l_{aa}]$$

The physical and generalized degrees of freedom are mass-coupled. The terms are formed from the equations,

$$[M_{qq}] = [G_{oq}]^T [M_{oo}][G_{oq}]$$

$$[M_{qt}] = [G_{oq}]^T [M_{ot} + M_{oo} \quad G^t_{oa}]$$

$$[M^l_{aa1}] = \begin{bmatrix} M_{qq} & M_{qt} \\ M_{tq} & 0 \end{bmatrix}$$

Virtual mass effects, if any, are added to $[M^l_{aa1}]$ to form $[M^l_{aa}]$.

The total mass matrix is formed by:

$$[M^m_{aa}] = [M_{aa}] + [M^l_{aa}]$$

For the rigid formats $[M_{aa}]$ is formed directly from its partitions,

$$[M_{aa}] = \begin{bmatrix} M_{qq} & M_{qt} \\ M_{tq} & M_{tt} \end{bmatrix}$$

The a-set matrices do not contain static effects in the t-set components, and dynamic effects in the q-set components. Virtual mass effects, if any, are added to $[M_{aa}]$ to form $[M^m_{aa}]$.

The damping matrices $[B_{gg}]$ and $[K^4_{gg}]$ are reduced to the a-set by the MATREDU module which performs the following operations:

1. Eliminate multipoint constraints

$$[B_{gg}] \Rightarrow \begin{bmatrix} \overline{B}_{nn} & B_{nm} \\ B_{mn} & B_{mm} \end{bmatrix}$$

$$[B_{nn}] = [G_{mn}]^T [B_{mn} G_{mn} + B_{mn}] + [B_{mn}]^T [G_{mn}] + [B_{nn}]$$

2. Eliminate single point constraints

$$[B_{nn}] \Rightarrow \begin{bmatrix} B_{ff} & B_{fs} \\ B_{sf} & B_{ss} \end{bmatrix}$$

3. Partition omitted degrees of freedom

$$[B_{ff}] \Rightarrow \begin{bmatrix} \overline{B}_{aa} & B_{ao} \\ B_{oa} & B_{oo} \end{bmatrix}$$

4. Perform static condensation

$$[B_{aa}] = [G_{oa}]^T [B_{oo} G_{oa} + B_{oa}] + [\overline{B}_{oa}]^T [G_{oa}] + [\overline{B}_{aa}]$$

Formulation of Dynamic Equations in SubDMAP GMA

The tables prepared by the geometry processors are modified by DPD (Dynamic Pool Distributor) to include the extra points (e-points) introduced for dynamic analysis.

$[K^m_{aa}]$, $[M^m_{aa}]$, $[B_{aa}]$, and $[K^4_{aa}]$ are expanded to the d-set to form $[K^1_{dd}]$, $[M^1_{dd}]$, $[B^1_{dd}]$, and $[K^4_{dd}]$. The direct input matrices $[K^2_{pp}]$, $[M^2_{pp}]$, and $[B^2_{pp}]$ requested by the K2PP, M2PP, and B2PP Case Control commands are reduced to the d-set to form $[K^2_{dd}]$, $[M^2_{dd}]$, and $[B^2_{dd}]$, respectively.

The general dynamic equation used in the direct methods is

$$[[M_{dd}]p^2 + [B_{dd}]p + [K_{dd}]]\{U_d\} = \{P_d\}$$

p is the derivative operator

$$\{U_d\} = \begin{Bmatrix} U_a \\ U_e \end{Bmatrix}$$

where:

$\{U_d\}$ = the extra points

For direct frequency response and complex eigenvalue analysis the stiffness, damping, and mass matrices are generated as follows:

$$[K_{dd}] = (1 + ig)[K^1_{dd}] + [K^2_{dd}] + i[K^4_{dd}]$$

$$[B_{dd}] = [B^1_{dd}] + [B^2_{dd}]$$

and

$$[M_{dd}] = [M^1_{dd}] + [M^2_{dd}]$$

For direct transient response the stiffness, damping and mass matrices are generated as follows:

$$[K_{dd}] = [K^1_{dd}] + [K^2_{dd}]$$

$$[B_{dd}] = [B^1_{dd}] + [B^{2x}_{dd}] + \frac{g}{\omega_3}[K^1_{dd}] + \frac{1}{\omega_4}[K^4_{dd}]$$

and

$$[M_{dd}] = [M^1_{dd}] + [M^2_{dd}]$$

where:

- $[K^1_{dd}]$ is the reduced structural stiffness matrix plus the reduced direct input K2GG (symmetric).
- $[K^2_{dd}]$ is the reduced direct input matrix K2PP plus the reduced transfer function input (symmetric oar unsymmetric).
- $[K^4_{dd}]$ is the reduced structural damping matrix obtained by multiplying the stiffness matrix $[K_e]$ of an individual structural element by an element damping factor g_e and combining the results for all structural elements (symmetric).
- $[B^1_{dd}]$ is the reduced viscous damping matrix plus the reduced direct input B2GG (symmetric).
- $[B^2_{dd}]$ is the reduced direct input matrix B2PP plus the reduced transfer function input (symmetric or unsymmetric).
- $[M^1_{dd}]$ is the reduced mass matrix plus the reduced direct input M2GG (symmetric).
- $[M^2_{dd}]$ is the reduced direct input matrix M2PP plus the reduced transfer function input (symmetric or unsymmetric).
- g is the structural damping coefficient on the PARAM Bulk Data entry G. The frequencies ω_3 and ω_4 are specified on the PARAM Bulk Data entries as W3 and W4, respectively.

For direct solutions $[K_{dd}]$, $[M_{dd}]$, and $[B_{dd}]$ are examined to identify rows and columns which are null in all three matrices. If the parameters ASING = 0 (the default) the singularities caused by the null rows and columns are removed as follows:

- For direct frequency and direct transient response, the stiffness matrix $[K^x_{dd}]$ is formed from $[K_{dd}]$ by placing unity on the diagonal for each null row and column.

- For direct complex eigenvalue analysis null rows and columns are discarded from $[K_{dd}]$, $[M_{dd}]$, and $[B_{dd}]$ to form $[K^d_{xx}]$, $[M^d_{xx}]$, and $[B^d_{xx}]$. This is performed in subDMAP DCEIGRS. See Complex Eigenvalue Analysis in SubDMAP CEIGRS, 416.

If parameter ASING = -1 and null rows and/or columns exist, a fatal error will result.

In modal frequency response, and complex eigenvalue analysis, structural damping and viscous damping are included in the dynamic matrices as follows:

$$[K^2_{dd}] = [K^{2x}_{dd}] + ig[K^1_{dd}] + i[K^4_{dd}]$$

and

$$[B^2_{dd}] = [B^1_{dd}] + [B^{2x}_{dd}]$$

In modal transient response, the structural damping and viscous damping are included as follows:

$$[K^2_{dd}] = [K^{2x}_{dd}]$$

and

$$[B^2_{dd}] = [B^1_{dd}] + [B^{2x}_{dd}] + \frac{g}{\omega_3}[K^1_{dd}] + \frac{1.0}{\omega_4}[K^4_{dd}]$$

The general dynamic equation used in the modal method is:

$$[[M_{hh}]p^2 + [B_{hh}]p + [K_{hh}]]\{U_h\} = \{P_h\}$$

where p is the derivative operator

$$\{U_h\} = \begin{Bmatrix} \zeta_i \\ U_e \end{Bmatrix}$$

ζ_i are the modal coordinates.

The transformation from $\{U_h\}$ to $\{U_d\}$ is

$$\{U_d\} = [\phi_{dh}]\{U_h\}$$

where:

$$[\phi_{dh}] = \begin{bmatrix} \phi_{ai} & 0 \\ 0 & I_{ee} \end{bmatrix}$$

The dynamic matrices expressed in modal coordinates are assembled in GKAM (General K Assembler-Modal) as follows:

$$[K_{hh}] = [k] + [\phi_{dh}]^T[K^2_{dd}][\phi_{dh}]$$

$$[M_{hh}] = [m] + [\phi_{dh}]^T[M^2_{dd}][\phi_{dh}]$$

and

$$[B_{hh}] = [b] + [\phi_{dh}]^T [B_{dd}^2][\phi_{dh}]$$

where, if PARAM,KDAMP = 1 (default)

m_i = modal mass

$b_i = 2\pi f_i g(f_i) m_i$

$k_i = 4\pi^2 f_i^2 m_i$

If PARAM,KDAMP = -1 in complex eigenvalue analysis, modal frequency response, aerodynamic flutter, and aeroelastic response the viscous modal damping will be incorporated into the complex stiffness matrix as follows:

m_i = modal mass

$b_i = 0$

$k_i = [1 + ig(f_i)]4\pi^2 f_i^2 m_i$

Values of $g(f_i)$ are specified on the TABDMP1 Bulk Data entry, selected by the SDAMPING Case Control command. If KDAMP = 1 (default), the matrices $[b]$ and $[k]$ are formulated as in modal transient response.

In SubDMAP CMPMODE damping of CMS(o-set) modes is calculated by similar techniques. This q-set damping is expanded to g-set size, and placed in the B_{aa} matrix for summing into downtown superelements.

Data Recovery Operations in SubDMAP SEDISP

The dependent displacements and the single-point forces of constraint are calculated in SDR1 (Stress Data Recovery — Phase 1). The inputs to SDR1 are columns of the solution vectors corresponding to each eigenvalue, loading condition, or output time step. In the case of transient analysis, there are three columns, corresponding to displacement, velocity and acceleration, for each output time step.

Load Modification

If PARAM,INREL,-1 or PARAM,INREL,-2 are specified, then any component of the load vector is first multiplied by $[C_{rx}]$ to produce a set of loads that are a linear combination of applied and inertial loads called $\{P_l^i\}$

$$\{P_l^i\} \leftarrow [P_l][C_{rx}]$$

Static Analysis

In the case of static analysis and PARAM,ALTRED,NO, SDR1 recovers dependent displacements

$$-\left\{\begin{matrix} u_l \\ \hline u_r \end{matrix}\right\} \Rightarrow \{u_a\}$$

$$-\left\{\frac{u_a}{u_o}\right\} \Rightarrow \{u_f\}$$

where

$$\{u_o\} = [G_{oa}^t]\{u_a\} + \{u_o^o\}$$

If PARAM,ALTRED,YES is specified, then the displacements $\{u_o\}$ are obtained from a backward pass from the equation

$$[L_{oo}]^T\{u_o\} = \{u_o^y\}$$

where $\{u_o^y\} = \{u_o^x\} - [L_{ao}]^T\{u_a\}$

Enforced displacements are combined with u_f

$$-\left\{\frac{u_f}{Y_s}\right\} \Rightarrow \{u_n\}$$

$$-\left\{\frac{u_n}{u_m}\right\} \Rightarrow \{u_g\}$$

where

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

and recovers single point forces of constraint

$$\{q_s\} = [K_{fs}]^T[u_f] + [K_{ss}]\{Y_s\} - \{P_s\}$$

and multipoint forces of constraint

$$q_{mg} = \begin{bmatrix} I_{mm} \\ -G_{mn}^t \end{bmatrix}[[K_{mg}]\{u_g\} - \{P_m\}]$$

In superelement static analysis, if PARAM,FIXEDB,-1 is present, then u_a is set to zero, and data recovery is performed using the modified equation

$$\{u_o\} = \{u_o^o\}$$

This is equivalent to obtaining a solution with all exterior ("boundary") degrees of freedom fixed. Displacements are due only to interior displacements due to interior loads, that is, the $\{u_o^o\}$ component.

Eigenvalue Analysis

In the case of eigenvalue analysis SDR1 recovers dependent components of the eigenvectors

$$\{\phi_o\} = [G_{oa}]\{\phi_a\}$$

$$\left\{\frac{\phi_a}{\phi_o}\right\} = \{\phi_f\}$$

$$\left\{\frac{\phi_f}{\phi_s}\right\} = \{\phi_n\}$$

$$\{\phi_m\} = [G_{mn}]\{\phi_n\}$$

$$\left\{\frac{\phi_n}{\phi_m}\right\} = \{\phi_g\}$$

and single-point forces of constraint

$$\{q_s\} = [K_{fs}]^T\{\phi_f\}$$

If PARAM,DYNSPCF,NEW (default) then the single-point forces are computed as follows:

$$\{q_s\} = [[K_{sf}] - \omega^2[M_{sf}]]\{\phi_f\}$$

In superelement analysis, if PARAM,FIXEDB,-1 is present, the uncoupled eigenvectors are recovered from the modified equations

$$\{\phi_a\} = \{\phi_{az}\}$$

$$\{\phi_o\} = \{\phi_{oz}\}$$

with $\{\phi_{az}\}$ and $\{\phi_{oz}\}$ from Superelement Analysis, 419.

The multipoint forces of constraint qm_q are recovered by

$$qm_q = \begin{bmatrix} I_{mm} \\ -G_{mn}^T \end{bmatrix}(-\omega^2 M_{mg} + K_{mg})$$

Dynamic Analysis

In the case of dynamic analysis with the direct formulation the extra points are partitioned out of $\{u_d\}$.

$$\left\{\frac{u_a}{u_e}\right\} = \{u_d\}$$

SDR1 recovers the dependent components

$$\{u_o\} = [G_{oa}]\{u_a\}$$

$$\begin{Bmatrix} u_a \\ u_o \end{Bmatrix} = \{u_f\}$$

and recovers single point forces of constraint

$$\{q_s\} = [K_{fs}]^T\{u_f\} - \{P_s\}$$

If PARAM,DYNSPCF,NEW (default) then the forces are computed as follows.

In transient response analysis, the formulation is

$$q_s = M^x_{sf}\ddot{u}_f + B^x_{sf}\dot{u}_f + K^x_{sf}u_f - P_s$$

where:

$$\begin{aligned} M^x_{sf} &= M_{sf} + M^{2pp}_{sf} \\ B^x_{sf} &= B^x_{sf} + B^{2pp}_{sf} + \frac{g}{\omega_3}K_{sf} + \frac{1}{\omega_4}K^{4gg}_{sf} \\ K^x_{sf} &= K_{sf} + K^{2pp}_{sf} \end{aligned}$$

In frequency response and complex eigenvalue analysis, the formulation is

$$q_s = (-\omega^2 M^x_{sf} + i\omega B^x_{sf} + K^x_{sf})u_f - P_s$$

where:

$$\begin{aligned} M^x_{sf} &= M_{sf} + M^{2pp}_{sf} \\ B^x_{sf} &= B^x_{sf} + B^{2pp}_{sf} \\ K^x_{sf} &= (1 + ig)K_{sf} + K^{2pp}_{sf} + iK^{4gg}_{sf} \end{aligned}$$

The multipoint forces of constraint qm_g are recovered

$$qm_g = \begin{bmatrix} I_{mm} \\ -G^T_{mn} \end{bmatrix}(X_m - P_m)$$

where qm_g is the reactions expanded to the g-set and X_m and P_m are the internal and external loads on the m-set degrees of freedom, respectively. (See also Eq. 9 in Section 3.5.1 in *The NASTRAN Theoretical Manual*).

In transient response analysis,

$$X_m = M^x_{mg}\ddot{u}_g + B^x_{mg}\dot{u}_g + K^x_{mg}u_g$$

where:

$$\begin{aligned} M^x_{mg} &= M_{mg} + M^{2pp}_{mg} \\ B^x_{mg} &= B_{mg} + B^{2pp}_{mg} + \frac{g}{\omega_3} K_{mg} + \frac{1}{\omega_4} K^{4gg}_{mg} \\ K^x_{mg} &= K_{mg} + K^{2pp}_{mg} \end{aligned}$$

In frequency response and complex eigenvalue analysis,

$$X_m = (-\omega^2 M^x_{mg} + i\omega B^x_{mg} + K^x_{mg}) u_g$$

where:

$$\begin{aligned} M^x_{mg} &= M_{mg} + M^{2pp}_{mg} \\ B^x_{mg} &= B_{mg} + B^{2pp}_{mg} \\ K^x_{mg} &= (1 + ig) K_{mg} + K^{2pp}_{mg} + iK^{4gg}_{mg} \end{aligned}$$

For frequency response there is one complex vector for each loading condition. For complex eigenvalue analysis there is one complex eigenvector for each complex root and there is no loading term $\{P_s\}$ associated with the determination of the single point constraints. For transient analysis there is a displacement, velocity, and acceleration associated with each output time step.

In the case of dynamic analysis with the modal formulation, the user may obtain improved element stresses by requesting the mode acceleration data recovery technique.

The use of the mode acceleration technique requires the decomposition of the l-set stiffness matrix $[K_{ll}]$. This decomposition is not otherwise performed in dynamic analysis unless the model has rigid body modes. Also, the technique requires a forward/backward substitution operation with a number of right-hand sides equal to the number of solution vectors. A subset of the solution vectors can be obtained can be obtained by using the OTIME or OFREQUENCY entries in the Case Control Section. The subset of solution vectors is prepared by the MODACC functional module.

In the case of dynamic analysis with the modal formulation, two methods of data recovery are available. The default procedure is usually much more efficient for data recovery operations when the modal formulation is used for dynamic analysis. In this procedure, the dependent components of the eigenvectors used in the modal formulation are first recovered by SDR1 using the same sequence of operations described for problems in real eigenvalue analysis (after eliminating unwanted modes). The complete eigenvectors are used as the input to SDR2 in order to determine the requested forces and stresses in terms of modal coordinates.

In dynamic analysis, the user may restrict the output request to the response of selected points in the solution set. In this case, the time-consuming operations in the recovery of the dependent degrees of freedom can be avoided. This solution set processing is performed by VDR (Vector Data Recovery).

The modal data generated by SDR1 and SDR2 are formed into a data matrix by DDRMM (Dynamic Data Recovery Matrix Method). The output quantities are determined by multiplying the data matrix by the modal solution vectors. The computer time for this matrix multiplication is proportional to the number of output times, the number of modes used in the modal formulation, and the number of output frequencies or output time steps. If both the number of output items and the number of modes are small, the computer time required will not be significant when compared to the total problem

solution time. Although the DDRMM method of data recovery is more efficient, a complete $\{u_g\}$ is not formed. For this reason the method does not permit the creation of deformed structure plots, grid point force balance or the use of the mode acceleration technique.

The alternate procedure may be invoked using PARAM,DDRMM,-1. With this method the solution vectors are first transformed from modal to physical coordinates

$$\{u_d\} = [\phi_{dh}]\{u_h\}$$

This operation is performed by MPYAD in subDMAP MODACC. Having determined the vectors $\{u_d\}$ in physical coordinates, the data recovery operations can be completed using the same sequence of operations described for the direct formulation of dynamics problems.

Most of the operations in the recovery of the dependent components do not require significant amounts of computer time. However, in the case of the multiply-add operation associated with the recovery of the omitted coordinates $([G_{oa}]\{u_a\}, [G_{oa}]\{\phi_a\})$ a significant amount of computer time will be required if there is a large number of vectors and a large number of components in the a-set. This operation is likely to be especially costly for transient problems because there is usually a large number of output time steps and there are three vectors for each time step.

Data Recovery Operations in SubDMAP SEDRCVR

The element forces and stresses are calculated in SDR2 (Stress Data Recovery — Phase 2). The matrix operations in this phase of data recovery are similar to those performed in EMG for stiffness matrix generation. The computer time will be proportional to the number of elements for which output is requested.

If static loads are applied to intermediate points on the CBAR element, the user may request the output of stresses and forces at a selected set of intermediate points. The additional calculations required for output at intermediate points on CBAR elements is performed in functional module SDRX.

The solutions and data recovery operations for transient analysis are initially sorted by the time step (SORT1). A transpose operation is performed by SDR3 (Stress Data Recovery — Phase 3) in order to prepare the requested output sorted by external point identification number or external element identification number (SORT2). In frequency response problems, the user may request either SORT1 or SORT2 output. The SORT2 output will usually be desirable if the solution is made for a large number of frequencies. In any event, the SORT2 output must be prepared if XY-plots are requested.

Computer time for the preparation of SORT2 output will be directly related to the number of items requested for output. If a large number of items are requested for output, the transpose operation will require a significant amount of computer time.

The user may request the output of a grid point force balance table for a selected set of points. This table lists the forces acting at each selected point due to element forces, single point forces of constraints and applied loads. The grid point force balance table is prepared by GPFDR (Grid Point Force Data Recovery).

The user may also request the output of an element strain energy table and/or an element kinetic energy table for a selected set of elements. (See the Case Control commands, EDE (Case), 311, EKE (Case), 314, and ESE (Case) (p. 334) in the *MSC Nastran Quick Reference Guide*.) The tables include the energy for each element, the percentage of the total energy for each element, and the energy density for each element. The energy density is computed by dividing the element energy by the

element volume. Some elements do not have enough data input to allow calculation of element volume, such as CELASi elements. The density output for these elements is set to zero. Simplified equations for other element volumes may be used. For example, offsets are ignored for bar-like elements, and midside node locations are ignored for plate and solid elements. Element volumes are calculated by the ELPRT module, and can be output by use of PARAM,EST,1. The strain energy table is prepared by functional module GPFDR.

Summary of Solution Sequence Operations

This section describes the data processing and algorithms that are performed by special purpose functional modules, sequences of functional modules (subDMAPs) and complete solution sequences.

Detailed descriptions of the functional modules are given in the MSC Nastran DMAP Programmer's Guide and Chapter 4 of the *MSC Nastran Programmer's Manual*.

Static Solutions

There are several major paths for static solutions and some minor ones. For standard static analysis the stiffness matrix is decomposed in SubDMAP SEKRRS, in preparation for a linear solution in SubDMAP STATRS. There is also a form of static analysis called inertia relief analysis.

There are now several paths through inertia relief analysis. This revision of this manual will discuss inertia relief in SOL 101 and a little in SOL 111. There is still another method in SOL 144, Static Aeroelasticity. It is documented elsewhere.

General Formulation

Consider common 3-D unconstrained structure with six 0 Hz rigid body modes.

Rigid body mechanics loads balance (small motion):

$$\sum \bar{F}_i - \sum m_i \bar{a}_i = \bar{0} \text{ and } \sum (\bar{M}_i + \bar{r}_{i/o} \cdot \bar{F}_i) - \sum (I_i \bar{\alpha}_i + \bar{r}_{i/o} \cdot m_i \bar{a}_i) = \bar{0}$$

In finite element matrix notation:

$$[R]^T\{P\} - [R]^T[M][R]\{\ddot{u}_o\} = \{0\}$$

R is a (a-degree of freedom x 6) geometric rigid body matrix resulting from unit displacements in each basic direction with respect to GRDPNT or (0,0,0). R provides summation and cross-product utilities for loads and motion at each degree of freedom i. Rigid body accelerations $\vec{a}_i$ and $\vec{\alpha}_i$ are represented by 6 x 1 $\ddot{u}_o$ at PARAM,GRDPNT. All $\vec{F}_i$ and $\vec{M}_i$ are entered into load vector P; m_i and I_i are entered into a-set mass matrix M. Solve for the rigid body accelerations:

$$\{\ddot{u}_o\} = ([R]^T[M][R])^{-1}[R]^T\{P\} \qquad (7\text{-}22)$$

$R^T MR$ is the total 6 x 6 a-set mass, nonsingular for normal 3-D models with appropriate mass properties.

Apply the balanced loads to the finite element structure in linear statics formulation. This form is employed by the PARAM,INREL,-2 method:

$$[K]\{U\} = \{P\} - [M][R]\{\ddot{u}_o\} \qquad \text{(Inrel = -2)} \tag{7-23}$$

In contrast, older method INREL = -1 and SOL 111 employ the following:

$$[K]\{U\} = \{P\} - [M][D]\{\ddot{U}_r\} \qquad \text{(Inrel = -1)} \tag{7-24}$$

$$[K]\{U\} = \{P\} - [M][\phi_r]\{\ddot{q}_r\} \qquad \text{(SOL 111 free-free RESVEC's)} \tag{7-25}$$

Each method uses a different representation for the rigid body matrix and accelerations. Stiffness matrix K is singular (i.e., rank $l = a - 6$), and each method likewise employs different techniques to solve for displacement shape U.

Inrel = -2

Inrel = -2 with auto-suport is designed for 3-D models with exactly 6 rigid body modes. The geometric R matrix is generated about reference GRDPNT, used instead of suport entries. R is a linear combination of normal rigid body modes ϕ_r and can be used to decouple rigid motion and provide a unique inertia relief solution, convergent to the SOL 111 $f \to 0$ result.

SOL 111 Basis

In MSC Nastran modal frequency response SOL 111, the elastic solution on a free-free structure converges statically as follows (using elastic modes notation e):

$$\{U_e\} = \lim_{f \to 0}\{U_e(f)\} = [\phi_e][k_{ee}]^{-1}[\phi_e]^T\{P\}$$

U_e also satisfies equation (7-25). It is obtained in MSC Nastran SOL 111 doing the following:

- Set PARAM,LFREQ or EIGRL f1 to low frequency value (i.e., 0.1 Hz) to exclude use of rigid body modes.
- Remove stiffness damping or load phasing to get real-valued, true static result.

An important observation is that U_e is orthogonal or "decoupled" from eigensolution rigid body modes ϕ_r. For "clean" low-strain rigid modes $r \geq 6$, geometric rigid body vectors R can be expressed as a linear combination of ϕ_t. SOL 111 static U_e is likewise decoupled from rigid motion R:

$$[\phi_r]^T[M]\{U_e\} = \{0\} \quad \rightarrow \quad [R]^T[M]\{U_e\} = \{0\} \tag{7-26}$$

SOL 101 Implementation

SOL 101 can use geometric R at low computational cost without having to compute normal rigid modes ϕ_r. For INREL = -2, impose the rigid body decoupling constraint observed in equation (7-26). Re-arrange equations (7-23), (7-22), and (7-26):

$$[K]\{U\} + [M][R]\{\ddot{u}_o\} = \{P\}$$

$$[R]^T[M][R]\{\ddot{u}_o\} = [R]^T\{P\}$$

$$[R]^T[M]\{U\} = \{0\}$$

Add together to obtain:

$$\begin{bmatrix} K & MR \\ R^TM & R^TMR \end{bmatrix} \begin{Bmatrix} U \\ \ddot{u}_o \end{Bmatrix} = \begin{Bmatrix} P \\ R^TP \end{Bmatrix} \tag{7-27}$$

Equation (7-27) is the resultant INREL = -2 formulation. Augmented equation (7-27) matrix is nonsingular for models with 6 rigid mode singularities in matrix *K*, because of the addition of 6 acceleration + 6 decoupling equations. For this scenario the solution is unique and identical to the SOL 111 elastic convergence solution. If *K* has more than 6 singularities - local, massless, or true mechanisms - equation (7-27) will become singular. Auto-support removes extra singularities in the matrix reduction above, but the quality of the inertia relief solution could reduce. Check for high factor diagonal ratios in the .f06 output. It is recommended that *K* have only the 6 rigid mode singularities for inertia relief with auto-suport to work most effectively.

Inrel = -1

Suport DOF (r-set) are commonly used in MSC Nastran to define reference DOF and partition singular stiffness K to calculate a relative displacement shape U. They are part of structure and solution (U_r), and must be independent DOF in the a-set. In inertia relief, Suport DOF also define the rigid body accelerations of the structure with nonzero $\ddot{U}_r$. INREL = -1 uses a stiffness/suport-generated rigid body matrix *D* applying unit displacements at each r-DOF in the flexible body:

$$[D] = \begin{bmatrix} -K_{ll}^{-1}K_{lr} \\ I_{rr} \end{bmatrix} = \begin{bmatrix} D_{lr} \\ I_{rr} \end{bmatrix} \tag{7-28}$$

INREL= -1, equation (7-24) becomes:

$$\begin{bmatrix} K_{ll} & K_{lr} \\ K_{rl} & K_{rr} \end{bmatrix} \begin{Bmatrix} U_l \\ U_r \end{Bmatrix} = \begin{Bmatrix} P_l \\ P_r \end{Bmatrix} - \begin{bmatrix} M_{ll} & M_{lr} \\ M_{rl} & M_{rr} \end{bmatrix} \begin{bmatrix} D_{lr} \\ I_{rr} \end{bmatrix} \{\ddot{U}_r\} \tag{7-29}$$

Premultiply by D^T and solve for acceleration $\ddot{U}_r$ similar to equation (7-22):

$$\{\ddot{U}_r\} = ([D]^T[M][D])^{-1}[D]^T\{P\} \tag{7-30}$$

Note: $[D]^T[K] = [0]$ is assumed, and valid when the structure is not grounded, and support DOF are a well-chosen statically determinate set, i.e., at 6 stiff DOF on the structure. *D* is comprised of strain-free rigid body vectors.

Solve top portion of equation (7-24) for U_i in terms of U_r :

$$\{U_l\} = [D_{lr}]\{U_r\} + [K_{ll}]^{-1}(\{P_l\} - [\overline{M}_{lr}]\{\ddot{U}_r\})$$

Setting $U_r = 0$ yields following relative displacement solution U:

$$\{U\} = \begin{bmatrix} U_i \\ U_r \end{bmatrix} = \begin{bmatrix} [K_{ll}]^{-1}\{\overline{P}_l\} \\ 0 \end{bmatrix}, \qquad \{\overline{P}_l\} = \{P_l\} - [(M_{ll}D_{lr} + M_{lr})]\{\ddot{U}_r\}$$

A significant property of INREL = -1 is that displacement U solution is not unique and differs with selection of suport DOF. But deformed shape of U and corresponding internal loads and stresses should not vary with suport sets. For well-chosen, statically determinate, stiff suport sets 1a and 1b:

$$\{U^{1a}\} \neq \{U^{1b}\} \quad shape\{U^{1a}\} = shape\{U^{1b}\} \quad \sigma\{U^{1a}\} = \sigma\{U^{1b}\}$$

On standard unconstrained 6 rigid-body-mode structure, ideal suport selections are:

- All six degrees of freedom at stiff independent grid at or near CG.
- Six degrees of freedom spread out to 3 distant stiff locations on the structure, in 3-2-1 statically determinate balance (i.e., support 123 on one corner, 23 and 3 on other corners).

Internal Loads and Accuracy

Inertia relief theory is predicated upon strain-free rigid body vectors available in the finite element model to formulate the free-free deformed solution:

$$\sigma[R] \approx \sigma[D] \approx \sigma[\phi_r] \approx [0]$$

When this assumption is valid, the following convergence and accuracy can be obtained:

$$\sigma\{U^2\} = \sigma\{U^1\} = \sigma\{U_e^{111}\}$$

INREL = -1 and matrix D tend to be more susceptible to inaccuracies above, because of the vagaries of suport selection, and the potential for “soft” suports. INREL = -2 tends to be more numerically robust for internal loads as well as displacement, as long as 6 rigid body modes exist which remain distinct from the 7th mode. Ideally rigid body mode frequencies should be on the order E-3 or E-4 less than first flexible mode. If difficult to achieve, both methods may be required to increase confidence in the analyses.

General Discussion

All methods of inertia relief analysis take the standard static analysis equation

$$K_{aa} \cdot u_a = P_a$$

where P_a are the applied loads, and add a set of constraint equations R_{ra} and automatically-computed inertial loads P_i. The constraints are required because K_{aa} is for a free structure and cannot be solved reliably without some form of constraint.

$$K_{aa} \cdot u_a = P_a - P_i = P_{ai}$$

$$R_{ra} \cdot u_a = 0$$

In theory, if P_{ai} is a set of balanced loads (no resultant) the solution of u_a is possible without the need for constraint equations. In practice, P_{ai} may have some unbalance due to numerical truncation effects. Even a small resultant load will provide a displacement solution at infinity in theory, or with very large numbers in practice, usually. The same truncation effects that unbalance the loads may provide enough "ground springs" in the stiffness matrix to allow a solution, but it will be a poor quality solution. Everything will first move 1.E5 units in the same direction. Stresses will be determined by subtracting numbers of this size to find element displacement perhaps on the order of 1.E-3. Ironically, if all goes well the forces in the support system are miniscule, but it must be present to control stability when truncation errors are present.

Types of Inertia Relief

Inertia Relief Analysis is a quasi-dynamic analysis. It is assumed that the structure is not tied to ground and is in a state of steady acceleration caused by the applied loads. The displacements measured relative to ground are constantly increasing and cannot be determined. Displacements are therefore determined with respect to an accelerating reference frame that is attached to the structure. The applied loads are balanced by inertial loads that are calculated automatically from the steady acceleration acting on the mass properties of the model. Inertia relief analysis is requested with PARAM,INREL,[x], where [x] may have the values -1 or -2.

The reference frame may be obtained by two different methods. For the manual support method (PARAM,INREL,-1) the user selects 6 or fewer DOFs using SUPORTi entries. The reference frame is defined from these DOFs. For the auto support method (PARAM,INREL,-2) 6 constraint equations are written that constrain the structure to zero motion, on average, while leaving it free to "breath" and deflect due to the applied loads and inertial loads. Both methods find it convenient to have "rigid body modes shapes" that represent a set of vectors that represent all possible strain-free motion of the structure when not loaded. The technique used to obtain rigid body mode shapes differs between the two methods.

Rigid Body Modes Using Manual Support (PARAM,INREL,-1 and SUPORTi)

The user prescribes a set of DOFs that constrain the structure in a statically-determinate manner on SUPORTi entries. These entries define the *r*-set. There must be six or less DOFs listed and they must constrain all rigid body modes that are not constrained by SPCs or other modeling devices. The rigid body modes are computed from the static analysis equation. There are no loads applied and I_{rr} is the identity matrix. Given

$$\begin{bmatrix} K_{ll} & K_{lr} \\ K_{lr}^T & K_{rr} \end{bmatrix} \begin{bmatrix} D_{lr} \\ I_{rr} \end{bmatrix} = \begin{bmatrix} O_{lr} \\ O_{rr} \end{bmatrix}$$

D_{lr} is found by solving the first row of this equation, then merged with I_{rr} to form D_{ar},

$$K_{ll} \cdot D_{lr} \leq -K_{lr} \cdot I_{rr}$$

$$[D_{ar}] = \begin{bmatrix} D_{lr} \\ I_{rr} \end{bmatrix}$$

Rigid Body Modes Using Auto-Support (PARAM,INREL,-2)

Vectors of rigid body motion D_{gr} for six unit motions of a reference point are computed from grid point geometry in module VECPLOT. The D_{ar} component is partitioned from this matrix.

The Basic Equation of Motion for Inertia Relief

Inertia relief analysis starts with the a-set equation

$$K_{aa} \cdot u_a \leq P_{ai}$$

and adds a constraint equation

$$R_{ar}^T \cdot u_a = 0$$

The last equation implies a similar equation on a generalized constraint force variable vector, q_r

$$R_{ar} \cdot q_r = P_{ai} - K_{aa} \cdot u_a$$

If P_{ai} represents balanced loads, q_r is zero. If there is a small unbalance due to numerical truncation effects, q_r is then small, but finite. There are two methods for defining R_{ar}, as discussed below. Both have the property of defining some part of u_a to have zero motion, so that the other parts of u_a are in effect measured relative to the constrained DOFs.

On of the solution methods partitions K_{aa} into its *r*-set and *l*-set components. The first equation is partitioned, and the a-set equations and the constraint equation may combine into one equation,

$$\begin{bmatrix} K_{ll} & K_{lr} & R_{rl}^T \\ K_{lr}^T & K_{rr} & R_{rr}^T \\ R_{rl} & R_{rr} & O_{rr} \end{bmatrix} \cdot \begin{bmatrix} u_l \\ u_r \\ -q_r \end{bmatrix} = \begin{bmatrix} P_{ail} \\ P_{air} \\ Y_r \end{bmatrix}$$

The methods of computing P_i are described below. Y_r, the generalized displacement input, is set to zero.

Constraint Equations Using Manual Supports

In the constraint equation $R_{rr} = I_{rr}$, the identity matrix. R_{rl} is null. The last row can then be solved for $u_r = Y_r = 0$. The second equation can then be used to eliminate q_r,

$$K_{lr}^T \cdot u_l - I_{rr} \cdot q_r = P_{air}$$

$$q_r = K_{lr}^T \cdot u_l - P_{air}$$

Because R_{rl} is null, the first equation then becomes

$$K_{ll} \cdot u_l = P_{ail}$$

This equation can be solved for u_l reduced from u_a directly. It will be shown later that P_r terms are transferred to P_l terms in the inertial loads. u_a is formed from a merge of u_l and null terms for u_r.

Constraint Equations Using Automatic Supports

A constraint equation is written that states that the average weighted motion of the system is zero,

$$R_{ra} = D_{ar}^T \cdot M_{aa}$$

$$R_{ra} \cdot u_{ar} = 0$$

R_{ra} is generally a dense matrix, with nonzero terms at most DOFs with mass. It could be used to reduce the K_{aa} matrix with equations similar to those used for multi-point constraints, but that would produce a dense K_{ll} matrix, much more expensive to decompose and solve than the K_{ll} used in the manual support method. A solution method is used which increases solution costs only slightly.

The *r*-set *l*-set partitioning is not needed. The constraint equation is appended to the a-set equations to define the *l*-set equations in their augmented form,

$$K_{ll} \cdot u_l = P_l$$

$$\begin{bmatrix} K_{aa} & R_{ra}^T \\ R_{ra} & 0_{rr} \end{bmatrix} \begin{bmatrix} u_a \\ -q_r \end{bmatrix} = \begin{bmatrix} P_{al} \\ 0_r \end{bmatrix}$$

q_r are the unknown forces required on the system to produce equilibrium. They should be computational zeros, and are output with the SPC forces. The equation involving the augmented K_{ll} is solved directly for u_l. u_a and q_r are then partitioned from u_l. In superelement analysis the constraints are applied in the residual structure only.

Matrix Decomposition in SubDMAP SEKRRS

There are two major branches through this SubDMAP, standard static analysis and static analysis with inertia relief effects.

Inertia Relief Not Selected

When inertia relief is not selected and there are no SUPORTi entries present the a-set is decomposed directly in the DCMP module,

$$K_{aa} = L_{aa} \cdot D_{aa} \cdot L_{aa}^T$$

L_{aa} is lower triangular, and D_{aa} is diagonal. Both are stored in an unconventional ("packed") format, and output in a single data block named LLL. This name is used because the *l*-set and the *a*-set are synonymous when no SUPORTi entries exist. The factor matrices L_{aa} and D_{aa} are passed on to SubDMAP STATRS for solution of u_a.

When inertia relief is not selected but there are SUPORTi entries present, the a-set equations are partitioned into the *r*-set, defined on the SUPORTi entries, and the *l*-set, what is "left over" after the *r*-set is removed.

$$\begin{bmatrix} K_{ll} & K_{lr} \\ K_{rl} & K_{rr} \end{bmatrix} \begin{bmatrix} u_l \\ u_r \end{bmatrix} \geq \begin{bmatrix} P_l \\ P_r \end{bmatrix}$$

In this context, the *r*-set is treated as an alternate form of single-point constraints. The partitions of *K* involving *r* are discarded. u_r is set to zero. u_l is found from the first row of this equation. Loads in P_r go directly to ground, and are output with the SPC forces, along with the product $-K_{rl} \cdot u_l$ from the second equation. K_{ll} is decomposed with the same equations used for conventional static analysis.

Decomposition of K_{ll}

For the manual support method K_{ll} is defined above. Its factors, used to compute the rigid body modes are also used later to compute the static solution. For the auto support method the factors of the expanded form of K_{ll} are passed on.

Solution in SubDMAP STATRS

Computation of Inertial Loads

The basic assumption of inertia relief is that the structure is in a state of steady acceleration and that all transients due to dynamic effects are negligible. The solution is obtained in an accelerating reference frame from which all displacements are measured. Inertial loads are automatically computed that exactly balance the applied loads. A support system is applied to make the system numerically stable. The inertial loads assume rigid body accelerations when computing the loads.

The derivations given here assume that the system mass matrix M_{gg} has been reduced to M_{aa} and is available, as described elsewhere. Unit inertial loads are computed from M_{gg}, then reduced by load reduction techniques, avoiding the need for M_{aa}. M_{aa} can be a large size data block, and costly to compute. It can be shown that the results provided by the two derivations are identical. For example, the rigid body mass matrix M_{rr}, defined below as

$$M_{rr} = D_{ar}^T \cdot M_{aa} \cdot D_{ar}$$

can just as easily by formed from the *g*-set equations

$$M_{rr} = D_{gr}^T \cdot M_{gg} \cdot D_{gr}$$

D_{ar} is used to compute the rigid body mass M_{rr} about the reference point.

$$M_{rr} = D_{ar}^T \cdot M_{aa} \cdot D_{ar}$$

M_{rr} is printed out with the matrix name of QRR.

The acceleration of the reference point a_r due to the applied loads is computed,

$$M_{rr} \cdot a_r = D_{ar}^T \cdot P_a$$

The acceleration a_r are printed with the matrix name of URA.

The resulting inertial loads P_{ai} and total loads P_{al} are

$$P_i = M_{aa} \cdot a_r$$

$$P_{ai} = P_a - P_i$$

For both methods, the mass matrix is reduced from M_{gg} size to M_{rr} size with load vector operations that can be shown to be identical to the M_{aa} operations described here, except for the order of operations. The load and mass transformations are based on rigid body modes produced from geometry.

A Comparison of Auto- and Manual-Support Solutions

When the same model is analyzed with both methods you should find that stresses, constraint forces, and other internal quantities are identical except for truncation errors, but the displacements may differ appreciably. This is because they use a different reference frame. For the manual supports, the reference frame is set up so that the motion at the *r*-set DOFs are identically zero. One can think of a reference frame drawn through these points, and all displacements measured relative to it. This reference frame is moving through space at a constant acceleration rate, so that displacements are relative rather than inertial displacements. There are many sets of SUPORT i entries valid for one model. If you change SUPORTi DOFs you may also expect to see changes in displacements, but not in internal loads or element stresses.

For the auto-support option, one can visualize the reference frame as a set of axes connected to every point with mass with a very complicated set of levers that allow the structure to “breathe” as loads are applied to it, but keep the weighted average motion to a zero value. If the model happens to have a grid point at the reference point, you will observe very small motion there, computational zeros but not the binary zeros of the manual support method. This is because this motion is computed for the auto-support method, but assumed to be zero for the manual method.

The choice of an *r*-set as defined by SUPORTi DOFs requires meeting both a geometric requirement and a stiffness requirement. This geometric requirement is that the set should form a statically determinate tie to ground. This can be difficult to do with complicated geometries. The stiffness requirement is that the DOFs should be stiff enough to provide a stable solution. A method that has been used by some modelers is to pick points stiff points do not exist. The difficulties in picking a stable, reliable support system was one of the main motivations for development of the more modern auto-support system. SUPORTi entries are not allowed for the auto support method, and will cause a fatal error exit when present.

PARAM,GRDPNT,GID is optional for both methods when superelements are not present. It is required when superelements are present, and must be on a boundary point attached to all superelements. This same point should be used for the SUPORTi DOFs for the manual support method. It defines the reference point used for generating the geometry-based rigid body vectors. The default for GRDPNT is GID=-1, which is a flag to use the basic origin as the reference point

when computing rigid body modes. Use of a GID near the c.g. of the structure produces more esthetically pleasing results. The rigid body accelerations produced by the solution are then measured near the center of the structure, rather than at a point that may be outside the structure.

Applications of Inertia Relief

Although this capability could be used in an environment where the reference frame is accelerating steadily, such as a free-falling elevator, the more common usage is when the modeler attempts to apply loads in equilibrium. He wishes the net acceleration of the structure to be zero. A ship floating in the sea, with pressure loads balancing gravity loads, is an example. Developing balanced load sets can be tedious on large models with complex geometry. The accelerations printed out in the URA matrix can be used as a measure of load quality. If the accelerations are negligibly small, the load balance is adequate. If the accelerations are large, some adjustment of the loads sets may be necessary. For good inertia relief analysis a good mass distribution is needed. As the structural weight is usually only a small proportion of the total weight, this means that a realistic payload mass distribution must be modeled also.

Another use of inertia relief analysis becoming more prevalent with the advent of auto support is in checkout of a new model. Although six and only six rigid body modes are provided, they are provided in the form of a "flying waterbed". To take an extreme example, suppose that a structure is made up of two components bolted together, but the bolts were not modeled, inadvertently. This model has twelve rigid body modes before the constraints are applied. Both components sit on the same waterbed. This means that all singularities will be constrained, and the model will run to completion on the first pass. If loads are applied to one component only, the constraints will cause the other component to move too, in a strain-free manner. A plot of the motion of the model should reveal that there is no displacement continuity between the two components. This should make the cause of the unexpected behavior apparent. As the goal of this analysis is model checkout only the structure mass distribution is usually adequate for stable solution. The payload masses may be added later.

Real Eigenvalue Analysis in SubDMAPs SEMR3 and MODERS

The basic equation solved in these SubDMAPs is

$$[K - \lambda \cdot M] \cdot \phi = 0$$

where K and M are real and symmetric, and M is positive semi-definite. This equation is defined for the a-set variables when called from MODERS for overall structural modes of vibration, and for o-set variables when called from SEMR3 during CMS (Component Mode Synthesis) analysis. Many steps are taken in DMAP to better automate the removal of DOFs that are ill-defined, and to augment the mode shapes with residual flexibility effects. The SubDMAPs that perform these operations are described below.

Although a knowledge of DMAP is not necessary for dynamic analysis, such a study can increase the understanding of the steps that are performed.

Massless Mechanism Identification in SubDMAP MMFIND

SubDMAP MMFIND identifies potential instabilities in the matrices used for real eigensolutions. The K and M matrices presented to the READ module for real eigensolutions may contain massless mechanisms (MMs). A massless mechanism has a constraint shape X such that

$$X^T \cdot K \cdot X = \varepsilon_k$$

$$X^T \cdot M \cdot X = \varepsilon_m$$

$$X^T \cdot X > 0.0$$

If ϕ contains the eigenvectors of the system, then

$$X^T \cdot \phi = 0$$

$$X^T \cdot M \cdot \phi = 0$$

The constraint shapes have nonzero terms only where the eigenvectors have zero terms. The massless mechanism shapes can be thought of as "deficient" eigenvectors describing parts of the system that are unconnected (or very weakly connected) to other parts of the structure that have both stiffness and mass that are active in the well-conditioned eigenvectors. ε_k and ε_m may be binary zeroes, or computational zeroes due to truncation effects. An "eigenvalue" λ associated with massless mechanism shapes is

$$\lambda = \varepsilon_k / \varepsilon_m$$

λ is indeterminate in the limit as the ε_i approach zero. It can be a very large number, a very small number, and either negative or positive. It is likely to change a large amount when small changes are made to the model. This is the definition of an unstable eigensolution. The assumption taken here is that the cause of this condition is either stiffness and/or mass effects inadvertently left out of this part of the model, with the result that the load paths are ill-defined. This load path is constrained, with a warning message to the user. That is, this mode of motion is later removed from the K and M matrices prior to eigensolution by steps equivalent to single point constraint elimination.

The massless mechanisms are defined from the shifted stiffness matrix $A = K + s \cdot M$, where s is a shift that the user may change from its default value with user parameter PARAM, SHIFT1, (default=1.234 Hz.). The default is chosen to make detection of rigid body modes as MMs unlikely. A is decomposed into its factors,

$$A = L \cdot D \cdot L$$

The ratios of the diagonal terms of A over D are stored in a vector Ry. Terms of D approach a value of zero at the ends of massless mechanisms. Terms of Ry larger in magnitude than user parameter MAXRATIO are reset to unity and are kept in vector R1. It can be shown that each of these terms is at the end of a massless mechanism constraint shape. R1 is expanded into a set of loading vectors P, with one unit term per column. P is used along with the factors to find the constraint shapes,

$$A \cdot u \leq P$$

The constraint shapes are normalized, and small numbers are discarded. Each column of u is then a potential constraint mode. It could also be a rigid body mode. The rigid body modes are detected by computing the diagonal terms of the generalized mass matrix for these shapes,

$$Mg = u \cdot M \cdot u$$

Terms of $Mg(i, i)$ greater than PARAM, MECHFIL, default value = 1.e-6 are an indication that the corresponding column is a rigid body mode. Columns of this type in u are discarded to produce u1, the matrix of constraint mode shapes. The location of the end of the constraint shapes is passed back in R. The constraint mode shapes themselves are passed back in MECHU. R is used later to remove rows and columns in K and M to constrain the massless mechanisms. The nonzero terms in MECHU are used to indicate which DOFs will discard loads applied to them, because they will have zero coefficients in the true eigenvectors of the system, leading to a null contribution to generalized force for all modes. MECHLD contains P, and is intended for a feature not yet implemented.

Massless Mechanism Control in SubDMAP XREAD

At present, the auto-omit DMAP is duplicated at several places, followed by a call to XREAD. It is therefore an implicit part of the XREAD operations now, and may be moved to XREAD in a later version to reduce the duplication of code.

SubDMAP XREAD modifies the input matrices used for eigensolution, when needed, then restores the modified solutions back to the original size of the matrices. Solutions for both structural vibration and buckling analyses are provided. This discussion describes vibration analysis first, with a concluding section on changes made for a buckling solution.

Auto-Omit Operation

All of the eigensolution methods require that each column of $[K]$ plus $[M]$ have some nonnull terms. Any degree of freedom that has only null terms in both matrices is regarded as undefined and causes a fatal error. The GIV and HOU methods of eigenvalue extraction also require that the mass matrix be positive-definite. The MGIV, MHOU, and the Lanczos methods allow null columns in the mass matrix but are generally faster if the massless degrees of freedom are removed by the static reduction method described in Static Solutions in SubDMAP SEKRRS, 376. However, the static reduction tends to increase the number of active columns in the a-set matrices. The INV, SINV, and Lanczos methods, by contrast, are generally slower if the static reduction is performed on massless degrees of freedom, because these methods take advantage of sparse matrices, whereas the other methods do not.

Three options are provided to deal with the various null column possibilities. If PARAM,ASING,-1 is used, any degrees of freedom with null columns in both the stiffness and mass matrices (“undefined degrees of freedom”) are identified and a fatal error exit is taken. This is an option to use to identify modeling errors before performing an expensive eigenvalue solution.

If ASING = 0 (the default) the method used depends on the eigenvalue extraction method. In the tridiagonal methods of eigenvalue extraction, degrees of freedom with null columns in the mass matrix are identified and placed in the w-set.

$$[M] = \left[\begin{array}{c|c} M_{xx} & M_{xw} \\ \hline M_{wx} & M_{ww} \end{array}\right]$$

where $[M_{ww}]$ and $[M_{wx}]$ are null by definition. Degrees of freedom in the r-set (SUPORT Bulk Data entry) must have mass, or a fatal error will result. The stiffness matrix is similarly partitioned.

$$[K] = \left[\begin{array}{c|c} K_{xx} & K_{xw} \\ \hline K_{wx} & K_{ww} \end{array}\right]$$

with until diagonal terms placed on null columns of $[K_{ww}]$. The static condensation matrices are formed

$$[K_{ww}][G_{wx}] = -[K_{wx}]$$

and

$$[K_{xx}] = [\overline{K}_{xx}] + [K_{wx}]^T[G_{wx}]$$

This method will prevent errors due to mass matrix singularly in the tridiagonal methods if the singularity is caused by null columns in $[M]$. It will not detect singularities due to mechanisms (caused, for example, by a point mass offset from a grid point). This type of singularity does not necessarily prevent obtaining eigensolutions, but will cause poor numerical stability. The MGIV or MHOU methods do not suffer from numerical instability due to singular or nearly singular mas matrices. They are therefore more reliable than the GIV HOU methods. In well conditioned problems, however, it is unnecessary to automatically select the appropriate method. The AHOU method is the most modern, reliable method.

The static reduction will reduce the cost of solution and minimize the number of uninteresting high roots found at the value of computed infinity due to the singularity of the mass matrix. Note that the static reduction used here does not introduce any approximation, as the "omitted" that is w-set terms have null mass values. The introduction of unit diagonal terms on null columns of $[K_{ww}]$ does not introduce approximations because the terms that multiply these diagonal terms (the corresponding rows of $[G_{??}]$ are null. The same results can be achieved at higher cost by first eliminating null rows and columns of $[K_{ww}]$ by the single point constraint operations of Single Point Constraint Operations in SubDMAPs SEKR, SEMR2 and SEMRB, 369 followed by static reduction.

The solution for the system $[K_{xx} - \lambda \cdot M_{xx}] \cdot \phi_x = 0$ is provided. Rigid body shapes D_{xr} are provided when SUPORTi entries exist, causing an r-set. The factor of K_{xx}, named LLL, is used for computing approximate solutions for buckling analysis, to better set the range of eigenvalues to be searched. For the tridiagonal methods, K and M have DOFs with zero mass removed by an auto-omit method in a prior subDMAP. For other methods of analysis, K_{aa} and M_{aa} are used directly for residual structure modes, and K_{vv} and M_{vv} for CMS (Component Mode Synthesis) modes.

The modes of the system are computed in the READ eigensolver module. If the solution appears to be of good quality the subDMAP returns with no other operations performed. If the modes can not be found, or appear to be of low quality, the module sets a flag (Neigv parameter). This flag causes the MMFIND subDMAP to be called. Given K and M, it finds a high-ratio vector R and a matrix of constraint mode shapes MMECHA. These terms are defined in the MMFIND section above.

There are nonzero terms in R at the end of massless mechanisms. These mechanisms can be removed with SPC-like operations using R as a partitioning vector,

$$K_{xx} = \begin{bmatrix} K22 & K21 \\ K12 & K11 \end{bmatrix}$$

M_{xx} is partitioned similarly, to obtain $M22$. Partitions involving the "1" subscript are discarded. The modified eigenproblem solved is then $[K22 - \lambda \cdot M22] \cdot \phi_2 = 0$. ϕ_2 is merged to the size of the original K matrix by inserting null rows for the "1-set" DOFs,

$$\phi_x = \begin{bmatrix} \phi_2 \\ 01 \end{bmatrix}$$

The auto-omit reduction is backed out by the operations

$$\phi_w = G_{wx} \cdot \phi_x$$

$$\phi = \begin{bmatrix} \phi_x \\ \phi_w \end{bmatrix}$$

Real Eigenvalue Analysis in the READ Module

Real eigenvalues and their associated eigenvectors are determined in READ (Real Eigenvalue Analysis -- Displacement). In the case of normal mode analysis, READ extracts real eigenvalues from the equation

$$[K - \lambda M]\{u\} = 0$$

There are seven basic eigensolution methods. The Inverse (INV and SINV) Method obtains eigensolutions by iterations based on the equation

$$[K - \lambda_i M]\{\phi_{i+1}\} = [M][\phi_i]$$

where λ_i is an estimate of the eigenvalue. This method is best suited to large problems with sparse matrices, where only a few eigenvectors are desired.

The tridiagonal methods, Givens (GIV and MGIV) and Householder (HOU and MHOU), first mass-scale the equation

$$[M] = [L][L]^T$$

where $[L]$ is a lower-triangular matrix. The $[K]$ and $[M]$ matrices are pre- and post-multiplied by $[L]^{-1}$ and $[L^{-1}]^T$, respectively, to obtain the standard form

$$[J - \lambda I]\{\bar{\phi}\} = 0$$

where $[J] = [L^{-1}][K][L^{-1}]^T$ and $[I]$ is the identity matrix. Note that the mass matrix is in effect inverted, so that it must be nonsingular. The $[J]$ matrix is transformed to tridiagonal form for economical computation of eigenvalues and eigenvectors. The eigenvectors in the tridiagonal basis are then back-transformed to the initial basis. The Modified Givens (MGIV) and Modified Householder (MHOU) Methods use the shifted matrix

$$[K + \lambda_s M]$$

in place of $[M]$ in the discussion above, thereby allowing the mass matrix to be singular if there are compensating terms in the stiffness matrix. The GIV, MGIV, and MHOU methods are best suited to small problems, or for large problems where many eigenvectors are required, after extensive static reduction or dynamic reduction is performed.

The Lanczos method uses a block shifted algorithm. Sets of vectors obtained by a recursive form are used to reduce the problem to a reduced block tridiagonal form. The eigensolutions are computed in the reduced basis by a QL algorithm,

then back-transformed to the original basis. This is currently the most modern method, and should be considered for all large-size problems. Since the method takes advantage of sparsity in the input matrices, it is most economical when used without static or dynamic reduction. It is requested with either the EIGRL or EIGR Bulk Data entries.

The AHOU method combines the best features of all of the tridiagonal methods, and is the recommended method at present. It first decomposes M. If the factors are stable, the HOU method is used. If the factors may be unstable, the MHOU method is used instead.

Modal Solution Preparation in the MODERS subDMAP

MODERS is called when solving for the eigensolution of the residual structure, with a-set operations. It is used for both structural mode and buckling analysis.

- In buckling, the r-set caused by SUPORTi entries are treated as equivalent SPCs. The rows and columns associated with r-set DOFs are discarded before eigensolution.
- For structural analysis, the auto-omit operations take place for the tridiagonal methods. See the XREAD Section for details.

There are also operations for ACMS and indexing of output tables present in MODERS. They can also be found in many other SubDMAPs. Their description is beyond the scope of this manual, but briefly, ACMS (Automated Component Mode Synthesis) is based on breaking a large model into superelements automatically, solving it in pieces, then re-assembling the pieces so that the output appears to be from a conventional model. Some of the partitioning burden that was placed on the user in manual superelement analysis proved difficult to automate, so more automatic alternate methods have been developed for ACMS that require less user input. The automatic support system described in the RESVEC SubDMAP section is a case in point. Most of the ACMS innovations are moved to conventional analysis, over time, after they have been tested in field usage.

Table indexing is a method to allow direct access to data in large sequential files, to speed up interactive analysis. This work is being done as part of the API project.

Calculation of Residual Vectors in MSC Nastran

Modal solutions are commonly used to reduce large-size models to relatively smaller representations that include the important dynamics of the original model. In addition to reducing the model size, modal reduction has the added advantage of producing diagonal mass and stiffness matrices for solution calculation. Although modal reduction may capture most of the dynamic response of the structure, the static response may not be complete. This ‘error’ is due to modal truncation of higher frequency modes that may contribute ‘statically’ to the total response. Residual vectors are used to improve the results of modal solutions by attempting to account for the response of these higher frequency modes.

Residual vectors can be determined from any set of base vectors. Base vectors are the raw material used to generate the residual vectors in MSC Nastran. Although any vector can be used as a residual vector, as long as it is partially independent of the modal vectors, it may produce coupling with the modal vectors when added to the modes for matrix reduction. This coupling destroys the diagonal properties of the reduced mass and stiffness matrices.

To ensure that the residual vector used in the reduction process result in diagonal matrices, and better results, the following steps are performed.

1. Ensure that loads are linearly independent with the modal inertial forces.
2. Determine base vectors from static response due to loads.
3. Ensure that base vectors are linearly independent.
4. Orthogonalize the base vectors with respect to the modal vectors to produce residual vectors. These vectors will result in diagonal mass and stiffness matrices.

The closer the base vectors approximate the actual deformation due to the load, the better the residual vectors and the better the results. For this reason, base vectors in MSC Nastran are determined from the static response of the structure to the following loads:

- Inertial forces due to rigid-body motion
- Applied loads
- Structural, viscous, and inertial forces due to enforced motion
- Forces at user specified discrete degrees of freedom (RVDOFi entries)
- Discrete damping forces due to viscous elements (CDAMPi and CVISC entries)

A new case control command, RESVEC, allows the user to specify the loads used for residual vector calculation. The format for the RESVEC is as follows:

$$\text{RESVEC}\left[\left(\begin{bmatrix}\text{INRLOD}\\ \text{NOINRL}\end{bmatrix},\begin{bmatrix}\text{APPLOD}\\ \text{NOAPPL}\end{bmatrix},\begin{bmatrix}\text{RVDOF}\\ \text{NORVDO}\end{bmatrix},\begin{bmatrix}\text{DMPLOD}\\ \text{NODMP}\end{bmatrix}\right)\right]=\left\{\begin{array}{c}\text{SYSTEM/NOSYSTEM}\\ \text{COMPONENT/NOCOMPONENT}\\ \text{BOTH or YES}\\ \text{NO}\end{array}\right\}$$

Examples:

RESVEC=SYSTEM
RESVEC(NOINRL)=COMPONENT
RESVEC=NO

Describer	Meaning
INRLOD/ NOINRL	Controls calculation of residual vectors based on inertia relief. (Default=INRLOD)
APPLOD/ NOAPPL	Controls calculation of residual vectors based on applied loads. (Default=APPLOD)
RVDOF/ NORVDOF	Controls calculation of residual vectors based on RVDOFi entries (Default=RVDOF)
DMPLOD/ NODMP	Controls calculation of residual vectors based on viscous damping.

Describer	Meaning
SYSTEM/ NOSYSTEM	Controls calculation of residual vectors for system (a-set) modes. For NOSYSTEM, describers inside the parentheses are ignored. See Remark 2 for default.
COMPONENT/ NOCOMPONENT	Controls calculation of residual vectors for component (superelement or o-set) modes. For NOCOMPONENT, describers inside the parentheses are ignored. See Remark 2 for default.
BOTH or YES	Requests calculation of residual vectors for both system modes and component modes. See Remark 2 for default.
NO	Turns off calculation of residual vectors for both system and component modes and describers inside the parentheses are ignored. See Remark 2 for default.

The inertial forces due to the normal mode shapes are removed from the loads used for residual vector calculation using the following:

$$\{\bar{P}\} = \{P\} - [M][\phi]([\phi][M][\phi])^{-1}[\phi]\{P\}$$

The modified loads are applied to the structure to determine the deformations that serve as the base vectors for residual vector calculation.

$$[K]\{V_{load}\} = \{\bar{P}\}$$

Additionally, for transient response, initial conditions are also added to the set of base vectors. Before the initial condition vectors are added, any mode shape content is removed.

$$\{\bar{V}_{initial}\} = \{V_{initial}\} - [\phi]([\phi][M][\phi])^{-1}[\phi][M]\{V_{initial}\}$$

The initial condition vectors are appended to the load vectors and the complete set of base vectors are made linearly independent. This process removes vectors that are linear combinations of other vectors in the base vector set. The remaining base vectors are then made orthogonal with respect to the mass and stiffness matrices. The resulting orthogonal vectors are the residual vectors.

The residual vectors are appended to the modes and the reduced mass and stiffness matrices are modified to include the residual vector additions.

$$[\bar{\phi}] = [\phi_{\text{modes}} | \phi_{\text{residual}}]$$

$$[\bar{M}] = [\bar{\phi}][M][\bar{\phi}]$$

$$[\bar{K}] = [\bar{\phi}][K][\bar{\phi}]$$

The augmented set of modes/residual vectors is used to reduce the damping, applied loads, and any external matrix input.

Residual vector calculation is available for all modal solutions in MSC Nastran. They can be calculated for the residual structure and superelements to improve the analysis results.

New C-Set and R-Set Reduction with Residual Vectors

In previous version of MSC Nastran, the o-set (interior dofs) motion associated with the b-set, c-set, and r-set (boundary dofs) was generated from the static reduction of the stiffness matrix.

$$[G_{ot}] = [G_{ob} | G_{ow}] = [-K_{oo}]^{-1}[K_{ob} | K_{ow}]$$

where the w-set = c-set + r-set.

The boundary motion determines the response of the o-set.

$$\{u_0\} = [G_{ob} | G_{ow}] \begin{Bmatrix} u_b \\ u_w \end{Bmatrix}$$

For dynamic analyses, the o-set motion can be augmented by the use of component modes to improve the response estimate. The motion of the component modes is represented by the q-set.

$$\{u_0\} = [G_{ob} | G_{ow} | \bar{\phi}_{oq}] \begin{Bmatrix} u_b \\ u_w \\ \xi_q \end{Bmatrix}$$

where the component modes, $\bar{\phi}_{oq}$, are determined from modes of the v-set (v-set = o-set + c-set + r-set).

The modes of the v-set are calculated in the standard fashion.

$$(-\lambda[M_{vv}] + [K_{vv}])\{\phi_v\} = \{0\}$$

The motion of the boundary grids are then removed from the calculated modes

$$\{\bar{\phi}_o\} = \{\phi_o\} - [G_{ow}]\{\phi_w\}$$

where

$$\{\phi_v\} = \begin{Bmatrix} \phi_o \\ \phi_w \end{Bmatrix}$$

$\{\phi_v\}$ are modes of the v-set degrees of freedom.

$\{\bar{\phi}_o\}$ are the v-set modes with the w-set boundary motion removed.

The previous operation decouples the w-set boundary motion $[G_{ow}]$ from the v-set modes $[\phi_v]$. This decoupling allows the boundary and component mode dofs to move independently and the total motion of the structure to be written in terms of the a-set (b-set + w-set + q-set).

$$\begin{Bmatrix} u_b \\ u_w \\ u_o \end{Bmatrix} = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ G_{ob} & G_{ow} & \bar{\phi}_{oq} \end{bmatrix} \begin{Bmatrix} u_b \\ u_w \\ \xi_q \end{Bmatrix}$$

If residual vectors are requested, they are calculated using the o-set matrices and appended to the component modes.

The new w-set processing follows a slightly different path. The procedure is based on a method suggested by Dr. Arya Majed and Ed Henkel. This method is the new default procedure for c-set reduction. The previous method for c-set reduction can be selected by setting PARAM,MHRED to 'NO'.

The new c-set procedure uses static reduction to determine the motion oft the v-set due to boundary b-set motion.

$$\{u_v\} = [G_{vb}]\{u_b\}$$

where

$$[G_{vb}] = [-K_{vv}]^{-1}[K_{vb}]$$

As previously was done, modes of the v-set are also calculated.

$$(-\lambda[M_{vv}] + [K_{vv}])\{\phi_v\} = \{0\}$$

If residual vectors are requested, they are calculated using the v-set size matrices and appended to the v-set modes.

At this point, the b-set boundary vectors and the v-set modes can describe the motion of the structure.

$$\begin{Bmatrix} u_b \\ u_v \end{Bmatrix} = \begin{bmatrix} I & 0 \\ G_{vb} & \phi_{vi} \end{bmatrix} \begin{Bmatrix} u_b \\ \xi_q \end{Bmatrix}$$

or splitting the v-set into its o-set and w-set partitions,

$$\begin{Bmatrix} u_b \\ u_w \\ u_o \end{Bmatrix} = \begin{bmatrix} I & 0 \\ G_{wb} & \phi_{wq} \\ G_{ob} & \phi_{oq} \end{bmatrix} \begin{Bmatrix} u_b \\ \xi_q \end{Bmatrix}$$

In the above representation, the w-set dofs are not independent dofs, but are dependent on b-set and q-set motion. In order to make the w-set independent, vectors that associate o-set motion with individual w-set dofs must be found. Once these vectors are found, the components are removed from the b-set and q-set motion resulting in w-set vectors that are independent.

The w-set vectors are determined using the flexibility of the v-set.

$$\begin{bmatrix} Y_{oo} & Y_{ow} \\ Y_{wo} & Y_{ww} \end{bmatrix} \begin{Bmatrix} P_o \\ P_w \end{Bmatrix} = \begin{Bmatrix} u_o \\ u_w \end{Bmatrix}$$

If loads are applied only to the w-set, the displacements are dependent only on P_w.

$$\begin{bmatrix} Y_{ow} \\ Y_{ww} \end{bmatrix} \{P_w\} = \begin{Bmatrix} u_o \\ u_w \end{Bmatrix}$$

Vectors that associate o-set motion with individual w-set dofs can be found by setting $P_w = Y_{ww}^{-1}$.

$$\begin{bmatrix} Y_{ow} \\ Y_{ww} \end{bmatrix} [Y_{ww}^{-1}] = \begin{bmatrix} Y_{ow} Y_{ww}^{-1} \\ I_{ww} \end{bmatrix}$$

The above vectors are used to make the w-set an independent set.

$$\begin{Bmatrix} u_b \\ u_w \\ u_o \end{Bmatrix} = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ G_{ob} - Y_{ow} Y_{ww}^{-1} G_{wb} & Y_{ow} Y_{ww}^{-1} & \phi_{oq} - Y_{ow} Y_{ww}^{-1} \phi_{wq} \end{bmatrix} \begin{Bmatrix} u_b \\ u_w \\ \xi_q \end{Bmatrix}$$

The flexibility matrices Y are determined from displacements due to unit forces on the w-set.

$$[K_{vv}]^{-1} \begin{bmatrix} 0 \\ I_{ww} \end{bmatrix} = [\bar{Y}_{vv}]$$

The modal flexibility is removed from the displacement response to determine the flexibility matrices used in .

$$[Y_{vv}] = [\bar{Y}_{vv}] - [\phi_{vq}]([\phi_{vq}]^T [M_{vv}][\phi_{vq}])^{-1} [\phi_{vq}]^T [M_{vv}][\bar{Y}_{vv}]$$

$$[Y_{vv}] = \begin{bmatrix} Y_{ow} \\ Y_{ww} \end{bmatrix}$$

Static Solution for Shape Functions

If the structure contains rigid body modes it is not stable for static analysis. This problem is solved by one of two methods:

Manual SUPORT (SubDMAP SEKRRS)

If the user supplies SUPORTi entries, the r-set DOFs it produces are used as “temporary” SPCs, applied to allow solving for the static shapes, but removed before computing modes. Partitions involving “r” are discarded.

$$K_{aa} \geq \begin{bmatrix} K_{ll} & K_{lr} \\ K_{rl} & K_{rr} \end{bmatrix}$$

$$P_{a2} \geq \begin{bmatrix} P_{l2} \\ P_{r2} \end{bmatrix}$$

$$K_{ll} \cdot ul \leq P_{l2}$$

$$u_{a2} = \begin{bmatrix} u_{l2} \\ 0_{r2} \end{bmatrix}$$

Auto SUPORT (SubDMAPs MODEFSRS, SEMR3) ACMS Only

If no SUPORTi entries or an inadequate set are supplied for constraining all rigid body modes, and rigid body modes are found when solving $[K_{aa} - \lambda \cdot M_{aa}] \cdot \phi_a = 0$, the rigid body modes ϕ_r are separated from the flexible modes based on their natural frequency being smaller than a user parameter, PARAM, FZERO, (default = 1.E-4 Hz). ϕ_r is used to make constraint equations MP_{ar},

$$MP_{ar} = M_{aa} \cdot \phi_r$$

These equations are appended to K_{aa} using Lagrange Multiplier Techniques, and solved directly,

$$\begin{bmatrix} K_{aa} & MP_{ar} \\ MP_{ar}^T & O_{rr} \end{bmatrix} \begin{bmatrix} u_{a2} \\ -q_{r2} \end{bmatrix} = \begin{bmatrix} P_{a2} \\ O_{r2} \end{bmatrix}$$

The matrix on the left is named KLL in the subDMAP. It is similar to the KLL produced above by partitioning the larger K_{aa} matrix in that it is stable for static solution, but differs in that it is larger than K_{aa} by the addition of the constraint equations. These equations enforce the condition expressed by the bottom row,

$$MP_{ar}^T \cdot u_{a2} = 0$$

This is a statement that the c.g. of the structure is at rest on average, but the structure is free to deflect about it. This approach will produce different u_{a2} vectors than those produced by the partitioned solution. In the partitioned solution one set of points is grounded, and all loads are transmitted through the structure to them. A load on an explicit r-set point, for example, will go directly to ground, and will cause zero deflection. This implies that some skill is required in selecting DOFs for SUPORTi entries. With the LMT approach the constraints to ground are distributed, so that loads applied anywhere will tend to load the entire structure.

Solution for Trial Vectors by Static Analysis

Inertia relief analysis is NOT performed, despite the presence of SUPORTi entries, usually used in the context of inertia relief. In this context, SUPORTi data or its equivalent for the auto-support feature is used for SPC-type operations. In CMS at present, K_{oo} may not contain rigid body modes. There is no provision to constrain them with SUPORT-type inputs. DOFs may be removed from the o-set to make it stable. For example, in superelement analysis, an interior point active in the rigid body mode, which by default is in the o-set, can be made an exterior point. This changes it to the a-set, which is held fixed when computing CMS modes. If the rigid body modes still exist downstream, they can be constrained in residual structure calculations.

The u_{a2} trial vectors, or “raw static shapes”, as they are called in the SubDMAP comments, may not be linearly independent of the mode shapes, and are certainly not orthogonal to the mode shapes or each other. An orthogonalization process is now used to insure that the resulting shapes are orthogonal to the mode shapes and each other. Shapes that have little information after this process are discarded.

Orthogonalization

The trial vectors u_{a2} may not be linearly independent of each other or the mode shapes. They are processed to find a set of combined mode shapes and linear combinations of the trial shapes that are all orthogonal to each other with respect to the mass matrix. This reduces the risk of ill-conditioning when using these matrices for solution later.

The trial vectors are appended to ϕ_a to form the combined set ϕ_{ua1}

$$\phi_{ua1} = \left[\phi_a \middle| u_{a1}\right]$$

An optional pre-sweeping of trial vectors is done when requested by PARAM, RESVSO, YES. The derivation of the sweeping equation is similar to that for sweeping loads above, with the equation for x, the vector of linear combination coefficients, being

$$x = M_{i-1} \cdot \phi_a \cdot M_{aa} \cdot u_{a2s}$$

The swept displacement shape vectors ϕ_{ua1} are then determined by eliminating x,

$$\phi_{ua1} = u_{a2s} - \phi_{a2} \cdot M_i^{-1} \cdot \phi_{a2} \cdot M_{aa} \cdot u_{a2s}$$

If the optional sweeping is not selected ϕ_{a1} is merely equal to ϕ_{a2}.

The u_{a1} vectors are mass-normalized into ϕ_{ua} such that the diagonal terms of the product on the left side are unity

$$\phi_{ua} \cdot M_{aa} \cdot \phi_{ua} = M_{zz}$$

$$\text{diag}(M_{zz}) = I_{zz}$$

M_{zz} is generally fully coupled at this point. It is decomposed with an LTL factorization,

$M_{zz} = L \cdot D \cdot L$, where L is lower triangular and D is diagonal.

The ratio of D over the diagonal terms of M_{zz} is calculated in the vector R,

$$R = \left\{ \frac{\text{diag}(M_{zz})}{\text{diag}(D)} \right\}$$

Rows of R whose value exceeds PARAM, RESVRAT (default=1.e8) indicate columns of ϕ_{ua} that have little new content, and are discarded to produce U_{a3} and M_{zz3}. M_{zz3} is diagonal in the region of the eigenvectors, but coupled in the region of the auxiliary shapes. It is then re-named as M_{zz}.

The generalized stiffness corresponding to M_{zz} is calculated,

$$K_{zz} = u_{a3} \cdot K_{aa} \cdot u_{a3}$$

The eigensolution of the z-set is obtained,

$$[K_{zz} - \lambda_z \cdot M_{zz}] \cdot \phi_z = 0$$

The eigenvectors ϕ_z of this reduced basis are used as a transformation to orthogonalize U_{a3},

$$\phi_a = \phi_{ua} \cdot \phi_z$$

The eigenvalue table of the combined system, λ_z, replace the prior eigenvalue table that contained modes only. λ_z includes both structural modes and the effects of orthogonalized residual flexibility vectors.

Model Partitioning in the MODEFSRS subDMAP

MODEFSRS is used in the residual structure to separate the fluid and structural components of the a-set. The modes of these components are computed independent of one another to find ϕ_{sa} and ϕ_{fa}, the structural and fluid modes, respectively. Once computed, residual flexibility effects are added with calls to the RESVEC SubDMAP.

The eigenvectors with residual flexibility effects appended to them ϕ_a are then merged together to produce one eigenvector matrix,

$$\phi_a = \begin{bmatrix} \phi_{sa} & 0 \\ 0 & \phi_{fa} \end{bmatrix}$$

For the cases where only fluid or only structural modes are requested, the identity matrix is substituted for the eigenvector matrix that is not calculated.

Complex Eigenvalue Analysis in SubDMAP CEIGRS

Complex eigenvalues and the associated eigenvectors are calculated in CEAD (Complex Eigenvalue Analysis -- Displacement) using the inverse power method, the determinant method, the Hessenberg method, or the Lanczos method, as requested by the user on the EIGC Bulk Data entry. For direct complex eigenvalue analysis and if PARAM,ASING = 0 (default) then null rows and columns are discarded from $[K_{dd}]$, $[M_{dd}]$, and $[B_{dd}]$ to form $[K^d_{xx}]$, $[M^d_{xx}]$, and $[B^d_{xx}]$.

In the case of a direct formulation CEAD extracts the eigenvalues from the following equation.

$$[M^d_{xx}p^2 + B^d_{xx}p + K^d_{xx}]\{u^d_x\} = 0$$

$\{u^d_x\}$ is then augmented with null rows to form $\{u_d\}$.

In the case of a modal formulation the following equation is used:

$$[M_{hh}p^2 + B_{hh}p + K_{hh}]\{u_h\} = 0$$

CEAD also normalizes the eigenvectors according to one of the following user requests:

- Unit magnitude of a selected coordinate (“POINT”).
- Unit magnitude of the largest component (“MAX”).

as specified on the EIGC Bulk Data entry.

The complex Lanczos method is recommended for large problems with sparse matrices. The HESS method is best suited to small dense matrices, typical of small models or those typical in a modal formulation. The INV method is obsolete, and is retained as an alternate method when all else fails.

Frequency Response and Random Analysis in SubDMAP FREQRS

The frequency-dependent loads are generated in FRLG (Frequency Response Load Generation) from the `RLOAD1` and `RLOAD2` Bulk Data entries. Constraints are applied to reduce the loads to the d-set, and in the case of a modal formulation, the loads are transformed to the h-set.

The solution for the response is performed in FRRD1 or FRRD2 (Frequency Response Solution). In the case of a direct formulation the following equation is solved.

$$[-M_{dd}\omega^2 + iB_{dd}\omega + K_{dd}]\{u_d\} = \{P_d\}$$

and in the case of a modal formulation, the following equation is used in:

$$[-M_{hh}\omega^2 + iB_{hh}\omega + K_{hh}]\{u_h\} = \{P_h\}$$

The arithmetic used in the solution may be real or complex and the solution procedure may be symmetric or unsymmetric. The choice of arithmetic and solution procedure is made by the program depending on the form of the dynamic matrices.

Power spectral density functions and autocorrelation functions are computed in RANDOM (Random Analysis) according to the information on RANDPS and RANDT1 Bulk Data entries. Random analysis calculations are made for selected displacements, loads, single point constraint forces, element stresses and element forces. The selection of items for random analysis is made in the XY-output request part of the Case Control Section.

Transient Response in SubDMAPs SEDTRAN and SEMTRAN

The linear time-dependent loads are generated in TRLG (Transient Response Load Generation) from the `TLOAD1` and `TLOAD2` Bulk Data entries. The nonlinear loads are generated from the NOLINi Bulk Data entries. Constraints are applied to reduce the loads to the d-set, and in the case of a modal formulation, the loads are transformed to the h-set.

The solution for the response is performed in TRD1 (Transient Response Solution). In the case of a direct formulation the following equation is integrated over the time periods specified via the TSTEP Bulk Data entry in subDMAP DTRANRS:

$$[M_{dd}p^2 + B_{dd}p + K^x_{dd}]\{u_d\} = \{P_d\} + \{P^{nl}_d\}$$

in order to determine the displacement velocity and acceleration response.

In the case of a modal formulation, the following equation is used in subDMAP MTRANRS:

$$[M_{hh}p^2 + B_{hh}p + K_{hh}]\{u_h\} = \{P_h\} + \{P^{nl}_h\}$$

If there are no direct input matrices and no nonlinear loads, an analytical integration procedure is used rather than numerical integration. The numerical integration procedure may use either symmetric or unsymmetric routines. The choice is made by the program depending on the form of the dynamic matrices.

Buckling Analysis in SubDMAP MODERS

In linear buckling, a static load is applied to the structure and the following eigenvalue problem is solved subsequently,

$$[K_{aa} + \lambda_i K^d_{aa}]\{\phi_i\} = 0$$

where $[K_{aa}]$ is the linear stiffness, $[K^d_{aa}]$ is the differential stiffness, λ_i are the eigenvalues and $\{\phi_i\}$ are the eigenvectors (buckling modes). The critical buckling loads are

$$\{P^{crit}_i\} = \lambda_i\{P_a\}$$

where $\{P_a\}$ is the vector of applied static loads including reaction forces.

The basic assumption in linear buckling is that the differential stiffness $[K^d_{aa}]$ is proportional to the applied load $\{P_a\}$. The assumption implies a linear material law and small deformations. For buckling in nonlinear problems see Additional Topics, 511.

The linear stiffness $[K_{aa}]$ is derived from the variation of the stresses, the differential stiffness $[K^d_{aa}]$ is derived from the variation of the strains keeping the stresses constant.

The Case Control Section must contain at least two subcases. A static loading condition is applied in the first subcase. The load is defined with a LOAD, TEMP(LOAD), or DEFORM Case Control command. The load may also be defined with enforced deformations on the SPC Bulk Data entries. In the second subcase, a METHOD command must appear to solve the eigenvalue problem. The Lanczos method (EIGR Bulk Data entry) is the recommended eigenvalue solver for linear buckling.

Linear buckling is available in the following solution sequences.

SOL 105	Linear buckling in statics with superelements, including original design sensitivity (DSA).

The linear buckling capability is available for the following elements: CONROD, CROD, CTUBE, CBAR, CBEAM, CBEND, CQUAD4, CTRIA3, CTRIA6, CSHEAR, CHEXA, CPENTA, AND CTETRA.

The following recommendations apply to linear buckling analysis.

- Use at least five elements per half sine wave to get reasonable results in the buckling load.
- For buckling of 3-D shell structures, use PARAM,K6ROT,100. to assign a stiffness to the sixth degree of freedom. The default is PARAM,K6ROT,0.

The following restrictions apply to linear buckling analysis.

- Offsets should not be used in beam, plate or shell elements. The buckling loads for structures with offsets are incorrect.
- Follower force effects are not included in the differential stiffness. The calculated buckling loads are incorrect. The Bulk Data entries FORCE1, FORCE2, MOMENT1, MOMENT2, PLOAD, PLOAD2, PLOAD4, RFORCE, TEMP, TEMPD, TEMPP1, TEMPP3, TEMPRB describe follower forces.

Superelement Analysis

The theoretical aspects of superelement analysis are discussed in the MSC Nastran Superelements and Modules User's Guide. Superelements are mathematically equivalent to substructures. The major difference is in a more convenient interface for the user and in the ability to process a much broader class of problems than is practical for most other substructuring capabilities.

There are now two distinct paths through superelements in the solution sequences. There are also two major types of superelements. When a BEGIN SUPER entry is present the program uses the more modern SEP1X module to make the SEMAP table used to control partition of superelements. When there is a BEGIN BULK entry but no BEGIN SUPER entries a parallel path using the older SEP1 module is used instead. While the newer module supports most features of both part and grid list superelements, it does not yet have some of the advanced features supported in the older module. When starting a new project you should decide at modeling initiation which path is most suitable because transitioning from one to another can be painful, particularly when the model is in development. If one starts on the old path, for example, and then adds a new feature that causes the program to automatically switch to the new path, some of your modeling input may be invalidated. This may cause considerable, confusing, and unnecessary work to transition to the new path.

The short answer is that the new path should be the best path for most projects, based on the appeal of its advanced features and the likelihood that future improvements in MSC Nastran may require use of the new path. The new path is forced by replacing the BEGIN BULK entry with a BEGIN SUPER entry in your Main Bulk Data file. It should by the first entry in your bulk data. It allows grid list superelements, but they need not be present.

The exception for the path chosen is when your project requires the use of features available in the old path but not yet enabled in the new path. Some of these features may be used in the new paths by special avoidance techniques, but these techniques can be cumbersome. It may be more convenient to stay with the older methods until the newer methods catch up.

Terminology

- Newest Option: The partitioned bulk data file superelements (parts) are the form introduced in Version 69. They are sometimes called substructures. A major feature of parts is ID independence. That is, element IDs and grid points IDs may be duplicated across different parts, although they must be unique within a part.
- Intermediate: The Main Bulk Data Superelements (superelements) introduced in Version 47 are sometimes called grid list superelements (list s.e.s). When used in the new path, they have more restrictive rules than in the old path.
- Oldest Path: List s.e.s only are used in the old path. There are no formal names at present to differentiate the intermediate and old flavors. Some advanced features are available only in this path, at present.

A subtle difference between the two paths is that the new path will always use a single level configuration for list s.e.s if there is no SETREE-type entry. It does this by defining new boundary points in the residual structure when some are needed but not supplied by the user. The old path automatically converts to a multilevel configuration when required, regardless of the presence of DTI, SETREE entries. It has no ability to generate new boundary points. This type of change when switching from the old path to the new, on purpose or by accident can be confusing.

Criteria for Selecting Part Versus List Superelements

Each of the two major forms has overwhelming advantages in certain contexts. Some of these advantages are of a temporary nature, as the rest of the technology in supporting FE tools catch up to the newest methods. However, other relative advantages are due to the inherent difference in concept between the two major branches that will be relevant into the foreseeable future. It is likely that both branches will be maintained and extended because of their unique capabilities. The following criteria should be considered when making a choice:

- Converting an existing non-superelement model to a part configuration is a labor-intensive process at present. No migration aides are provided. Converting it to a list superelement requires addition of only SESET entries, a trivial task by comparison.
- When debugging a new model it is much easier to comprehend and more efficient when it is in a superelement configuration. Errors in one area will not obscure errors in other areas, and the model solved is smaller, leading to faster turnaround. It may be more economical to run it in a non-superelement configuration for production runs after the model is stable and requires no more changes. If it is a list superelement this means merely removing the SESET entries, after which it is automatically an exactly equivalent non-superelement model if advanced features such as those provided by the RELEASE and CSUPEXT entries are avoided. If it is a part superelement there can be a large labor cost for conversation, as all duplicated IDs will need to be changed, duplicate points at boundaries will need to be resolved, etc. A list s.e. model, to make virtue of necessity, enforces rules such as unique IDs when updating the model that make re-conversion to a non-s.e. configuration an almost automatic process.
- When starting a new modeling project where more than one person provides input, and most particularly when more than one organization contributes to the model, the higher degree of data control for part superelements is attractive. There is no need to legislate acceptable ID ranges for each part. It is possible to lock the description of a part into its boundary matrices, in a data base marked with a time stamp, so that there is little questions about who is responsible for the behavior of each part.
- Many GUIs can be used to modify list superelements because the rules for ID independence are the same for GUIs as they are for list s.e.s Part superelements can be modeled in GUIs only one at a time. This makes boundary matching, in particular, cumbersome.

Features Not Supported in the New Path at Present

- Fluid Structure Analysis: This capability is supported in the new path with the “old” path rules:
  a. All fluid elements must be in a special list superelement, and/or in the residual structure
  b. This special superelement may have fluid elements only; structural elements or grid points are not allowed
  c. One or several fluid cavities may be in the fluid superelement
  d. Fluid elements may be connected to structural elements in list superelements, or in the residual structure.
  e. Elements in part superelements may not be connected to fluid elements.
- p-Elements: Not supported at all. It may be possible to segregate the portions of the model containing p-elements into external superelements, but this will be a labor-intensive process. No documentation is available on this topic at present.

- RELEASE Entries: Allowed only in the old path. There are semi-heroic modeling techniques that may be used to simulate this feature in the new path, but they are not documented.
- SExSET Entries: Most set entries for non-superelement models such as ASET have synonyms for use in list s.e.s such as SEASET. The synonym is not accepted in PARTs, but the simpler format of ASET, for example, is accepted.

Features Not Supported in the Old Path

In general, features associated with automatic boundary control are not supported. The QRG is not explicit on this issue. A list of entries not supported in the old path is:

SETREE, SENQSET, SEBNDRY, SEBULK, SECONCT, SEEXCLD, SELABLE, SELOC, SEMPLN

The DTI, SETREE entry is used in place of SETREE. The action of the SEELT entry is somewhat different. For the new path, the SEMAP will be changed to account for the action of SEELT entries. For the old path, it is assumed that the element being moved is one that can be moved without changing the SEMAP, such as an element attached only to boundary points. If the SEELT requires change of the SEMAP the program processes conventionally until it encounters the s.e. with the "illegal" SEELT entry, then gives a fatal error exit stating that the connectivity of the superelement is incorrect.

Check List for Transitioning an Old Model to the New Path

- Replace the BEGIN BULK entry with a BEGIN SUPER entry.
- Check the SEMAP table from an old run to determine if it is in a multilevel state. When it is, add SETREE entries that define the tree in the old run, or convert it to a single level model if multilevel is not essential. If you have many multilevel models to convert, there is a parameter that will request that SETREE entries be made from the SEMAP, and placed in the.pch (punch) file.
- Discard all RELEASE and CSUPEXT entries and p-elements. If you cannot find reasonable replacements for them you may be forced to stay in the old path.

Decide if this is the right time to convert to a PART model, or whether the list s.e. capability in the new path is more advantageous, based on the discussion above.

Solution sequences that support superelement analysis are listed in Output Description, 340.

The Case Control commands for superelement processing selection all begin with the letters SE (SEMG, SELG, SEKR, SEMR, SELR, SEALL, and SEEX). The SEFINAL command is used to control the order of processing of the superelements. A subcase structure is required that defines constraints, loads, and data recovery options for each superelement. Each subcase includes the Case Control command SUPER in order to relate the subcase to a particular superelement and load condition. Examples of Case Control (including plot requests) are given in the MSC Nastran Superelements and Modules User's Guide. A set of plotting commands is required for each superelement to be plotted, separated by the plotter commands SEPLOT and SEUPPLOT.

Each superelement is given an identification number (SEID). The user assigns each grid point to a superelement on a `GRID` or SESET Bulk Data entry. Identical image, mirror image, and external superelements are defined by the CSUPER and optional SEQSEP Bulk Data entries. The SEELT Bulk Data entry may be used to change the superelement membership of boundary elements. The RELEASE Bulk Data entry may be used to disconnect selected degrees of freedom from exterior grid points of a superelement. Use of Parameters, 32 lists all PARAMeters used in superelement analysis.

The mesh generator option, called by MESH on the NASTRAN statement, supports the superelement capability fully, in that all GRID entries generated for a field can be given the same SEID, and the SESET Bulk Data entry can change these SEIDs at boundaries. The automatic grid point resequencing features are fully implemented for superelement models.

The superelement partitioning module passes on resequencing data from SEQGP Bulk Data entries only for grid points of a superelement which are not exterior points for any superelement. This class of grid points can be determined from the SEMAP table printed by the program, but may change if the order of processing is changed. The automatic resequencing option processes each superelement separately.

Component mode synthesis uses SEBSETi, SECSETi, SESUP and SEQSETi entries to define the boundary condition when computing component modes. Uncoupled solutions for statics and modes can be obtained by use of PARAM,FIXEDB,-1.

Superelement Bulk Data entries may be specified in nonsuperelement solution sequences, but those entries pertaining to the superelement capability (e.g., the CSUPER entry) are ignored, as well as some fields on other entries (e.g., Field 9 on the GRID entry). However, better automatic resequencing may be possible if the SEID data is removed.

Superelement Processing in the Structured Solution Sequences

An overview of the superelement processing in SOLutions 101-200 is shown in Figure 7-3. Phase 0 performs the preparation and bookkeeping function. The input data is stored on the database and scanned to prepare the SEMAP table. This table lists the superelement to which each Bulk Data entry is to be applied. Phase 0 also partitions out the data that defines each superelement and performs automatic restart processing. In Phase I this data is used to generate the boundary matrices to be assembled when combining the superelements into the total structure. Phase II performs calculations done only on the residual structure (the superelement processed last), such as transient response calculations or eigensolutions. Phase III expands boundary solutions to the interior grid points of the superelement, performs the data recovery operations for internal forces, stresses, and strain energy, and outputs these quantities. Phase IV combines plot vectors from upstream superelements to provide deformed structural plots for all or part of the structure.

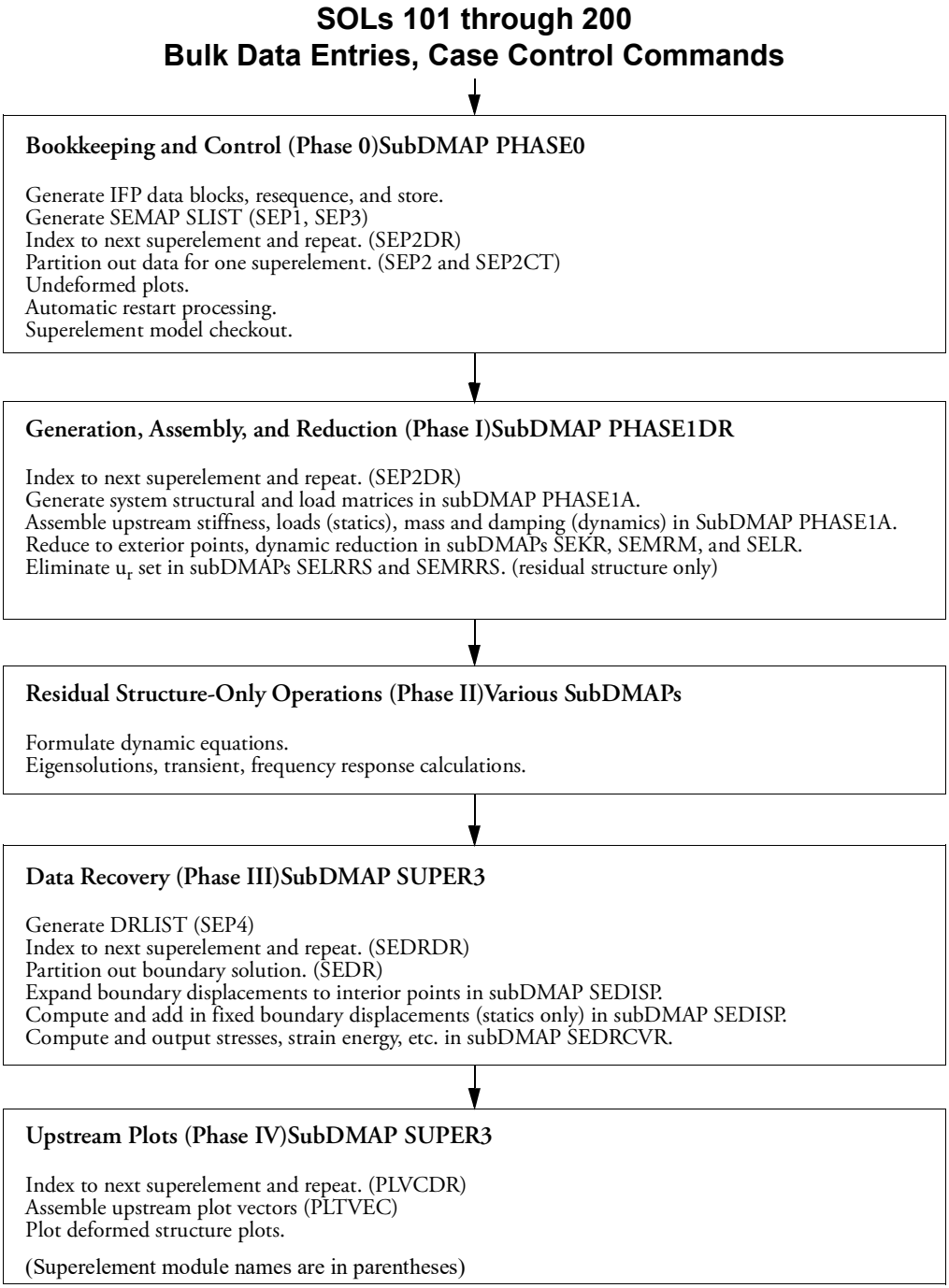

SOLs 101 through 200
Bulk Data Entries, Case Control Commands
Bookkeeping and Control (Phase 0)SubDMAP PHASE0
Generate IFP data blocks, resequence, and store.
Generate SEMAP SLIST (SEP1, SEP3)
Index to next superelement and repeat. (SEP2DR)
Partition out data for one superelement. (SEP2 and SEP2CT)
Undeformed plots.
Automatic restart processing.
Superelement model checkout.
Generation, Assembly, and Reduction (Phase I)SubDMAP PHASE1DR
Index to next superelement and repeat. (SEP2DR)
Generate system structural and load matrices in subDMAP PHASE1A.
Assemble upstream stiffness, loads (statics), mass and damping (dynamics) in SubDMAP PHASE1A.
Reduce to exterior points, dynamic reduction in subDMAPs SEKR, SEMRM, and SELR.
Eliminate u_r set in subDMAPs SELRRS and SEMRRS. (residual structure only)
Residual Structure-Only Operations (Phase II)Various SubDMAPs
Formulate dynamic equations.
Eigensolutions, transient, frequency response calculations.
Data Recovery (Phase III)SubDMAP SUPER3
Generate DRLIST (SEP4)
Index to next superelement and repeat. (SEDRDR)
Partition out boundary solution. (SEDR)
Expand boundary displacements to interior points in subDMAP SEDISP.
Compute and add in fixed boundary displacements (statics only) in subDMAP SEDISP.
Compute and output stresses, strain energy, etc. in subDMAP SEDRCVR.
Upstream Plots (Phase IV)SubDMAP SUPER3
Index to next superelement and repeat. (PLVCDR)
Assemble upstream plot vectors (PLTVEC)
Plot deformed structure plots.
(Superelement module names are in parentheses)

Figure 7-3 Structured Solution Sequence Flow Chart

The input file processor operations are carried out in a conventional manner. The grid point resequencer module resequences each superelement individually.

Some of the geometry processor operations of Geometry Processing in SubDMAP PHASE0, 363 are carried out to prepare tables of all grid points in the entire model, useful for generation of the SEMAP table. After the SEMAP is available, the SLIST table is assembled. This list controls the operations to be performed in that particular run for generating boundary matrices, as selected with Case Control commands. The GP4 module generates constraint data in the form of the USET table, as described in Geometry Processing in SubDMAP PHASE0, 363. A separate USET table is generated for each superelement. It is similar to the USET table of the older rigid formats except that the exterior points are automatically placed in the a-set.

Assembly and Reduction (Phase I) in SubDMAP Phase1DR

The following operations may be carried out, as selected by the user with superelement-type Case Control commands. By default, all superelements and the residual structure are given these operations serially. If requests for the SEMA (or SEMR) operation are made on an upstream superelement and the residual structure, all superelements in the tree between the selected superelement and the residual structure will also be given the SEMA operation.

Superelement Matrix Generation (SEMG) in SubDMAP PhaselA

The element structural matrices (stiffness, mass and damping) are generated and stored. The direct matrix input option (DMIG Bulk Data entry) can be used to add stiffness, mass or damping terms to the system matrices in this operation. These terms are selected by the Case Control commands K2GG, M2GG, and B2GG. The parameters CB1, CB2, CK1, CK2, CM1 and CM2 can be used to scale the structural matrices.

The SEMA module adds in stiffness terms from any upstream superelements. The SEEX Case Control command may be used to eliminate stiffness contributions from selected upstream superelements. The superelements that are upstream are defined by the current SEMAP. The number of rows and columns of the upstream superelements must be consistent with the number of exterior grid points, as defined by the SEMAP In order to complete execution of the SEMA module, the boundary matrices of all upstream superelements must be present in the database, and consistent with the current SEMAP.

The SELA module adds loads from upstream superelements to the load vector for the current superelement. The SEEX Case Control command may be used to eliminate load contributions from upstream superelements. The requirements for conformity of load data in the database with the current state of the SEMAP are similar to those for stiffness, except that only rows, not columns of the load vectors are mapped using data in the SEMAP table. The requirement on columns of all load matrices is that they be equal to the number of columns (number of load condition) in the residual structure. If the number of columns in all superelements and the residual structure is not the same, a user fatal error results. Static loads may be applied to any grid point, whether exterior or interior to a superelement. Dynamic loads may be applied directly to residual structure points. Dynamic load entries may also reference static load sets applied to the superelements via the `LOADSET` Case Control command and LSEO Bulk Data entries.

Superelement Load Generation (SELG) in SubDMAP PHASE1A

The load generation module SSG1 is executed in the static and dynamic response solutions. This operation is not executed in the normal modes and complex eigenvalue solution sequences. As loads may depend on structural matrices (i.e., thermal or inertial loads), an SELG operation must be preceded by an SEMG operation. The resultants of these loads about the grid point listed on the PARAM,GRDPNT Bulk Data entry are output.

Superelement Stiffness Reduction (SEKR) in SubDMAPs SEKR and SEKRRS

The user may change SPCs and/or MPCs between subcases (“boundary condition changes”). The GPSP module identities grid point singularities, and at user's option provides SPCs for singular degrees of freedom. This is followed by the constraint eliminations described in Multipoint Constraint Operations in SubDMAPs SEKR and SEMR2 and SEMRB, 366, Single Point Constraint Operations in SubDMAPs SEKR, SEMR2 and SEMRB, 369, and Constraint and Mechanism Problem Identification in SubDMAP SEKR, 370. If the omitted degrees of freedom contain mechanism-type singularities, the grid and component number of all downstream ends of the mechanism are output in the matrix named MECHS, and a user fatal error stops execution. The approximate fixed-boundary natural frequencies of the superelement are printed in the FAPPROX matrix. The boundary stiffness matrix $[K_{aa}]$ is stored in the database. Mass and damping matrices are not processed in this phase.

SUPORTi Bulk Data entries, which define the r-set, may reference only grid points in the residual structure. The rigid body operations for the stiffness are performed as described in Static Solutions in SubDMAP SEKRRS, 376.

Superelement Load Reduction (SELR) in SubDMAP SELR

This operation processes loads in the static and dynamic response solution sequences. This operation is not executed in the normal modes and complex eigenvalue solution sequences. The static load reduction of Static Condensation in SubDMAPs SEKR and SEMR2, 373 is applied, stopping after generation of the $[P_a]$ matrix for superelements and $[P_l]$ for the residual structure.

Superelement Mass Assembly and Reduction (SEMR) in SubDMAP SEMRM

In the dynamics solution sequences, the constraint operations of Multipoint Constraint Operations in SubDMAPs SEKR and SEMR2 and SEMRB, 366, Single Point Constraint Operations in SubDMAPs SEKR, SEMR2 and SEMRB, 369, and Constraint and Mechanism Problem Identification in SubDMAP SEKR, 370 are applied to the mass and damping matrices. The rigid body operations of Static Solutions in SubDMAP SEKRRS, 376 are applied to the mass of the residual structure only. Dynamic reduction and component mode calculations are performed. This allows a mass-change-only type of restart by use of the SEMG and SEMR options and avoids the costly reduction of the stiffness matrix in the SEKR operation if no stiffness changes are made.

The optional Bulk Data entry PARAM,ERROR,0 affects operations in Phase I. If it is not used, the default action is to terminate the run after the first user fatal error, as in the older rigid formats. If the parameter is used, a user fatal error causes a branch to the end of Phase I for the superelement, and continuation to the generation, assembly, and reduction operations for any remaining superelements. The five SE-type commands may be executed on one run, or may be done separately. The following sequence must be observed if these commands are executed separately:

SEKR after SEMG
SELG after SEMG
SELR after SEKR, SELG, SEMR
SEMR after SEKR

Residual Structure-Only (Phase II)

The operations on the superelements described above are also performed on the residual structure. This is followed by operations unique to the residual structure. Eigensolutions and direct and modal forced response calculations take place at this point. Direct input matrices, when selected by the Case Control commands M2PP, K2PP, and B2PP are constrained and added as in the Formulation of Dynamic Equations in SubDMAP GMA, 385.

Bulk Data entries used for static loads (i.e., FORCE, GRAV, LOAD, etc.) may be used in direct and modal dynamic analysis. They are interchangeable with DAREA Bulk Data entries and may be referenced by LSEQ Bulk Data entries.

Data Recovery Operations (Phase III) in SubDMAP SUPER3

The total solution vectors of the downstream superelement are partitioned to generate the boundary solution vectors for the superelement being processed. The boundary solution is expanded to all interior points of the superelement. The fixed boundary component is computed and added.

If SPC force output is requested, the resultants of the SPC forces about PARAM,GRDPNT or the basic coordinate system origin are also output. The data recovery operations of Static Solutions in SubDMAP SEKRRS, 376 are carried out. (Note that solution sequences involving complex eigenvalue or frequency response analysis may produce complex displacement vectors. GPFDR module output (grid point force balance, element strain energy) may be requested for complex vectors, but only the real component of the vector is processed.) Deformed structure plots of only elements in the superelement are generated here.

Upstream Plots (Phase IV) in SubDMAP SUPER3

The displacement vectors of a superelement and all of its upstream members are combined and plotted at the user's request.

Superelement Processing

An overview of the superelement processing is shown in Figure 7-4. Phase 0 performs the preparation and bookkeeping function. The input data is stored on the database and scanned to prepare the SEMAP table. This table lists the superelement to which each Bulk Data entry is to be applied. Phase I partitions out the data that defines one superelement and processes this data to generate the boundary matrices to be assembled when combining the superelements into the total structure. Phase II performs calculations done only on the residual structure (the superelement processed last), such as transient response calculations or eigensolutions. Phase III expands boundary solutions to the interior grid points of the superelement, performs the data recovery operations for internal forces, stresses, and strain energy, and outputs these quantities. Phase IV combines plot vectors from upstream superelements to provide deformed structural plots for all or part of the structure.

The input file processor operations are carried out in a conventional manner. The grid point resequencer module resequences each superelement individually.

Some of the geometry processor operations of Geometry Processing in SubDMAP PHASE0, 363 are carried out to prepare tables of all grid points in the entire model, useful for generation of the SEMAP table. After the SEMAP is available, the SLIST table is assembled.

This list controls the operations to be performed in that particular run for generating boundary matrices as selected with Case Control options.

Assembly and Reduction (Phase I)

The following operations may be carried out as selected by the user on superelement-type Case Control commands. By default, all superelements and the residual structure are given these operations serially. If requests for the SEMA (or SEMR) operation are made on an upstream superelement and the residual structure, all superelements in the tree between the selected superelement and the residual structure will also be given the SEMA operation.

Superelement Matrix Generation (SEMG)

The element structural matrices (stiffness, mass and damping) are generated and stored. The direct matrix input option (DMIG Bulk Data entry) can be used to add stiffness, mass or damping terms to the system matrices in this operation. These terms are selected by the Case Control commands K2GG, M2GG, and B2GG. The parameters CB1, CB2, CK1, CK2, CM1, and CM2 can be used to scale the structural matrices.

Superelement Load Generation (SELG)

The load generation module SSGI is executed in the statics and dynamic response solutions. This operation is not executed in the normal modes and complex eigenvalue solution sequences. As loads may depend on structural matrices (i.e., thermal or inertial loads), an SELG operation must be preceded by an SEMG operation. The resultants of these loads about the grid point listed on the PARAM,GRDPNT Bulk Data entry are output.

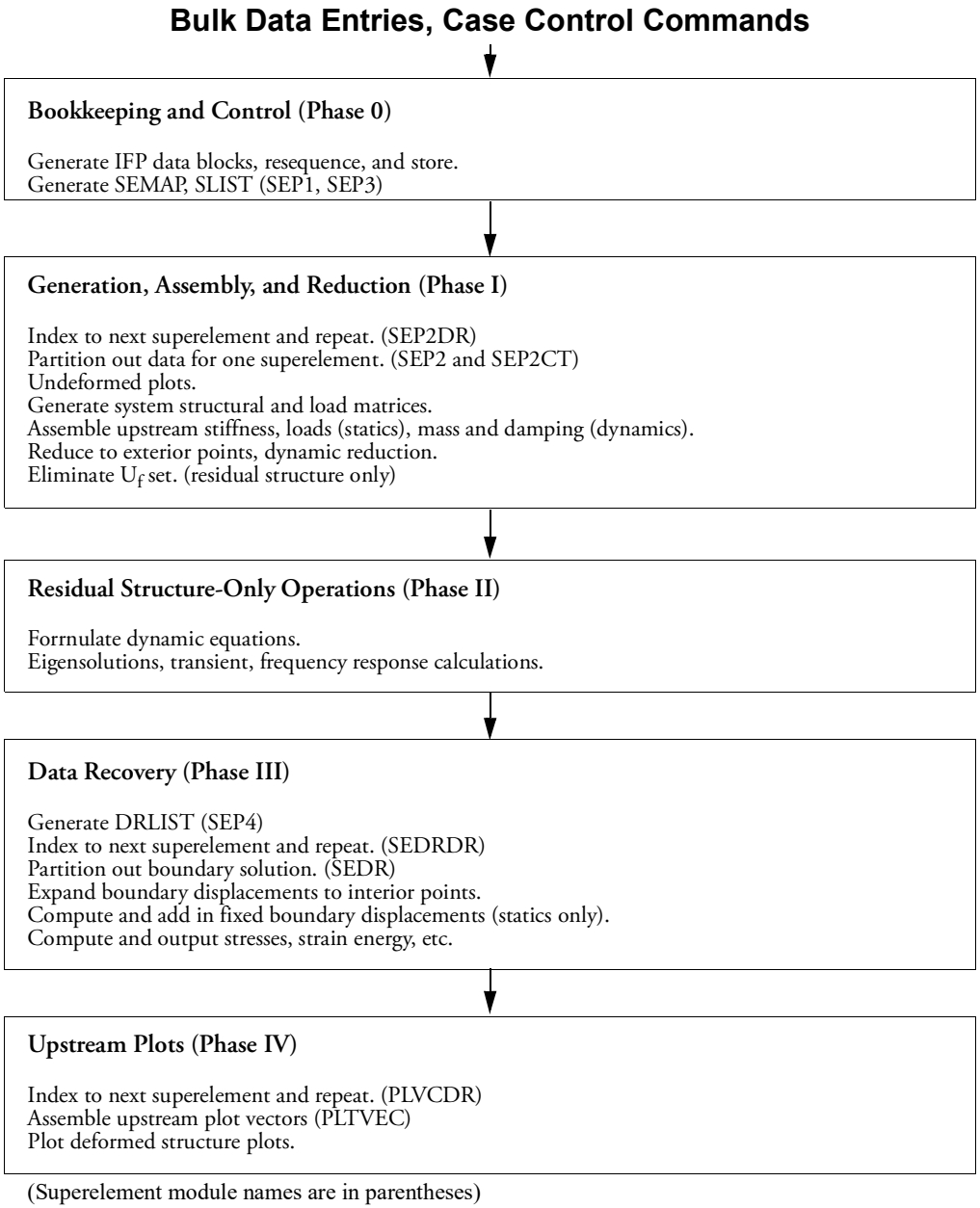


Figure 7-4 Solution Sequence Flow Chart

Superelement Stiffness Matrix Assembly and Reduction (SEKR)

The SEMA module adds in stiffness terms from any upstream superelements. The SEEX Case Control command may be used to eliminate stiffness contributions from selected upstream superelements. The superelements that are upstream are defined by the current SEMAP. The number of rows and columns of the upstream superelements must be consistent with the number of exterior grid points, as defined by the SEMAP. In order to complete execution of the SEMA module, the boundary matrices of all upstream superelements must be present in the database, and consistent with the current SEMAP. The GP4 module generates constraint data in the form of the USET table, as described in Geometry Processing in SubDMAP PHASE0, 363. A separate USET table is generated for each superelement. It is similar to the USET table of the older rigid formats except that the exterior points are automatically placed in the a-set.

The user may change SPCs and/or MPCs between subcases ("boundary condition changes") in SOLs 101 and 103 only. The GPSP module identifies grid point singularities, and at user's option provides SPCs for singular degrees of freedom. This is followed by the constraint eliminations described in Multipoint Constraint Operations in SubDMAPs SEKR and SEMR2 and SEMRB, 366, Single Point Constraint Operations in SubDMAPs SEKR, SEMR2 and SEMRB, 369, and Constraint and Mechanism Problem Identification in SubDMAP SEKR, 370. If the omitted degrees of freedom contain mechanism-type singularities, the grid and component number of all downstream ends of the mechanism are output in the matrix named MECHS, and a user fatal error stops execution. The approximate fixed-boundary natural frequencies of the superelement are printed in the FAPPROX matrix. The boundary stiffness matrix $[KAA]$ is stored in the database. Mass and damping matrices are not processed in this phase.

SUPORT Bulk Data entries, which define the r-set, may reference only grid points in the residual structure. The rigid body operations for the stiffness are performed as described in Static Solutions in SubDMAP SEKRRS, 376.

Superelement Load Assembly and Reduction (SELR)

This operation processes loads in the static and dynamic response solution sequences. This operation is not executed in the normal modes and complex eigenvalue solution sequences. The SELA module adds loads from upstream superelements to the load vector for the current superelement. The SEEX Case Control command may be used to eliminate load contributions from upstream superelements. The requirements for conformity of load data in the database with the current state of the SEMAP are similar to those of the SEKR operation above, except that only rows, not columns of the load vectors are mapped using data in the SEMAP table. The requirement on columns of all load matrices is that they be equal to the number of columns (number of load condition) in the residual structure. If the number of columns in all superelements and the residual structure is not the same, a user fatal error results. Static loads may be applied to any grid point, whether exterior or interior to a superelement. Dynamic loads may be applied directly to residual structure points. Dynamic load entries may also reference static load sets applied to the superelements via the `LOADSET` Case Control command and `LSEQ` Bulk Data entries. The static load reduction of Static Condensation in SubDMAPs SEKR and SEMR2, 373 is applied, stopping after generation of the $[P_a]$ matrix for superelements and $[P_l]$ for the residual structure.

Superelement Mass Assembly and Reduction (SEMR)

In the dynamics solution sequences, the constraint operations of Multipoint Constraint Operations in SubDMAPs SEKR and SEMR2 and SEMRB, 366, Single Point Constraint Operations in SubDMAPs SEKR, SEMR2 and SEMRB, 369 and Constraint and Mechanism Problem Identification in SubDMAP SEKR, 370 are applied to the mass and damping matrices. The rigid body operations of Static Solutions in SubDMAP SEKRRS, 376 are applied to the mass of the residual structure only. Dynamic reduction and component mode calculations of Static and Dynamic Load Generation, 379 are performed. This allows a mass-change-only type of restart by use of

the SEMG and SEMR options and avoids the costly reduction of the stiffness matrix in the SEKR operation if no stiffness changes are made.

The optional Bulk Data entry PARAM,ERROR,0 affects operations in Phase I. If it is not used, the default action is to terminate the run after the first user fatal error, as in the older rigid formats. If the parameter is used, a user fatal error causes a branch to the end of Phase I for the superelement, and continuation to the generation, assembly, and reduction operations for any remaining superelements.

The five SE-type commands may be executed in one run, or may be done separately. The following sequence must be observed if these commands are executed separately:

SEKR after SEMG
SELG after SEMG
SELR after SEKR, SELG, SEMR
SEMR after SFKR

Residual Structure-Only (Phase II)

The operations on the superelements described above are also performed on the residual structure. This is followed by operations unique to the residual structure. Eigensolutions and direct and modal forced response calculations take place at this point. Direct input matrices, when selected by the Case Control commands M2PP, K2PP, and B2PP are constrained and added as in Formulation of Dynamic Equations in SubDMAP GMA, 385.

Bulk Data entries used for static loads (i.e., `FORCE`, GRAV, LOAD, etc.) may be used in direct and modal dynamic analysis. They are interchangeable with DAREA Bulk Data entries and may be referenced by `LSEQ` Bulk Data entries.

Data Recovery Operations (Phase Ill)

The total solution vectors of the downstream superelement are partitioned to generate the boundary solution vectors for the superelement being processed. The boundary solution is expanded to all interior points of the superelement. The fixed boundary component is computed and added.

If SPC force output is requested, the resultants of the SPC forces about PARAM,GRDPNT or the basic coordinate system origin are also output. The data recovery operations of Data Recovery Operations in SubDMAP SEDISP, 388 are carried out. (Note that solution sequences involving complex eigenvalue or frequency response analysis may produce complex displacement vectors. GPFDR module output (grid point force balance, element strain energy) may be requested for complex vectors, but only the real component of the vector is processed.) Deformed structure plots of only elements in the superelement are generated here.

Upstream Plots (Phase IV)

The displacement vectors of a superelement and all of its upstream members are combined and plotted at user's request.

Superelement Alternate Statics Solution

SOL 101 provides an alternate method of performing the superelement static condensation. PARAM,ALTRED,YES must be entered in the Bulk Data Section. In some cases, the alternate solution can result in significant reduction in computer cost for static analysis.

The stiffness and load reduction and displacement recovery are described in Static and Dynamic Load Generation, 379, Static Condensation in SubDMAPs SEKR and SEMR2, 373 and Data Recovery Operations in SubDMAP SEDISP, 388.

Static Superelement Analysis With Inertia Relief

Inertia relief analysis is available in SOL 101. PARAM,INREL,-1 or PARAM,INREL, -2 must be present for either of these options. The following method transforms static loads based on model geometry rather than the stiffness matrix as described in Static Solutions in SubDMAP SEKRRS, 376.

The additional operations required to perform inertia relief are described in Static and Dynamic Load Generation, 379 and Static Solutions in SubDMAP SEKRRS, 376.

Assumptions and Limitations

- The model must have six and only six rigid body modes for a continuous, three-dimensional structure.
- Masses on scalar points are ignored in calculating inertia loads.
- The structural model, when given enforced displacements at the reference point, is assumed to produce the same motion as a rigid body model. This will be true for any structural or rigid element. It is possible, if unconventional, to use CELASi or general elements, MPC equations, or DMIG coefficients that will not meet this assumption.
- Identical and mirror image superelements are not processed correctly.

Nonlinear Heat Transfer Analysis

Steady State Heat Transfer Analysis

The steady state heat balance equation is given by

$$[K]\{u\} + [\mathfrak{R}]\{u + T_{abs}\}^4 = \{P\} + \{N\}$$

where:

$[K]$	=	a heat conduction matrix
$[\mathfrak{R}]$	=	a radiation exchange matrix
$\{P\}$	=	a vector of applied heat flows that are constant
$\{N\}$	=	a vector of nonlinear heat flows that depend on temperature
$\{u\}$	=	a vector of grid point temperature
T_{abs}	=	the absolution temperature

The components of the applied heat flow vector, $\{P\}$, are associated either with surface heat transfer or with heat generated inside the volume heat conduction elements. The vector of nonlinear heat flows, $\{N\}$, results from boundary radiation, surface convection, and temperature dependent thermal loads.

The equilibrium equation is solved by a Newton iteration scheme, where the tangential stiffness matrix is approximated by

$$[K_T]^i \approx [K]^i + 4[\Re]^i \{u^i + T_{abs}\}^3 - \left\{\frac{\partial N}{\partial u}\right\}^i$$

and the residual vector is

$$\{R\}^i = \{P\} + \{N\}^i - [K]^i \{u\}^i - [\Re]^i \{u^i + T_{abs}\}^4$$

Transient Heat Transfer Analysis

The general equation solved in transient analysis has the form

$$[B]\{\dot{u}\} + [K]\{u\} + [\Re]\{u + T_{abs}\}^4 = \{P\} + \{N\}$$

To take phase change into consideration, the heat diffusion equation is converted into

$$\{\dot{H}\} + [K]\{u\} + [\Re]\{u + T_{abs}\}^4 = \{P\} + \{N\}$$

where:

$[B]$	=	heat capacity matrix
$[K]$	=	heat conduction matrix
$[\Re]$	=	radiation matrix
$\{P\}$	=	vector of applied heat flows that are constant or functions of time
$\{N\}$	=	vector of nonlinear heat flows that depend on temperature
$\{H\}$	=	enthalpy vector
$\{\dot{H}\}$	=	$\{dH/dt\}$
$\{u\}$	=	grid point temperatures
$\{\dot{u}\}$	=	$\{du/dt\}$
T_{abs}	=	absolute temperature scale factor

The equilibrium equation is solved by Newmark's method with adaptive time stepping. Base on this one-step integration scheme, the time derivative of the nodal temperatures at the $(i+1)$ th iteration of the time step $(n+1)$ is expressed as

$$\{\dot{u}_{n+1}\}^{i+1} = \frac{1}{\theta \Delta t}\{u_{n+1}^{i+1} - u_n\} + \left(1 - \frac{1}{\theta}\right)\{\dot{u}_n\}$$

where

$$\{u_{n+1}\}^{i+1} = \{u_{n+1}\}^i + \{\Delta u_{n+1}\}^i$$

and

$$\frac{1}{\theta} = 2 - 2\eta$$

The parameter η is specified on the PARAM,NDAMP Bulk Data entry. When $\eta = 0$ ($\theta = 0.5$), no numerical damping is requested. In this case, Newmark's method is equivalent to the Crank-Nicolson method.

8 Element Data Recovery Resolved at Grid Points

- Introduction
- Stress Recovery at Grid Points
- Mesh Stress Discontinuities at Grid Points

Introduction

Element stresses may be averaged at grid points and then used to calculate estimates of stress discontinuities in the model mesh. Resolving element data at the grid points is supported only for selected linear elastic element types (see Table 8-1 for supported element types). The calculation of average stresses at grid points and the calculation of estimates of stress discontinuities use a common mesh definition method. The mesh is defined in the OUTPUT(POST) section of the Case Control Section. There are two methods available for mesh definition: SURFACE input for 2-D shell element meshes and VOLUME input for 3-D solid element meshes. The averaging of grid point stresses (and strains) and stress discontinuities are calculated independently for meshes composed of only 2-D shell elements and those composed of only 3-D solid elements. The element stress and strain data used in the calculation of grid point stresses and stress discontinuities must be requested in the Case Control Section using the STRESS command.

The request for grid point stresses for reports is made by the GPSTRESS command in Case Control. The calculation of strains may be substituted for stresses using the STRAIN command in place of the STRESS command in the Case Control Section (see STRAIN (Case) (p. 589) in the *MSC Nastran Quick Reference Guide* command for element strain output requests).

The calculation of stress discontinuities is supported for both elements and grid points associated with an element mesh by the ELSDCON and GPSDCON commands, respectively.

Stress Recovery at Grid Points, 436 and Mesh Stress Discontinuities at Grid Points, 447 explain in more detail the requesting and calculation of grid point stresses and stress discontinuities, respectively.

Table 8-1 Elements Types Supported in Element Data Resolved at Grid Point Calculations (Hyperelastic Elements Not Included)

Element Type	2-D Shell Stress	3-D Solid Stress	Stress Discontinuity
CQUAD4	x		x
CQUADR	x		x
CQUAD8	x		x
CTRIA3	x		x
CTRIAR	x		x
CTRIA6	x		x
CHEXA		x	x
CPENTA		x	x
CTETRA		x	x

Stress Recovery at Grid Points

Stresses may be computed at grid points on surfaces of two-dimensional plate elements, namely, the CQUAD4, CQUADR, CQUAD4, CTRIA3, CTRIAR, and CTRIA6, and in volumes containing CHEXA, CPENTA, and CTETRA solid elements.

By default, stresses are output at the center of each of these elements and also at the vertices of CQUADR, CQUAD8, CTRIAR, and CTRIA6 shell elements and the CHEXA, CPENTA, and CTETRA solid elements. The center output is not universally satisfactory because in the case concerning the surfaces of CQUAD4 and/or CTRIA3 elements, the analyst often has a requirement for realistic stress values at the grid points. Also, in the case of surfaces of CQUADR, CQUAD8, CTRIAR, and CTRIA6 elements or of volumes containing CHEXA, CPENTA, and CTETRA elements, the vertex stresses output for the elements that connect to a common grid point are often not identical. The option described in this section provides a rational method that yields accurate and unique stresses at each vertex grid point in user-defined surfaces or volumes. Grid point stress recovery is only available in linear static analysis, real eigenvalue analysis, and transient analysis. Grid point stress recovery is not available for lamina stresses.

An option is also provided to estimate the probable error in these grid point stresses (see Mesh Stress Discontinuities at Grid Points, 447). Estimates are generated for each stress component. The root mean square error of these estimates is then computed to provide a single measure of the error in the computation of the stresses at a grid point. This data provides a gross indication of the adequacy of a finite element model to represent the physical phenomena under investigation. The error estimator is available in SOLution Sequence 101.

Description of Method

A scheme to interpolate and/or extrapolate over a surface from a known set of stresses (the element stresses) requires that these known stress components be transformed into a consistent coordinate system prior to interpolation and extrapolation. This coordinate system is defined by the analyst and it should be generally compatible with the surface of interest in order to obtain accurate grid point stresses. Generally, such a coordinate system will be the natural coordinate system that the analyst used in the generation of the model of the surface or volume in question. Calculated grid point stress components are output in this coordinate system, called the output coordinate system of the surface or volume.

Each stress component is treated independently of other stress components in the calculation of average grid point stress components. The stress invariants at grid points (i.e., principal stresses, the von Mises stress, and the mean pressure) are evaluated from the average values of the stress components at grid points. The grid point stress components are output in the output coordinate system of the surface or volume and the stress invariants are oriented relative to this output coordinate system.

Two methods are used to calculate grid point stress components for plate and shell elements: topological and geometric. A single, simplified, topological method is used to calculate grid point stress components in volumes containing solid elements. However, prior to a discussion of these methods, it is necessary to define the four categories into which grid points are divided.

1. Interior Grid Points

   Grid points that are connected only to interior line segments; i.e., line segments that are coincident with the edges of two or more elements.

2. Corner Grid Points

   Grid points that are only connected to a single element.

3. Edge Grid Points

Grid points that are connected to both interior and exterior line segments. An exterior line segment is coincident with the edge of only one element.

4. Exception Grid Points

   Grid points at which discontinuities in stress can occur. Exception points will be described more completely under Treatment of Exception Points: BRANCHing, 443.

Stresses at interior, corner, or edge grid points are considered to be continuous between directly connected elements. Stresses at exception points may be discontinuous between connected elements and, as will be seen, a different grid point stress may be output for each of the connected elements. Engineering judgment must be exercised in assessing the validity of grid point stresses at exception points.

Topological Method: Plate Elements and Solid Elements

The topological method for the evaluation of grid point stresses only recognizes the existence of interior, edge, and corner points; that is, exception points are not considered as a distinct category of grid point (exception points arise as a result of geometrical rather than topological characteristics). The relations used to compute average grid point stresses in surfaces of plate elements by the topological method are described in Table 8-2. The relations used to compute the average grid point stresses in volumes containing solid elements are described in Table 8-3. Only the topological method is available for solid elements.

Table 8-2 Evaluation of Average Grid Point Stresses for Plate and Shell Elements by the Topological Method

Type of Grid Point	Average Grid Point Stress Component	Remarks
Interior	$\sigma_I = \frac{1}{N_e}\sum_{i=1}^{N_e}\sigma_{ei}$ where: σ_I = average stress at interior grid point N_e = number of directly connected elements σ_{ei} = element center stresses if only CQUAD4 and/or CTRIA3 elements are connected; element vertex stresses if CQUAD8 and/or CTRIA6 are connected	If CQUAD4 and/or CTRIA3 elements connect to the same grid point as CQUAD8 and/or CTRIA6 elements, then only the CQUAD8 and or CTRIA6 elements are used in the calculation of average grid point stresses.

Table 8-2 Evaluation of Average Grid Point Stresses for Plate and Shell Elements by the Topological Method (continued)

Type of Grid Point	Average Grid Point Stress Component	Remarks
Edge	$\sigma_E = \frac{2}{N_e}\sum_{i=1}^{N_e}\sigma_{ei} - \frac{1}{N_I}\sum_{i=1}^{N_I}\sigma_{Ii}$ where: σ_E = average stress at edge grid point N_I = number of interior points connected by line segments σ_{Ii} = average stress at ineterior grid points	If an edge point is not connected to an interior point by line segments, then $N_I = 0$, the second term in the relation σ_E is set to zero, and the factor 2 in the first term is replaced by 1. Same as remark for interior grid points.
Corner	Corner points connected to CQUAD4 element with the grid point diagonally opposite the corner point, an interior point, or an edge point. $\sigma_c = 2\sigma_1 - \sigma_D$ where: σ_c = average stress at corner grid point σ_i = CQUAD4 center stress σ_D = average stress at grid point diagonally opposite corner point (If point diagonally opposite corner point is not an interior or edge point, then $\sigma_c = \sigma_1$.	Same as remark for interior points

Table 8-2 Evaluation of Average Grid Point Stresses for Plate and Shell Elements by the Topological Method (continued)

Type of Grid Point	Average Grid Point Stress Component	Remarks
Corner	Corner points connected to CTRIA3 elements and the other two grid points, namely, E1 and E2, of the connected CTRIA3 are edge points: $\sigma_c = 3\sigma_1 - (\sigma_{E1} + \sigma_{E2})$ where: σ_c = average stress at corner grid point σ_1 = CTRIA3 center stress σ_{E1}, σ_{E2} = stresses at edge points E1 and E2 (if either E1 or E2 is not an edge point, then $\sigma_c = \sigma_1$) Corner points connected to one CQUAD4 or one CTRIA6 element: $\sigma_c = \sigma_r$ where: σ_r = stress at element vertex connected to corner point Corner points connected to two or more CTRIA6 elements: $\sigma_c = \frac{1}{N_c} \sum_{i=1}^{N_c} \sigma_{r_i}$ where: N_c = number of CTRIA6 elements connected to corner point.	Same as remark for interior points
Interior and Edge	$\sigma_I = \frac{1}{N_e} \sum_{i=1}^{N_e} \sigma_{ei}$ where: σ_I = average stress at interior or edge grid points N_e = number of directly connected elements σ_{ei} = element vertex stresses	Mixtures of CHEXA, CPENTA, and CTETRA elements may connect to an interior or edge grid point.

Table 8-2 Evaluation of Average Grid Point Stresses for Plate and Shell Elements by the Topological Method (continued)

Type of Grid Point	Average Grid Point Stress Component	Remarks
Corner	$\sigma_c = \sigma_r$ where: σ_r = stress at element vertex connected to corner point	The same relation is used whether a CHEXA, CPENTA, or CTETRA element is connected to the grid point.

Table 8-3 Evaluation of Average Grid Point Stresses for Plate and Shell Elements by the Geometric Method

Type of Grid Point	Number of Elements Involved in Stress Calculation
Interior	All directly connected elements
Edge	Two directly connected elements: Use the two known element stresses plus the stresses at all interior points connected by line segments of the directly connected elements. Three or more directly connected elements: Use the known element stresses of the directly connected elements.
Corner	Same as topological method (see Table 8-1).

Geometric Method: Surface Elements

In the geometric method, grid point stresses for points which connect to CQUAD8 or CTRIA6 elements use the average of the element corner stresses. If any CQUAD4 or CTRIA3 elements connect to these points, their influence is not considered.

The geometric method for the determination of average grid point stresses which connect only to CQUAD4 or CTRIA3 elements utilizes the numerical technique of least squares. The "best fit" value of the stress Σ at a point (x, y) is functionally denoted as:

$$\sum(x,y) = A_0 + A_1 x + A_2 y \tag{8-1}$$

where the coefficients A_0, $A_1 x$, and A_2 must be determined. The coordinates x and y are the locations of points at which known stresses exist. In accordance with the least squares technique, it is required to minimize the function:

$$f = \frac{1}{2}\left\{\sigma - \sum\right\}^T \left\{\sigma - \sum\right\} \tag{8-2}$$

with respect to $\{A\}$. Here σ represents the known element stresses.

To perform the minimization, equation (8-1) is written in matrix form as:

$$\left\{\sum\right\} = [T]\{A\}$$

where:

$$[T] = \begin{bmatrix} 1 & X_1 & X_2 \\ \vdots & \vdots & \vdots \\ i & X_i & X_i \end{bmatrix}$$

and equation (8-2) is differentiated with respect to the coefficients A_i.

$$\frac{\partial f}{\partial A_i} = [T]^T[T]\{A\} - [T]^T\{\sigma\} = 0 \tag{8-3}$$

The coefficients $\{A\}$ are then determined from equation (8-3) to be:

$$\{A\} = ([T]^T[T])^{-1}[T]^T\{\sigma\} \tag{8-4}$$

The average grid point stress at a given grid point can then be calculated by substituting equation (8-4) into the matrix representation of equation (8-1) as follows:

$$\sum = [1 \;\; X_o \;\; Y_o]([T]^T[T])^{-1}[T]^T\{\sigma\} = [Q]\{\sigma\} \tag{8-5}$$

where:

Q	=	matrix of interpolation factors
$X_o \;\; Y_o$	=	locations of grid points at which stress is required

Note that if $([T]^T[T])^{-1}$ is singular, the matrix of interpolation factors is taken to be $1/N_I$ so that equation (8-5) becomes:

$$\sum = \frac{1}{N_I}\sum_{i=1}^{N_I}\sigma_{ei} \tag{8-6}$$

where N_I = number of known stress points associated with the grid point. In most cases, equation (8-6) is equivalent to the topological interpolation method.

The strategy used in the evaluation of equation (8-5) for interior, edge, and corner grid points is presented in Table 8-3.

As previously mentioned, the coordinates x and y of equation (8-1) are the locations of points at which known stresses exist. These locations are actually taken as the projected coordinates of the known stress points on the average surface evaluated at the grid point at which the stress is to be computed. This average surface is determined to be the surface whose normal is the average of the unit normals to the elements connected to the grid point of interest. The local x-axis used in the evaluation of the geometric coefficients of the matrix $[T]$ is taken as the projection of the x-axis of the output coordinate system on the average surface evaluated at the grid point of interest.

Treatment of Exception Points: BRANCHing

Exception points have been defined as grid points at which stress discontinuities may occur. Such discontinuities may occur because a substantial difference in slope exists between adjacent elements directly connected to the same line segment or if more than two elements are directly connected to the same line segment. Examples of these two cases are illustrated below.

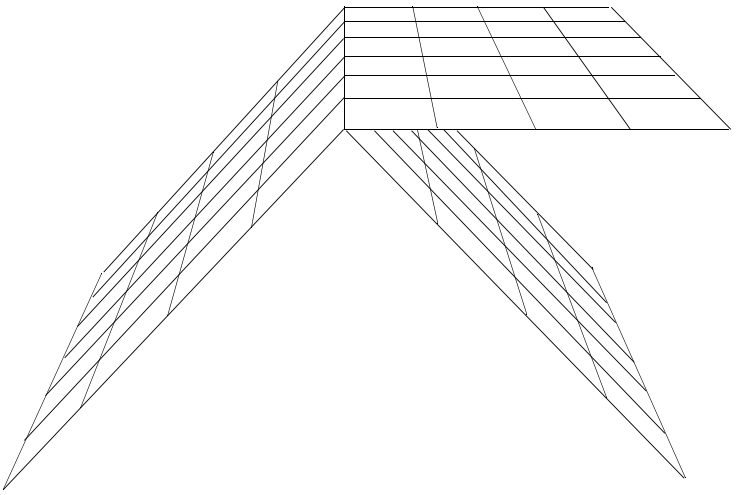


Figure 8-1 Three or More Elements Connected to the Same Line Segment

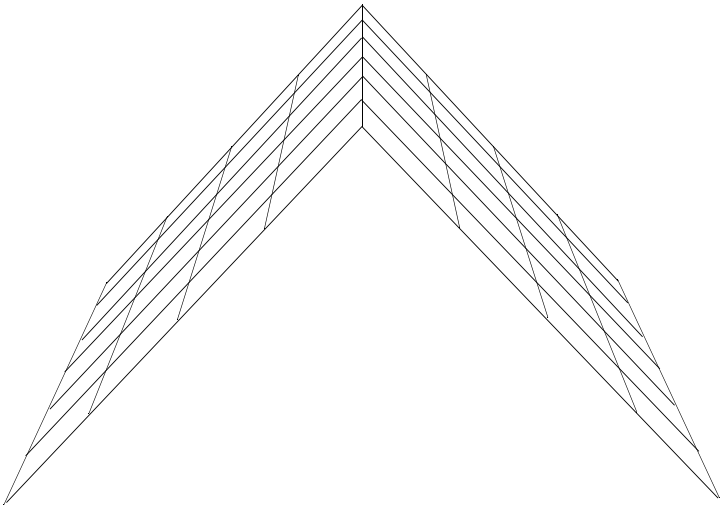


Figure 8-2 Slope Difference between Adjacent Elements Connected to the Same Line

Note: That these two situations are geometric rather than topological in nature and, therefore, are only considered under the geometric interpolation option.

Since exception points are defined as grid points at which stress discontinuities may exist, it is conceivable that a computed average grid point stress at an exception point may be different for each of the directly connected elements. However, it is more common to find that a subset of the elements that are directly connected to an exception point forms a subsurface on which stresses may be considered to be continuous. A unique average grid point stress may then be computed for the elements of the subsurface.

The calculation of average grid point stresses for a subsurface that contains two or more elements is performed with the geometric interpolation method used for edge points. If a subsurface consists of a single element, the calculation of an average grid point stress is performed in the same manner as utilized in the calculation of average grid point stresses at corner points. It must be noted, however, that σ_D, σ_{E1}, and σ_{E2} may be interior, exterior, or corner points (see Table 8-3).

User Information

The use of the grid point stress recovery option is demonstrated in this section through the consideration of several example problems. These problems are discussed separately after the presentation of the requirements that must be met in the Executive Control Section, the Case Control Section, and the Bulk Data Section.

Executive Control Section

A typical Executive Control Section would be of the following form:

```
SOL 101
CEND
```

Case Control Section

The option to obtain grid point stress output is exercised through commands in the Case Control Section. The general request for the recovery of grid point stresses is accomplished using commands with the form:

```
GPSTRESS = ALL
```

or

```
SET j = k, l, m, . . .
GPSTRESS = j
```

The k, *l*, m, . . . reference the identification numbers of the surfaces or volumes of elements that are defined in the OUTPUT(POST) section of the Case Control Section. Grid point stress data will be output only for those surfaces or volumes referenced through the GPSTRESS Case Control command.

Note:	If grid point stress data is desired, an ELSTRESS (or STRESS) request for all elements in the surfaces or volumes of interest must be included in the Case Control Section. Such requests will cause the output of element stress data as well as grid point stress data. If the output of element stress data is not desired, the requests ELSTRESS(PLOT) = or STRESS(PLOT) = may be used.

In static analysis, one may place the GPSTRESS command above the subcase level or in individual subcases. If the GPSTRESS command is above the subcase level, it will be utilized by all subcases and subcoms unless overridden by a GPSTRESS command in a subcase or subcom.

The OUTPUT(POST) section of the Case Control Section contains specific requests for grid point stress data for the surfaces or volumes of elements. Any number of element surfaces or volumes may be defined, but only those surfaces or volumes that are referenced through the GPSTRESS command will have grid point stress data generated and output.

Element surfaces and volumes are defined through a SURFACE and VOLUME commands in the OUTPUT(POST) section of the Case Control Section.

Grid Point Stress Output Description

Listing 8-1 through Listing 8-3 are examples of grid point stress output produced for two-dimensional plate elements requested on the SURFACE Case Control command. Listing 8-1 is the most common form of output and is always the result when the TOPOLOGICAL method is used, and when the GEOMETRIC method is used and no surface "BRANCHing" is detected. Note that when the "ELEMENT ID" value is zero, it always indicates that all elements in the SURFACE connected to the grid point contribute to the average grid point stress and lie in the "best" subsurface. Listing 8-2 and Listing 8-3 show examples of the output when the GEOMETRIC method is used and when "BRANCHing" occurs. The negative value for the "ELEMENT ID" indicates the "best" average grid point stress and the element identification number of an element associated with "best" subsurface. Similarly a positive element identification numbers indicate another average grid point stress for a "less best" grid point stress and the element identification number associated with it's

subsurface. When TOLERANCE is set positive as in Listing 8-2 only the “best” grid point stress is calculated when “BRANCHing” is detected and when TOLERANCE is set negative all subsurface average grid point stress are calculated. See SURFACE (p. 728) in the .

Listing 8-1 Grid Point Stress Output – Typical

```
SQUARE PLATE WITH CIRCULAR HOLE                                          JUNE  24, 1991  MSC/NASTRAN  6/14/91   PAGE    27
  GRID POINT STRESS TEST CASE
 UNIFORM LOAD ALONG X=5.0.                                                                              SUBCASE 1
   SUBCASE =              1
                                S T R E S S E S   A T   G R I D   P O I N T S   - -     S U R F A C E       4
                     SURFACE X-AXIS X  NORMAL(Z-AXIS)  Z        REFERENCE COORDINATE SYSTEM FOR SURFACE DEFINITION CID        1
  GRID       ELEMENT            STRESSES IN SURFACE SYSTEM          PRINCIPAL STRESSES            MAX
   ID          ID     FIBER   NORMAL-X   NORMAL-Y   SHEAR-XY     ANGLE      MAJOR      MINOR      SHEAR     VON MISES
 10304          0     MID     4.965E+03  4.041E+03 -6.431E+03  -42.9450  1.095E+04 -1.944E+03  6.447E+03   1.204E+04
 10305          0     MID     6.132E+03  3.839E+03 -5.518E+03  -39.1310  1.062E+04 -6.503E+02  5.636E+03   1.096E+04
 10306          0     MID     7.033E+03  3.408E+03 -4.246E+03  -33.4429  9.837E+03  6.037E+02  4.617E+03   9.549E+03
 10400          0     MID     6.101E+02  1.631E+04 -1.038E+03  -86.2353  1.638E+04  5.418E+02  7.919E+03   1.612E+04
 10401          0     MID     1.463E+03  1.417E+04 -3.402E+03  -75.9185  1.503E+04  6.098E+02  7.209E+03   1.473E+04
 10402          0     MID     2.228E+03  1.211E+04 -5.354E+03  -66.3558  1.446E+04 -1.156E+02  7.286E+03   1.452E+04
 10403          0     MID     2.809E+03  1.019E+04 -6.381E+03  -60.0211  1.387E+04 -8.721E+02  7.371E+03   1.433E+04
 10404          0     MID     3.159E+03  8.514E+03 -6.308E+03  -56.5007  1.269E+04 -1.016E+03  6.852E+03   1.323E+04
 10405          0     MID     3.307E+03  7.318E+03 -5.482E+03  -55.0445  1.115E+04 -5.249E+02  5.837E+03   1.142E+04
```

Listing 8-2 Grid Point Stress Output with “BRANCHing” and TOLERANCE > 0

```
INTERSECTING PLATES                                                      JUNE  24, 1991  MSC/NASTRAN  6/14/91   PAGE    34
  GRID POINT STRESS TEST CASE

   SUBCASE =              1
                                S T R E S S E S   A T   G R I D   P O I N T S   - -     S U R F A C E       4
                     SURFACE X-AXIS X  NORMAL(Z-AXIS)  Z        REFERENCE COORDINATE SYSTEM FOR SURFACE DEFINITION CID        0
  GRID       ELEMENT            STRESSES IN SURFACE SYSTEM          PRINCIPAL STRESSES            MAX
   ID          ID     FIBER   NORMAL-X   NORMAL-Y   SHEAR-XY     ANGLE      MAJOR      MINOR      SHEAR     VON MISES
 10304          0     Z1      9.107E+02  1.209E+03  2.799E+03   46.5224  3.863E+03 -1.743E+03  2.803E+03   4.969E+03
                      Z2      1.052E+03  1.571E+03  2.913E+03   47.5464  4.236E+03 -1.613E+03  2.925E+03   5.233E+03
                     MID      9.813E+02  1.390E+03  2.856E+03   47.0453  4.049E+03 -1.678E+03  2.863E+03   5.099E+03
 10305          0     Z1      1.906E+03  5.699E+02  1.789E+03   34.7594  3.147E+03 -6.714E+02  1.909E+03   3.531E+03
                      Z2      2.032E+03  5.648E+02  1.934E+03   34.6114  3.367E+03 -7.698E+02  2.068E+03   3.811E+03
                     MID      1.969E+03  5.674E+02  1.861E+03   34.6824  3.257E+03 -7.206E+02  1.989E+03   3.671E+03
 10306          0     Z1      2.771E+03 -1.983E+02  2.678E+02    5.1132  2.795E+03 -2.223E+02  1.508E+03   2.912E+03
                      Z2      3.024E+03 -1.118E+02  4.889E+02    8.6589  3.098E+03 -1.862E+02  1.642E+03   3.196E+03
                     MID      2.897E+03 -1.550E+02  3.783E+02    6.9615  2.944E+03 -2.012E+02  1.572E+03   3.049E+03
 10400       -119     Z1     -5.775E+04  1.181E+03 -2.060E+04  -72.5215  7.667E+03 -6.423E+04  3.595E+04   6.839E+04
                      Z2     -5.821E+04  1.207E+04 -1.647E+04  -77.4454  1.573E+04 -6.188E+04  3.880E+04   7.106E+04
                     MID     -5.798E+04  6.623E+03 -1.853E+04  -75.0777  1.156E+04 -6.292E+04  3.724E+04   6.942E+04
 10401       -119     Z1     -1.856E+04  7.060E+03 -4.939E+03  -79.4558  7.979E+03 -1.948E+04  1.373E+04   2.446E+04
                      Z2     -1.714E+04  1.656E+04 -3.881E+03  -83.5131  1.700E+04 -1.758E+04  1.729E+04   2.995E+04
                     MID     -1.785E+04  1.181E+04 -4.410E+03  -81.7177  1.245E+04 -1.849E+04  1.547E+04   2.696E+04
```

Listing 8-3 Grid Point Stress Output with “BRANCHing” and TOLERANCE < 0

```
INTERSECTING PLATES                                                    JUNE  24, 1991  MSC/NASTRAN  6/14/91   PAGE    40
     GRID POINT STRESS TEST CASE

      SUBCASE =              1
                                  S T R E S S E S   A T   G R I D   P O I N T S   - -     S U R F A C E      5
                        SURFACE X-AXIS X  NORMAL(Z-AXIS)  Z        REFERENCE COORDINATE SYSTEM FOR SURFACE DEFINITION CID        0
     GRID      ELEMENT            STRESSES IN SURFACE SYSTEM          PRINCIPAL STRESSES              MAX
     ID          ID     FIBER    NORMAL-X    NORMAL-Y    SHEAR-XY     ANGLE      MAJOR      MINOR      SHEAR     VON MISES
    10304         0      Z1     9.107E+02   1.209E+03   2.799E+03    46.5224  3.863E+03 -1.743E+03  2.803E+03   4.969E+03
                         Z2     1.052E+03   1.571E+03   2.913E+03    47.5464  4.236E+03 -1.613E+03  2.925E+03   5.233E+03
                        MID     9.813E+02   1.390E+03   2.856E+03    47.0453  4.049E+03 -1.678E+03  2.863E+03   5.099E+03
    10305         0      Z1     1.906E+03   5.699E+02   1.789E+03    34.7594  3.147E+03 -6.714E+02  1.909E+03   3.531E+03
                         Z2     2.032E+03   5.648E+02   1.934E+03    34.6114  3.367E+03 -7.698E+02  2.068E+03   3.811E+03
                        MID     1.969E+03   5.674E+02   1.861E+03    34.6824  3.257E+03 -7.206E+02  1.989E+03   3.671E+03
    10306         0      Z1     2.771E+03  -1.983E+02   2.678E+02     5.1132  2.795E+03 -2.223E+02  1.508E+03   2.912E+03
                         Z2     3.024E+03  -1.118E+02   4.889E+02     8.6589  3.098E+03 -1.862E+02  1.642E+03   3.196E+03
                        MID     2.897E+03  -1.550E+02   3.783E+02     6.9615  2.944E+03 -2.012E+02  1.572E+03   3.049E+03
    10400      -119      Z1    -5.775E+04   1.181E+03  -2.060E+04   -72.5215  7.667E+03 -6.423E+04  3.595E+04   6.839E+04
                         Z2    -5.821E+04   1.207E+04  -1.647E+04   -77.4454  1.573E+04 -6.188E+04  3.880E+04   7.106E+04
                        MID    -5.798E+04   6.623E+03  -1.853E+04   -75.0777  1.156E+04 -6.292E+04  3.724E+04   6.942E+04
                        219     Z1    -5.772E+04   1.152E+03   2.064E+04    72.4819  7.667E+03 -6.423E+04  3.595E+04   6.839E+04
                         Z2    -5.819E+04   1.204E+04   1.652E+04    77.4058  1.573E+04 -6.188E+04  3.880E+04   7.106E+04
                        MID    -5.795E+04   6.597E+03   1.858E+04    75.0381  1.156E+04 -6.292E+04  3.724E+04   6.942E+04
    10401      -119      Z1    -1.856E+04   7.060E+03  -4.939E+03   -79.4558  7.979E+03 -1.948E+04  1.373E+04   2.446E+04
                         Z2    -1.714E+04   1.656E+04  -3.881E+03   -83.5131  1.700E+04 -1.758E+04  1.729E+04   2.995E+04
                        MID    -1.785E+04   1.181E+04  -4.410E+03   -81.7177  1.245E+04 -1.849E+04  1.547E+04   2.696E+04
                        120     Z1    -1.856E+04   7.060E+03  -4.939E+03   -79.4558  7.979E+03 -1.948E+04  1.373E+04   2.446E+04
                         Z2    -1.714E+04   1.656E+04  -3.881E+03   -83.5131  1.700E+04 -1.758E+04  1.729E+04   2.995E+04
                        MID    -1.785E+04   1.181E+04  -4.410E+03   -81.7177  1.245E+04 -1.849E+04  1.547E+04   2.696E+04
                        219     Z1    -1.855E+04   7.053E+03   4.957E+03    79.4163  7.979E+03 -1.948E+04  1.373E+04   2.446E+04
                         Z2    -1.713E+04   1.655E+04   3.905E+03    83.4735  1.700E+04 -1.758E+04  1.729E+04   2.995E+04
                        MID    -1.784E+04   1.180E+04   4.431E+03    81.6781  1.245E+04 -1.849E+04  1.547E+04   2.696E+04
                        220     Z1    -1.855E+04   7.053E+03   4.957E+03    79.4163  7.979E+03 -1.948E+04  1.373E+04   2.446E+04
                         Z2    -1.713E+04   1.655E+04   3.905E+03    83.4735  1.700E+04 -1.758E+04  1.729E+04   2.995E+04
                        MID    -1.784E+04   1.180E+04   4.431E+03    81.6781  1.245E+04 -1.849E+04  1.547E+04   2.696E+04
```

General Remarks

Each element stress component (independently of all other stress components) is transformed into the output coordinate system before interpolation. It is necessary that this output coordinate system vary slowly over the surface for the interpolated stresses to be accurate. This approach is certainly valid for stress interpolation over elements that are in a common plane and it should be adequate for engineering analysis when the difference in slope between adjacent elements is small. However, when the elements do vary substantially from a smooth surface, the analyst must utilize engineering judgment relative to the accuracy of the resultant stress data.

Average grid point stresses are only computed at element vertices. For elements with mid-edge grid points, one can estimate the stress at these mid-edge grid points to be the average of the computed stresses at the grid point at the ends of the edge in question.

Mesh Stress Discontinuities at Grid Points

Error Estimates for Grid Point Stress Data

Stress Recovery at Grid Points, 436 is devoted to a discussion of the averaging procedures utilized to provide meaningful stresses at the grid points of finite element models in MSC Nastran. This section will focus on the description of error estimators to assist the analyst in the identification of regions of his model that may require refinement.

The essence of the previous averaging procedures used to obtain grid point stress data is to:

1. Convert these local element stress components into a common user-defined coordinate system.
2. Average the several values of each stress component to obtain a unique value of the stress component that is to be associated with the grid point in question.
3. Compute the stress invariants at the grid points from the stress components at the grid points.

In the general case, the stress components are σ_x, σ_y, σ_z, τ_{xy}, τ_{xz}, and τ_{yz}.

For discussion purposes, the averaging process used to compute the stress components at the grid points can be represented in the form of equation (8-7).

$$\sigma_g = \sum_{i=1}^{N_e} (W_i \sigma_{ei}) \tag{8-7}$$

where:

σ_g	=	weighted mean value of the stress component computed at the grid point.
σ_{ei}	=	value of the stress component in the ith element (i = 1,2,...,N_e) in the neighborhood of the grid point. δ_{gi} is in the same coordinate system as δ_g.
W_i	=	weighting factor assigned to the ith element. The sum of the N_e values of W_i must equal 1. (This requirement assures that all computed statistics will be unbiased.) The attribute of being unbiased implies that the variance is equal to the mean square error. Equal weighting (i.e., $W_i = 1/N_e$) is assumed.

An estimate of the error in a particular component of stress at a grid point can then be computed by assuming that the values of the corresponding stress components computed for the elements in the neighborhood of the grid point are data points with uncorrelated random errors. It then follows that an estimate of the probable error in the stress component δ_g at the grid point is:

$$\begin{aligned} \delta_g &= \sqrt{\sum_{i=1}^{N_e} (W_i \delta_{ei})^2} \\ &= \frac{1}{\sqrt{N_e}} \sqrt{\frac{\sum_{i=1}^{N_e} (\delta_{ei})^2}{N_e}} \end{aligned} \tag{8-8}$$

where $\delta_{ei} = \sigma_{ei} - \sigma_g$. Thus, the probable error δ_g is the root mean square error in δ_{ei} divided by $\sqrt{N_e}$.

It should be noted that the standard deviation or root mean square error is a reasonable measure of precision in many practical cases, but it is easy to provide examples in which the standard deviation is a poor measure of the concentration of the distribution about the mean. Equation (8-8) is assumed to provide an approximate error estimator for the grid point stress data.

Element and Grid Point Stress Discontinuity Output Requests

It should be clear that element and grid point stress discontinuity output can only be obtained if the analyst has requested grid point stress output via the GPSTRESS command in the Case Control Section of the MSC Nastran input file. The GPSTRESS command, of course, requires that the analyst define all required SURFACEs and VOLUMEs in the OUTPUT(POST) portion of the Case Control Section.

To output Grid Point Stress DisCONtinuities:

GPSDCON = ALL

or

SET j = k, *l*, m, ...
GPSDCON = j

To output ELement Stress DisCONtinuities:

ELSDCON = ALL

or

SET j = k, *l*, m, ...
ELSDCON = j

The following remarks should be noted relative to the use of the GPSDCON and/or ELSDCON commands.

1. The GPSDCON and ELSDCON commands are honored only in SOLution Sequences 101, 114, 144, and 200.
2. The GPSDCON and ELSDCON commands may be placed above the SUBCASE level or in individual SUBCASEs and/or `SUBCOM`s. The use of these commands above the SUBCASE level causes stress discontinuity data to be output for ALL SUBCASEs in the MSC Nastran input file. If the analyst wishes to restrict stress discontinuity output to specific SUBCASEs, GPSDCON and/or ELSDCON commands should appear only under these specific SUBCASEs.
3. The analyst is cautioned that these commands currently produce a substantial amount of output.
4. The GPSDCON and/or ELSDCON commands may be effectively used in restarts. For example, one could include requests for GPSTRESS data during the initial run of a project and then obtain data on stress discontinuities on a subsequent restart by including GPSDCON and/or ELSDCON commands in the Case Control Section.
5. Stress discontinuity output will only be provided when grid point stresses have been previously computed by the methods described in this article. Thus, stress continuity will not be computed when, for example, both plate and solid elements are connected to a grid point that is involved in stress discontinuity calculations.

Discussion of Error Measures

Large values of error estimators for `CQUAD4` and `CTRIA3` elements are not uncommon occurrences in statistical error measures. For example, the mean and variance have direct analogs in engineering mechanics. The mean is analogous to the centroid of a body and is thus a measure of where the mass is centered. The variance is the second `moment` about the mean, and it tends to be small if the majority of the mass is concentrated about the centroid. As the mass is dispersed further

from the centroid, the moment of inertia tends to increase. If the mass is concentrated at the centroid (as it might be in some idealized lumped mass models), the moment of inertia becomes zero.

Thus, the error estimates for CQUAD4 and CTRIA3 elements will tend toward larger values because the data used to compute the estimators are dispersed relatively far from the mean. For those elements that provide stress data at vertices, the estimators will tend toward smaller values because the stress data tend to be concentrated about the mean in well-designed finite element models.

Generally, the goal is to design a finite element mesh for static analysis so that all important stress gradients will be adequately represented. If the mesh is not sufficiently detailed, the stress data at element vertices or element centroids will result in the relatively large values of the error estimators.

Very inaccurate values of these error estimators may be obtained at the edges of defined SURFACEs and on the faces of defined VOLUMEs. Such edges and surfaces will often correspond with the boundaries of superelements.

In summary, error estimators can, in some cases, be highly inaccurate. Nevertheless, error estimate data is quite useful when properly interpreted by the analyst.

Grid Point Stresses and Mesh Stress Discontinuities

Grid point stresses may be requested for grid points connected by CTRIA3, CTRIA6, CTRIAR, CQUAD4, CQUAD8, CQUADR, CHEXA, CPENTA, and CTETRA elements (with linear material properties only). In linear static analysis only, a mesh stress discontinuity error analysis may then be performed based on the computed element and grid point stresses.

Grid point stresses are not computed for laminar stresses.

The grid point stresses are calculated in the user-specified coordinate system. The following real membrane stresses are output on request:

- Normal stresses in the x and y directions.
- Shear stresses on the x face in the y direction.
- Angle between the x axis and the major principal axis.
- Major and minor principal stresses.
- von Mises equivalent or maximum shear stress (see STRESS Case Control command).

The following real stresses are output on request for grid points connected to solid elements:

- The three normal stresses.
- The three shear stresses.
- Magnitude and direction of the three principal stresses.
- The mean pressure.
- The von Mises equivalent or octahedral stress (see STRESS Case Control command).

Only real stresses are available at the grid points. The user has the following options:

- The capability to vary output requests by SUBCASE.
- Choice of elements in stress field.

- Choice of topological or geometric interpolation (surface fields only).
- Choice of output coordinate system.
- Choice of output stress fiber locations (surface fields only).

The grid point stresses are requested in the Case Control Section for printing and postprocessing using the Case Control commands GPSTRESS (Grid Point Stress) and STRFIELD (Stress Field). The user is required to request element stress output (STRESS = N) for all elements referenced by selected SURFACE and VOLUME commands. The grid point stress requests a set of surface or volume fields which are defined in the OUTPUT(POST) part of the Case Control Section on the SURFACE or VOLUME commands, respectively. The postprocessing output requests are separated from case control and plot requests by the OUTPUT(POST) command.

The general rules for Case Control syntax are the same as for plotting (see Plotting, 460). The grid point stress output is defined on the SURFACE and VOLUME commands. The SURFACE command defines the following: output coordinate system, list of elements in the stress surface field, output stress fiber location for stress evaluation and the choice of topological or geometric interpolation method used to determine the grid point stresses from the elements. The VOLUME command defines the following: output coordinate system, list of elements in the stress volume, and the choice of DIRECT and/or PRINCIPAL stresses.

In the linear static structured solution sequences only (SOLutions 101, 114, 144, and 200) mesh stress discontinuities may be requested to estimate errors due to the coarseness of the finite element mesh. The estimation of the error is computed for the element and grid point stresses and requested by the ELSDCON and GPSDCON Case Control commands, respectively. These commands select the desired surfaces and volumes which are defined in the OUTPUT(POST) Section of Case Control. This output option also requires a corresponding GPSTRESS or STRFIELD request for the desired surfaces and volumes and a STRESS Case Control command for the elements in those surfaces and volumes. If PARAM,POST,0 is specified, then this output will also be written to the DBC database.

The following is computed and output for stress discontinuities on shell elements:

- Normal stress discontinuities in the x and y directions
- Shear stress discontinuities on the x-y direction
- Major and minor principal stress discontinuities
- Von Mises and maximum shear stress discontinuities
- Error estimate

The following is computed and output for stress discontinuities on solid elements:

- Normal stress discontinuities in the x, y, and z directions
- Shear stress discontinuities on the x-y, y-z, and x-z directions
- Major and minor principal stress discontinuities
- Von Mises stress and mean pressure discontinuities
- Error estimate

9 DMAP

- Introduction

Introduction

MSC Nastran provides a wide variety of solution sequences. Each solution sequence consist of a series of DMAP (Direct Matrix Abstraction Program) statements. MSC Nastran allows the user to modify the solution sequences or to write his or her own solution sequences using DMAP. DMAP is a high level language with its own compiler and grammatical rules. The *MSC Nastran DMAP Programmer's Guide* provides detailed information on these modules.

10 Database Concepts

- Introduction
- Delivery Databases
- Dbset Deletion
- Database Autoassignment
- Database Project and Version
- Database Archival, Compression, and Transfer
- Database Compression
- Database Transfer Across Different Computer Types

Introduction

All MSC Nastran runs require a database. A database allows for the storage and retrieval of data: data blocks, parameters, scratch files, and DMAP object and source files. During a run data may be written onto only one database. This is called the primary database which is attached automatically or by the INIT MASTER statement. Additional databases may be attached for “read-only” purposes. This includes a “delivery” database and one or more “located” databases. The delivery database contains solution sequences and is automatically attached by the SOL Executive statement or the ACQUIRE FMS statement. “Located” databases are attached by the DBLOCATE statement.

A database is divided into several DBsets which are initialized by the INIT statement. There are two types of DBsets: permanent and scratch. Permanent DBsets may be saved at the end of the run and reused in a restart run. Scratch DBsets are automatically deleted at the end of the run. A DBset may be composed of up to twenty concatenated physical files or DBset members. Each member is assigned to a physical file by the ASSIGN statement.

By default, there are four DBsets - two permanent and two scratch - which are predefined and automatically assigned by the program. The DBset names of the predefined permanent DBsets are: MASTER and DBALL. The DBset-names of the predefined scratch DBsets are: SCRATCH and OBJSCR. The DBsets are briefly described below:

- The MASTER DBset is the directory of the database. It contains the names of all DBsets, DBset members and their physical file names, and a directory of projects, versions, data blocks, parameters, DMAP source and object files. It also contains the NDDL scheme used to describe the database.
- The DBALL DBset contains all the DMAP data blocks which may be saved permanently for reuse in a subsequent run. In the solution sequences in which restarts are not possible, (for example, SOL 1) this DBset is empty.
- The SCRATCH DBset is a temporary DBset for all scratch data blocks and files. It has two partitions: one for DMAP data blocks and one for DMAP module internal scratch files.
- The OBJSCR DBset is a temporary DBset for DMAP compilation.
- If the user desires to create his or her own DMAP source and object files, two other DBsets (USROBJ and USRSOU) need to be allocated.
- The USROBJ and USRSOU DBsets contain DMAP source and object files that may be saved permanently for execution in a subsequent run. See the COMPILE Executive Control statement.

If an INIT and ASSIGN statement are not specified for one or more of the predefined DBsets, then the corresponding statement(s) are provided by default.

```
ASSIGN MASTER='dbs-name.MASTER'
INIT    MASTER(RAM)   LOGICAL=(MASTER(5000))
ASSIGN DBALL='dbs-name.DBALL'
INIT    DBALL LOGICAL=(DBALL(25000))
ASSIGN USROBJ='dbs-name.USROBJ'
INIT    USROBJ LOGICAL=(USROBJ(5000))
ASSIGN USRSOU='dbs-name.USRSOU'
INIT    USRSOU LOGICAL=(USRSOU(5000))
ASSIGN SCRATCH='temp-name.SCRATCH'
ASSIGN SCR300='temp-name.SCR300'
INIT    SCRATCH(MEM) LOGICAL=(SCRATCH(175000)),
                     SCR300=(SCR300(175000))
```

```
ASSIGN OBJSCR='temp-name.OBJSCR'
INIT    OBJSCR LOGICAL=(OBJSCR(5000))
```

The numbers inside the parentheses are the maximum sizes of the DBsets in GINO blocks. By default, one member with the same log-name as the DBset-name is assigned to each DBset. (For a description of a GINO block, DBset-name and log-name, see the INIT statement). For example, the DBALL DBset has one member also called DBALL. The dbs-name is determined from the dbs keyword on the nastran command (see Executing MSC Nastran (p. 1) in the *MSC Nastran Quick Reference Guide*) and temp-name is generated by the command procedure (see the MSC Nastran 2020 Installation and Operations Guide). The scratch DBsets are assigned special names on a scratch disk. (For a description of the dbs-name and temp-name, see the MSC Nastran 2020 Installation and Operations Guide). The filenames above are those that would be generated on Linux-type computers. For example, if the name of the input file is called MYJOB.DAT and the dbs keyword is not specified, then the default filenames for the permanent DBsets would be:

```
MYJOB.MASTER
MYJOB.DBALL
```

Also, they would have the same directory or path as the input file.

A physical filename is automatically created for any new DBsets specified by an INIT statement. For example, to create a new DBset called DBUP for split database operations, specify:

```
INIT DBUP
```

The physical filename will then be: MYJOB.DBUP. For another example, to create two members for DBALL, specify

```
INIT DBALL LOGI=(DB1 ,DB2)
```

Their physical filenames will be MYJOB.DB1 and MYJOB.DB2.

The ASSIGN statement may be used to override these default names. For example, to rename MYJOB.DBUP to FENDER.DBUP, specify

```
ASSIGN DBUP='FENDER.DBUP'
INIT DBUP
```

To rename MYJOB.DB1 and MYJOB.DB2, specify

```
ASSIGN DB1 ='FENDER.DBALL1'
ASSIGN DB2='FENDER.DBALL2'
INIT DBALL LOGI=(DB1,DB2)
```

The INIT statement may be used to change the maximum size of the predefined DBsets. (The size may be specified in blocks, kilowords, megawords, kilobytes, or megabytes.) For example, to specify a size of 50,000 blocks for DBALL,

```
INIT DBALL LOGICAL=(DBALL(50000))
```

or 50,000 bytes,

```
INIT DBALL LOGICAL=(DBALL(50KB))
```

The INIT and ASSIGN statement may be used to specify more than one member for a DBset which may exist on different physical devices.

```
ASSIGN DB1 ='physical file name of DB1 on disk 1'
ASSIGN DB2='physical file name of DB2 on disk 2'
INIT DBALL LOGICAL=(DB1(5000),DB2(5000))
```

Delivery Databases

The Delivery Database contains the solution sequences. This database, in addition to the primary database, is automatically assigned in order to execute an MSC Nastran solution sequence. For example, on Linux-type computers, the filenames of the database are

```
SSS.MASTERA
SSS.MSCOBJ
SSS.MSCSOU
```

If the solution sequence is not an MSC Nastran solution sequence, as in the case of a user's solution sequence; then the ACQUIRE FMS statement may be used to select the appropriate database. ACQUIRE NDDL selects the SSS database. The ACQUIRE statement is required when the DBLOAD FMS statement is used with the ENDJOB FMS statement.

Dbset Deletion

The disposition of permanent DBsets in the primary database at the end of the run is controlled in several ways.

- If scr=yes is specified on the nastran command (see Executing MSC Nastran (p. 2) in the *MSC Nastran Quick Reference Guide*) or the INIT MASTER(S) statement appears in the FMS Section then all DBsets in the primary database are deleted.
- An individual DBset member is automatically deleted at the end of the run by specifying the TEMP keyword on the ASSIGN statement. For example, to delete the USROBJ and USRSOU DBsets at the end of the run, specify:

  ```
  ASSIGN   USROBJ='A' TEMP
  ASSIGN   USROBJ='B' TEMP
  ```

  However, DBsets deleted in this manner or manually cannot be recreated in a future run.
- The DBSETDEL FMS statement provides a better method than the TEMP keyword on the ASSIGN statement except that the DBsets are deleted at the beginning of the run. The deleted DBsets may be recreated in a future run. For example:

  ```
  DBSETDEL USROBJ,USRSOU
  ```

Database Autoassignment

When a database is created, the physical filenames of all DBset members are loaded into the MASTER DBset. Therefore, when using DBLOCATE or RESTART, it is sufficient to assign the MASTER DBset member; and all other DBset members previously initialized; that is, DBALL, DBUP, etc., are automatically assigned by the program. For example, on restart, the following input will automatically assign the DBALL, USROBJ, and USRSOU DBsets:

```
ASSIGN MASTER='filename of the MASTER DBset member'
RESTART
```

If DBLOCATE is used, then:

```
ASSIGN MSTR1 ='filename of the MASTER DBset member'
DBLOCATE LOGI=MSTR1
```

The autoassignment feature cannot assign the physical file of a DBset member that has been renamed. This is because the MASTER DBset member contains the old physical filename. Therefore, an additional ASSIGN statement is required to specify the new filename of the DBset member. If in the previous restart example DBALL is renamed, then the following input is required:

```
ASSIGN MASTER='filename of the MASTER DBset member'
ASSIGN DBALL='new filename of the DBALL DBset member'
RESTART
```

In split database operations, it is often necessary to assign only some of the DBsets because the others are "offline." The NASTRAN AUTOASGN statement is used to specify the types of databases that will be autoassigned:

- AUTOASGN = 0: No databases will be autoassigned. This includes the primary database, Delivery Database, and any located databases.
- AUTOASGN = 1: All databases will be autoassigned. This is the default.
- AUTOASGN = 2: Only the Delivery Database will be autoassigned.
- AUTOASGN = 4: Only located databases will be autoassigned.

If some combination of the options above is desired, then sum their values. For example, if the delivery and located databases are to be autoassigned, then specify NASTRAN AUTOASGN = 6.

Database Project and Version

All data stored on the primary database is assigned to a particular project identification string (ID) and version identification number (ID). The current project ID is specified on the PROJECT FMS statement. The current version ID is an integer value assigned by the program each time the database is attached as the primary database. When a primary database is initialized or a new project is specified, then all items stored on the database during the run are stored under a version ID of integer 1. Then, in subsequent runs a new version is created with an ID equal to the last version ID incremented by one. If the RESTART FMS statement is specified, then a new version of data is created which is equivalent to the last version. In other words, all items in the old version are made available for reuse in the new version.

Database Archival, Compression, and Transfer

Database archival, compression and transfers are performed using the DBUNLOAD and DBLOAD FMS statements. The user may select all items on the database or selected items according to the WHERE clause, see "WHERE and CONVERT Clauses" in the File Management Statements (p. 41) in the *MSC Nastran Quick Reference Guide*.

Database Compression

When data is deleted from the database, its space is released for the storage of new data and this old space will be reused before any new space is used. However, even if the space is not reused, the size of the database is not reduced. After several restarts or the execution of the DBCLEAN FMS statement, the database may contain a significant amount of released or "dead" space. There are two methods of removing dead space or "compressing" the database.

The first method involves the DBUNLOAD and DBLOAD FMS statements, and the examples shown above will also perform a database compression. The second and simplest method involves the DBLOCATE statement specified with the COPY keyword. For example, if the database was created with SOLs 100 through 200, then the input file is:

```
ACQUIRE NDDL
ASSIGN MYDB='physical file name of the MASTER DBset'
DBLOCATE LOGI=MYDB COPY
ENDJOB
```

It is also possible to compress an individual DBset and change its maximum size. For example, the input file is:

```
ACQUIRE NDDL
ASSIGN MYDB='physical file name of the MASTER DBset'
DBLOCATE LOGI=MYDB COPY WHERE(DBSET='DBALL')
INIT DBALL LOGICAL=(DBALL(10000))
ENDJOB
```

Database Transfer Across Different Computer Types

It is often desirable to copy a database from one computer to a dissimilar computer (e.g., copying a database created on a Windows computer to a Linux workstation). A method is available using the DBUNLOAD and DBLOAD FMS statements. The DBUNLOAD statement is first used to convert the database to a "neutral" file on the first computer. This neutral file is now suitable for a simple copy to the second computer or may be accessed through a network if the two computers are so connected. On the second computer, the DBLOAD statement is then used to convert the neutral file back to a database. It is also necessary to execute the solution sequence DBTRANS (or 190) as described under Database Migration, because some data blocks cannot be converted to a neutral format. (In SOL DBTRANS; data blocks EST, EMAP, ETT, PTELEM, SLT, KDICT, KELM, MDICT, MELM, BDICT, BELM, and ACPT are regenerated.)

The ASSIGN statement is also required to assign the physical filename of the neutral file and specify that the file is formatted since the default format (UNFORMATTED) is not transferable across dissimilar computers. The following example unloads a database created by one of the unstructured solution sequences on the first computer to a neutral file:

```
ASSIGN DBUNLOAD='physical filename of neutral file' FORMATTED
DBUNLOAD FORMAT=NEUTRAL
ENDJOB
```

Then the neutral file is copied to, or accessed through a network by, the second computer:

```
ASSIGN DBLOAD='physical filename of neutral file' FORMATTED
DBLOAD FORMAT=NEUTRAL
SOL DBTRANU
CEND
```

With the DBTRANS solution sequence it is assumed that the neutral file contains all of the data from a completed and successful analysis. If not, then the solution sequences may fail and the transfer may not be successful. In this case, it is possible to alter DBTRANS with the ALTER statement to obtain the appropriate data.

If the database was created in a heat transfer analysis then NASTRAN HEAT = 1 must be specified.

If the database on the first computer was not created by a MSC Nastran solution sequence, then it may be necessary to develop a new transfer solution sequence.

11 Plotting

- General Capability
- Superelement Plotting
- Postprocessor

General Capability

MSC Nastran can generate the following kinds of plots:

- Undeformed geometric projections of the structural model.
- Static deformations of the structural model by either displaying the deformed shape (alone or superimposed on the undeformed shape), or displaying the displacement vectors at the grid points (superimposed on either the deformed or undeformed shape).
- Modal deformations resulting from real or complex eigenvalue analysis by the same options stated in 2 above. Complex modes for flutter analysis may be plotted for any user-chosen phase lag.
- Deformations of the structural model for transient response or frequency response by displaying either vectors or the deformed shape for specified times or frequencies.
- X-Y graphs of transient response or frequency response.
- V-F and V-G graphs for flutter analysis.
- Contour plots of displacements, temperature and stress on the structure.

Structural plots (items 1 through 4) are discussed in the Case Control Commands in the *MSC Nastran Quick Reference Guide*. X-Y plots (item 5) are discussed in X-Y PLOT Commands (p. 651) in the . Requests for structure plots or X-Y plots are made in the Case Control Section by submitting a structure plot request or an X-Y output request. The optional PLOTID command is considered to be part of the plot request although it must precede any OUTPUT(PLOT), OUTPUT(XYOUT) or OUTPUT(XYPLOT) commands. See PLOTID (Case) (p. 541) in the *MSC Nastran Quick Reference Guide* for a description of the command.

Plot requests are separated from Case Control by the OUTPUT(PLOT), OUTPUT(XYPLOT) or OUTPUT(XYOUT) commands. Data above this command (except PLOTID) will not be recognized by the plotter, even though it may have the same name (for example, the SET command).

Superelement Plotting

Plotting can occur at four different places in the superelement solution sequences. There are two plot commands, SEPLOT and SEUPPLOT, used with other Case Control and PARAM commands to control the type of plot to be prepared. A flow chart of the solution process is given in Figure 11-1.

Undeformed structure plots are made early during Phase 0 in the structured solution sequences and during Phase l in the unstructured solution sequences. They can be made for either one superelement only, or for a superelement and all of its upstream members, as controlled by PARAM,PLOTSUP (see Figure 11-1). Plots are made for superelements selected by the SEMG command, and listed in a plot request headed by SEPLOT SEID, where SEID is the superelement identification number. In the structured solution sequences, the SEMG command is not required and all superelements which appear on SEPLOT commands will be plotted (see Figure 11-1, block 0). Undeformed plots can be used to check geometry and connectivity, and do not require the presence of property or material commands. In the unstructured solution sequences, a branch to the end of the loop immediately after the plot module can be made by using the PARAM,PLOT,-1 command.

In dynamic analysis, solution set XY-plots are requested by use of SEPLOT 0. In the data recovery phase, XY-plots and deformed structure plots for elements in one superelement only are requested by the SEPLOT command. Deformed structure plots for a superelement and all its upstream plots are requested with the SEUPPLOT command.

XY-plots and deformed structure plots are regarded as output requests and will result in automatic execution of the data recovery loop as is required to produce the plots requested. For example, the command SEUPPLOT 0 will result in data recovery being performed on the entire model, even in the absence of any other output requests. For SEUPPLOT requests, if SUBCOM or SYMCOM subcases are used, each superelement must have identical SUBCOM and SYMCOM structure.

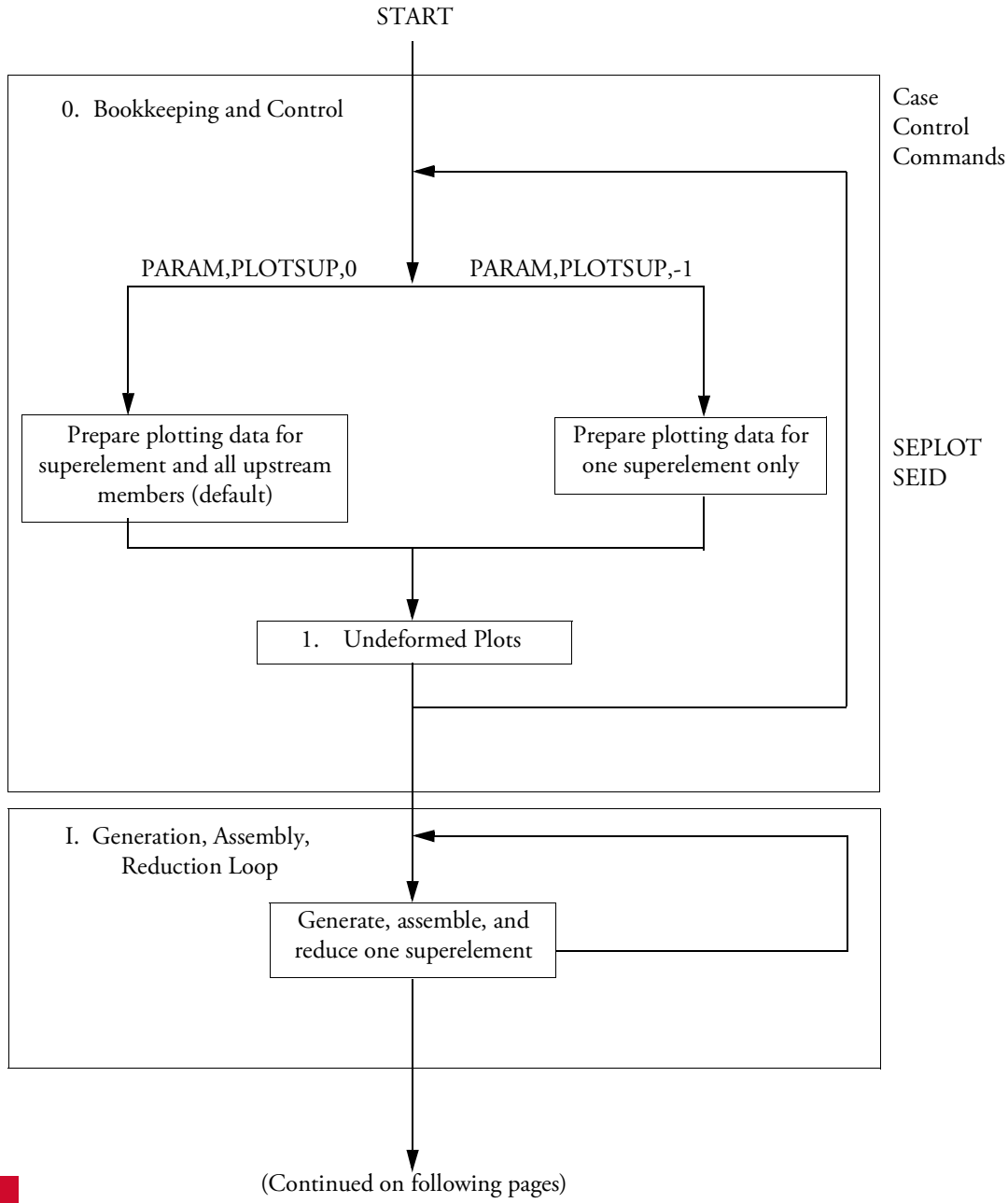


Figure 11-1 Phase 0 and I Superelement Plot Control In the Structured Solution Sequences

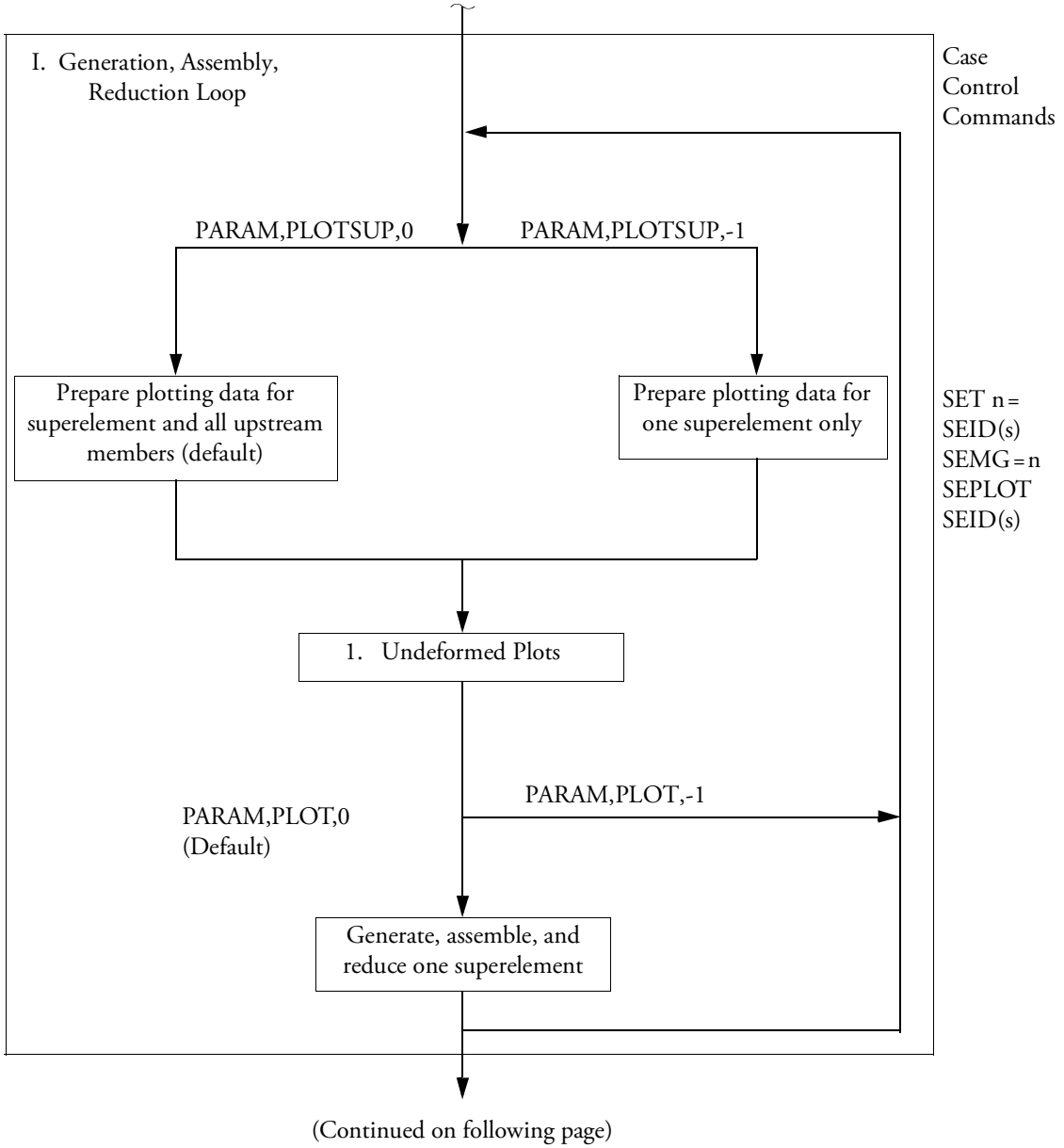


Figure 11-2 Phase I Superelement Plot Control In the Unstructured Solution Sequences (cont.)

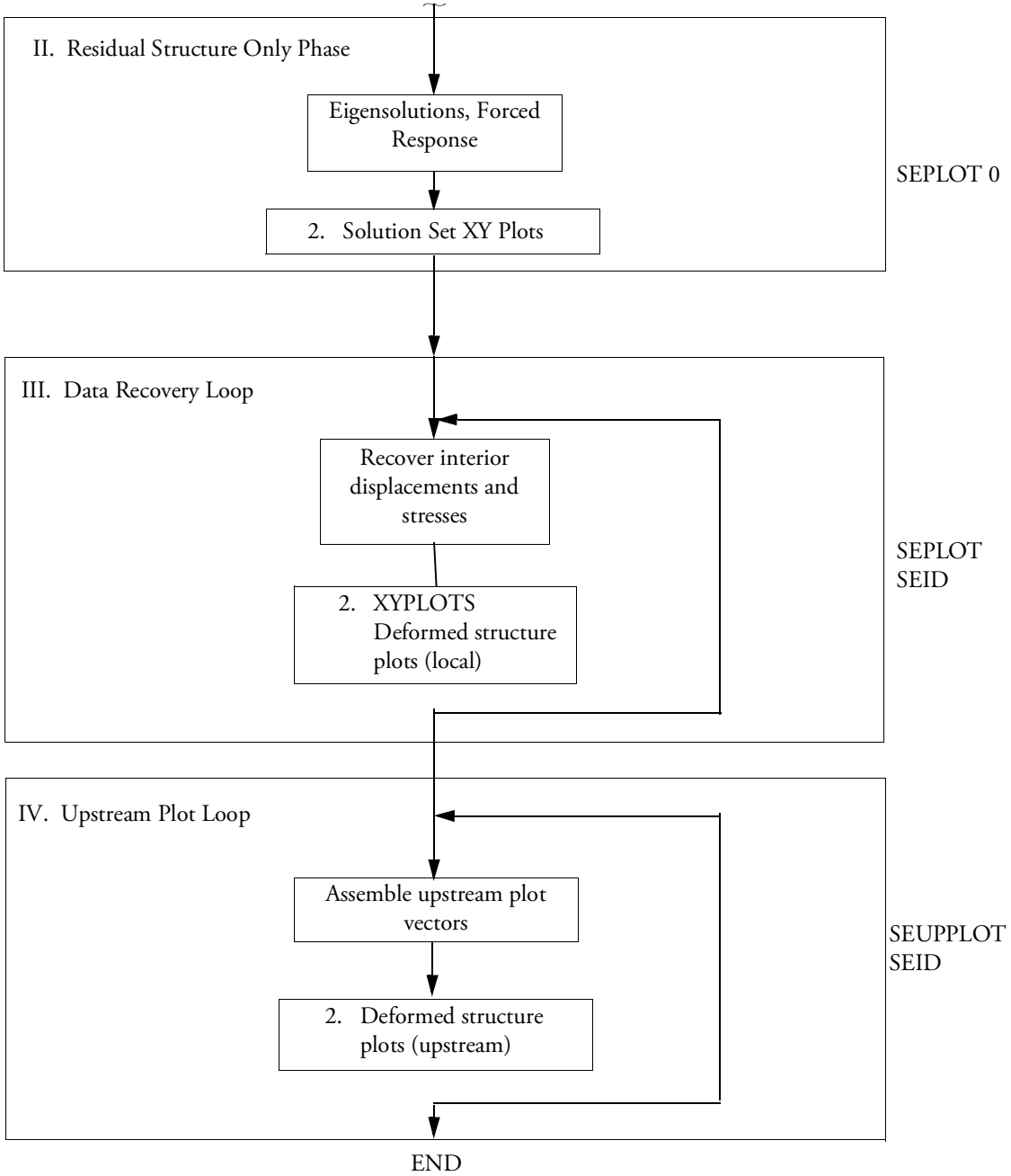


Figure 11-3 Phase II, III, and IV Superelement Plot Control in All Solution Sequences (cont.)

Structure Plotter Coordinate System and Orthographic Projection

In order to define the coordinates of an orthographic projection of the structural model, an R, S, T plotter coordinate system is defined as shown in Figure 11-4.

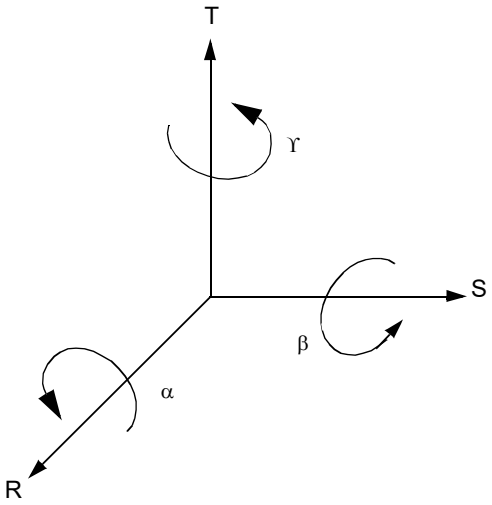


Figure 11-4 Plotter Coordinate System

The S-T plane is taken as the plan of projection. The structural model is defined in the basic coordinate system, which is denoted as the X, Y, Z coordinate system. The user specifies the position of the structural model with respect to the S-T projection plane by the angles Υ, β, and α. These angles position the X, Y, Z coordinate systems with respect to the R, S, T coordinate system. The two coordinate systems are coincident for $\Upsilon = \beta = \alpha = 0$. The sequence in which the rotations are taken is crucial and has been arbitrarily chosen as Υ, the rotation about T-axis followed by β, the rotation about the S-axis, followed by α, the rotation about the R-axis. Normally, α is not used since it does not affect the appearance of the S-T projection, but only its orientation on the page.

The orthographic projection is obtained by computing the S and T coordinates of each point having coordinates X, Y, Z from the transformation equation

$$\begin{Bmatrix} R \\ S \\ T \end{Bmatrix} = [A_\alpha][A_\beta][A_\Upsilon]\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}$$

where:

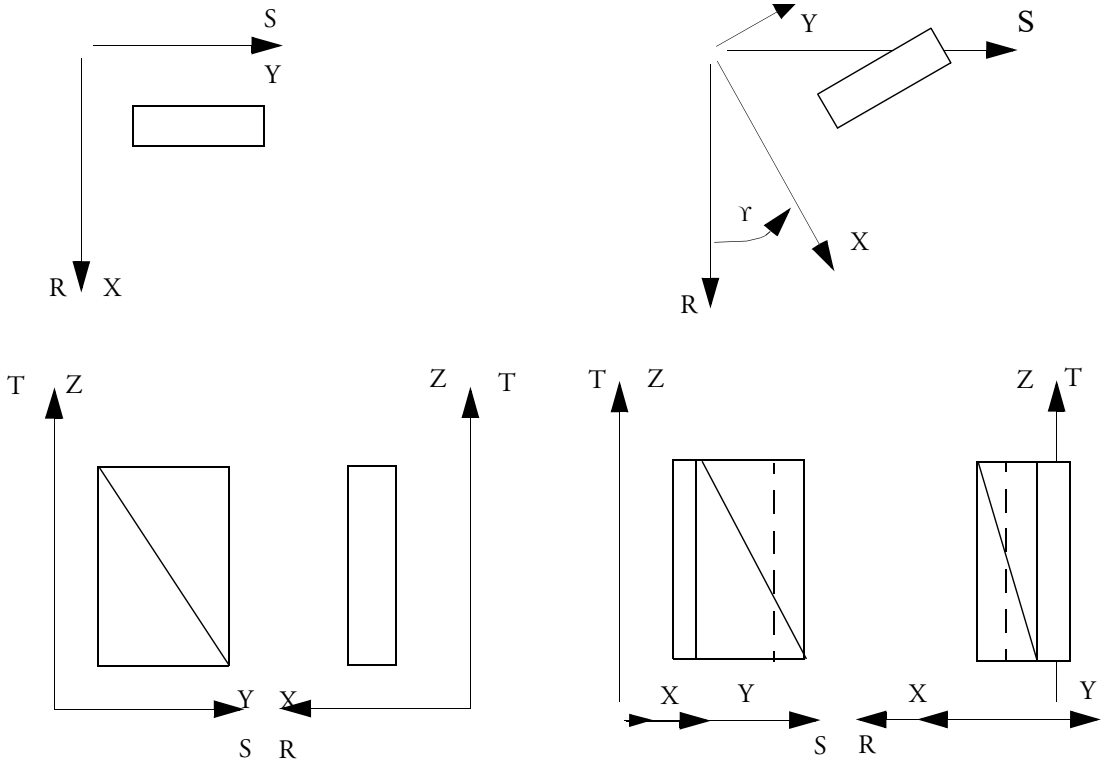
$$[A_\Upsilon] = \begin{bmatrix} cos\,\Upsilon & -sin\,\Upsilon & 0 \\ sin\,\Upsilon & cos\,\Upsilon & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$[A_\beta] = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix},$$

and

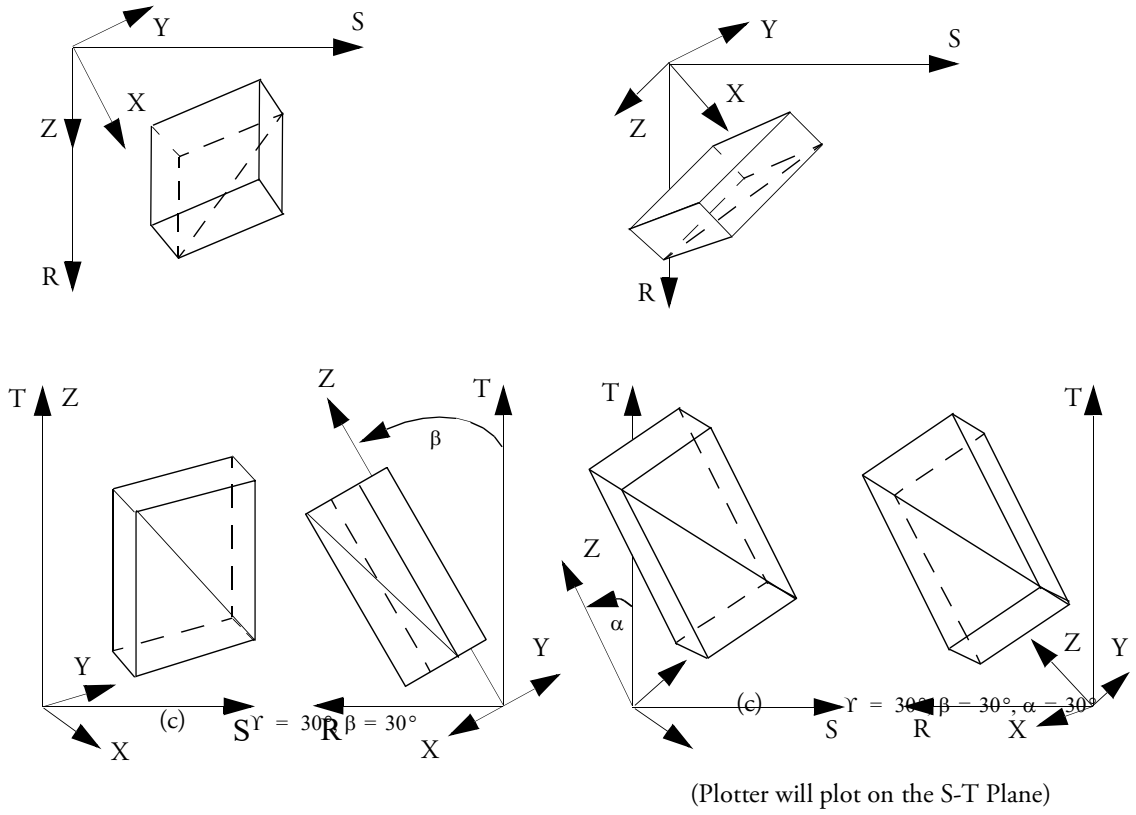
$$[A_\alpha] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

In order to illustrate clearly the orientation process, Figure 11-5 shows a rectangular parallelepiped as it is rotated through the Υ, β, α sequence. The final S-T plane shown in Figure 11-5b contains the desired orthographic projection.



(a) Initial Position

(b) $\Upsilon = 30°$



(Plotter will plot on the S-T Plane)

Figure 11-5 Plotter -- Model Orientation

Perspective Projection

In addition to the three angular relationships required for orthographic projection, the perspective projection requires knowledge of the vantage point in the R-S-T system (that is, the three coordinates of the observer), and the location of the projection plane (plotter surface). The vantage point is selected by the user (or automatically by the program), and lies in the positive R half space as shown in Figure 11-6. The projection plane is chosen to lie between the observer and the S-T plane.

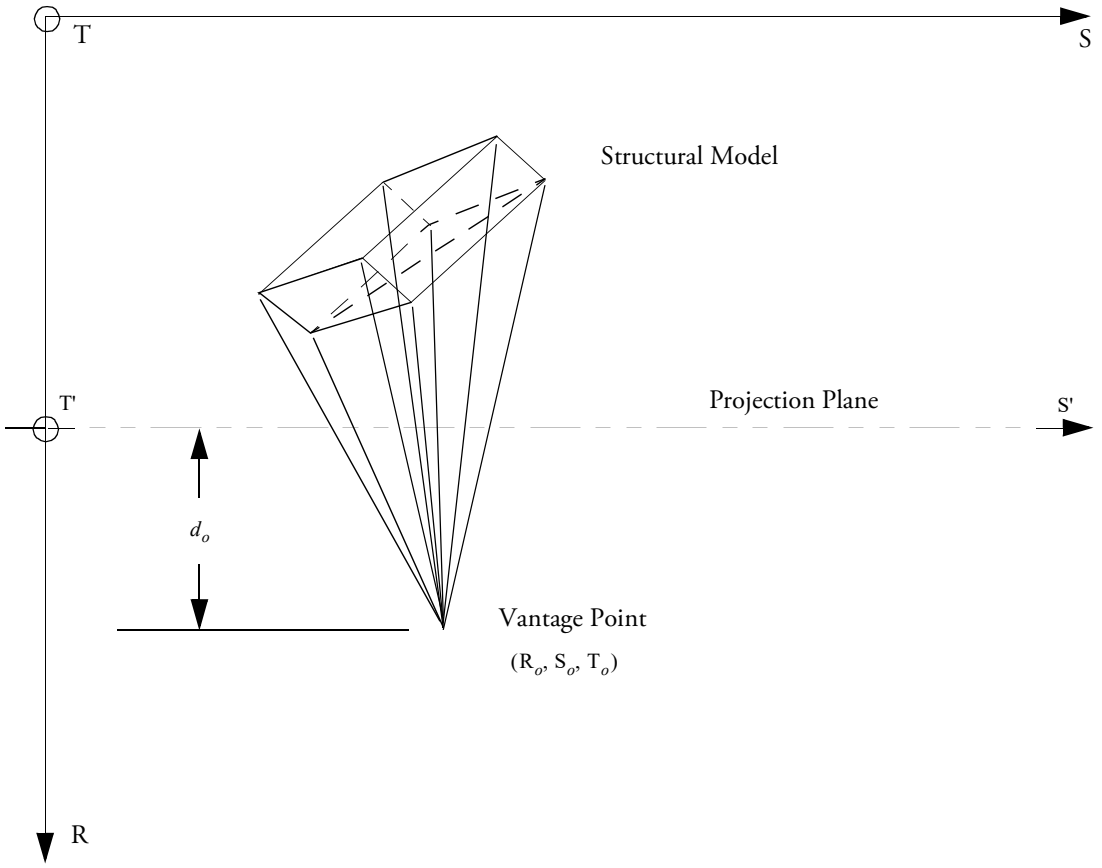


Figure 11-6 Perspective Projection Geometry

For each point, the coordinates S′ and T′ T’ on the projection plane (see Figure 11-6) are obtained from the orthographic projection coordinates R, S, T by

$$\begin{Bmatrix} S' \\ T' \end{Bmatrix} = \begin{Bmatrix} S_o \\ T_o \end{Bmatrix} + \frac{d_o}{R_o - R} \begin{Bmatrix} S - S_o \\ T - T_o \end{Bmatrix}$$

where R_o, S_o, T_o are the coordinates of the vantage point and d_o is the separation distance between the vantage point and the projection plane.

Stereoscopic Projection

The stereoscopic effect is obtained through the differences in images received by the left and right eyes. Each is a perspective image, but with a different vantage point. The two vantage points are separated horizontally by 70 mm (2.756 inches), the nominal ocular separation standard used in commercially available stereoscopic cameras and viewers. To plots are produced for viewing with a stereoscopic viewer.

Projection Plane to Plotter Transformations

Since the plotter surface is defined differently for each of the plotters used in MSC Nastran, and since it is desirable to minimize the amount of special coding for each plotter, a common interface with the plotter routines is provided wherein the plotted surface is assumed to have a lower left corner defined by x, y coordinates (0,0) and an upper right corner defined by x, y coordinates (1,1). Plotter utility routines are utilized to convert from this system to each individual plotter.

Since the coordinates of points in any of the projection planes previously discussed may have arbitrary numerical values, a linear mapping of the form

$$\begin{Bmatrix} x \\ y \end{Bmatrix} = [A_{ij}] \begin{Bmatrix} S \\ T \end{Bmatrix} + \begin{Bmatrix} x_o \\ y_o \end{Bmatrix}$$

is used, where the transformation constants $[A_{ij}]$ and x_o, y_o are determined as shown below so as to fill that portion of the x, y space required by the suer or automatically selected by the program. In the case of a perspective projection, $\{S\}$ and $\{T\}$ are replaced by $\{S'\}$ and $\{T'\}$. Since we desire no distortion of the plotted object and the orientation has already been specified, we set

$$[A_{ij}] = A \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Let

$$M = \frac{1}{2}[(T_{max} - T_{min}) - (S_{max} - S_{min})]$$

Then, to fill the available x, y space we require, for $M > 0$,

$$\begin{aligned} 0 &= AT_{min} + y_o \\ 1 &= AT_{max} + y_o \\ m &= AS_{min} + x_o \\ 1 - m &= AS_{max} + x_o \end{aligned} \tag{11-1}$$

or, for $M < 0$

$$\begin{aligned} m &= \mathrm{AT}_{min} + y_o \\ 1 - m &= \mathrm{AT}_{max} + y_o \\ 0 &= \mathrm{AS}_{min} + x_o \\ 1 &= \mathrm{AS}_{max} + x_o \end{aligned} \tag{11-2}$$

where m is the horizontal margin for $M > 0$ and the vertical margin for $M < 0$.

From equations (11-1) and (11-2) the quantities m, A, x_o, y_o, and thus the plotter coordinates, are determined.

Deformation Scaling

In plotting deformed structures, the components of displacement are added to the coordinates of the undeformed grid points to obtain the coordinates of the grid points of the deformed structure. Since the numerical magnitude of the physical structural displacements is usually much smaller than the size of the structure itself, additional scaling must be performed on the displacement vectors in order to obtain a viewable plot. In MSC Nastran this is done by the user who specified a value for the magnitude of the maximum structural deflection in units of length of the undeformed structural model. Thus, if max $\{\mathrm{T}_{max} - \mathrm{T}_{min}, \mathrm{S}_{max} - \mathrm{S}_{min}\}$ were, say, 1000 units, a specification of 50 would result in a plotted maximum deformation equal to 5% of the maximum plot size. In addition, the scale of the deformed structure, described in Projection Plane to Plotter Transformations, 469, is reduced by 5% to accommodate the deformation vectors.

Postprocessors

The plotps postprocessor reads plotting commands from a single MSC Nastran binary or neutral format plot file and produces a file which can be printed or viewed on a PostScript device. Each image will be oriented to best correlate the plot's page size with the printable page size (assumed to be 7.5 inches wide and 10.0 inches tall). Command line arguments can be specified in any order, command keywords can be abbreviated by truncation. These command keywords are discussed in PLOTPS (p. 35) in the .

Examples

- Translate a binary format plot file into PostScript:

  mscplotps example
- Translate a neutral format plot file into PostScript:

  mscplotps example

12 Interface With Other Programs

- Introduction 472
- POST Case Control Command 472
- PUNCH Output 478
- Patran Support of Heat Transfer Analysis 509
- The Nastran HDF5 Result Database (NH5RDB) 511
- NH5RDB Data Access 522

Introduction

MSC Nastran provides for direct support of interface to MSC's interactive graphics products, MSC Access, Patran, and SDRC I-DEAS programs, and to other outside vendor programs.

For Patran, the support is provided via the MSC Nastran DBC module, which creates a "graphics" database. The interface to the DBC module is described in the POST (p. 974) in the *MSC Nastran Quick Reference Guide*.

The graphics database created by the MSC Nastran DBC module, can also be read by the MSC Access library of object routines. This object library can be linked with a user-created program that extracts data from the database. This is an indirect method for outside vendors to obtain MSC Nastran model and results information. See the *MSC Access Reference Guide* for more information.

Communication with Patran, SDRC I-DEAS, LMS International's MSC_NF interface, and Dynamic Design Solutions' SYSTUNE interface, is through the MSC Nastran OUTPUT2 module, which creates a FORTRAN readable file that is converted by these outside programs to their own particular data formats. The user interface and the supported data for these outside programs are described in POST (p. 974) in the *MSC Nastran Quick Reference Guide* (-1, -2, -4, and -5).

Sometimes just a direct punch output file is useful for creating input to external programs.

POST Case Control Command

Introduction

Finite element analysis of large structural components sometimes requires the consideration of many different loading scenarios to complete the simulation. This can be especially true for static analysis, which may require several thousand different load cases to be evaluated. During this process, recovery and postprocessing of large amounts of response data usually occurs. Many analysis organizations use commercially available pre-/postprocessors, like Patran, to graphically display results data. These programs typically read the results data from a fortran file generated by the OUTPUT2 module in MSC Nastran. This file contains all of the output requested by the case control data recovery requests in each and every subcase. Occasionally, output data is required for a subcase that does not need to be handled by the post-processing program.

Theory

The amount of data recovered for each subcase is dependent upon the case control commands that are present. If a request is placed above all subcases, that request applies to all of the subcases. Data is generated and produced one time for all subcases by examining the case control requests in each subcase. Each separate type of response is stored in its own data block. For example, displacement results are computed and stored separately from element stress results. These data blocks will include results for any subcase that contains a data recovery request with a print, plot or punch destination. The data blocks are then written to an external file by the OUTPUT2 module. There is no way to control the placement of the results data on the output file except through the case control request for that data. For example, if there are three subcases and displacement results for the first and third subcases are required for post-processing, but not the second subcase, one could simply request displacement output for only the first and third subcases. However, if punched displacement results for the second subcase were also required, displacements for all three subcases would be placed on the output file, even though

results for the second subcase are not needed nor wanted in the postprocessor. This applies to all output quantities requested by case control commands. There is no way to handle this situation except to run the analysis twice and change the data recovery requests. A simple method is needed to exercise more control over the contents of the output file.

The new POST Case Control command gives the user more control over the contents of the output file produced by the MSC Nastran program when a PARAM,POST Bulk Data entry is present in the input. It does not affect the contents of the results data blocks, only the amount of data that is transferred from those data blocks to the output file. This is accomplished by giving the OUTPUT2 module access to the case control requests for each subcase. The POST command is examined for the output requested in each subcase. Only that output requested by the POST command is placed on the output file for the subcase. Note that the data must be made available by the presence of a data recovery command in the subcase. For example, the POST command cannot output stress data if no stress data has been requested.

The operations associated with the POST Case Control command are closely related to the operations described under the PARAM POST discussion in Parameters (p. 785) in the *MSC Nastran Quick Reference Guide*. All of the DMAP parameters discussed in the QRG related to the POST parameter are honored and can be used to control the output produced. The POST parameter value itself is supplied by the POST Case Control command through selection of one of the supported post-processor programs shown in the following table.

ppname	Product	PARAM,POST,Value
PATRAN	MSC/PATRAN V3	-1
SDRC	SDRC IDEA-S	-2
NF	MSC/LMS NF	-4
FEMTOOLS	DDS/FemTools	-5
UNIGRAHICS	EDS/Unigraphics	-6

Inputs

The POST Case Control command is designed to give the user a way to limit the output of data recovery results for a particular subcase. The general format of the POST command is:

POST [ppname] [TOFILE furn] oplist

The POST command recognizes only a sub-set of all of the results data that can be produced by MSC Nastran. This sub-set is summarized in the following table.

Output Item	oplist Keyword	Case Command
Displacements	[NO]DISPLACE	DISP
Forces of Single Point Constraint	[NO]SPCFORCE	SPCFORCE
Element Forces	[NO]FORCES	ELFO/FORCE
Element Stresses	[NO]STRESS	ELST/STRESS

Output Item	oplist Keyword	Case Command
Element Strain Energy	[NO]ESE	ESE
Grid Point Force Balance	[NO]GPFORCE	GPFORCE
Stress at Grid Points	[NO]GPSIGMA	STRESS
Strain/Curvature at Grid Points	[NO]GPEPSILON	STRAIN
Composite Element Failure Indices	[NO]PLYFAILURE	STRESS
Element Kinetic Energy	[NO]EKE	EKE
Element Energy Loss	[NO]EDE	EDE
Multi-point Constraint Forces	[NO]MPCFORCE	MPCFORCE
Composite Lamina Stresses	[NO]PLYSIGMA	STRESS
Composite Lamina Strains	[NO]PLYEPSILON	STRAIN
Element Strains	[NO]STRAIN	STRAIN
Grid Point Stresses	[NO]GPSTRESS	GPSTRESS
Grid Point Strains	[NO]GPSTRAIN	GPSTRAIN
Applied Loads	[NO]LOAD	OLOAD
No items to be output	NONE	----------

The POST command permits the optional selection of the target postprocessor program as well as the target fortran file unit reference number where the output will be stored.

Outputs

There are no special output considerations when using the POST Case Control command.

Guidelines and Limitations

The new POST Case Control command is intended primarily for static analysis. It is considered to be an extension of the parameters associated with the POST DMAP parameter discussed in Parameters (p. 785) in the *MSC Nastran Quick Reference Guide*. The POST command recognizes specification of any of the commercial post-processor products currently supported by the MSC Nastran solution sequences. The following guidelines should be kept in mind when using the POST command.

- Use of the POST command is entirely optional. If it is not used and a PARAM,POST,x entry is present in the input file, a postprocessor output file is generated just as in previous releases of the program.
- If any POST command operations are to be performed, a POST command must be specified above all subcases.
- When a POST command is encountered, a “PARAM,POST,x” entry is inserted in the Case Control at that point. All other POST-related DMAP parameters are available to the user to control output of data as before.

- Once a particular postprocessor program is specified, it cannot be changed in any subsequent subcases.
- The POST command does not allow a change in specification of the output fortran unit reference number if it is associated with a form=formatted OUTPUT2 file.
- POST command options are limited to SORT1 formatted output data at the present time.
- As is presently the case, only data that has been generated by the presence of a case control data recovery request command will be available for postprocessing output.
- There is no control over any geometry data output or over any data blocks that are not generated from a case control request. These data blocks will continue to be placed on the OUTPUT2 unit specified by the OUNIT2 parameter depending upon the value of the POST parameter (and the values of any related parameters).
- The placement of the POST command above the subcase level causes a cumulative effect on POST commands in subsequent subcases. Any options specified above the subcase level propagate down into the POST command within a subsequent subcase. Thus, if a POST command specifies NODISP (no displacement output wanted) above the subcase level, then a POST command with the DISP option would be required within a subcase to produce any output on the OUTPUT2 file for displacements. This also implies that changing the OUTPUT2 file unit reference number with the TOFILE option in a subcase causes **all** output quantities currently scheduled for output to be switched to the new unit number, not just those in the option list for the current POST command.

Demonstration Example

A simple model is presented to demonstrate the usage of the POST Case Control command to eliminate unwanted output from being placed on the postprocessor data file generated by MSC Nastran. The model data itself is not important for this example. It is the subcase structure that is the essence of the discussion. As an example of how this new feature might be used, consider the case of a static simulation performed on a model requiring three load cases. The output required is displacements at three grid points in the model. Furthermore, the output for the second load case must also include punched displacement data for all grid points. Displacement results for subcases 100 and 300 are to be placed on the default OUTPUT2 file for use by the Patran postprocessor. No output for subcase 200 is required for post-processing by Patran. The subcase structure might look something like:

```
SET 1000  =  10, 11, 12
DISP = 1000
$
POST$ use all defaults: Patran v3.0, unit=12
$
SUBCASE 100
LOAD = 100
$
SUBCASE200
LOAD = 200
DISP(PUNCH) = ALL
POST  NODISP  $  stop any displacement output from going to POST file
$
SUBCASE300
LOAD = 300
```

In this example, the presence of the POST command above all of the subcases indicates that all output requested (DISP for SET 1000 is the only output requested and generated) is to be stored on the default OUTPUT2 fortran unit for use by MSC.Patran V3.0. The presence of the POST command in SUBCASE 200 with the NODISP option prevents the large output produced by the DISP(PUNCH)=ALL request in SUBCASE 200 from being placed on the OUTPUT2 file.

Model Description

The model used for this example is very simple as the intent is to focus on the subcase structure. It is a cantilever plate structure consisting of only six CQUAD4 shell elements and fourteen grid points. There are three subcases used to apply loads and recover displacements.

Input File

```
$
$*********************************************************************
$
$Minor Enhancement A01130/A01131 - new POST case control capability
$VERSION: 2004
$TEST DECK NAME: postrnex.dat
$
$PURPOSE:
$   Simple case control setup to demonstrate usage of the new POST Case
$   Control Command for the Release Notes
$
$DESCRIPTION:
$   An OP2 file is to be generated for the MSC.Patran V3.0 post-
$   processor.  Three subcases are present in case control.
$   Displacement output is requested above the subcase level.
$   Displacements for subcases 100 and 300 are to be placed on the
$   default OP2 file.  Displacements for subcase 200 are to be placed on
$   system punch file but not on the OP2 file.
$
$EXPECTED RESULTS:
$   The following UIM 4114 should be present in the .f06 output:
$
$*** USER INFORMATION MESSAGE 4114 (OUTPX2)
$    DATA BLOCK OUGV1    WRITTEN ON FORTRAN UNIT 12, TRL =
$                  101    0  160    0    0    0    0
$    (MAXIMUM POSSIBLE FORTRAN RECORD SIZE =    xxxxx WORDS.)
$    (MAXIMUM SIZE OF FORTRAN RECORDS WRITTEN =      146 WORDS.)
$    (NUMBER OF FORTRAN RECORDS WRITTEN =       30 RECORDS.)
$    (TOTAL DATA WRITTEN FOR DATA BLOCK =      381 WORDS.)
$    (DATA FOR 2 CASE(S) WAS OUTPUT TO THIS UNIT. CASE ID LIST FOLLOWS.)
$               100       300
$
$*********************************************************************
$
ID MSC,POSTRNEX
SOL 101
TIME 5
CEND
TITLE = POST CASE CONTROL COMMAND EXAMPLE FOR RELEASE NOTES
SUBTITLE = DEMONSTRATE POST COMMAND USAGE
$
```

```
SPC = 1
SEALL = ALL
SET 1000 = 10,11,12
DISP   = 1000
$
$  place post command above subcase level.  Nothing specified causes
$  all defaults to be taken:
$  POST PATRAN TOFILE 12 (all outputs requested via CC commands)
$  or, for the example here where only DISP is requested,
$  POST PATRAN TOFILE 12 DISP
$  Note that POST command options specified above the subcase level
$  flow down into all of the subcases.
$
POST  patran tofile 12
$
$-----------------------------------------------------------------
subcase 100
LOAD = 100
$-----------------------------------------------------------------
subcase 200
LOAD = 200
DISP(PUNCH)=ALL    $  request punched displacement output for all grids
POST NODISP        $  no displacement output to default OP2 file.
$-----------------------------------------------------------------
subcase 300
LOAD = 100
$-----------------------------------------------------------------
BEGIN BULK
CQUAD4  1       1       1       4       5       2
CQUAD4  2       1       5       6       3       2
CQUAD4  3       1       4       7       8       5
CQUAD4  4       1       8       9       6       5
CQUAD4  5       1       7       10      11      8
CQUAD4  6       1       11      12      9       8
$
PSHELL  1       100     0.001   100
MAT1    100     10.7+6          0.33
$
FORCE   1       10              1.                      10.
FORCE   1       11              1.                      20.
FORCE   1       12              1.                      10.
GRID    1               0.      1.75    0.
GRID    2               0.      2.275   0.
GRID    3               0.      2.8     0.
GRID    4               0.6     1.75    0.
GRID    5               0.6     2.275   0.
GRID    6               0.6     2.8     0.
GRID    7               1.2     1.75    0.
GRID    8               1.2     2.275   0.
GRID    9               1.2     2.8     0.
GRID    10              1.8     1.75    0.
GRID    11              1.8     2.275   0.
GRID    12              1.8     2.8     0.
GRID    13              0.0     1.75    1.
GRID    14              1.0     1.75    1.
LOAD    100     1.      1.      1
LOAD    200     1.      2.      1
```

```
LOAD     300      1.       3.       1
PARAM    AUTOSPC  YES
SPC      1        1        13456
SPC      1        2        123456
SPC      1        3        13456
ENDDATA
```

PUNCH Output

Description of the MSC Nastran System Cell

System cell 210 is used to control the punch formatting, OFP tables being processed, and determine if the line number is placed in columns 73 through 80. It should be noted that this only effects the OFP punch capability and does not impact other punch options such as ECHO = PUNCH, Composite Element, synthesized element, and material property values, and other areas addressed outside of the OFP environment. The system cell values are:

Value	Selection
0	“OLD” Punch, default in V2001 and earlier
1	“NEW” Punch, default in V2004 and uses NDDL
2	Same as 1 except the line number is eliminated

The term “OLD” punch refers to the algorithm used to process the entries of the output table. In MSC Nastran releases prior to the current release, assumed was that the first item of an entry could be either integer or real depending upon the table sort condition. The remaining entry items were assumed to be real. This led to incorrect translation of the binary when the entry item was either integer or character in data value. This could cause “not a number” or numerical range value exceptions in the translated punch file.

The “New” punch has been implemented in the time frame of the new executive system and uses the Nastran Data Definition Language (NDDL) interface to correctly translate the entry data items. The implication is that the reading program must better understand the true format of the entry within an output table class.

Review of OFP Table Codes and Punch Header Conventions

Table 12-1 Structural Punch Table Support and Descriptor Entry

Table Code	Table Support	Table Content and Descriptor
1	Old	$DISPLACEMENTS
2	Old	$OLOADS
3	Old	$SPCF
4	Old	$ELEMENT FORCES
5	Old	$ELEMENT STRESSES
6	None	$EIGENVALUE SUMMARY
7	Old	$EIGENVECTOR
8	None	$GRID POINT SINGULARITY TABLE
9	None	$EIGENVALUE ANALYSIS SUMMARY
10	Old	$VELOCITY
11	Old	$ACCELERATION
12	Old	$NON-LINEAR-FORCES
13	None	$GRID POINT WEIGHT OUTPUT
14	Old	$EIGENVECTOR (SOLUTION SET)
15	Old	$DISPLACEMENTS (SOLUTION SET)
16	Old	$VELOCITY (SOLUTION SET)
17	Old	$ACCELERATION (SOLUTION SET)
18	Old	$ELEMENT STRAIN ENERGIES
19	Old	$GRID POINT FORCE BALANCE
20	None	$STRESS AT GRID POINTS
21	None	$STRAIN/CURVATURE AT GRID POINTS
22	None	$ELEMENT INTERNAL FORCES AND MOMENTS
23	None	$ELEMENT ORIENTED FORCES
24	None	$ELEMENT PRESSURES
25	None	$COMPOSITE FAILURE INDICIES
26	New	$GRID POINT STRESS/PLANE STRESS
27	New	$GRID POINT STRESS VOLUME DIRECT
28	New	$GRID POINT STRESS VOLUME PRINCIPAL

Table 12-1 Structural Punch Table Support and Descriptor Entry (continued)

Table Code	Table Support	Table Content and Descriptor
29	New	$ELEMENT STRESS DISCONTINUITIES
30	New	$ELEMENT STRESS DISCONTINUITIES DIRECT
31	New	$ELEMENT STRESS DISCONTINUITIES PRINCIPAL
32	New	$GRID POINT STRESS DISCONTINUITIES
33	New	$GRID POINT SRESS DISCONTINUITIES DIRECT
34	New	$GRID POINT STRESS DISCON PRINCIPAL
35	New	$GRID POINT STRESS/PLAIN STRAIN
36	Old	$ELEMENT KINETIC ENERGY
37	Old	$ELEMENT ENERGY LOSS PER CYCLE
38	New	$MAX/MIN SUMMARY INFORMATION
39	Old	$MPCF
40	Old	$MODAL GRID POINT KINETIC ENERGY

Note:

- The “new” table format supports all of the “old” table formats for punch.
- The word, “stress” in referenced table codes can be replaced by “strain”.
- The “none” table format implies that there is no Case Control interface to select punch as an option, only print is supported.
- When RANDOM is selected, then the Table Descriptor of Table codes 1 through 5 can be augmented by the strings of - PSDF, - AUTO, - RMS, - NO, or - CRMS, to denote the entry formulation.

Other impacted aspects of RANDOM analysis are the replacement of the $SUBCASE ID entry with $RANDOM ID entry and two additional Table Descriptions that have no base table augmentation. The table descriptors are:

```
$CROSS-PSDF
$CROSS-CORRELATION FUNCTION
```

Starting with the current version, when thermal analysis has been selected, the structural table descriptor strings have been replaced by their correct thermal designation. Table 12-2 associates the table code to the thermal table descriptor string. In addition, with the exception of temperature, which formats the punch output to bulk data conventions, the scalar class entries have been shortened to a single data item.

Table 12-2 Alternate Punch Descriptor for Thermal Analysis

Table Code	Thermal Descriptor Entry
1	$TEMPERATURE
2	$HEAT FLOW AT LOAD POINTS
3	$HEAT FLOW AT CONTSTRAINT POINTS
4	$ELEMENT GRADIENTS AND FLUXES
10	$ENTHALPY
11	$H DOT

Description of an OFP Punch Header Block

The OFP Punch header is composed of six to nine entries, depending upon sort conditions, analysis approach, grid versus element responses, and other conditions. In addition, column one has the following meaning. A "$" indicates the entry belongs to the header section. A "-" indicates the continuation of item information for an entry. A blank in column one implies the start of an entry. In general, the punch header entries are:

```
$TITLE = Case Control Title
$SUBTITLE = Case Control Subtitle
$LABEL = Case Control Lable
$Table Descriptor
$MAXMIN (Table code = 38) additional information
$Output Format
$SUBCASE ID or RANDOM ID, missing for SORT 2 STATICS
$ELEMENT TYPE = missing for grid table codes
```

Approach or Sort Specific Information

Note that entries one through four are always be present. The MAXMIN additional information entry is described in "MAX/MIN/RMS Methodology in Data Recovery" in *MSC.Nastran 2004 Release Guide*.

The Table Descriptor entry has been augmented for the energy recovery tables 18, 36, and 37 to delineate the element type for non frequency response analysis approaches, and to include the calculation option for the Frequency Response formulation.

$ELEMENT STRAIN ENERGIES		BAR	OPT=AVERAGE =AMPLITUDE =PEAK

The Output Format entry has been augmented, when necessary, to include the superelement identification. The area remains blank when a residual structure is detected.

$REAL OUTPUT	SEID=10

The SUBCASE entry has been augmented to include the SURFACE or VOLUME designations when Grid Point Stress or Discontinuity tables are being processed. The keyword FIBER or STRCUR will be present for strain table recovery.

$SUBCASE ID =	1 SURFACE= VOLUME=	1 FIBER

The Element Type line has the most significant augmentation. The element name has now been appended to further identify the element enumeration. Because certain elements may contain variable user selectable responses, additional keywords are appended, when necessary, to reflect the entry content. The keywords VONM and SHEAR deal with the invariant selected; STRCUR or FIBER for the strain recovery, CUBIC for the QUAD4 corner extrapolation method, and MATERIAL for shell elements, when not in the element reference system.

$ELEMENT TYPE =	144	QUAD4C	VONM SHEAR	SRCUR FIBER	BILIN SGAGE CUBIC	MATERIAL

The last augmentation is for the sort 2 table condition and deals with the issues of approach and whether the entry identification is integer or real.

$SPOINT ID =	1	IDENTIFIED BY SUBACASE FREQUENCY TIME LOAD FACTOR

Example

The following are examples of header and entry changes between the MSC Nastran 2004 and later releases and previous releases of the OFP punch formats.

```
Sort2 Statics, the SUBCASE line was deleted since the SUBCASE identifies the entry and SUBCASE is not
global for all entries.
$TITLE   =CANTILEVER BEAM CONSTRUCTED OF PLATES, STRESS CONTOURS   F10         1
$SUBTITLE= REF: SINGER,STRENGTH OF MATERIALS, ART 5-2,PG 133                   2
$LABEL   =                                                                     3
$DISPLACEMENTS                                                                 4
$REAL OUTPUT                                                                   5
$SUBCASE ID =           1                                                      6
$POINT ID =          1                                                         6
         1       G      0.000000E+00      0.000000E+00      0.000000E+00       7
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00       8

$TITLE   =CANTILEVER BEAM CONSTRUCTED OF PLATES, STRESS CONTOURS   F10         1
$SUBTITLE= REF: SINGER,STRENGTH OF MATERIALS, ART 5-2,PG 133                   2
$LABEL   =                                                                     3
$DISPLACEMENTS                                                                 4
$REAL OUTPUT                                                                   5
$POINT ID =          1  IDENTIFIED BY SUBCASE                                  5
         1       G      0.000000E+00      0.000000E+00      0.000000E+00       6
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00       7
```

```
In superelement processing the block content was not identified.

$TITLE   =RESULT CASE NAME                                                    1
$SUBTITLE=LOAD CASE NAME 1                                                    2
$LABEL   =RESTRAINT CASE NAME 1                                               3
$DISPLACEMENTS                                                                4
$REAL OUTPUT                                                                  5
$SUBCASE ID =           1                                                     6
       101       G      0.000000E+00      9.994075E-04      0.000000E+00      7
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00      8

$TITLE   =RESULT CASE NAME                                                    1
$SUBTITLE=LOAD CASE NAME 1                                                    2
$LABEL   =RESTRAINT CASE NAME 1                                               3
$DISPLACEMENTS                                                                4
$REAL OUTPUT            SEID=10                                               5
$SUBCASE ID =           1                                                     6
       101       G      0.000000E+00      9.994075E-04      0.000000E+00      7
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00      8
```

```
When Multi-point Constraint Forces were recovered, they were incorrectly identified.
$SUBTITLE=LOAD CASE NAME 1                                                    2
$LABEL   =RESTRAINT CASE NAME 1                                               3
$SPCF                                                                         4
$REAL OUTPUT                                                                  5
$SUBCASE ID =           1                                                     6
         1       G      0.000000E+00      0.000000E+00      0.000000E+00      7
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00      8

$TITLE   =RESULT CASE NAME                                                    1
$SUBTITLE=LOAD CASE NAME 1                                                    2
$LABEL   =RESTRAINT CASE NAME 1                                               3
$MPCF                                                                         4
$REAL OUTPUT                                                                  5
$SUBCASE ID =           1                                                     6
         1       G      0.000000E+00      0.000000E+00      0.000000E+00      7
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00      8
```

To assist the user with the element type enumeration, the GPTABD name values were appended to the white area of the ELEMENT TYPE entry.

```
$TITLE   =RESULT CASE NAME                                                    1
$SUBTITLE=LOAD CASE NAME 1                                                    2
$LABEL   =RESTRAINT CASE NAME 1                                               3
$ELEMENT FORCEs                                                                 4
$REAL OUTPUT                                                                  5
$SUBCASE ID =           1                                                     6
$ELEMENT TYPE =          33                                                   7
         1             -3.906332E+01     -9.898468E+02     -3.342360E+01      8
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00       9
-CONT-                  0.000000E+00      0.000000E+00                        10

$TITLE   =RESULT CASE NAME                                                  745
$SUBTITLE=LOAD CASE NAME 1                                                  746
$LABEL   =RESTRAINT CASE NAME 1                                             747
$ELEMENT FORCEs                                                               748
$REAL OUTPUT                                                                749
$SUBCASE ID =           1                                                   750
$ELEMENT TYPE =          33  QUAD4                                          751
         1             -3.906332E+01     -9.898468E+02     -3.342360E+01    752
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00     753
-CONT-                  0.000000E+00      0.000000E+00                       754
```

When element classes had optional output selections the user could select from the Case Control entry, these selections were not indicated in the punch heading block.

```
$TITLE   =RESULT CASE NAME                                                    287
$SUBTITLE=LOAD CASE NAME 1                                                    288
$LABEL   =RESTRAINT CASE NAME 1                                               289
$ELEMENT STRAINS                                                              290
$REAL OUTPUT                                                                  291
$SUBCASE ID =           1                                                     292
$ELEMENT TYPE =          33                                                   293
         1             -5.000000E-02      8.596358E-05     -3.260426E-04     294
-CONT-                 -2.896712E-05     -2.010852E+00      8.647210E-05     295
-CONT-                 -3.265511E-04      2.515292E-04      5.000000E-02     296
-CONT-                  8.596358E-05     -3.260426E-04     -2.896712E-05     297
-CONT-                 -2.010852E+00      8.647210E-05     -3.265511E-04     298
-CONT-                  2.515292E-04                                         299

$TITLE   =RESULT CASE NAME                                                    287
$SUBTITLE=LOAD CASE NAME 1                                                    288
$LABEL   =RESTRAINT CASE NAME 1                                               289
$ELEMENT STRAINS                                                              290
$REAL OUTPUT                                                                  291
$SUBCASE ID =           1                                                     292
$ELEMENT TYPE =          33  QUAD4     VONM   FIBER                           293
         1             -5.000000E-02      8.596358E-05     -3.260426E-04     294
-CONT-                 -2.896712E-05     -2.010852E+00      8.647210E-05     295
-CONT-                 -3.265511E-04      2.515292E-04      5.000000E-02     296
-CONT-                  8.596358E-05     -3.260426E-04     -2.896712E-05     297
-CONT-                 -2.010852E+00      8.647210E-05     -3.265511E-04     298
-CONT-                  2.515292E-04                                         299

$TITLE   =RESULT CASE NAME                                                    287
$SUBTITLE=LOAD CASE NAME 1                                                    288
$LABEL   =RESTRAINT CASE NAME 1                                               289
$ELEMENT STRAINS                                                              290
$REAL OUTPUT                                                                  291
$SUBCASE ID =           1                                                     292
$ELEMENT TYPE =          33  QUAD4     VONM   STRCUR                          293
         1              0.000000E+00      8.596358E-05     -3.260426E-04     294
-CONT-                 -2.896712E-05     -2.010852E+00      8.647210E-05     295
-CONT-                 -3.265511E-04      2.515292E-04     -1.000000E+00     296
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00     297
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00     298
-CONT-                  0.000000E+00                                         299
```

When an element had other than real data items, the values were incorrectly formatted. In this example the integer are outside the IEEE exponent range.

```
$TITLE   =RESULT CASE NAME                                                      1
$SUBTITLE=LOAD CASE NAME 1                                                      2
$LABEL   =RESTRAINT CASE NAME 1                                                 3
$ELEMENT STRAINS                                                                4
$REAL OUTPUT                                                                    5
$SUBCASE ID =           1                                                       6
$ELEMENT TYPE =         144                                                     7
        1               1.973054E+02      5.605194E-45      0.000000E+00      8
-CONT-                  8.596358E-05     -3.260426E-04     -2.896712E-05      9
-CONT-                 -2.010852E+00      8.647210E-05     -3.265511E-04     10
-CONT-                  2.515292E-04     -1.000000E+00      0.000000E+00     11
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00     12
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00     13
-CONT-                  1.401298E-45      0.000000E+00      8.924844E-05     14
-CONT-                 -4.056804E-04     -1.781213E-04     -9.896718E+00     15
-CONT-                  1.047868E-04     -4.212187E-04      3.214852E-04     16
-CONT-                 -1.000000E+00      0.000000E+00      0.000000E+00     17
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00     18
-CONT-                  0.000000E+00      0.000000E+00      2.802597E-45     19
-CONT-                  0.000000E+00      8.894981E-05     -2.488935E-04     20
-CONT-                 -1.678175E-04     -1.320753E+01      1.086421E-04     21
-CONT-                 -2.685857E-04      2.242232E-04     -1.000000E+00     22
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00     23
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00     24
-CONT-                  0.000000E+00      1.261169E-44      0.000000E+00     25
-CONT-                  8.297735E-05     -2.488935E-04      1.065242E-04     26
-CONT-                  8.897819E+00      9.131589E-05     -2.572320E-04     27
-CONT-                  2.086957E-04     -1.000000E+00      0.000000E+00     28
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00     29
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00     30
-CONT-                  1.121039E-44      0.000000E+00      8.267872E-05     31
-CONT-                 -4.056804E-04      1.236546E-04      7.104448E+00     32
-CONT-                  9.038459E-05     -4.133863E-04      3.101407E-04     33
-CONT-                 -1.000000E+00      0.000000E+00      0.000000E+00     34
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00     35
-CONT-                  0.000000E+00      0.000000E+00                       36

$TITLE   =RESULT CASE NAME                                                      1
$SUBTITLE=LOAD CASE NAME 1                                                      2
$LABEL   =RESTRAINT CASE NAME 1                                                 3
$ELEMENT STRAINS                                                                4
$REAL OUTPUT                                                                    5
$SUBCASE ID =           1                                                       6
$ELEMENT TYPE =         144  QUAD4C    VONM   STRCUR  BILIN                     7
        1         CEN/                      4           0.000000E+00      8
-CONT-                  8.596358E-05     -3.260426E-04     -2.896712E-05      9
-CONT-                 -2.010852E+00      8.647210E-05     -3.265511E-04     10
-CONT-                  2.515292E-04     -1.000000E+00      0.000000E+00     11
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00     12
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00     13
-CONT-                     1              0.000000E+00      1.131398E-04     14
-CONT-                 -4.066659E-04     -2.896712E-05     -1.594806E+00     15
-CONT-                  1.135430E-04     -4.070691E-04      3.160993E-04     16
-CONT-                 -1.000000E+00      0.000000E+00      0.000000E+00     17
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00     18
-CONT-                  0.000000E+00      0.000000E+00         2             19
-CONT-                  0.000000E+00      6.580506E-05     -2.497893E-04     20
-CONT-                 -2.896712E-05     -2.622126E+00      6.646837E-05     21
-CONT-                 -2.504526E-04      1.929787E-04     -1.000000E+00     22
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00     23
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00     24
-CONT-                  0.000000E+00         9              0.000000E+00     25
-CONT-                  5.983260E-05     -2.479976E-04     -2.896712E-05     26
-CONT-                 -2.687879E+00      6.051255E-05     -2.486775E-04     27
-CONT-                  1.892093E-04     -1.000000E+00      0.000000E+00     28
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00     29
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00     30
-CONT-                     8              0.000000E+00      1.065701E-04     31
-CONT-                 -4.046950E-04     -2.896712E-05     -1.621391E+00     32
-CONT-                  1.069800E-04     -4.051049E-04      3.119066E-04     33
-CONT-                 -1.000000E+00      0.000000E+00      0.000000E+00     34
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00     35
-CONT-                  0.000000E+00      0.000000E+00                       36
```

```
Again the coordinate system was also not correctly indicated.

$TITLE   = MSC/NASTRAN JOB CREATED ON 11-NOV-99 AT 14:47:27                    24
$SUBTITLE=SUBCASE1                                                             25
$LABEL   =                                                                     26
$ELEMENT STRESSES                                                              27
$REAL OUTPUT                                                                   28
$SUBCASE ID =           1                                                      29
$ELEMENT TYPE =          33                                                    30
         1             -5.000000E-02      3.225000E+04      3.225000E+04      31
-CONT-                 -2.775000E+04     -4.500000E+01      6.000000E+04      32
-CONT-                  4.500000E+03      5.788134E+04      5.000000E-02      33
-CONT-                 -3.225000E+04     -3.225000E+04      2.775000E+04      34
-CONT-                  4.500000E+01     -4.500000E+03     -6.000000E+04      35
-CONT-                  5.788134E+04                                          36

$TITLE   = MSC/NASTRAN JOB CREATED ON 11-NOV-99 AT 14:47:27                    24
$SUBTITLE=SUBCASE1                                                             25
$LABEL   =                                                                     26
$ELEMENT STRESSES                                                              27
$REAL OUTPUT                                                                   28
$SUBCASE ID =           1                                                      29
$ELEMENT TYPE =          33  QUAD4     VONM                    MATERIAL      30
         1             -5.000000E-02      3.225000E+04      3.225000E+04      31
-CONT-                 -2.775000E+04     -4.500000E+01      6.000000E+04      32
-CONT-                  4.500000E+03      5.788134E+04      5.000000E-02      33
-CONT-                 -3.225000E+04     -3.225000E+04      2.775000E+04      34
-CONT-                  4.500000E+01     -4.500000E+03     -6.000000E+04      35
-CONT-                  5.788134E+04                                          36
```

```
This checks that SHEAR versus VONM is correctly identified.

$TITLE   =RESULT CASE NAME                                                    7509
$SUBTITLE=LOAD CASE NAME 1                                                    7510
$LABEL   =RESTRAINT CASE NAME 1                                               7511
$ELEMENT STRESSES                                                             7512
$REAL OUTPUT                                                                  7513
$SUBCASE ID =           1                                                     7514
$ELEMENT TYPE =          33                                                   7515
         1             -5.000000E-02     -3.906332E+02     -9.898468E+03    7516
-CONT-                 -3.342360E+02     -2.010852E+00     -3.788981E+02    7517
-CONT-                 -9.910203E+03      4.765653E+03      5.000000E-02    7518
-CONT-                 -3.906332E+02     -9.898468E+03     -3.342360E+02    7519
-CONT-                 -2.010852E+00     -3.788981E+02     -9.910203E+03    7520
-CONT-                  4.765653E+03                                         7521

$TITLE   =RESULT CASE NAME                                                    8433
$SUBTITLE=LOAD CASE NAME 1                                                    8434
$LABEL   =RESTRAINT CASE NAME 1                                               8435
$ELEMENT STRESSES                                                             8436
$REAL OUTPUT                                                                  8437
$SUBCASE ID =           1                                                     8438
$ELEMENT TYPE =          33  QUAD4      SHEAR                                 8439
         1             -5.000000E-02     -3.906332E+02     -9.898468E+03    8440
-CONT-                 -3.342360E+02     -2.010852E+00     -3.788981E+02    8441
-CONT-                 -9.910203E+03      4.765653E+03      5.000000E-02    8442
-CONT-                 -3.906332E+02     -9.898468E+03     -3.342360E+02    8443
-CONT-                 -2.010852E+00     -3.788981E+02     -9.910203E+03    8444
-CONT-                  4.765653E+03                                         8445
```

```
Check if element class name exists for Laminated Composite and that the invariant
is eliminated since SHEAR context is only available.

$TITLE   =SQUARE PLATE WITH CIRCULAR HOLE - COMPOSITES SOLUTION                1
$SUBTITLE=MEMBRANE PROPERTIES OF QUAD4 AND TRIA3 ELEMENTS                      2
$LABEL   =UNIFORM LOAD ALONG X=5.0.                                            3
$ELEMENT STRESSES                                                              4
$REAL OUTPUT                                                                   5
$SUBCASE ID =           1                                                      6
$ELEMENT TYPE =          95                                                    7
         1                1              6.713920E+03     -2.537445E+03       8
-CONT-                  1.848406E+02      0.000000E+00      0.000000E+00       9
-CONT-                  1.144151E+00      6.717611E+03     -2.541137E+03      10
-CONT-                  4.629374E+03                                          11

$TITLE   =SQUARE PLATE WITH CIRCULAR HOLE - COMPOSITES SOLUTION                1
$SUBTITLE=MEMBRANE PROPERTIES OF QUAD4 AND TRIA3 ELEMENTS                      2
$LABEL   =UNIFORM LOAD ALONG X=5.0.                                            3
$ELEMENT STRESSES                                                              4
$REAL OUTPUT                                                                   5
$SUBCASE ID =           1                                                      6
$ELEMENT TYPE =          95  QUAD4LC                                           7
         1                1              6.713920E+03     -2.537445E+03       8
-CONT-                  1.848406E+02      0.000000E+00      0.000000E+00       9
-CONT-                  1.144151E+00      6.717611E+03     -2.541137E+03      10
-CONT-                  4.629374E+03                                          11
```

```
The GPSDCON entry caused unidentified output to be created in the punch stream.

$TITLE   =RESULT CASE NAME                                                    831
$SUBTITLE=LOAD CASE NAME 1                                                    832
$LABEL   =RESTRAINT CASE NAME 1                                               833
$REAL OUTPUT                                                                  834
$SUBCASE ID =           1                                                     835
         1              1.358237E-19      3.892658E+02      2.825602E+03     836
-CONT-                  5.154498E+02      3.298798E+02      2.884989E+03     837
-CONT-                  1.277554E+03      2.729644E+03      1.673444E+03     838
         1              1.358237E-19      3.892658E+02      2.825602E+03     839
-CONT-                  5.154498E+02      3.298798E+02      2.884989E+03     840
-CONT-                  1.277554E+03      2.729644E+03      1.673444E+03     841
         1              1.738842E-19      3.892658E+02      2.825602E+03     842
-CONT-                  5.154498E+02      3.298798E+02      2.884989E+03     843
-CONT-                  1.277554E+03      2.729644E+03      1.673444E+03     844

$TITLE   =RESULT CASE NAME                                                    831
$SUBTITLE=LOAD CASE NAME 1                                                    832
$LABEL   =RESTRAINT CASE NAME 1                                               833
$GRID POINT STRESS DISCONTTINUITIES                                           834
$REAL OUTPUT                                                                  835
$SUBCASE ID =           1  SURFACE=            1                              836
         1          Z1                    3.892658E+02      2.825602E+03     837
-CONT-                  5.154498E+02      3.298798E+02      2.884989E+03     838
-CONT-                  1.277554E+03      2.729644E+03      1.673444E+03     839
         1          Z2                    3.892658E+02      2.825602E+03     840
-CONT-                  5.154498E+02      3.298798E+02      2.884989E+03     841
-CONT-                  1.277554E+03      2.729644E+03      1.673444E+03     842
         1          MID                   3.892658E+02      2.825602E+03     843
-CONT-                  5.154498E+02      3.298798E+02      2.884989E+03     844
-CONT-                  1.277554E+03      2.729644E+03      1.673444E+03     845
```

```
The ELSDCON entry caused unidentified output to be created in the punch stream.

$TITLE   =RESULT CASE NAME                                                    1867
$SUBTITLE=LOAD CASE NAME 1                                                    1868
$LABEL   =RESTRAINT CASE NAME 1                                               1869
$REAL OUTPUT                                                                  1870
$SUBCASE ID =           1                                                     1871
         1              5.724519E+10     1.491285E-07     1.358237E-19      1872
-CONT-                  3.240547E+02     2.101422E+03     6.456028E+02      1873
-CONT-                  2.751211E+02     2.147283E+03     9.770737E+02      1874
-CONT-                  2.051576E+02     1.282938E+03                       1875

$TITLE   =RESULT CASE NAME                                                    1869
$SUBTITLE=LOAD CASE NAME 1                                                    1870
$LABEL   =RESTRAINT CASE NAME 1                                               1871
$ELEMENT STRESS DISCONTINUITIES                                               1872
$REAL OUTPUT                                                                  1873
$SUBCASE ID =           1  SURFACE=             1                             1874
         1                      QUAD4                    Z1                   1875
-CONT-                  3.240547E+02     2.101422E+03     6.456028E+02      1876
-CONT-                  2.751211E+02     2.147283E+03     9.770737E+02      1877
-CONT-                  2.051576E+02     1.282938E+03                       1878
```

```
The GPSTRESS entry caused unidentified output to be created in the punch stream.

$TITLE   =RESULT CASE NAME                                                    287
$SUBTITLE=LOAD CASE NAME 1                                                    288
$LABEL   =RESTRAINT CASE NAME 1                                               289
$REAL OUTPUT                                                                  290
$SUBCASE ID =           1                                                     291
         1              0.000000E+00      1.358237E-19     -7.940824E+02     292
-CONT-                 -1.270989E+04      1.011802E+03      4.819154E+00     293
-CONT-                 -7.087778E+02     -1.279519E+04      6.043207E+03     294
-CONT-                  1.245593E+04                                         295

$TITLE   =RESULT CASE NAME                                                   2913
$SUBTITLE=LOAD CASE NAME 1                                                   2914
$LABEL   =RESTRAINT CASE NAME 1                                              2915
$GRID POINT STRESS/PLANE STRESS                                              2916
$REAL OUTPUT                                                                 2917
$SUBCASE ID =           1  SURFACE=            1                             2918
         1                 0          Z1                -7.940824E+02     2919
-CONT-                 -1.270989E+04      1.011802E+03      4.819154E+00     2920
-CONT-                 -7.087778E+02     -1.279519E+04      6.043207E+03     2921
-CONT-                  1.245593E+04                                        2922
         1                 0          Z2                -7.940824E+02     2923
-CONT-                 -1.270989E+04      1.011802E+03      4.819154E+00     2924
-CONT-                 -7.087778E+02     -1.279519E+04      6.043207E+03     2925
-CONT-                  1.245593E+04                                        2926
         1                 0         MID                -7.940824E+02     2927
-CONT-                 -1.270989E+04      1.011802E+03      4.819154E+00     2928
-CONT-                 -7.087778E+02     -1.279519E+04      6.043207E+03     2929
-CONT-                  1.245593E+04                                        2930
```

```
The GPSTRAIN entry caused unidentified output to be created in the punch stream. The
GPSTRESS phase II module also did not react to FIBER/STRCUR correctly.

$TITLE   =RESULT CASE NAME                                                    5665
$SUBTITLE=LOAD CASE NAME 1                                                    5666
$LABEL   =RESTRAINT CASE NAME 1                                               5667
$REAL OUTPUT                                                                  5668
$SUBCASE ID =           1                                                     5669
         1              0.000000E+00     1.358237E-19     1.006295E-04     5670
-CONT-                 -4.157221E-04     8.768948E-05     4.819154E+00     5671
-CONT-                  1.043260E-04    -4.194186E-04     5.237446E-04     5672
-CONT-                  4.801584E-04                                       5673
         1              0.000000E+00     1.358237E-19     1.006295E-04     5674
-CONT-                 -4.157221E-04     8.768948E-05     4.819154E+00     5675
-CONT-                  1.043260E-04    -4.194186E-04     5.237446E-04     5676
-CONT-                  4.801584E-04                                       5677
         1              0.000000E+00     1.738842E-19     1.299615E+04     5678
-CONT-                 -2.920491E+03    -1.808289E+03    -6.400696E+00     5679
-CONT-                  1.319900E+04    -3.123345E+03     8.161175E+03     5680
-CONT-                  1.500647E+04                                       5681

$TITLE   =RESULT CASE NAME                                                    5673
$SUBTITLE=LOAD CASE NAME 1                                                    5674
$LABEL   =RESTRAINT CASE NAME 1                                               5675
$GRID POINT STRAIN/PLANE STRESS                                               5676
$REAL OUTPUT                                                                  5677
$SUBCASE ID =           1  SURFACE=            1  FIBER                       5678
         1                 0          Z1                  1.006295E-04     5679
-CONT-                 -4.157221E-04     8.768948E-05     4.819154E+00     5680
-CONT-                  1.043260E-04    -4.194186E-04     5.237446E-04     5681
-CONT-                  4.801584E-04                                       5682
         1                 0          Z2                  1.006295E-04     5683
-CONT-                 -4.157221E-04     8.768948E-05     4.819154E+00     5684
-CONT-                  1.043260E-04    -4.194186E-04     5.237446E-04     5685
-CONT-                  4.801584E-04                                       5686
         1                 0          MID                 1.006295E-04     5687
-CONT-                 -4.157221E-04     8.768948E-05     4.819154E+00     5688
-CONT-                  1.043260E-04    -4.194186E-04     5.237446E-04     5689
-CONT-                  4.801584E-04                                       5690

$TITLE   =RESULT CASE NAME                                                    5673
$SUBTITLE=LOAD CASE NAME 1                                                    5674
$LABEL   =RESTRAINT CASE NAME 1                                               5675
$GRID POINT STRAIN/PLANE STRESS                                               5676
$REAL OUTPUT                                                                  5677
$SUBCASE ID =           1  SURFACE=            1  STRCUR                      5678
         1                 0          MID                 1.006295E-04     5679
-CONT-                 -4.157221E-04     8.768948E-05     4.819154E+00     5680
-CONT-                  1.043260E-04    -4.194186E-04     5.237446E-04     5681
-CONT-                  4.801584E-04                                       5682
         1                 0          CURV                 0.000000E+00    5683
-CONT-                  0.000000E+00     0.000000E+00     0.000000E+00     5684
-CONT-                  0.000000E+00     0.000000E+00     0.000000E+00     5685
-CONT-                  0.000000E+00                                       5686
```

```
The GPSTRESS entry caused unidentified output to be created in the punch stream.

$TITLE   = LINEAR ELEMENT CODE CHECK PROBLEM                                198
$SUBTITLE= ELASTIC LOADING BY FORCE                                         199
$LABEL   = CHECK GRID POINT STRESS OUTPUT FORMATS                           200
$REAL OUTPUT                                                                201
$SUBCASE ID =           2                                                   202
      3401              9.471862E+02      5.677020E+01                    203
-CONT-                  1.723746E+01              -NaNQ                   204

$TITLE   = LINEAR ELEMENT CODE CHECK PROBLEM                                199
$SUBTITLE= ELASTIC LOADING BY FORCE                                         200
$LABEL   = CHECK GRID POINT STRESS OUTPUT FORMATS                           201
$GRID POINT STRESS/PLAIN STRAIN                                             202
$REAL OUTPUT                                                                203
$SUBCASE ID =           2  SURFACE=            34                           204
      3401              9.471862E+02      5.677020E+01                    205
-CONT-                  1.723746E+01                                      206
```

```
The GPSTRESS entry caused unidentified output to be created in the punch stream.

$TITLE   = LINEAR ELEMENT CODE CHECK PROBLEM                                235
$SUBTITLE= ELASTIC LOADING BY FORCE                                         236
$LABEL   = CHECK GRID POINT STRESS OUTPUT FORMATS                           237
$REAL OUTPUT                                                                238
$SUBCASE ID =           2                                                   239
      3901              1.979894E+03      6.972885E+01     -2.529273E-14    240
-CONT-                  2.071350E-13     -3.198958E+00      2.251117E+01    241
-CONT-                 -6.832076E+02      1.946365E+03                      242

$TITLE   = LINEAR ELEMENT CODE CHECK PROBLEM                                237
$SUBTITLE= ELASTIC LOADING BY FORCE                                         238
$LABEL   = CHECK GRID POINT STRESS OUTPUT FORMATS                           239
$GRID POINT STRESS VOLUME DIRECT                                            240
$REAL OUTPUT                                                                241
$SUBCASE ID =           2   VOLUME=            39                           242
      3901              1.979894E+03      6.972885E+01     -2.529273E-14    243
-CONT-                  2.071350E-13     -3.198958E+00      2.251117E+01    244
-CONT-                 -6.832076E+02      1.946365E+03                      245
```

```
The GPSTRESS entry caused unidentified output to be created in the punch stream.

$TITLE   = LINEAR ELEMENT CODE CHECK PROBLEM                                  336
$SUBTITLE= ELASTIC LOADING BY FORCE                                           337
$LABEL   = CHECK GRID POINT STRESS OUTPUT FORMATS                             338
$REAL OUTPUT                                                                  339
$SUBCASE ID =           2                                                     340
      3901              1.980150E+03     -4.018156E-01      6.987475E+01     341
-CONT-                  9.999354E-01     -1.135504E-02     -5.369717E-04     342
-CONT-                 -1.903498E-05      4.556393E-02     -9.989614E-01     343
-CONT-                  1.136771E-02      9.988969E-01      4.556077E-02     344
-CONT-                 -6.832076E+02      1.946365E+03                       345

$TITLE   = LINEAR ELEMENT CODE CHECK PROBLEM                                  339
$SUBTITLE= ELASTIC LOADING BY FORCE                                           340
$LABEL   = CHECK GRID POINT STRESS OUTPUT FORMATS                             341
$GRID POINT STRESS VOLUME PRINCIPAL                                           342
$REAL OUTPUT                                                                  343
$SUBCASE ID =           2   VOLUME=           39                              344
      3901              1.980150E+03     -4.018156E-01      6.987475E+01     345
-CONT-                  9.999354E-01     -1.135504E-02     -5.369717E-04     346
-CONT-                 -1.903498E-05      4.556393E-02     -9.989614E-01     347
-CONT-                  1.136771E-02      9.988969E-01      4.556077E-02     348
-CONT-                 -6.832076E+02      1.946365E+03                       349
```

```
The element class did not identify the Element Strain Energy output.

$TITLE   =RESULT CASE NAME                                                    3043
$SUBTITLE=LOAD CASE NAME 1                                                    3044
$LABEL   =RESTRAINT CASE NAME 1                                               3045
$ELEMENT STRAIN ENERGIES                                                      3046
$REAL OUTPUT                                                                  3047
$SUBCASE ID =           1                                                     3048
         1              2.976484E-03      6.479465E-02      1.632466E+00    3049
         2              1.780166E-03      3.875217E-02      5.816491E-01    3050

$TITLE   =RESULT CASE NAME                                                    8977
$SUBTITLE=LOAD CASE NAME 1                                                    8978
$LABEL   =RESTRAINT CASE NAME 1                                               8979
$ELEMENT STRAIN ENERGIES                  QUAD4                               8980
$REAL OUTPUT                                                                  8981
$SUBCASE ID =           1                                                     8982
         1              2.976484E-03      6.479465E-02      1.632466E+00    8983
         2              1.780166E-03      3.875217E-02      5.816491E-01    8984
```

```
In eigenvalue analysis the eigenvalue was wrong and element class identifier
missing.

$TITLE   =RESULT_CASE_NAME                                                    2871
$SUBTITLE=LOAD CASE NAME                                                      2872
$LABEL   =RESTRAINT CASE NAME                                                 2873
$ELEMENT STRAIN ENERGIES                                                      2874
$REAL OUTPUT                                                                  2875
$SUBCASE ID =           1                                                     2876
$EIGENVALUE =    .3390093E+13  MODE =     1                                   2877
         1              5.414903E+02      6.852106E-02      4.061989E+02    2878
         2              3.771790E+04      4.772884E+00      2.829408E+04    2879

$TITLE   =RESULT_CASE_NAME                                                    4101
$SUBTITLE=LOAD CASE NAME                                                      4102
$LABEL   =RESTRAINT CASE NAME                                                 4103
$ELEMENT STRAIN ENERGIES                  TETRA                               4103
$REAL OUTPUT                                                                  4104
$SUBCASE ID =           1                                                     4105
$EIGENVALUE =  0.1580508E+07  MODE =     1                                    4106
         1              5.414903E+02      6.852106E-02      4.061989E+02    4107
         2              3.771790E+04      4.772884E+00      2.829408E+04    4108
```

```
The EKE entry did not identify the output, the eigenvalue was incorrect and the
element class type was missing.

$TITLE   =RESULT_CASE_NAME                                                    3389
$SUBTITLE=LOAD CASE NAME                                                      3390
$LABEL   =RESTRAINT CASE NAME                                                 3391
$REAL OUTPUT                                                                  3392
$SUBCASE ID =           1                                                     3393
$EIGENVALUE =   .3390093E+13  MODE =     1                                    3394
         1           6.581564E+04      8.328419E+00      4.937160E+04       3395
         2           3.031419E+02      3.836007E-02      2.274019E+02       3396

$TITLE   =RESULT_CASE_NAME                                                    4821
$SUBTITLE=LOAD CASE NAME                                                      4822
$LABEL   =RESTRAINT CASE NAME                                                 4823
$ELEMENT KINETIC ENERGY                  TETRA                                4823
$REAL OUTPUT                                                                  4824
$SUBCASE ID =           1                                                     4825
$EIGENVALUE =  0.1580508E+07  MODE =     1                                    4826
         1           6.581565E+04      8.328419E+00      4.937161E+04       4827
         2           3.031419E+02      3.836007E-02      2.274019E+02       4828
```

```
ESE in Frequency Response did not identify the method used for calculating the
response quantity.  The element class was also missing.

$TITLE   =                                                                    1
$SUBTITLE=                                                                    2
$LABEL   =                                                                    3
$ELEMENT STRAIN ENERGIES                                                      4
$REAL OUTPUT                                                                  5
$SUBCASE ID =           2                                                     6
$FREQUENCY =    .1000000E+01                                                  7
       101              2.102882E-05      1.820742E-06      1.051441E-07      8
       102              2.893353E-02      2.505157E-03      1.446677E-04      9
       103              2.029759E-01      1.757429E-02      1.014879E-03     10

$TITLE   =                                                                    1
$SUBTITLE=                                                                    2
$LABEL   =                                                                    3
$ELEMENT STRAIN ENERGIES                  BAR    OPT=AVERAGE                  4
$REAL OUTPUT                                                                  5
$SUBCASE ID =           2                                                     6
$FREQUENCY =   1.0000000E+00                                                  7
       101              3.086982E-05      2.673122E-06      1.543491E-07      8
       102              2.927053E-02      2.534634E-03      1.463527E-04      9
       103              2.036543E-01      1.763512E-02      1.018272E-03     10
```

```
In EKE output in Frequency Response the block was not identified, the method was
not indicated, and the element class was missing.

$TITLE   =                                                                    5666
$SUBTITLE=                                                                    5667
$LABEL   =                                                                    5668
$REAL OUTPUT                                                                  5669
$SUBCASE ID =           2                                                     5670
$FREQUENCY =    .1000000E+01                                                  5671
       101              4.627901E+01      5.389437E+00      2.313951E-01     5672
       102              3.740206E+01      4.355669E+00      1.870103E-01     5673
       103              2.966045E+01      3.454118E+00      1.483023E-01     5674

$TITLE   =                                                                    5668
$SUBTITLE=                                                                    5669
$LABEL   =                                                                    5670
$ELEMENT KINETIC ENERGY                    BAR    OPT=AVERAGE                5671
$REAL OUTPUT                                                                  5672
$SUBCASE ID =           2                                                     5673
$FREQUENCY =   1.0000000E+00                                                  5674
       101              4.627080E+01      5.390246E+00      2.313540E-01     5675
       102              3.739541E+01      4.356322E+00      1.869771E-01     5676
       103              2.965508E+01      3.454624E+00      1.482754E-01     5677
```

```
In EDE output in Frequency Response the block was not identified, the method was
not indicated, and the element class was missing.

$TITLE   =                                                                    9635
$SUBTITLE=                                                                    9636
$LABEL   =                                                                    9637
$REAL OUTPUT                                                                  9638
$SUBCASE ID =           2                                                     9639
$FREQUENCY =    .1000000E+01                                                  9640
      3042              6.970902E+02      2.664647E+01                      9641
      5042              1.918979E+03      7.335353E+01                      9642

$TITLE   =                                                                    9799
$SUBTITLE=                                                                    9800
$LABEL   =                                                                    9801
$ELEMENT ENERGY LOSS PER CYCLE              DAMP2  OPT=AVERAGE                9802
$REAL OUTPUT                                                                  9803
$SUBCASE ID =           2                                                     9804
$FREQUENCY =   1.0000000E+00                                                  9805
      3042              6.970200E+02      2.665085E+01                      9806
      5042              1.918356E+03      7.334915E+01                      9807
```

```
Nonlinear output blocks were not identified by the LOAD FACTOR.

$TITLE   =RESULT_CASE_1                                                      1
$SUBTITLE=                                                                   2
$LABEL   =                                                                   3
$ELEMENT STRESSES                                                            4
$REAL OUTPUT                                                                 5
$SUBCASE ID =           1                                                    6
$ELEMENT TYPE =          89                                                  7
        1              1.905082E+02      1.905082E+02      6.350274E-06      8
-CONT-                 6.350274E-06      0.000000E+00      0.000000E+00      9

$TITLE   =RESULT_CASE_1                                                      1
$SUBTITLE=                                                                   2
$LABEL   =                                                                   3
$ELEMENT STRESSES                                                            4
$REAL OUTPUT                                                                 5
$SUBCASE ID =           1                                                    6
$ELEMENT TYPE =          89 RODNL                                            7
$LOAD FACTOR =   1.0000000E-01                                               7
        1              1.905082E+02      1.905082E+02      6.350274E-06      8
-CONT-                 6.350274E-06      0.000000E+00      0.000000E+00      9
_____________________________________________________________________________Punch
format maintained, however block identifier changed to reflect content.

$TITLE   = RADIATION BOUNDARY CONDITION                                      1
$SUBTITLE=                                                                   2
$LABEL   =                                                                   3
$DISPLACEMENTS                                                               4
$REAL OUTPUT                                                                 5
$SUBCASE ID =           1                                                    6
$POINT ID =           1                                                      7
TEMP*                  1              1    7.270000E+02                     8
TEMP*                  2              1    7.270001E+02                     9
TEMP*                  3              1    7.270000E+02                    10

$TITLE   = RADIATION BOUNDARY CONDITION                                      1
$SUBTITLE=                                                                   2
$LABEL   =                                                                   3
$TEMPERATURE                                                                 4
$REAL OUTPUT                                                                 5
$SUBCASE ID =           1                                                    6
$POINT ID =           1 IDENTIFIED BY TIME                                   7
TEMP*                  1              1    7.270000E+02                     8
TEMP*                  2              1    7.270001E+02                     9
TEMP*                  3              1    7.270000E+02                    10
```

```
When THERMAL analysis, the problem is a single degree of freedom system, therefore
when the true content is identified, the scale of the output can be reduced.

$TITLE   = RADIATION BOUNDARY CONDITION                                          305
$SUBTITLE=                                                                       306
$LABEL   =                                                                       307
$VELOCITY                                                                        308
$REAL OUTPUT                                                                     309
$SUBCASE ID =           1                                                        310
$POINT ID =           1                                                          311
    0.000000E+00 S      1.817500E+00      0.000000E+00      0.000000E+00         312
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00         313

$TITLE   = RADIATION BOUNDARY CONDITION                                         1409
$SUBTITLE=                                                                      1410
$LABEL   =                                                                      1411
$ENTHALPY                                                                       1412
$REAL OUTPUT                                                                    1413
$SUBCASE ID =           1                                                       1414
$POINT ID =           1  IDENTIFIED BY TIME                                     1415
    0.000000E+00 S      1.817500E+00                                            1416
```

```
Same issues as previous.

$TITLE   = RADIATION BOUNDARY CONDITION                                    801
$SUBTITLE=                                                                 802
$LABEL   =                                                                 803
$ACCELERATION                                                              804
$REAL OUTPUT                                                               805
$SUBCASE ID =           1                                                  806
$POINT ID =           1                                                    807
    0.000000E+00 S      1.066395E-06      0.000000E+00      0.000000E+00   808
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00   809

$TITLE   = RADIATION BOUNDARY CONDITION                                   2817
$SUBTITLE=                                                                2818
$LABEL   =                                                                2819
$H DOT                                                                    2820
$REAL OUTPUT                                                              2821
$SUBCASE ID =           1                                                 2822
$POINT ID =           1  IDENTIFIED BY TIME                               2823
    0.000000E+00 S      2.730277E-04                                      2824
```

```
Same issues as ENTHLAPY.

$TITLE   =LARGE PROBLEM                                                    85
$SUBTITLE=                                                                 86
$LABEL   =                                                                 87
$OLOADS                                                                    88
$REAL OUTPUT                                                               89
$SUBCASE ID =           1                                                  90
       101       S      1.500000E+03      0.000000E+00      0.000000E+00   99
-CONT-                  0.000000E+00      0.000000E+00      0.000000E+00  100

$TITLE   =LARGE PROBLEM                                                    85
$SUBTITLE=                                                                 86
$LABEL   =                                                                 87
$HEAT FLOW AT LOAD POINTS                                                  88
$REAL OUTPUT                                                               89
$SUBCASE ID =           1                                                  90
$LOAD FACTOR =   1.0000000E+00                                             90
       101       S      1.500000E+03                                       95
```

```
Same issues as ENTHLAPY.

$TITLE   =LARGE PROBLEM                                                    247
$SUBTITLE=                                                                 248
$LABEL   =                                                                 249
$SPCF                                                                      250
$REAL OUTPUT                                                               251
$SUBCASE ID =           1                                                  252
        98       S     -2.466861E+02     0.000000E+00     0.000000E+00     257
-CONT-                  0.000000E+00     0.000000E+00     0.000000E+00     258
        99       S     -1.051691E+04     0.000000E+00     0.000000E+00     259
-CONT-                  0.000000E+00     0.000000E+00     0.000000E+00     260


$TITLE   =LARGE PROBLEM                                                    169
$SUBTITLE=                                                                 170
$LABEL   =                                                                 171
$HEAT FLOW AT CONTRAINT POINTS                                             172
$REAL OUTPUT                                                               173
$SUBCASE ID =           1                                                  174
$LOAD FACTOR =   1.0000000E+00                                             174
        98       S     -2.466861E+02                                      177
        99       S     -1.051691E+04                                      178
```

```
Thermal element data recovery contains mis-labeled and possible numerical errors.

$TITLE    = RADIATION BOUNDARY CONDITION                                    1649
$SUBTITLE=                                                                  1650
$LABEL    =                                                                 1651
$ELEMENT FORCES                                                             1652
$REAL OUTPUT                                                                1653
$SUBCASE ID =            1                                                  1654
$ELEMENT TYPE =           33                                                1655
$ELEMENT ID =          4                                                    1656
      0.000000E+00       5.724519E+10      1.491285E-07      0.000000E+00   1657
-CONT-                   0.000000E+00      1.401298E-45      0.000000E+00   1658
-CONT-                   0.000000E+00      1.401298E-45                     1659

$TITLE    = RADIATION BOUNDARY CONDITION                                    6233
$SUBTITLE=                                                                  6234
$LABEL    =                                                                 6235
$ELEMENT GRADIENTS AND FLUXES                                               6236
$REAL OUTPUT                                                                6237
$SUBCASE ID =            1                                                  6238
$ELEMENT TYPE =           33  QUAD4                                         6239
$ELEMENT ID =          4  IDENTIFIED BY TIME                                6240
      0.000000E+00              QUAD4                    0.000000E+00      6241
-CONT-                   0.000000E+00          1          0.000000E+00      6242
-CONT-                   0.000000E+00          1                            6243
```

Patran Support of Heat Transfer Analysis

Patran offers a comprehensive and integrated graphics interface to the linear, nonlinear, steady-state, and transient thermal analysis capabilities in MSC Nastran. From within the Patran environment, you can completely define your thermal analysis model, submit it to MSC Nastran for analysis, import MSC Nastran thermal analysis results into the Patran database, and visualize the results. Complete support is provided for analyses including conduction, convection, advection, and radiation.

The interface is designed for both new and advanced users of MSC Nastran and Patran. The online help system provides several guided example problems and describes all of the Patran menu interface to MSC Nastran's thermal solver. Basic and Advanced modes are included for most of the menu forms for thermal loads and boundary conditions. For example, the Basic convection form only requires you to define the convection coefficient and ambient fluid temperature, while the Advanced form includes options for a user-defined convection equation, time-dependent and temperature-dependent coefficients, and variable ambient temperatures.

A brief overview of capabilities and limitations is given in the sections that follow.

Loads and Boundary Conditions

The following thermal loads and boundary conditions are supported, each of which can vary both spatially and temporally:

- Prescribed nodal temperatures
- Heat flux normal to a surface
- Vector heat flux from a distant radiant heat source
- Heat applied directly to a node (nodal source)
- Volumetric heat generation within conduction elements
- Basic convection
- Advection with forced convection boundaries
- Radiation exchange with space
- Radiation exchange within enclosures (cavity radiation)
- Multiple enclosures

Results Processing

- The following results quantities can be postprocessed within Patran:
- Nodal temperatures
- Applied heat loads
- Heats of constraint for temperature boundary conditions
- Heat fluxes within conduction elements
- Heat fluxes at boundaries due to applied heat loads, free/forced convection, advection, and radiation
- Temperature gradients
- Enthalpies
- Rate of change of enthalpies

Limitations

When using Patran to pre- and postprocess heat transfer analyses in MSC Nastran, keep in mind the following limitations:

- There is currently no support of MSC Nastran heat transfer Bulk Data entries through Patran's NASTRAN Input File Reader.
- Emissivity as a function of wavelength is currently not supported. The RADBND Bulk Data entry is used to specify the wavelength break points for radiation problems.
- Automatic restart is not supported for the Patran interface to MSC Nastran heat transfer analysis; this includes:
  - Restarting from a previously converged time step (SOL 159 to SOL 159)
  - Using a steady-state analysis as a restart to a transient analysis (SOL 159 from SOL 153)

- Using a steady-state analysis as a restart to a thermal stress analysis (SOL 106 from SOL 153)
- Using various time steps from a transient thermal analysis as a restart to a quasi-static thermal stress analysis (SOL 106 from SOL 159)

To perform many of the restarts listed above, you must modify the cold start run and manually insert the bulk data entries for the restart.

- The global view factor is currently not supported. Specifically, the RADCAV Bulk Data entry, SETij, are not supported. Basic view-factor calculations are supported.
- NOLIN Bulk Data entries are not supported. The NOLIN entries are used for various thermal-state control functions (e.g., the nonlinear transient power function).
- The `TLOAD2` Bulk Data entry is not supported. This entry enables you to specify various functions such as SIN, COS, and EXP when applying a thermal load. One work-around is to use the Field capability to create these functions. For example, you can create a PCL (PATRAN Command Language) function that can then be mapped to a table using the Create/Non-Spatial/Tabular Input field capability.
- The ambient element (ELEAMB) field for the RADCAV Bulk Data entry is not supported. For radiation enclosure problems, this means that for an incomplete cavity, all energy exchange carried on outside the enclosure surface is lost to zero-degree space. Patran will not automatically generate an ambient element that could have a nonzero temperature and a distinct area associated with it.

Documentation

For detailed documentation, refer to the *Patran Interface to MSC Nastran Thermal,*. This document is available online as part of the help system for Patran.

The Nastran HDF5 Result Database (NH5RDB)

Introduction

The NH5RDB is a Nastran database file in Hierarchical Data Format (HDF5). The database organizes Nastran input and output data in a hierarchical structure. Nastran data is stored as dataset in database and can be accessed through standard HDF5 APIs or third party packages in multiple programming languages. The main features of NH5RDB include:

- A data type schema in XML defines database structure and datasets formats.
- All main Nastran input and output data are covered
- Supported by multiple Nastran solutions.
- Optional output by MDLPRM parameter and system cell.
- New DMAP modules to write Nastran data blocks in database.
- An open format and multiple programming language support for data access.

The NH5RDB supports wide range of data types and is capable of defining complicated data structures. In the database, the datasets are stored in a hierarchical structure, making it easy to add, remove or update datasets in applications. The database supports high precision, compression and unlimited amount of data. Its open format and multiple programming languages support make it ideal for FEA applications.

The NH5RDB has modal input and result data. It organizes data in groups and datasets. The input data has groups like NODE, ELEMENT, CONSTRAINT, LOAD, etc. The result data includes ACOUSTIC, CONTACT, NODAL, ELEMTENTAL and other groups. In each group, data are stored in table like datasets. A dataset in NH5RDB is defined by a compound structure with member fields. For example, the `GRID` dataset in the input NODE group has fields for grid number and coordinates, it will have all grid point definitions from the modal. The DISPLACEMENT dataset in RESULT group has fields for displacement components and will contain all nodal displacement from all output steps. A dataset in NH5RDB could be identified by its dataset name and group path defined in the schema.

There are some options available for NH5RDB output. The HDF5 parameter in MDLPRM is used to control NH5RDB generation. By default, NH5RDB will not be generated. Data compression can be selected for NH5RDBB. Also the input data can be selected not to be included in NH5RDB by HDF5 parameter options.

Since HDF5 is an open format architecture. The NH5RDB data can be accessed by standard HDF5 library APIs or third party packages. Multiple programming languages, such as C/C++, FORTRAN, Python and Java, are supported. The example section gives examples of NH5RDB data access in multiple languages.

Data Schema

The data schema is used to define database structure and its datasets formats. The database is in a tree structure with datasets as its leaf nodes. The dataset path in the tree forms a unique identifier. The dataset data is in a table format with member fields as columns. The schema is defined in XML with elements <typedef>, <group>, <dataset>, etc. The following schema definitions show the top levels of the schema and the input `GRID` and stress output HEXA datasets.

```
<?xml version="1.0" encoding="utf-8"?>
<crdb schema="0">
  <typedefs>...</typedefs>
  <groups>
    <group name="NASTRAN">
      <group name="INPUT">
        <group name="CONSTRAINT">...</group>
        <group name="CONTACT">...</group>
        <group name="COORDINATE_SYSTEM">...</group>
        <group name="DESIGN">...</group>
        <dataset name="DOMAINS">...</dataset>
        <group name="DYNAMIC">...</group>
        <group name="ELEMENT">...</group>
        <group name="FATIGUE">...</group>
        <group name="LOAD">...</group>
        <group name="MATERIAL">...</group>
        <group name="MATRIX">...</group>
        <group name="NODE">...</group>
        <group name="PARAMETER">...</group>
        <group name="PARTITION">...</group>
```

```
        <group name="PROPERTY">...</group>
        <group name="TABLE">...</group>
        <group name="UDS">...</group>
      </group>
      <group name="RESULT">
        <group name="ACOUSTIC">...</group>
        <group name="CONTACT">...</group>
        <dataset name="DOMAINS">...</dataset>
        <group name="ELEMENTAL">...</group>
        <group name="FATIGUE">...</group>
        <group name="MODAL">...</group>
        <group name="NLOUT">...</group>
        <group name="NODAL">...</group>
        <group name="OPTIMIZATION">...</group>
        <group name="SUMMARY">...</group>
        <group name="MONITOR">...</group>
      </group>
    </group>
  </groups>
</crdb>
```

Schema 1: The top level groups

```
<?xml version="1.0" encoding="utf-8"?>
<crdb schema="0">
  <typedefs>
    <typedef name="GRID_SS" description ="Grid strain and stress structure">
      <field name="GRID" type="integer"/>
      <field name="X" type="double"/>
      <field name="Y" type="double"/>
      <field name="Z" type="double"/>
      <field name="TXY" type="double"/>
      <field name="TYZ" type="double"/>
      <field name="TZX" type="double"/>
    </typedef>
  </typedefs >
    <groups>
    <group name="NASTRAN">
      <group name="INPUT">
        <group name="NODE">
          <dataset name="GRID" version="1">
            <field name="ID" type="integer"/>
            <field name="CP" type="integer"/>
            <field name="X" type="double" size="3"/>
            <field name="CD" type="integer"/>
            <field name="PS" type="integer"/>
            <field name="SEID" type="integer"/>
            <field name="DOMAIN_ID" type="integer"/>
          </dataset>
        </group>
    </group>
```

```
      <group name="RESULT">
        <group name="ELEMENTAL">
          <group name="STRESS">
            <dataset name="HEXA">
              <field name="EID" type="integer"/>
              <field name="CID" type="integer"/>
              <field name="CTYPE" type="character" size="4"/>
              <field name="NODEF" type="integer"/>
              <field name="SS" type="GRID_SS" size="9"/>
              <field name="DOMAIN_ID" type="integer"/>
            </dataset>
          </group>
        </group>
      </group>
    </group>
  </groups>
</crdb>
```

Schema 2: The input GRID and output HEXA stress dataset definitions

Nastran Input and Output in NH5RDB

The NH5RDB includes Nastran input and output data. The supported data types are summarized in the following table.

Table 12-3 NH5RDB data categories

Category	Description
Input:	
Case control and parameters	Case control option and parameter values
Constraint	Constraint entries like SPC and MPC
Contact	Contact entry like BSURF
Coordinate system	Coordinate system entry like CORD1C
Design optimization	Design optimization entry like BEADVAR
Dynamic	Dynamic entries like EIGC
Element	Element entries like CHEXA and CQUAD4
Fatigue	Fatigue related entries like MATFTG
Load	Load entries like FORCE and PLOAD4
Material	Materials like MAT1, MAT2 and MAT8
Matrix	Direct matrix input like CONM1
Node	Node entries like GRID and SPOINT
Partition	Define set or list like SET1

Category	Description
Property	Property like PSHELL and PSOLID
Table	Table values in TABLEM1 and TABLEM2
Output:	
Acoustic output	Acoustic power, participation factor, etc.
Contact output	Contact force and stress output
Elemental output	Stress, strain, element force, etc.
Fatigue output	Fatigue analysis output
Modal	Modal response
Monitor	Monitor output
Nodal output	Displacement, velocity, acceleration, etc.
Optimization data	Objective function, constraint, history values, etc.
Special nonlinear format data	NLOUT format data

Nastran Solution Support

The NH5RDB is supported by multiple Nastran solutions. The table below shows all supported solution sequences.

Table 12-4 NH5RDB support solutions

Solution Number	Description
101	Statics
103	Normal modes
105	Buckling
107, 110	Complex eigenvalues
108, 111	Frequency response
109, 112	Transient response
144, 145, 146	Aerodynamic analysis
200	Design Optimization
400	Nonlinear analysis

NH5RDB Optional Output

The NH5RDB is an optional output. By default, the database will not be generated by Nastran. A parameter HDF5 is added in MDLPRM for NH5RDB generation. In BULK data section, setting HDF5 parameter in MDLPRM will ask Nastran to create the database.

Table 12-5 The HDF5 parameter in MDLPRM

Parameters Name	Description, Type, and Default Values	
HDF5	Control NH5RDB database creation.	
	-1	Do not create NH5RDB database (Default).
	0	Create NH5RDB database without compression.
	1	Create NH5RDB database with compression.
	2	Create uncompressed NH5RDB database without input data.
	3	Create compressed NH5RDB database without input data.

The usage of HDF5 parameter of MDLPRM in input file is shown as the following:

```
BEGIN BULK
MDLPRM,HDF5,1
```

In general, particularly for solid models, the user does not need rotational components in nodal vector type datasets as they are not easy or useful to visualize in post processor, suppressing them can reduce result file size. An option is provided to active this.

The option is an integer expressed by the MDLPRM keyword H5NORDOF or system cell 730.

Table 12-6 The H5NORDOF parameter in MDLPRM

Parameters Name	Description	
H5NORDOF	Parameter to suppress output rotational components to NH5RDB database	
	-1	Do not create NH5RDB database (Default).
	0	Create NH5RDB database without compression.

The option is only used when MDLPRM keyword HDF5 (or system cell 702) is 0 or 1, and applies to datasets for displacement, velocity, acceleration and eigenvector.

To distinguish datasets with both translational and rotational components, the datasets without rotational have "_TRANS" post prefix. For example, the corresponding dataset of DISPLACEMENT dataset is DISPLACEMENT_TRANS, and the one for DISPLACEMENT_CPLX is DISPLACEMENT_TRANS_CPLX.

The NH5RDB is also optional for intermediate output through OUTCTRL keyword in NLOPRM. If this option is selected, multiple NH5RDB files will be generated for output steps. Each NH5RDB file will contain data from its corresponding output step. The data in these generated NH5RDB files can be accessed during Nastran run, allowing post processing applications to access available data during a time consuming model run.

NH5RDB DMAP modules

There are several DMAP modules to write Nastran input and output data into database. The descriptions of these modules are as the following:

CRDB_IN	Write IFP data blocks into the NH5RDB database

Write IFP data blocks in an HDF5 format database. A maximum of five data blocks can be passed in one call.

Format:

```
CRDB_IN     DB1,DB2,DB3,DB4,DB5//
            NDDLNAM1/ DDLNAM2/NDDLNAM3/NDDLNAM4/NDDLNAM5/
            SEID/AFPMID/TRIMID/ $
```

Input Data Blocks:

DBi	Input data blocks, see remark 1 for supported data blocks.

Parameters:

NDDLNAMi	Char8, default=blank	NDDL data block name corresponding to DBi
SEID	Integer, default=0	Super element id of the data block
AFPMID	Integer, default=0	Acoustic field point mesh id
TRIMID	Trim id, default=0	Trim component id

Remarks:

1. The supported input data blocks are GEOM1, GEOM2, GEOM3, GEOM4, EPT, EDT, MPT, DIT, DYNAMIC, CONTACT, MATPOOL, EDOM, FATIGUE, PVT, CASECC and SPECSEL.
2. The SEID, AFPMID and TRIMID indicate which super element, acoustic field point mesh or trim component the data blocks are from.
3. NDDLNAMi must be provided for all data blocks.

CRDB_OUT	Write OFP data blocks into NH5RDB database.

Write OFP data blocks in an HDF5 format database. A maximum of five data blocks can be passed in one call.

Format:

CRDB_OUT	DB1,DB2,DB3,DB4,DB5// NDDLNAM1/NDDLNAM2/NDDLNAM3/NDDLNAM4/NDDLNAM5/ SEID/AFPMID/TRIMID/DESCYCLE/$

Input Data Blocks:

DBi	OFP data blocks, see remark 1 for supported data blocks.

Parameters:

NDDLNAMi	Char8, default=blank	NDDL data block name corresponding to DBi
SEID	Integer, default=0	Super element id of the data block
AFPMID	Integer, default=0	Acoustic field point mesh id
TRIMID	Trim id, default=0	Trim component id
DESCYCLE	Integer, default=0	Design cycle number in design optimization

Remarks:

1. The supported output data blocks are OUG. OVG, OAG, OES, OEF, OPG, OQG, OEFTG, OEFTGM, OUGFP, OMPF, OERP, OARPWR, OVGFP, OAPWR2, OAIG, OCOMP, OCOMPM, OMCFRAC, DBCOPT, NLOUT, OFVCCT and OFCON3D.
2. The SEID, AFPMID, TRIMID and DESCYCLE indicate which super element, acoustic field point mesh, trim component and design cycle the data blocks are from.
3. NDDLNAMi must be provided for all data blocks.

CRDB_DBK	Write special data blocks into the NH5RDB database

For some special datasets in NH5RDB, several Nastran data blocks and parameters are involved when writing them. The CRDB_DBK is defined for these kinds of data output. The module accepts up to ten data blocks, five integer and five real type parameters in its arguments. A string parameter is used in this module to indicate what data these input arguments are for. The implementation of the module will check the string indicator and interpret the input arguments accordingly. Currently only CLAMA and DSCM are supported in the string parameter, they are for complex eignvalue summary and design sensitivity coefficients output.

Format:

CRDB_DBK	DB1,DB2,DB3,DB4,DB5,DB6,DB7,DB8,DB9,DB10// TYPE/PI1/PI2/PI3/PI4/PI5/ PR1/PR2/PR3/PR4/PR5/ $

Input Data Blocks:

DBi	Data blocks to be processed.

Parameters:

TYPE	Char8, default=blank	Option indicating the to-do process action
PIi	Integer, default=0	Integer parameters used with data blocks
PRi	Real, default=0.0	Real parameters used with data blocks

Remarks:

1. The module will interpret the TYPE parameter and process data blocks accordingly. Only the complex eigenvalue CLAMA and design sensitivity coefficient DSCM are supported. The input arguments for the module are defined as the following.

Type	Data Block	Integer Parameter	Real Parameter	Note
CLAMA	DB1=CLAMA	N/A	PR1=Spin speed	CLAMA=Complex eigenvalue summary data block
DSCM	DB1=DSCMCOL DB2=DSCM DB3=DESTAB DB4=R1VALRG DB5=R2VALRG DB6=R3VALRG	PI1=SEID PI2=AFPMID PI3=TRIMID PI4=DESCYCLE	N/A	DSCMCOL=Design sensitivity parameters data block DSCM=Design sensitivity coefficient matrix DESTAB=Design variables attributes R1VALRG=Response type1 value matrix R2VALRG=Response type 2 value matrix R3VALRG=Response type3 value matrix SEID=Super element id AFPMID=Acoustic field point id TRIMID=Trim component id DESCYCLE=Design cycle

CRDB_MCR	Create intermediate HDF5 format database

Create intermediate HDF5 format database. The following calls of HDF5 output module, like CRDB_IN and CRDB_OUT, will write data into the created intermediate database.

Format:

CRDB_MCR	//NUM $

Parameters:

NUM	Integer, default=0	Sequence number of the intermediate database

Remarks:

1. This module will create intermediate database with the job name and sequence number. All following HDF5 output will be put into the intermediate database.
2. The intermediate database close module CRDB_MCL should be used with CRDB_MCR as pair.

CRDB_MCL	Close intermediate HDF5 format database

Close intermediate HDF5 format database.

Format:

CRDB_MCL	//

Remarks:

1. This module will close intermediate database. All following HDF5 output will be put into the primary database.
2. The intermediate database creation module CRDB_MCR should be used with CRDB_MCL as pair.

Test Case

The test model below shows the usage of HDF5 parameter and structure of NH5RDB database, the database is opened in HDF5 browser HDFView.

Test model file: test.dat

```
SOL 101
TIME 5
CEND
DISP = ALL
STRESS = ALL
FORCE = ALL
```

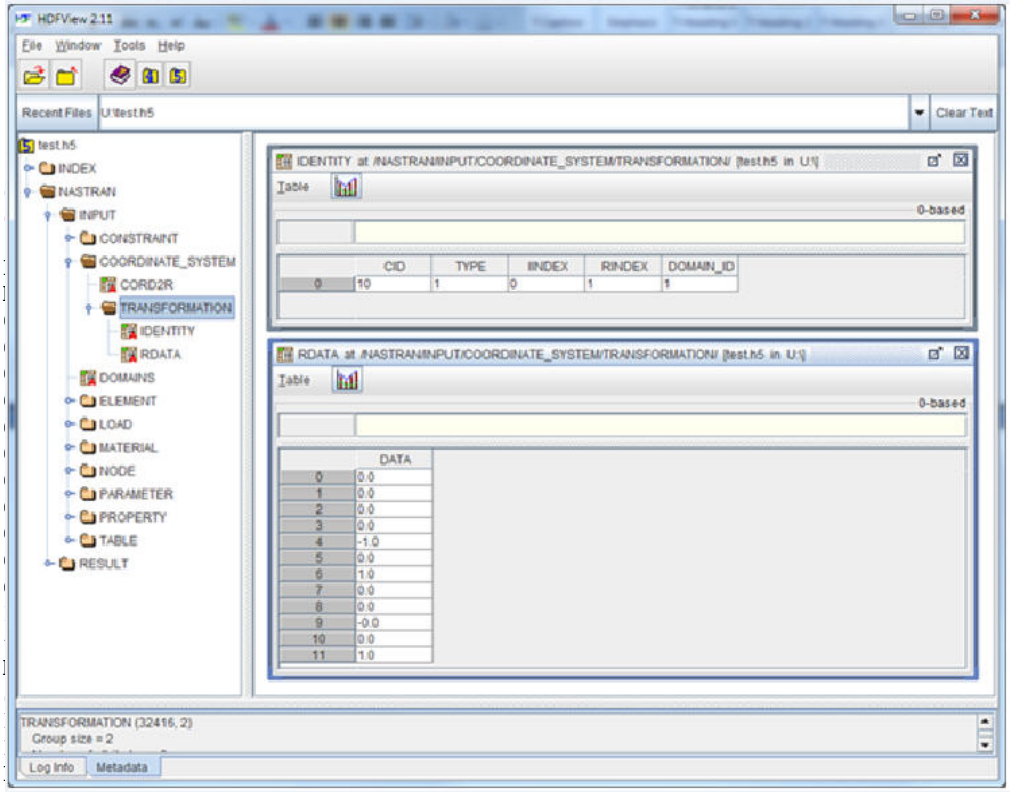


Figure 12-1 NH5RDB in HDF View browser

NH5RDB Data Access

The NH5RDB is a Hierarchical Data Format (HDF5) database; it organizes and stores data in groups and datasets. For Nastran data, the groups and datasets are defined in a schema file. The NH5RDB data structures will be created from these definitions in the schema.

The HDF5 is an open format architecture; it provides multiple language programming interfaces for data operations. The following shows how to access data in NH5RDB by examples in Python and C languages, a Java program using Java HDF object package is also given.

Data schema

The data schema is an XML file. Its elements and attributes define file structure and data format in NH5RDB. A brief summary of the schema elements and attributes is shown here.

Table 12-7 The schema XML elements and attributes

Element	Description	Element Attribute	Attribute Value
<crdb>	Root element	schema	Schema version number
<typedefs>	Parent element of <typedef>	description	Description string
<typedef>	Define structure type used in schema	name	Structure name
		description	Description string
<groups>	Parent element of <group>	description	Description string
<group>	Define a group	name	Group name
		description	Description string
<dataset>	Define dataset format	name	Dataset name
		description	Description string
<field>	Structure member field	name	Field name
		type	Field type
		size	Field size
		description	Description string

Like folders in a file system, the <group> element contains subgroups and datasets belong to it. The <dataset> element is used to define storing data format, it has <field> element to define each field properties. The <typedef> defines data structure element that will be used in the field type attribute in schema.

Below is a snippet of data schema that defines `GRID` input entry, nodal displacement and HEXA element stress output. A special note is for DOMAINS dataset and DOMAIN_ID field in input and output data. NH5RDB will store same type data in one dataset, for example, all displacement data will be put in the DISPLACEMENT dataset, no matter from which sub case and time step the displacement is. To distinguish data from different cases, the domain concept is introduced. The DOMAINS dataset consists of fields like ID, SUBCASE, STEP, etc. The ID field gives a unique number of the domain. The combination of all other fields makes one domain entry, which indicates the source of data. Correspondingly, a DOMAIN_ID field is attached to input and output data. This DOMAIN_ID is the ID number in DOMAINS dataset. From the DOMAIN_ID, the data source information can be obtained. There is an example that shows how to relate data with domains in example section.

```
<?xml version="1.0" encoding="utf-8"?>
<crdb schema="0">
  <typedefs>
    <typedef name="GRID_SS" description ="Grid strain and stress structure">
      <field name="GRID" type="integer"/>
      <field name="X" type="double"/>
      <field name="Y" type="double"/>
      <field name="Z" type="double"/>
      <field name="TXY" type="double"/>
```

```
      <field name="TYZ" type="double"/>
      <field name="TZX" type="double"/>
    </typedef>
  </typedefs >
  <groups>
    <group name="NASTRAN">
      <group name="INPUT">
        <group name="NODE">
          <dataset name="GRID" version="1">
            <field name="ID" type="integer"/>
            <field name="CP" type="integer"/>
            <field name="X" type="double" size="3"/>
            <field name="CD" type="integer"/>
            <field name="PS" type="integer"/>
            <field name="SEID" type="integer"/>
            <field name="DOMAIN_ID" type="integer"/>
        </dataset>
         </group>
      </group>
      <group name="RESULT">
              <dataset name="DOMAINS">
                <field name="ID" type="integer"/>
                <field name="SUBCASE" type="integer"/>
                <field name="STEP" type="integer"/>
                <field name="ANALYSIS" type="integer"/>
                <field name="TIME_FREQ_EIGR" type="double"/>
                <field name="EIGI" type="double"/>
                <field name="MODE" type="integer"/>
                <field name="DESIGN_CYCLE" type="integer"/>
                <field name="RANDOM" type="integer"/>
                <field name="SE" type="integer"/>
                <field name="AFPM" type="integer"/>
                <field name="TRMC" type="integer"/>
                <field name="INSTANCE" type="integer"/>
                <field name="MODULE" type="integer"/>
              </dataset>
        <group name="NODAL">
          <dataset name="DISPLACEMENT">
            <field name="ID" type="integer"/>
            <field name="X" type="double" description="X component"/>
            <field name="Y" type="double" description="Y component"/>
            <field name="Z" type="double" description="Z component"/>
            <field name="RX" type="double" description="RX component"/>
            <field name="RY" type="double" description="RY component"/>
            <field name="RZ" type="double" description="RZ component"/>
            <field name="DOMAIN_ID" type="integer"/>
          </dataset>
        </group>
        <group name="ELEMENTAL">
          <group name="STRESS">
            <dataset name="HEXA">
```

```
                <field name="EID" type="integer"/>
                <field name="CID" type="integer"/>
                <field name="CTYPE" type="character" size="4"/>
                <field name="NODEF" type="integer"/>
                <field name="SS" type="GRID_SS" size="9"/>
                <field name="DOMAIN_ID" type="integer"/>
              </dataset>
            </group>
          </group>
        </group>
      </group>
    </groups>
</crdb>
```

Schema 3. Dataset and domains

NH5RDB

The NH5RDB database stores datasets as defined in the schema. A NH5RDB database snapshot is shown below. The tables in the snapshot show the nodal displacement and HEXA element stress datasets.

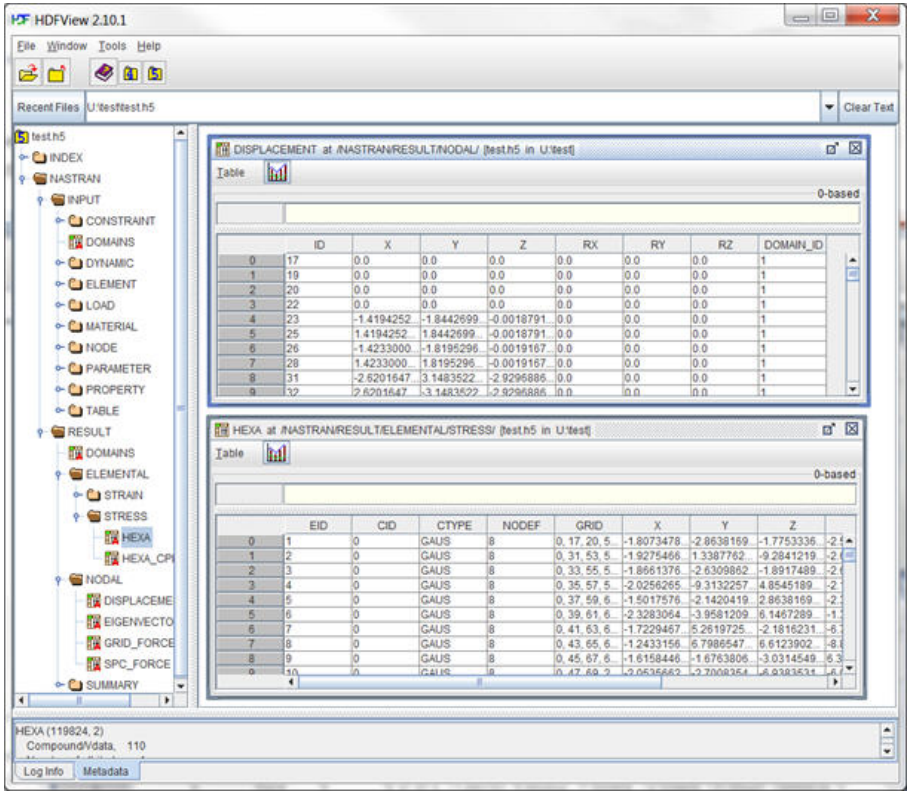


Figure 12-2 NH5RDB snapshot

Dataset domain index

As explained above about DOMAIN_ID, NH5RDB associates all data with domain id and put the same type of data in one dataset. Therefore, a dataset may have data with multiple domain ids. For a large dataset, one common operation is to access partial data from interested domain. To improve data retrieval efficiency, NH5RDB will generate domain id index for corresponding output datasets. Under the root group, an INDEX group will be created for domain index datasets. Corresponding to each output dataset, a domain index dataset with the same group and dataset name will be generated in the INDEX group. For example, the nodal displacement dataset is located at /NASTRAN/RESULT/NODAL/DISPLACEMENT in NH5RDB, its domain index dataset will be created as /INDEX/NASTRAN/RESULT/NODAL/DISPLACEMENT. All the index datasets have the same format with three fields DOMAIN_ID, POSITION and LENGTH. The DOMAIN_ID is the domain id number, the POSITION is the domain id start position in corresponding output dataset, and the LENGTH is the number of rows for the domain id. From the start position and number of rows, the data for the domain id can be located and accessed.

Records of a domain index dataset are sorted based on domain id, records with the same domain in a result dataset are sorted, which are similar to MSC Nastran sort1 option. Client code can use this feature to implement binary search algorithm for fast search. In most cases, the sort key is entity id; however there are exceptions, for example, result dataset /NASRAN/RESULT/ELEMENTAL/VIBRATION_INTENSITY has two keys for sorting, the first key is element type and the second is element id.

The figure below shows the displacement and its index datasets. From the index dataset, the displacement data for domain id 5 can be found from position 76 and has 44 rows.

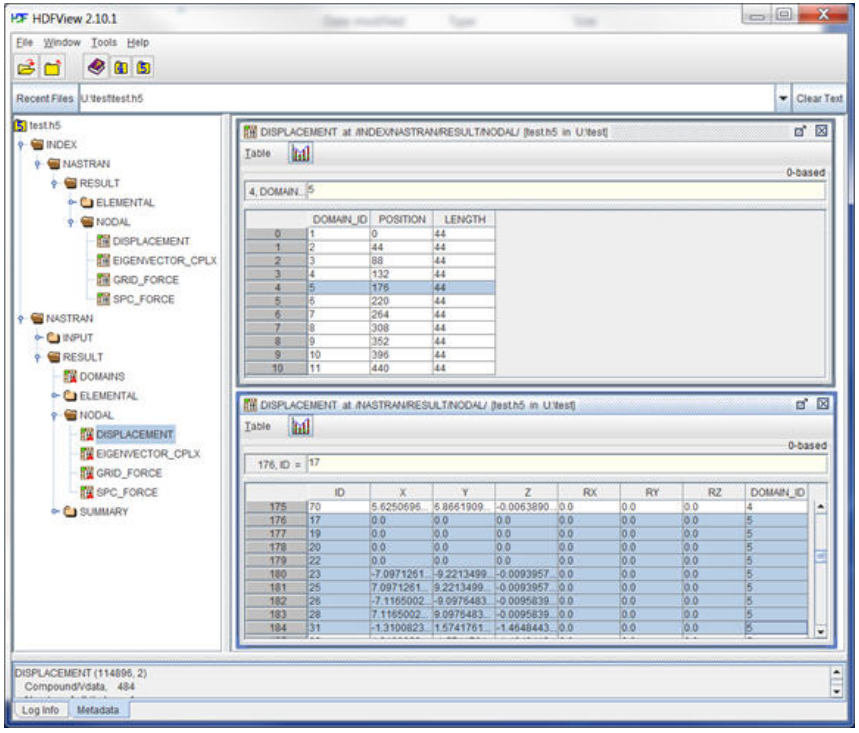


Figure 12-3 Domain index dataset

Database access examples

The HDF5 is platform independent and provides multiple programming language APIs for data operations. The examples here are given in Python, C\C++ and Java to show dataset access in NH5RDB. In addition to source files, each example gives build script and readme files for reference.

The Python examples uses PyTable package for HDF5 file. The PyTable and NumPy packages are required to be installed with Python. The examples show how easy it is to access the datasets.

- **Requirements of C and C++ examples**

  To build and run C and C++ examples, HDF5 library and a C++ compiler is required. All below examples are built with Visual C++ 2013 on Windows and Intel C++ 2015 and HDF5 library 1.8.14 or 1.8.15-patch1, a later version of the C++ compiler and HDF5 library should work.

- **Example 1: Read nodal displacements**

  This example reads nodal displacement dataset and prints the grid id and displacements from the second row of the table.

```
import tables

# Open database file
file = tables.open_file("test.h5")

# Get displacement table
disp = file.root.NASTRAN.RESULT.NODAL.DISPLACEMENT

# Get row number 2 of the table
row = disp[2]

# Print grid id and its displacement
print row['ID'], row['X'], row['Y'], row['Z'], row['RX'], row['RY'],
row['RZ']

# Close file
file.close()
```

Example 1: Read nodal displacement in Python

- **Example 2: Extract nodal displacements at time 3.0 and print in CSV format**

  This example has two steps, it first gets domain id belongs to time step 3.0, then extracts displacement for this time step and writes to a CSV file. In the example, the where clause is used to extract data for specified criterion. The time step and its corresponding displacements are shown in figure 4 and the converted CSV file is shown in figure 5.

```
import tables as pt

# ------------------------------------------------------------------------
--
# Extract nodal displacements at time = 3.000 and print as CSV
# ------------------------------------------------------------------------
--

#  --- open h5 input file
fname = "ldr2s400utp02.h5"
h5 = pt.open_file(fname)

# --- step 1: get "domain id" for t = 3.000
#  -- path to table with domain ids
tblDOMAINS = h5.root.NASTRAN.RESULT.DOMAINS

#  -- where clause
t   = 3.000
tol = 0.005
swhere = "(TIME_FREQ_EIGR > %6.4f)&(TIME_FREQ_EIGR < %6.4f)"%(t-tol,
t+tol)

#  -- extract domain id
for row in tblDOMAINS.where(swhere) :
    did = row["ID"]
```

```
    break

# --- step 2: extract the nodal displacements
#   -- path to table with nodal displacements
tblDISPS   = h5.root.NASTRAN.RESULT.NODAL.DISPLACEMENT

#   -- open CSV output file
fp = open("displacements.csv","w")

#    -- where clause
swhere = "DOMAIN_ID == %d"%(did)

#   -- loop over table
for row in tblDISPS.where(swhere):
    gid  = row["ID"]
    x = row["X"]
    y = row["Y"]
    z = row["Z"]
    rx = row["RX"]
    ry = row["RY"]
    rz = row["RZ"]
    fp.write("%d,%14.5e,%14.5e,%14.5e,%14.5e,%14.5e,%14.5e\n"%\
         (gid,x, y, z, rx, ry, rz) )

#   -- close files
fp.close()

h5.close()
```

Example 2: Extract displacement and convert to CSV format

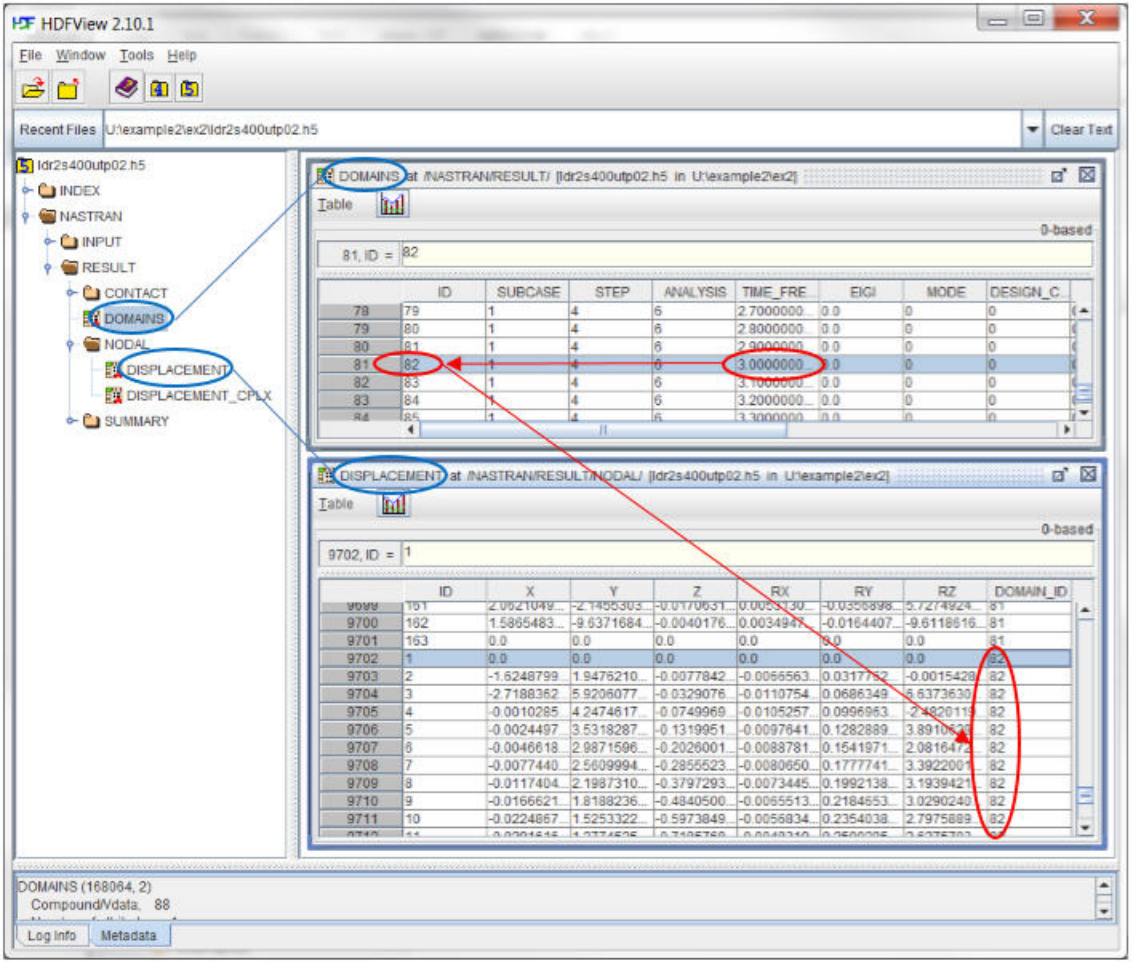
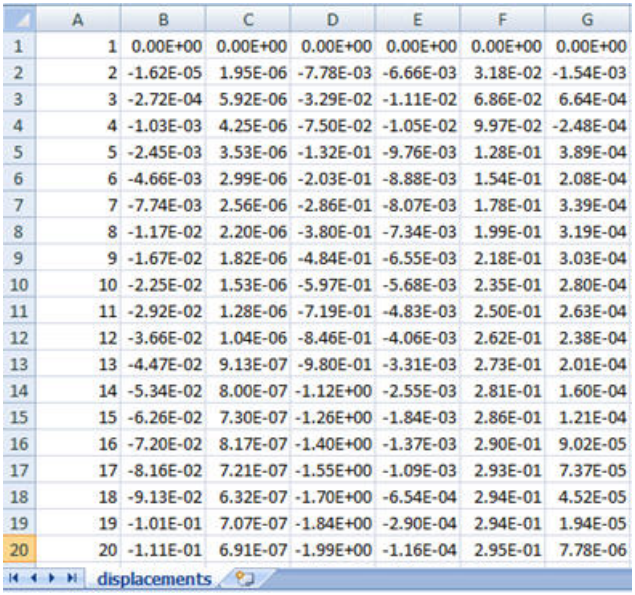


Figure 12-4 Time step and displacement data



	A	B	C	D	E	F	G
1	1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2	2	-1.62E-05	1.95E-06	-7.78E-03	-6.66E-03	3.18E-02	-1.54E-03
3	3	-2.72E-04	5.92E-06	-3.29E-02	-1.11E-02	6.86E-02	6.64E-04
4	4	-1.03E-03	4.25E-06	-7.50E-02	-1.05E-02	9.97E-02	-2.48E-04
5	5	-2.45E-03	3.53E-06	-1.32E-01	-9.76E-03	1.28E-01	3.89E-04
6	6	-4.66E-03	2.99E-06	-2.03E-01	-8.88E-03	1.54E-01	2.08E-04
7	7	-7.74E-03	2.56E-06	-2.86E-01	-8.07E-03	1.78E-01	3.39E-04
8	8	-1.17E-02	2.20E-06	-3.80E-01	-7.34E-03	1.99E-01	3.19E-04
9	9	-1.67E-02	1.82E-06	-4.84E-01	-6.55E-03	2.18E-01	3.03E-04
10	10	-2.25E-02	1.53E-06	-5.97E-01	-5.68E-03	2.35E-01	2.80E-04
11	11	-2.92E-02	1.28E-06	-7.19E-01	-4.83E-03	2.50E-01	2.63E-04
12	12	-3.66E-02	1.04E-06	-8.46E-01	-4.06E-03	2.62E-01	2.38E-04
13	13	-4.47E-02	9.13E-07	-9.80E-01	-3.31E-03	2.73E-01	2.01E-04
14	14	-5.34E-02	8.00E-07	-1.12E+00	-2.55E-03	2.81E-01	1.60E-04
15	15	-6.26E-02	7.30E-07	-1.26E+00	-1.84E-03	2.86E-01	1.21E-04
16	16	-7.20E-02	8.17E-07	-1.40E+00	-1.37E-03	2.90E-01	9.02E-05
17	17	-8.16E-02	7.21E-07	-1.55E+00	-1.09E-03	2.93E-01	7.37E-05
18	18	-9.13E-02	6.32E-07	-1.70E+00	-6.54E-04	2.94E-01	4.52E-05
19	19	-1.01E-01	7.07E-07	-1.84E+00	-2.90E-04	2.94E-01	1.94E-05
20	20	-1.11E-01	6.91E-07	-1.99E+00	-1.16E-04	2.95E-01	7.78E-06

displacements

Figure 12-5 Converted CSV data

- **Example 3: Extract time history of nodal translation displacements**

  This example extracts displacements of node number 63 for all time steps, the data is shown in figure 6.

```
import tables as pt
import numpy as np

# ------------------------------------------------------------------------
--
# Extract time history of translational displacements for node 63
# Modified to correct domain ids; IDs are read from DOMAIN table
# Correctly extracted values are: 5 to 86 for 2017, 2018
# Modified to accommodate 2016 or 2017++ DISPLACEMENT table format

# ------------------------------------------------------------------------
--

#  --- open h5 input file
fname = "ldr2s400_20160.h5"
h5 = pt.open_file(fname)

# --- step 1:  extract times from domains table
#  -- path to table with domain ids
tblDOMAINS = h5.root.NASTRAN.RESULT.DOMAINS
```

```
# -- create empty numpy arrays for domain ids and times
ids = np.array([], int)
t   = np.array([], float)

# -- construct where clause
#    (time steps have domain ids 5 to 86 as found in HDFVIEW)
# Modify for loop to use TABLE.where() to use ANALYSIS=6 (MTRAN steps),
# instead of hard coded Domain ids.
swhere = "(ID >= 5) & ( ID <= 86)"

#  -- loop over domains table
for row in tblDOMAINS.where("ANALYSIS==6"):
    t   = np.append(t, row["TIME_FREQ_EIGR"])
    ids = np.append(ids,row["ID"])

# -- need to get first/last domain ids from above for later use

min_mtran_id = ids[0]
max_mtran_id = ids[-1]

# -- create numpy arrays for results
nt   = t.size
disp = np.zeros((nt,3), float)


# --- step 2: extract the nodal displacements
#  -- path to table with nodal displacements
tblDISPS   = h5.root.NASTRAN.RESULT.NODAL.DISPLACEMENT

# -- create where clause for node 63
node = 63
swhere = "(ID == %d) & ( DOMAIN_ID >= %d) & (DOMAIN_ID <= %d)"%( node,
                min_mtran_id, max_mtran_id )

# Note that 2016 Displacement table has a VALUE column instead of X,Y,Z.
# Code modified to work for 2016 or 2017++ Displacement table
nast_ver = h5.root._v_attrs.SCHEMA
#  -- loop over table
i = 0
for row in tblDISPS.where(swhere):
  if (nast_ver>0 ) :
    disp[i,0] = row["X"]
    disp[i,1] = row["Y"]
    disp[i,2] = row["Z"]
  else :
    disp[i,0] = row["VALUE"] [0]
    disp[i,1] = row["VALUE"] [1]
    disp[i,2] = row["VALUE"] [2]

    i += 1
```

```
#  -- close file
h5.close()

# --- write extracted time history data to output file
#  -- open time history output file
fname = "disp_node_%d_mod3.csv"%(node)
fp = open(fname,"w")
fp.write(" TIME, XDISP, YDISP, ZDISP\n")
for i in np.arange(nt) :
    fp.write("%6.3f,%14.6e,%14.6e,%14.6e\n"%\
                (t[i], disp[i,0], disp[i,1], disp[i,2]))

# -- close output file
fp.close()
```

Example 3: Extract nodal displacement at all time steps

	A	B	C	D	E	F	G
1	TIME	XDISP	YDISP	ZDISP			
2	0	0.00E+00	0.00E+00	0.00E+00			
3	0.005	-1.07E-01	3.02E-06	-1.90E+00			
4	0.01	-4.74E-02	1.49E-06	-8.39E-01			
5	0.015	-3.46E-02	7.81E-08	-6.11E-01			
6	0.02	-1.15E-01	2.72E-06	-2.04E+00			
7	0.025	-2.32E-03	7.26E-07	-3.92E-02			
8	0.03	-9.70E-02	1.08E-06	-1.72E+00			
9	0.035	-6.11E-02	3.34E-06	-1.08E+00			
10	0.04	-2.21E-02	2.17E-07	-3.90E-01			
11	0.045	-1.19E-01	2.15E-06	-2.11E+00			
12	0.05	-7.76E-03	1.92E-06	-1.35E-01			
13	0.055	-8.53E-02	-5.85E-08	-1.51E+00			
14	0.06	-7.60E-02	2.46E-06	-1.34E+00			
15	0.065	-1.19E-02	1.56E-06	-2.11E-01			
16	0.07	-1.20E-01	8.38E-07	-2.13E+00			
17	0.075	-1.45E-02	3.15E-06	-2.55E-01			
18	0.08	-7.17E-02	8.04E-07	-1.27E+00			
19	0.085	-8.92E-02	1.01E-06	-1.58E+00			
20	0.09	-5.81E-03	2.44E-06	-1.03E-01			
21	0.095	-1.19E-01	1.19E-07	-2.10E+00			
22	0.1	-2.50E-02	1.99E-06	-4.44E-01			
23	0.105	-5.74E-02	2.44E-06	-1.01E+00			
24	0.11	-9.95E-02	5.89E-07	-1.76E+00			
25	0.115	-1.51E-02	2.47E-06	-2.29E-02			

disp_node_63

Figure 12-6 Nodal displacement at time steps

The following examples are programs in C language. The HDF5 table APIs are used in these examples. For description of these APIs, please refer to HDF5 reference manual.

- **Example 4: Read nodal displacements**

This example reads all data in the displacement dataset and prints the translation displacements. The dataset format is as its schema definition; its C type structure definition is shown in the example and can be found in dataset header file generated from schema.

```
#include "hdf5.h"
#include "hdf5_hl.h"
#include <cstdlib>

/*
 * Read all nodal displacements
 */
void printDisplacement()
{
  // displacement dataset name
  const char* table = "/NASTRAN/RESULT/NODAL/DISPLACEMENT";

  // displacement structure
  typedef struct {
    long long ID;       // grid id
    double X;    // displacement
    double Y;
    double Z;
    double RX;
    double RY;
    double RZ;
    long long DOMAIN_ID;// domain id
  } Type;

  // displacement structure size
  size_t ts = sizeof(Type);

  // field offset
  size_t offset[] = {
    HOFFSET(Type, ID),
    HOFFSET(Type, X),
    HOFFSET(Type, Y),
    HOFFSET(Type, Z),
    HOFFSET(Type, RX),
    HOFFSET(Type, RY),
    HOFFSET(Type, RZ),
    HOFFSET(Type, DOMAIN_ID)
  };

  Type type;
  // field size
  size_t size[] = {
    sizeof(type.ID),
    sizeof(type.X),
    sizeof(type.Y),
    sizeof(type.Z),
    sizeof(type.RX),
```

```
      sizeof(type.RY),
      sizeof(type.RZ),
      sizeof(type.DOMAIN_ID)
    };

    // open file
    hid_t fid = H5Fopen("test.h5", H5F_ACC_RDONLY, H5P_DEFAULT);
    if (fid >= 0) {
      hsize_t nField, nRecord;
      // get field and record number
      if (H5TBget_table_info(fid, table, &nField, &nRecord) >= 0) {
        if (nRecord > 0) {
          // allocate buffer
          Type* p = (Type*) malloc(ts * nRecord);
          if (p) {
            // read displacement dataset
            if (H5TBread_table(fid, table, ts, offset, size, p) >= 0) {
              // print translation displacement
              for (size_t i = 0; i < nRecord; i++) {
                printf("GRID %d : %e, %e, %e\n", p[i].ID, p[i].X,
                  p[i].Y, p[i].Z);
              }
            }
            free(p);
          }
        }
      }
      // close file
      H5Fclose(fid);
    }
}

int main()
{
  printDisplacement();
}
```

Example 4: Read and print nodal displacements

- **Example 5: Read records (rows) of nodal displacements**

  The above example reads all data in nodal displacement dataset. This example shows reading only specified records (rows) from a dataset. It will read the first 10 records of displacements.

```
/*
 * Read the first 10 records of nodal displacements
 */
void printDisplacement2()
{
  // displacement dataset name
  char* table = "/NASTRAN/RESULT/NODAL/DISPLACEMENT";
```

```
// displacement structure
typedef struct {
  long long ID;       // grid id
  double X;    // displacement
  double Y;
  double Z;
  double RX;
  double RY;
  double RZ;
  long long DOMAIN_ID;// domain id
} Type;

// displacement structure size
size_t ts = sizeof(Type);

// field offset
size_t offset[] = {
  HOFFSET(Type, ID),
  HOFFSET(Type, X),
  HOFFSET(Type, Y),
  HOFFSET(Type, Z),
  HOFFSET(Type, RX),
  HOFFSET(Type, RY),
  HOFFSET(Type, RZ),
  HOFFSET(Type, DOMAIN_ID)
};

Type type;
// field size
size_t size[] = {
  sizeof(type.ID),
  sizeof(type.X),
  sizeof(type.Y),
  sizeof(type.Z),
  sizeof(type.RX),
  sizeof(type.RY),
  sizeof(type.RZ),
  sizeof(type.DOMAIN_ID)
};

// open file
hid_t fid = H5Fopen("test.h5", H5F_ACC_RDONLY, H5P_DEFAULT);
if (fid >= 0) {
  hsize_t nField, nRecord;
  // get field and record number
  if (H5TBget_table_info(fid, table, &nField, &nRecord) >= 0) {
    if (nRecord > 0) {
      // set record number to 10
      if (nRecord > 10) {
        nRecord = 10;
      }
```

```
            // allocate buffer
            Type* p = (Type*) malloc(ts * nRecord);
            if (p) {
              // read the first 10 records
             if (H5TBread_records(fid, table, 0, nRecord, ts, offset, size, p)
                >= 0) {
                // print translation displacement
                for (size_t i = 0; i < nRecord; i++) {
                  printf("GRID %d : %e, %e, %e\n", p[i].ID, p[i].X,
                    p[i].Y, p[i].Z);
                }
              }
              free(p);
            }
          }
        }
        // close file
        H5Fclose(fid);
    }
}
```

Example 5: Read nodal displacement records

- **Example 6: Read dataset fields (columns) of HEXA element stress by name**

  The above example shows reading records (rows) of dataset. This example will show how to read some fields (columns) in dataset. It will read data fields from HEXA element stress output. The specified fields are given by their field names, which are defined in schema.

```
/*
 * Read HEXA element stress data by field name
 */
void printHexaStress()
{
  /*
    The HEXA stress dataset structure in database

    typedef struct {
      long long EID;       // Element identification number
      long long CID;       // Stress Coordinate System
      char CTYPE[4];       // Coordinate System Type (BCD)
      long long NODEF;     // Number of Active Points
      long long GRID[9];   // Number of active grids or corner grid ID
      double X[9];         // Normal X
      double Y[9];         // Normal Y
      double Z[9];         // Normal Z
      double TXY[9];       // Shear xy
      double TYZ[9];       // Shear yz
      double TZX[9];       // Shear zx
      long long DOMAIN_ID; // Domain identifier
    } Type;
  */
```

```
// HEXA stress dataset name
char* table = "/NASTRAN/RESULT/ELEMENTAL/STRESS/HEXA";

// fields to read from HEXA stress
typedef struct {
  long long EID;
  long long X[9];
  long long Y[9];
  long long Z[9];
  long long TXY[9];
  long long TYZ[9];
  long long TZX[9];
  long long DOMAIN_ID;
} Type;

size_t ts = sizeof(Type);
size_t offset[] = {
  HOFFSET(Type, EID),
  HOFFSET(Type, X),
  HOFFSET(Type, Y),
  HOFFSET(Type, Z),
  HOFFSET(Type, TXY),
  HOFFSET(Type, TYZ),
  HOFFSET(Type, TZX),
  HOFFSET(Type, DOMAIN_ID)
};

Type type;
size_t size[] = {
  sizeof(type.EID),
  sizeof(type.X),
  sizeof(type.Y),
  sizeof(type.Z),
  sizeof(type.TXY),
  sizeof(type.TYZ),
  sizeof(type.TZX),
  sizeof(type.DOMAIN_ID)
};

// field names of HEXA stress to read
char* name = "EID,X,Y,Z,TXY,TYZ,TZX,DOMAIN_ID";

hid_t fid = H5Fopen("test.h5", H5F_ACC_RDONLY, H5P_DEFAULT);
if (fid >= 0) {
  hsize_t nField, nRecord;
  if (H5TBget_table_info(fid, table, &nField, &nRecord) >= 0) {
    if (nRecord > 0) {
      Type* p = (Type*) malloc(ts * nRecord);
      if (p) {
        // read fields by name
```

```
            if (H5TBread_fields_name(fid, table, name, 0, nRecord, ts,
              offset, size, p) >= 0){
              for (size_t i = 0; i < nRecord; i++) {
                printf("EID %d : DOMAIN %d\n", p[i].EID, p[i].DOMAIN_ID);
                for (int j = 0; j < 9; j++) {
                  printf("\t%e, %e, %e\n", p[i].X[j], p[i].Y[j], p[i].Z[j]);
                }
              }
            }
            free(p);
          }
        }
      }
      H5Fclose(fid);
    }
  }
```

Example 6: Read HEXA element stress data by field name

- **Example 7: Read data fields (columns) of HEXA element stress by index**

  In addition to names, field index can be used to specify the fields to read. This example uses an index array to specify the fields.

```
/*
 * Read HEXA element stress data by field index
 */
void printHexaStress2()
{
  /*
    The HEXA stress dataset structure in database

    typedef struct {
      long long EID;       // Element identification number
      long long CID;       // Stress Coordinate System
      char CTYPE[4];       // Coordinate System Type (BCD)
      long long NODEF;     // Number of Active Points
      long long GRID[9];   // Number of active grids or corner grid ID
      double X[9];         // Normal X
      double Y[9];         // Normal Y
      double Z[9];         // Normal Z
      double TXY[9];       // Shear xy
      double TYZ[9];       // Shear yz
      double TZX[9];       // Shear zx
      long long DOMAIN_ID; // Domain identifier
    } Type;
  */

  // HEXA stress dataset name
  char* table = "/NASTRAN/RESULT/ELEMENTAL/STRESS/HEXA";
```

```
// fields to read from HEXA stress
typedef struct {
  long long EID;
  long long X[9];
  long long Y[9];
  long long Z[9];
  long long TXY[9];
  long long TYZ[9];
  long long TZX[9];
  long long DOMAIN_ID;
} Type;

size_t ts = sizeof(Type);
size_t offset[] = {
  HOFFSET(Type, EID),
  HOFFSET(Type, X),
  HOFFSET(Type, Y),
  HOFFSET(Type, Z),
  HOFFSET(Type, TXY),
  HOFFSET(Type, TYZ),
  HOFFSET(Type, TZX),
  HOFFSET(Type, DOMAIN_ID)
};

Type type;
size_t size[] = {
  sizeof(type.EID),
  sizeof(type.X),
  sizeof(type.Y),
  sizeof(type.Z),
  sizeof(type.TXY),
  sizeof(type.TYZ),
  sizeof(type.TZX),
  sizeof(type.DOMAIN_ID)
};

// field index of HEXA stress to read
int index[] ={0, 5, 6, 7, 8, 9, 10, 11};
int num = sizeof(index) / sizeof(int);

hid_t fid = H5Fopen("test.h5", H5F_ACC_RDONLY, H5P_DEFAULT);
if (fid >= 0) {
  hsize_t nField, nRecord;
  if (H5TBget_table_info(fid, table, &nField, &nRecord) >= 0) {
    if (nRecord > 0) {
      Type* p = (Type*) malloc(ts * nRecord);
      if (p) {
        // read fields by index
       if (H5TBread_fields_index(fid, table, num, index, 0, nRecord, ts,
          offset, size, p) >= 0){
          for (size_t i = 0; i < nRecord; i++) {
```

```
                printf("EID %d : DOMAIN %d\n", p[i].EID, p[i].DOMAIN_ID);
                for (int j = 0; j < 9; j++) {
                  printf("\t%e, %e, %e\n", p[i].X[j], p[i].Y[j], p[i].Z[j]);
                }
              }
            }
            free(p);
          }
        }
      }
      H5Fclose(fid);
    }
  }
```

Example 7: Read HEXA element stress data by field index

- **Example 8: Write displacement, HEXA element stress and domain datasets**

  The example writes the nodal displacement, hexahedral element stress and domain datasets. These datasets use the same formats as defined in NH5RDB schema and table data is appended row by row. Both Python and C programs are given below.

```
# Write displacement, hexa stress and domain datasets in Python
from tables import *
from numpy import *

# Domain structure
class DOMAINS(IsDescription):
    ID = Int64Col(pos = 0)
    SUBCASE = Int64Col(pos = 1)
    STEP = Int64Col(pos = 2)
    ANALYSIS = Int64Col(pos = 3)
    TIME_FREQ_EIGR = Float64Col(pos = 4)
    EIGI = Float64Col(pos = 5)
    MODE = Int64Col(pos = 6)
    DESIGN_CYCLE = Int64Col(pos = 7)
    RANDOM = Int64Col(pos = 8)
    SE = Int64Col(pos = 9)
    AFPM = Int64Col(pos = 10)
    TRMC = Int64Col(pos = 11)
    INSTANCE = Int64Col(pos = 12)
    MODULE = Int64Col(pos = 13)

# Displacement structure
class DISPLACEMENT(IsDescription):
    ID = Int64Col(pos = 0)
    X = Float64Col(pos = 1)
    Y = Float64Col(pos = 2)
    Z = Float64Col(pos = 3)
    RX = Float64Col(pos = 4)
    RY = Float64Col(pos = 5)
    RZ = Float64Col(pos = 6)
    DOMAIN_ID = Int64Col(pos = 7)

# Hexa element stress structure
class HEXA(IsDescription):
    EID = Int64Col(pos = 0)
    CID = Int64Col(pos = 1)
    CTYPE = StringCol(4, pos = 2)
    NODEF = Int64Col(pos = 3)
    GRID = Int64Col(shape = 9, pos = 4)
    X = Float64Col(shape = 9, pos = 5)
    Y = Float64Col(shape = 9, pos = 6)
    Z = Float64Col(shape = 9, pos = 7)
    TXY = Float64Col(shape = 9, pos = 8)
    TYZ = Float64Col(shape = 9, pos = 9)
    TZX = Float64Col(shape = 9, pos = 10)
```

```
    DOMAIN_ID = Int64Col(pos = 11)

# Open file
h5 = open_file('test.h5', mode = 'w')
# Create domain table
domain = h5.create_table('/NASTRAN/RESULT', 'DOMAINS',
                                DOMAINS, createparents = True)
# Set domain values
row = domain.row
for i in xrange(1, 2):
    row['ID'] = i
    row['SUBCASE'] = 1
    row['STEP'] = 0
    row['ANALYSIS'] = 1
    row['TIME_FREQ_EIGR'] = 0
    row['EIGI'] = 0
    row['MODE'] = 0
    row['RANDOM'] = 0
    row['DESIGN_CYCLE'] = 0
    row['SE'] = 0
    row['AFPM'] = 0
    row['TRMC'] = 0
    row['INSTANCE'] = 0
    row['MODULE'] = 0
    row.append()
domain.flush()  # Flush table to file

# Create displacement table
displacement = h5.create_table('/NASTRAN/RESULT/NODAL',
                'DISPLACEMENT', DISPLACEMENT, createparents = True)
# Populate displacement, the values are for example only.
row = displacement.row
for i in xrange(1, 10):
    row['ID'] = i       # grid id
    row['X'] = i * 1.0
    row['Y'] = i * 2.0
    row['Z'] = i * 3.0
    row['RX'] = 0.0
    row['RY'] = 0.0
    row['RZ'] = 0.0
    row['DOMAIN_ID'] = 1    # domain id
    row.append()
displacement.flush()

# Create hexa element stress table
hexa = h5.create_table('/NASTRAN/RESULT/ELEMENTAL/STRESS',
                       'HEXA', HEXA, createparents = True)
# Populate hexa stress, the values are for example only.
row = hexa.row
for i in xrange(1, 6):
    row['EID'] = i
```

```
        row['CID'] = 0
        row['CTYPE'] = 'GAUS'    # Gauss point
        row['NODEF'] = 8
        row['GRID'] = array(i * arange(9))  # grid ids including center
        row['X'] = array(i * arange(9))     # x normal stress of grids
        row['Y'] = array(i * arange(9))
        row['Z'] = array(i * arange(9))
        row['TXY'] = array(i * arange(9))   # xy shear stress of grids
        row['TYZ'] = array(i * arange(9))
        row['TZX'] = array(i * arange(9))
        row['DOMAIN_ID'] = 1    # domain id
        row.append()
    hexa.flush()

    # Close file
    h5.close()
```

Example 8.a: Write nodal displacement, hexahedral element stress and domain datasets in Python

```
// Write displacement, hexa stress and domain datasets in C
void writeStressHexa()
{
  const char* result = "/NASTRAN/RESULT";
  const char* nodal = "/NASTRAN/RESULT/NODAL";
  const char* stress = "/NASTRAN/RESULT/ELEMENTAL/STRESS";
  const char* dptable = "DISPLACEMENT";
  const char* hxtable = "HEXA";
  const char* dmtable = "DOMAINS";
  hsize_t chunk = 50;

  // displacement structure
  typedef struct {
    long long ID;       // grid id
    double X;    // displacement
    double Y;
    double Z;
    double RX;
    double RY;
    double RZ;
    long long DOMAIN_ID;// domain id
  } Displacement;

  // displacement structure size
  size_t dp = sizeof(Displacement);

  // displacement field offset
  size_t dpo[] = {
    HOFFSET(Displacement, ID),
    HOFFSET(Displacement, X),
    HOFFSET(Displacement, Y),
```

```
  HOFFSET(Displacement, Z),
  HOFFSET(Displacement, RX),
  HOFFSET(Displacement, RY),
  HOFFSET(Displacement, RZ),
  HOFFSET(Displacement, DOMAIN_ID)
};

// displacement field size
Displacement disp;
size_t dps[] = {
  sizeof(disp.ID),
  sizeof(disp.X),
  sizeof(disp.Y),
  sizeof(disp.Z),
  sizeof(disp.RX),
  sizeof(disp.RY),
  sizeof(disp.RZ),
  sizeof(disp.DOMAIN_ID)
};

// displacement field name
const char* dpn[] = {
  "ID",
  "X",
  "Y",
  "Z",
  "RX",
  "RY",
  "RZ",
  "DOMAIN_ID"
};

// displacemen field type
hid_t dpt[] = {
  H5T_NATIVE_LLONG,
  H5T_NATIVE_DOUBLE,
  H5T_NATIVE_DOUBLE,
  H5T_NATIVE_DOUBLE,
  H5T_NATIVE_DOUBLE,
  H5T_NATIVE_DOUBLE,
  H5T_NATIVE_DOUBLE,
  H5T_NATIVE_LLONG
};

// hexa stress structure
typedef struct {
  long long EID;      // Element identification number
  long long CID;      // Stress Coordinate System
  char CTYPE[4];      // Coordinate System Type (BCD)
  long long NODEF;     // Number of Active Points
  long long GRID[9];   // Number of active grids or corner grid ID
```

```
  double X[9];       // Normal X
  double Y[9];       // Normal Y
  double Z[9];          // Normal Z
  double TXY[9];      // Shear xy
  double TYZ[9];      // Shear yz
  double TZX[9];      // Shear zx
  long long DOMAIN_ID; // Domain identifier
} Hexa;

// hexa stress structure size
size_t hx = sizeof(Hexa);

// hexa field offset
size_t hxo[] = {
  HOFFSET(Hexa, EID),
  HOFFSET(Hexa, CID),
  HOFFSET(Hexa, CTYPE),
  HOFFSET(Hexa, NODEF),
  HOFFSET(Hexa, GRID),
  HOFFSET(Hexa, X),
  HOFFSET(Hexa, Y),
  HOFFSET(Hexa, Z),
  HOFFSET(Hexa, TXY),
  HOFFSET(Hexa, TYZ),
  HOFFSET(Hexa, TZX),
  HOFFSET(Hexa, DOMAIN_ID)
};

// hexa stress field size
Hexa hexa;
size_t hxs[] = {
  sizeof(hexa.EID),
  sizeof(hexa.CID),
  sizeof(hexa.CTYPE),
  sizeof(hexa.NODEF),
  sizeof(hexa.GRID),
  sizeof(hexa.X),
  sizeof(hexa.Y),
  sizeof(hexa.Z),
  sizeof(hexa.TXY),
  sizeof(hexa.TYZ),
  sizeof(hexa.TZX),
  sizeof(hexa.DOMAIN_ID)
};

// hexa stress field name
const char* hxn[] = {
  "EID",
  "CID",
  "CTYPE",
  "NODEF",
```

```
    "GRID",
    "X",
    "Y",
    "Z",
    "TXY",
    "TYZ",
    "TZX",
    "DOMAIN_ID"
  };

  // domain structure
  typedef struct {
    long long ID;              // Domain identifier
    long long SUBCASE;         // Subcase number
    long long STEP;            // Step number
    long long ANALYSIS;        // Analysis type
    double TIME_FREQ_EIGR;     // Time, frequency or real part of eigen
value
    double EIGI;           // Imaginary part if eigen value (if applicable)
    long long MODE;            // Mode number
    long long DESIGN_CYCLE;    // Design cycle
    long long RANDOM;          // Random code
    long long SE;              // Superelement number
    long long AFPM;            // acounstic field point mesh id
    long long TRMC;            // trim component id
    long long INSTANCE;        // Instance
    long long MODULE;          // Module
  } Domain;

  // domain structure size
  size_t dm = sizeof(Domain);

  // domain field offset
  size_t dmo[] = {
    HOFFSET(Domain, ID),
    HOFFSET(Domain, SUBCASE),
    HOFFSET(Domain, STEP),
    HOFFSET(Domain, ANALYSIS),
    HOFFSET(Domain, TIME_FREQ_EIGR),
    HOFFSET(Domain, EIGI),
    HOFFSET(Domain, MODE),
    HOFFSET(Domain, DESIGN_CYCLE),
    HOFFSET(Domain, RANDOM),
    HOFFSET(Domain, SE),
    HOFFSET(Domain, AFPM),
    HOFFSET(Domain, TRMC),
    HOFFSET(Domain, INSTANCE),
    HOFFSET(Domain, MODULE)
  };

  // domain field size
```

```
Domain domain;
size_t dms[] = {
  sizeof(domain.ID),
  sizeof(domain.SUBCASE),
  sizeof(domain.STEP),
  sizeof(domain.ANALYSIS),
  sizeof(domain.TIME_FREQ_EIGR),
  sizeof(domain.EIGI),
  sizeof(domain.MODE),
  sizeof(domain.DESIGN_CYCLE),
  sizeof(domain.RANDOM),
  sizeof(domain.SE),
  sizeof(domain.AFPM),
  sizeof(domain.TRMC),
  sizeof(domain.INSTANCE),
  sizeof(domain.MODULE)
};

// domain field name
const char* dmn[] = {
  "ID",
  "SUBCASE",
  "STEP",
  "ANALYSIS",
  "TIME_FREQ_EIGR",
  "EIGI",
  "MODE",
  "DESIGN_CYCLE",
  "RANDOM",
  "SE",
  "AFPM",
  "TRMC",
  "INSTANCE",
  "MODULE"
};

// domain field type
hid_t dmt[] = {
  H5T_NATIVE_LLONG,
  H5T_NATIVE_LLONG,
  H5T_NATIVE_LLONG,
  H5T_NATIVE_LLONG,
  H5T_NATIVE_DOUBLE,
  H5T_NATIVE_DOUBLE,
  H5T_NATIVE_LLONG,
  H5T_NATIVE_LLONG,
  H5T_NATIVE_LLONG,
  H5T_NATIVE_LLONG,
  H5T_NATIVE_LLONG,
  H5T_NATIVE_LLONG,
  H5T_NATIVE_LLONG,
```

```
    H5T_NATIVE_LLONG
  };

  // create file
  hid_t fid = H5Fcreate("test.h5", H5F_ACC_TRUNC, H5P_DEFAULT,
H5P_DEFAULT);
  if (fid >= 0) {
    hid_t lpid = H5Pcreate(H5P_LINK_CREATE);
    if (lpid >= 0) {
      if (H5Pset_create_intermediate_group(lpid, 1) >= 0) {
        // create domain dataset
        memset(&domain, 0, dm);
        hid_t gid = H5Gcreate2(fid, result, lpid, H5P_DEFAULT,
H5P_DEFAULT);
        if (gid >= 0) {
          if (H5TBmake_table("", gid, dmtable, 14, 0, dm, dmn, dmo, dmt,
                chunk, NULL, 1, NULL) >= 0) {
            domain.ID = 1;
            domain.SUBCASE = 1;
            domain.ANALYSIS = 1;
            H5TBappend_records(gid, dmtable, 1, dm, dmo, dms, &domain);
          }
          H5Gclose(gid);
        }

        // create displacement dataset
        gid = H5Gcreate2(fid, nodal, lpid, H5P_DEFAULT, H5P_DEFAULT);
        if (gid >= 0) {
          if (H5TBmake_table("", gid, dptable, 8, 0, dp, dpn, dpo, dpt,
                  chunk, NULL, 1, NULL) >= 0) {
              // populate displacement
              for (int i = 0; i < 10; i++) {
                disp.ID = i + 1;
                disp.DOMAIN_ID = domain.ID;
                disp.X = i * 1.0;
                      disp.Y = i * 2.0;
                      disp.Z = i * 3.0;
                      disp.RX = 0.0;
                      disp.RY = 0.0;
                      disp.RZ = 0.0;
                H5TBappend_records(gid, dptable, 1, dp, dpo, dps, &disp);
              }
          }
          H5Gclose(gid);
        }

        // create hexa dataset
        gid = H5Gcreate2(fid, stress, lpid, H5P_DEFAULT, H5P_DEFAULT);
        if (gid >= 0) {
          hsize_t dim[1] = {9};
          hid_t ai = H5Tarray_create2(H5T_NATIVE_LLONG, 1, dim);
```

```
            hid_t ad = H5Tarray_create2(H5T_NATIVE_DOUBLE, 1, dim);
            hid_t ct = H5Tcopy(H5T_C_S1);
            if (ai >= 0 && ad >= 0 && ct >= 0 && H5Tset_size(ct, 4) >= 0) {
              // data types
              hid_t hxt[12] = {H5T_NATIVE_LLONG, H5T_NATIVE_LLONG, ct,
                H5T_NATIVE_LLONG, ai, ad, ad, ad, ad, ad, ad,
H5T_NATIVE_LLONG};
              if (H5TBmake_table("", gid, hxtable, 12, 0, hx, hxn, hxo, hxt,
                    chunk, NULL, 1, NULL) >= 0) {
                for (int i = 0; i < 10; i++) {
                  memset(&hexa, 0, hx);
                  hexa.EID = i + 1;
                  hexa.NODEF = 8;
                  strncpy(hexa.CTYPE, "GAUS", 4);
                  hexa.DOMAIN_ID = domain.ID;
                  for (int j = 1; j < 9; j++) {
                    hexa.GRID[j] = i * 10 + j;
                    hexa.X[j] = i * 10 + j;
                    hexa.Y[j] = i * 20 + j;
                    hexa.Z[j] = i * 30 + j;
                    hexa.TXY[j] = i * 10 + j;
                    hexa.TYZ[j] = i * 20 + j;
                    hexa.TZX[j] = i * 30 + j;
                  }
                  H5TBappend_records(gid, hxtable, 1, hx, hxo, hxs, &hexa);
                }
              }
            }
            if (ai >= 0) H5Tclose(ai);
            if (ad >= 0) H5Tclose(ad);
            if (ct >= 0) H5Tclose(ct);
            H5Gclose(gid);
          }
        }
        H5Pclose(lpid);
      }
      H5Fclose(fid);
    }
}
```

Example8. Write nodal displacement, hexahedral element stress and domain datasets in C

- **Example 9: Query and print result value by user specified domain and entities**

  This C++ example shows how to search records by a specified domain id and entity ids from a result dataset in the database. This search procedure can be implemented by reading records one by one and check if ids are agree, however it is very slow when the dataset becomes huge. To improve performance, the program can first search for where the domain is located by using domain index dataset, which is prepared when creating NH5RDB database, then read all the records with the specified domain into memory and find the records with the specified entity ids. Both search procedures should use binary search because

1. Domain ids in index dataset are sorted.
2. Within the same domain id, the entity ids are also sorted.

   Here is an NH5RDB example mceig.h5 generated by a sol 400 perturbation analysis job (mceig.dat). It has one subcase with a nonlinear static step (analysis=nlstat) followed by a complex eigenvalue analysis step (analysis=mceig). This model includes 44 grids and 12 elements.

   The contents of the NH5RDB are shown in Figure 12-7, the first 11 result domains are from the nonlinear static step. Suppose we want to query displacement values of grid 25, 31 to 38 in the 4th increment of this step, which has domain id 4, the procedure is shown in Figure 12-7.

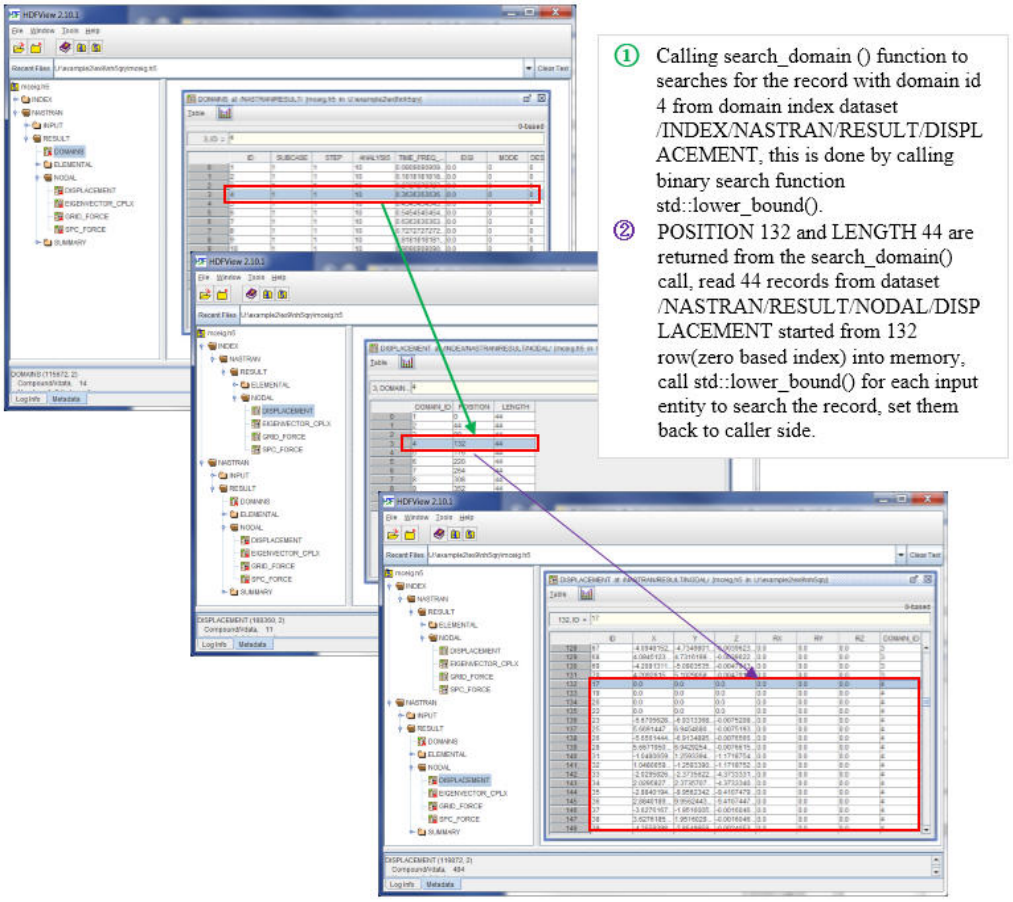


Figure 12-7 Process flow of the query

For details to build and run the program, read README.txt in example 9.

Here shows source code of below two functions, check nh5qry.cpp for other details in example 9. Below is the function to search domain:

```
//
// Purpose: search the start position and length (number of rows) of a
domain
//    in a dataset
//    Input:
//      hid_t file: hdf5 file id
//      const std::string& restype0: result type
//      long long domain_id: domain id to be searched
//    output:
//      long long& pos: the start row position of the domain
//      long long& length: number of rows of the domain
//    return:
//         0: successfull
//        !0: fail
//
int search_domain(hid_t file, const std::string& restype0, long long
domain_id,
    // output
    long long &pos, long long& length )
{
    hid_t index_dset = 0;
    std::string fullname;
    int err = get_dset(file, restype0, true, index_dset, fullname);
    if (err != 0)
        return err;

    //
    // read domain index dataset into memory
    //
    // get number of rows of domain index dataset
    hid_t space_id = H5Dget_space(index_dset);
    hsize_t  dims_out[4];
    int status_n = H5Sget_simple_extent_dims(space_id, dims_out, NULL);
    hsize_t total_row = dims_out[0];

    // read all rows in one call
    std::vector<DomainIndex_T> indics;
    err = read_h5(index_dset, 0, total_row, indics);
    if (err != 0)
        return err;

   // binary search the domain, domain dataset is sorted based on
DOMAIN_ID
    DomainIndex_T index = { domain_id, 0, 0 };
    auto bound = std::lower_bound(indics.begin(), indics.end(), index,
DomainIndex_Cmp());
    if (bound != indics.end() && bound->DOMAIN_ID == domain_id)
    {
```

```
        pos = bound->POSITION;
        length = bound->LENGTH;

        return 0;
    }
    else
    {
        fprintf(stderr, "No domain %d found in the damain index dataset
%s\n",
            domain_id, fullname.data());
        return -1;
    }
}
```

Below is the function to search records of specified ids in a domain:

```
//
// Purpose: search and print out records of specified ids in a domain
// Input:
//   hid_t file: hdf5 file id
//   long long domain_id: domain id to be searched
//   const std::string& restype0: result type
//        DISPLACEMENT, DISPLACEMENT_CPLX, QUAD4, QUAD_CN are valid words
//   const std::string& id_str: a srting to specify ids
//        "1:6 8 10" means 1 through 6, 8 and 10.
// return:
//     0: successful
//    Non 0: fail
//
template<typename T>
int do_search(hid_t file, long long domain_id, const std::string&
restype0,
    const std::string& id_str)
{
    long long pos, length;
    int err = search_domain(file, restype0, domain_id, pos, length);

    if(err == 0 && length > 0)
    {
        hid_t dset = 0;
        std::string fullname;
        int err = get_dset(file, restype0, false, dset, fullname);
        if (err != 0)
            return err;

        printf("Row range of domain %ld,  %ld %ld\n", domain_id,
            pos, pos + length);
        std::vector<T> displacements;
        print_head(displacements);

        //
        // read all rows into memory and call binary search algorithm.
```

```
        // if the number of rows is huge, it is better to implement a
binary
        // search algorithm to the hdf5 file directory for memory
efficiency
        //
        // std::vector<typename T> displacements;
        err = read_h5(dset, pos, length, displacements);
        if (err != 0)
        {
            H5Fclose(file);
            return err;
        }

        std::vector<long long> ids = parse_id_string(id_str);
        for(auto id_it = ids.begin(); id_it != ids.end(); ++id_it)
        {
            long long id = *id_it;
           //records in a dataset is sorted based on Nastran sort1,binary
search
            // the interesting ID
            T disp0 = { id };
            auto bound_rec = std::lower_bound(displacements.begin(),
                displacements.end(), disp0, CompareID<T>());
            if (bound_rec != displacements.end() && match_id(*bound_rec,
id))
            {
                auto const & disp = *bound_rec;
                size_t dist = std::distance(displacements.begin(),
bound_rec);
                print_row(disp);
            }
            else
               fprintf(stderr, "Cannot find record of id %ld from dataset
\n", id);
        }
    }

    return 0;
}
```

Example 9. Query and print result value by user specified domain and entities

- **Example 10: Read nodal displacement dataset in Java**

  This example demonstrates how to access dataset in NH5RDB in Java. The Java HDF Object Package, which wraps HDF Java interfaces in an object model, is used in the program. The package consists of several Java jar files and shared libraries. It can be obtained from HDF group website.

  In the example, the nodal displacement dataset is read in first. Then each field data, including ID, displacement component and DOMAIN_ID, is obtained and print out to the system output.

  ```
  /*
  ```

```
 * This example uses Java HDF Object Package to access NH5RDB.
 *
 */
import hdf.object.*;
import java.util.*;

/**
 * The NH5RDB class defines methods to access NH5RDB.
 *
 */
public class NH5RDB {
  /**
   * Read nodal displacement dataset and print to system output
   *
   * @param file the NH5RDB file name
   * @throws Exception
   */
  public void printDisplacement(String file) throws Exception {
    FileFormat ff = null;
    try {
      ff =
FileFormat.getFileFormat(FileFormat.FILE_TYPE_HDF5).createInstance(
                                                    file, FileFormat.READ);
      ff.open();
      CompoundDS ds = (CompoundDS)
ff.get("NASTRAN/RESULT/NODAL/DISPLACEMENT");
      List data = (List) ds.getData();
      long[] ID = (long[]) data.get(0);
      double[] X = (double[]) data.get(1);
      double[] Y = (double[]) data.get(2);
      double[] Z = (double[]) data.get(3);
      double[] RX = (double[]) data.get(4);
      double[] RY = (double[]) data.get(5);
      double[] RZ = (double[]) data.get(6);
      long[] DOMAIN_ID = (long[]) data.get(7);
      for (int i = 0; i < ID.length; i++) {
        System.out.println("ID: " + ID[i] + ", Domain: " + DOMAIN_ID[i]);
        System.out.println("\tX = " + X[i]);
        System.out.println("\tY = " + Y[i]);
        System.out.println("\tZ = " + Z[i]);
        System.out.println("\tRX = " + RX[i]);
        System.out.println("\tRY = " + RY[i]);
        System.out.println("\tRZ = " + RZ[i]);
      }
    } finally {
      if (ff != null) {
        ff.close();
      }
    }
  }
```

```
  /**
   * @param args the command line arguments
   */
  public static void main(String[] args) {
    try {
      NH5RDB h5 = new NH5RDB();
      h5.printDisplacement("test.h5");
    } catch (Exception e) {
      e.printStackTrace();
    }
  }
}
```

Example 10: Read nodal displacement in Java

- **Python utility function**

  For post processing demonstration, python functions to calculate principle and Von Mises stress are given here. Also a program to find the maximum Von Mises stress in frequency response analysis is shown below. These python functions can be used as utility functions for user in post processing applications.

  - Calculate principal and Von Mises stress from stress dataset

```
import sys
import os
import math
import numpy

class ZeroStressException(Exception) :
      # exception needed to jump out when all stress/strain components
      # are zero.
      def __str__(self):
           return repr(self.value)

def _cal_prstrs(stress) :
    # principalStress(stress, pstr, dirc, idg33, strn, sn, so, iloc) :
    # ----------------------------------------------------------------
    #

#*********************************************************************
    #
    #  Description: Principal and Invarient Values in 3D
    #
    #  Purpose: Given normal and shear stresses in STR this routine
    #  attempts to find principal stresses PSTR and their direction
    #  cosines DCOS.  Invarient for Mean Pressure and VonMises will also
    #  be returned.
    #
    #  Arguments:
    #   STRESS[6]    double,input   Normal and Shear Stresses
    #     PSTR[3]    double,output  Principal Stresses
```

```
#     DIRC[3][3] double,output  Direction Cosines
#    idg33       integer,input  Indicator for Invariant Type
#                               0->Oct  1->VonMises
#     strn       logical,input  Indicator for strain versus stress
#                               false->stress  true->strain
#       sn       double,output  Mean Pressure
#       so       double,output  Oct Shear or VonMises
#    iloc       integer,input  Location in entry for MAXMIN processing
#
#  Method: Cubic Equation for Principal Stresses in 3 dimensions is:
#
#   3            2                                 2     2     2
#  S -(S +S +S )S  + (S S  + S S  + S S  - T   - T   - T  )S
#       X  Y  Z        X Y    Y Z    Z X    XY    YX    ZX
#
#                                    2       2       2
#     - (S S S  + 2 T  T  T   - S T   - S T   - S T  ) = 0
#         X Y Z      XY YZ ZX    X YZ    Y ZX    Z XY
#
#  The three roots S , S , and S  are the Principal Stresses.
#                   1   2       3
#

#*****************************************************************************
    #
    # --- constants
    power = 1.0 / 3.0
    degrad =  math.pi / 180.

    # --- local arrays
    prin = [0.0, 0.0, 0.0 ]
    s    = [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
    temp = [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
    # --- removed parameters, now used as local variables
    dirc = [ [0.0, 0.0, 0.0 ],[0.0, 0.0, 0.0 ],[0.0, 0.0, 0.0 ]]
    pstr = [0.0, 0.0, 0.0 ]
    idg33 = 0
    strn  = False

    # --- scaling stresses to be in range -1.0 to +1.0
    equal = False
    rm    = 0.0
    fl    = ( stress[0] + stress[1] + stress[2] ) / 3.0
    for i in (0, 1, 2) :
        if ( abs(stress[i] - fl) > rm ) : rm = abs(stress[i] - fl)
        if ( abs(stress[i+3]) > rm ) : rm = abs(stress[i+3])
    #
    # --- create exception for zero stress/strain values
    try :
        if rm <= 0.0 :          # former goto 180
            raise ZeroStressException
```

```
            # --- special cases
            # --- normalize stresses
            # --- former label 180
            sx  = (stress[0] -fl )/rm
            sy  = (stress[1] -fl )/rm
            sz  = (stress[2] -fl )/rm
            txy = stress[3] / rm
            tyz = stress[4] / rm
            tzx = stress[5] / rm
            #
            #  The above cubic is of the form,
            #
            #             3     2
            #            Y  + PY  + QY + R = 0
            #
            #  and substituting for Y the value  X - P/3 gives,
            #
            #             3
            #            X  + AX + B = 0
            #
            #  Now finding the three roots (Values of X)
            #
            txysq  = txy**2
            tyzsq  = tyz**2
            tzxsq  = tzx**2
            sxsy   = sx * sy
            p      = -sx -sy -sz
            q      = sxsy + sy*sz + sz*sx - txysq - tyzsq - tzxsq
            r      = -sxsy*sz -2.0 *txy*tyz*tzx + sx*tyzsq + sy*tzxsq +
    sz*txysq
            smalla = q - p**2/3.0
            smallb = (2.0*p**3 - 9.0*p*q)/27.0 + r
            #
            # --- for principal stresses we assume no imaginary roots
            pover3 = p / 3.0
            quan   = smallb**2/4.0 + smalla**3/27.0
            if (quan < -0.0000001 ) :   # --- former goto 400
                # --- former label 400
                # here three real unequal roots (trig solution)
                quan   = 2.0 * math.sqrt(abs(smalla**3/27.0))
                if (quan <= 0.0 )  :
                    # -- former goto 150
                    raise ZeroStressException
                #
                phiov3  = math.acos(-smallb/quan)/3.0
                quan    = 2.0 * math.sqrt(abs(smalla/3.0))
                prin[0] = quan * math.cos(phiov3) - pover3
                prin[1] = quan * math.cos(phiov3 + 120.0 * degrad) - pover3
                prin[2] = quan * math.cos(phiov3 + 240.0 * degrad) - pover3
            else :
                # here at least two roots are equal
```

```
            # kpr will point to the unequal root after "label 1000" sort
            kpr = 0
            bover2 = -smallb/2.0
            aplusb = 2.0 * abs(bover2)**power
            if bover2 < 0.0 :
                aplusb = -aplusb
                kpr = 1
            prin[0] = aplusb - pover3
            prin[1] = -aplusb/2.0 - pover3
            prin[2] = prin[1]
            equal = True
            # --- former goto 1000
        # --- here principal stresses are known
        # --- former label 1000
        pstr[0] = prin[0]
        pstr[1] = prin[1]
        pstr[2] = prin[2]
        prin[0] = max( [ pstr[0], pstr[1], pstr[2] ] )
        prin[1] = min( [ pstr[0], pstr[1], pstr[2] ] )
        prin[2] = pstr[0] + pstr[1] + pstr[2] - prin[0] - prin[1]
        pstr[0] = rm * prin[0] + fl
        pstr[1] = rm * prin[1] + fl
        pstr[2] = rm * prin[2] + fl
        #
        #  ---- Solving for Direction COSINES. -------------------------
------
        #
        #  Where S is a Principal Stress and K,L,M are the Direction
Cosines
        #
        #  A solution of the Direction Cosines can be found by solving any
        #  two of the following three equations with the fourth.
        #
        #
        #     (S -S)K  +   T    L  +   T    M = 0              (1)
        #       X           XY          ZX
        #
        #
        #      T    K  +  (S -S)L  +   T    M = 0              (2)
        #       XY          Y           YZ
        #
        #
        #      T    K  +   T    L  +  (S -S)M = 0              (3)
        #       ZX          YZ          Z
        #
        #
        #       2    2    2
        #      K  + L  + M  = 1
        #
        #  ------------------------------------------------------------
------
```

```
            #
            for k in range(3) :      # former do 2000 k=1,3
                if ( equal and ( k != kpr ) ) :   # former  goto 1400
                    for j in ( 0, 1, 2) :
                        dirc[j][k] = 0.0
                else :
                    s[0] = sx - prin[k]
                    s[1] = txy
                    s[2] = tzx
                    s[3] = txy
                    s[4] = sy - prin[k]
                    s[5] = tyz
                    s[6] = tzx
                    s[7] = tyz
                    s[8] = sz - prin[k]
                    #
                   #   Select 2 vectors whose cross products magnitude is non-
zero.
                    #  itk: calling a daxb equivalent will not gonna work,
because
                    #  it depends on fortran call by reference convention. we
will
                    #       have to write out what's done inside daxb.
                   #       ( may be array striping could also have been used )
                    #          D(1) = A(2)*B(3) - A(3)*B(2)
                    #          D(2) = A(3)*B(1) - A(1)*B(3)
                    #          D(3) = A(1)*B(2) - A(2)*B(1)
                    # CALL DAXB( S(1), S(4), TEMP(1) )
                    temp[0] = s[1] * s[5] - s[2] * s[4]
                    temp[1] = s[2] * s[3] - s[0] * s[5]
                    temp[2] = s[0] * s[4] - s[1] * s[3]
                    # CALL DAXB( S(4), S(7), TEMP(4) )
                    temp[3] = s[4] * s[8] - s[5] * s[7]
                    temp[4] = s[5] * s[6] - s[3] * s[8]
                    temp[5] = s[3] * s[7] - s[4] * s[6]
                    # CALL DAXB( S(7), S(1), TEMP(7) )
                    temp[6] = s[7] * s[2] - s[8] * s[1]
                    temp[7] = s[8] * s[0] - s[6] * s[2]
                    temp[8] = s[6] * s[1] - s[7] * s[0]
                    #

                    jj   = 0
                    fmax = 0.0
                    # -- start former 1380 loop
                    # for j in range(0,8,3) :
                    for j in (0,3,6) :
                        fl = math.sqrt(temp[j]**2 + temp[j+1]**2 +
temp[j+2]**2)
                        if fl > fmax :
                           jj = j
                           fmax = fl
```

```
                # -- end of former 1380 loop

                # IF( FMAX .LE. 0.0000001D0 ) GO TO 1400
                if not (fmax <=  0.0000001 ) :
                    for j in (0,1,2) :
                        dirc[j][k] = temp[jj] / fmax
                        jj += 1
                # former goto 2000
                else :
                    #  Ill condition (can not determine unique direction
of this
                    #  Principal Stress.)
                    #  former 1400 label
                    for j in ( 0, 1, 2) :
                        dirc[j][k] = 0.0
                # former 2000 label


    except ZeroStressException :
        # --- special case when all six components are zero
        for i in range(3) : # former 150 do-loop
            pstr[i] = fl
            dirc[0][i] = 0.0
            dirc[1][i] = 0.0
            dirc[2][i] = 0.0

    # former 2500 label
    sn = -(pstr[0] + pstr[1] + pstr[2])/3.0
    voct =
math.sqrt(((pstr[0]+sn)**2+(pstr[1]+sn)**2+(pstr[2]+sn)**2)/3.0)
    vonm = voct*3.0/math.sqrt(2.0)
    # if ( strn ):
    #    vonm = 2.0*vonm/3.0
    #
    # if  idg33 == 0  :
    #      so = voct
    # else :
    #      so = vonm
    return pstr

def _cal_vonmises_stress(str) :
    """ Calculates Von Mises stress from six stress components.
        Arguments:
            str     tuple containing the six stress components
        Returns:
            vonmises stress
    """
    aa = (str[0] - str[1])**2 + (str[1] - str[2])**2 + (str[2] -
str[0])**2 +\
         6*( str[3]**2 + str[4]**2 + str[5]**2)
    if ( aa > 0.0 ) :
```

```
            vonm = math.sqrt(aa / 2.0)
        else :
            # -- in theory we should never get here
            vonm = 0.0
        return vonm

    def _cal_vonmises_prinstress(pstr):
        """ Calculates Von Mises stress from principal stresses.
            Arguments:
               pstrs    array containing the three principal stresses
            Returns:
               vonmises stress
        """
        sn = -(pstr[0] + pstr[1] + pstr[2])/3.0
        voct =
    math.sqrt(((pstr[0]+sn)**2+(pstr[1]+sn)**2+(pstr[2]+sn)**2)/3.0)
        vonm = voct*3.0/math.sqrt(2.0)
        return vonm

    if __name__ == "__main__" :

       def _check_stress( stresses ) :
            """ calculates and prints principal stresses and vonmises stress
                 from six stress components passed as tuple in stresses
            """
            pstr = _cal_prstrs( stresses)
            vonm = _cal_vonmises_prinstress(pstr)
            vonm2 = _cal_vonmises_stress(stresses)
            print("%15.6e  %15.6e %15.6e --> %15.6e %15.6e"% \
                            (pstr[0],pstr[1],pstr[2], vonm,vonm2))

       _check_stress((2.0, 0.0, 0.0, 0.0, 0.0, 0.0))
       # pstr = _cal_prstrs((-38971184.0, -450032.6875, 3.2782554626464844e-
    07, \
       #                    -1299021.125, -17.771186828613281, -200944.828125))
       _check_stress((2.0, 0.0, 0.0, 0.0, 0.0, 0.0))
       _check_stress((0.0, 2.0, 0.0, 0.0, 0.0, 0.0))
       _check_stress((0.0, 0.0, 2.0, 0.0, 0.0, 0.0))
       print(" --- ")
       _check_stress((1.0, 1.0, 0.0, 0.0, 0.0, 0.0))
       _check_stress((0.0, 1.0, 1.0, 0.0, 0.0, 0.0))
       _check_stress((1.0, 0.0, 1.0, 0.0, 0.0, 0.0))
       print(" --- ")
       _check_stress((0.0, 0.0, 0.0, 1.0, 0.0, 0.0))
       _check_stress((0.0, 0.0, 0.0, 0.0, 1.0, 0.0))
       _check_stress((0.0, 0.0, 0.0, 0.0, 0.0, 1.0))
       print(" --- ")
       _check_stress((0.0, 0.0, 0.0, 2.0, 0.0, 0.0))
       _check_stress((0.0, 0.0, 0.0, 0.0, 2.0, 0.0))
       _check_stress((0.0, 0.0, 0.0, 0.0, 0.0, 2.0))
       print(" --- ")
```

```
   _check_stress((1.0, -1.0, 0.0, 0.0, 0.0, 0.0))
```

- Calculate maximum Von Mises stress in frequency response analysis

The python program calculates maximum Von Mises stress from complex stress tensor in frequency analysis. Both 2D and 3D element calculation methods are given.

```
from math import *
import tables
# import numpy
#
# script to calculate true von Mises stress from complex stress tensor
# in frequency response analysis.
# CTRIA3 (top&bottom), CQADU4 (top&bottom), CTETRA (centroid),
# CPENTA(centroid) and CHEXA(centroid) are supported.
# Results are written to .csv file.
#
# Based on paper from MSC World Conference in 1993:
#  "EXACT CALCULATION OF MINIMUM MARGIN OF SAFETY FOR FREQUENCY RESPONSE
#   ANALYSIS STRESS RESULTS USING YIELDING OR FAILURE THEORIES"
#
# Usage :
#    python  calc_vm_from_h5_2d3d.py  <jobname.h5>  <vonmisesstress.csv>
# with
#    jobname.h5    the Hdf5 file resulting from the Nastran job
#    vonmises.csv  the csv output file with the Von Mises stresses
#                  default is the name of the Hdf5 file with the
#                  extension .h5 replaced by .csv

#
# 2D element complex von mises function
#
def vM2D(strD) :
# bottom
    s1=sqrt(strD[0]**2+strD[1]**2)
    a1=atan(strD[1]/strD[0])
    if strD[0] < 0.0 :
       a1=a1+pi
    s2=sqrt(strD[2]**2+strD[3]**2)
    a2=atan(strD[3]/strD[2])
    if strD[2] < 0.0 :
       a2=a2+pi
    s12=sqrt(strD[4]**2+strD[5]**2)
    a12=atan(strD[5]/strD[4])
    if strD[4] < 0.0 :
       a12=a12+pi
    rhs=(1.0*( (s1*s1*sin(2.0*a1)) + (s2*s2*sin(2.0*a2)) +
             3.0*(s12*s12*sin(2.0*a12)) - (abs(s1)*abs(s2)*sin(a1+a2)) )
/
             ( (s1*s1*cos(2.0*a1)) + (s2*s2*cos(2.0*a2)) +
```

```
                3.0*(s12*s12*cos(2.0*a12)) - (abs(s1)*abs(s2)*cos(a1+a2)) )
)
    o1=-0.5*atan(rhs)
    svm1=sqrt( (s1*s1*cos(a1+o1)*cos(a1+o1)) +
               (s2*s2*cos(a2+o1)*cos(a2+o1)) +
               3.0*(s12*s12*cos(a12+o1)*cos(a12+o1)) -
               (abs(s1)*abs(s2)*cos(a1+o1)*cos(a2+o1))  )
    o1=o1+pi/2.0
    svm2=sqrt( (s1*s1*cos(a1+o1)*cos(a1+o1)) +
               (s2*s2*cos(a2+o1)*cos(a2+o1)) +
               3.0*(s12*s12*cos(a12+o1)*cos(a12+o1)) -
               (abs(s1)*abs(s2)*cos(a1+o1)*cos(a2+o1))  )
    cvm1=max(svm1,svm2)
# top
    s1=sqrt(strD[6]**2+strD[7]**2)
    a1=atan(strD[7]/strD[6])
    if strD[6] < 0.0 :
       a1=a1+pi
    s2=sqrt(strD[8]**2+strD[9]**2)
    a2=atan(strD[9]/strD[8])
    if strD[8] < 0.0 :
       a2=a2+pi
    s12=sqrt(strD[10]**2+strD[11]**2)
    a12=atan(strD[11]/strD[10])
    if strD[10] < 0.0 :
       a12=a12+pi
    rhs=( 1.0 * ( (s1*s1*sin(2.0*a1)) + (s2*s2*sin(2.0*a2)) +
                3.0*(s12*s12*sin(2.0*a12)) - (abs(s1)*abs(s2)*sin(a1+a2))
) /
                ( (s1*s1*cos(2.0*a1)) + (s2*s2*cos(2.0*a2)) +
               3.0*(s12*s12*cos(2.0*a12)) - (abs(s1)*abs(s2)*cos(a1+a2))
) )
    o1=-0.5*atan(rhs)
    svm1=sqrt( (s1*s1*cos(a1+o1)*cos(a1+o1)) +
               (s2*s2*cos(a2+o1)*cos(a2+o1)) +
               3.0*(s12*s12*cos(a12+o1)*cos(a12+o1)) -
               (abs(s1)*abs(s2)*cos(a1+o1)*cos(a2+o1))  )
    o1=o1+pi/2.0
    svm2=sqrt( (s1*s1*cos(a1+o1)*cos(a1+o1)) +
               (s2*s2*cos(a2+o1)*cos(a2+o1)) +
               3.0*(s12*s12*cos(a12+o1)*cos(a12+o1)) -
               (abs(s1)*abs(s2)*cos(a1+o1)*cos(a2+o1))  )
    cvm2=max(svm1,svm2)

    return [cvm1,cvm2]
#
# 3D elements complex von mises function
#
def vM3D(strD) :
    s1=sqrt(strD[0]**2 + strD[1]**2)
    a1=atan(strD[1]/strD[0])
```

```
    if strD[0] < 0.0 :
       a1=a1+pi
    s2=sqrt(strD[2]**2 + strD[3]**2)
    a2=atan(strD[3]/strD[2])
    if strD[2] < 0.0 :
       a2=a2+pi
    s3=sqrt(strD[4]**2 + strD[5]**2)
    a3=atan(strD[5]/strD[4])
    if strD[4] < 0.0 :
       a3=a3+pi
    s12=sqrt(strD[6]**2 + strD[7]**2)
    a12=atan(strD[7]/strD[6])
    if strD[6] < 0.0 :
       a12=a12+pi
    s23=sqrt(strD[8]**2 + strD[9]**2)
    a23=atan(strD[9]/strD[8])
    if strD[8] < 0.0 :
       a23=a23+pi
    s31=sqrt(strD[10]**2 + strD[11]**2)
    a31=atan(strD[11]/strD[10])
    if strD[10] < 0.0 :
       a31=a31+pi
#
    rhs=(1.0*( (s1*s1*sin(2.0*a1)) + (s2*s2*sin(2.0*a2)) +
(s3*s3*sin(2.0*a3)) +
               3.0*((s12*s12*sin(2.0*a12)) + (s23*s23*sin(2.0*a23)) +
               (s31*s31*sin(2.0*a31))) - (abs(s1)*abs(s2)*sin(a1+a2)) -
              (abs(s1)*abs(s3)*sin(a1+a3)) - (abs(s2)*abs(s3)*sin(a2+a3))
) /
             ( (s1*s1*cos(2.0*a1)) + (s2*s2*cos(2.0*a2)) +
(s3*s3*cos(2.0*a3)) +
               3.0*((s12*s12*cos(2.0*a12)) + (s23*s23*cos(2.0*a23)) +
               (s31*s31*cos(2.0*a31))) - (abs(s1)*abs(s2)*cos(a1+a2)) -
              (abs(s1)*abs(s3)*cos(a1+a3)) - (abs(s2)*abs(s3)*cos(a2+a3))
)  )
    o1=-0.5*atan(rhs)
    svm1=sqrt( (s1*s1*cos(a1+o1)*cos(a1+o1)) +
               (s2*s2*cos(a2+o1)*cos(a2+o1)) +
               (s3*s3*cos(a3+o1)*cos(a3+o1)) +
               3.0*((s12*s12*cos(a12+o1)*cos(a12+o1)) +
                    (s23*s23*cos(a23+o1)*cos(a23+o1)) +
                    (s31*s31*cos(a31+o1)*cos(a31+o1))) -
                    (abs(s1)*abs(s2)*cos(a1+o1)*cos(a2+o1)) -
                    (abs(s1)*abs(s3)*cos(a1+o1)*cos(a3+o1)) -
                    (abs(s2)*abs(s3)*cos(a2+o1)*cos(a3+o1) )  )
    o1=o1+pi/2.0
    svm2=sqrt( (s1*s1*cos(a1+o1)*cos(a1+o1)) +
               (s2*s2*cos(a2+o1)*cos(a2+o1)) +
               (s3*s3*cos(a3+o1)*cos(a3+o1)) +
               3.0*((s12*s12*cos(a12+o1)*cos(a12+o1)) +
                    (s23*s23*cos(a23+o1)*cos(a23+o1)) +
```

```
                        (s31*s31*cos(a31+o1)*cos(a31+o1))) -
                        (abs(s1)*abs(s2)*cos(a1+o1)*cos(a2+o1)) -
                        (abs(s1)*abs(s3)*cos(a1+o1)*cos(a3+o1)) -
                        (abs(s2)*abs(s3)*cos(a2+o1)*cos(a3+o1) )  )
        cvm1=max(svm1,svm2)

        return cvm1
    # main function
    def main(infile,outfile) :
        #
        # Open database file
        #
        fh5 = tables.open_file(infile)
        outf=open(outfile,'w')
        # Get element stress tables
        try :
            stressQ4 = fh5.root.NASTRAN.RESULT.ELEMENTAL.STRESS.QUAD4_CPLX
            hasQ4  = True
        except  :
            hasQ4  = False
        try :
            stressT3 = fh5.root.NASTRAN.RESULT.ELEMENTAL.STRESS.TRIA3_CPLX
            hasT3  = True
        except  :
            hasT3  = False
        try :
            stressTet = fh5.root.NASTRAN.RESULT.ELEMENTAL.STRESS.TETRA_CPLX
            hasTet = True
        except  :
            hasTet = False
        try :
            stressPen = fh5.root.NASTRAN.RESULT.ELEMENTAL.STRESS.PENTA_CPLX
            hasPen = True
        except  :
            hasPen = False
        try :
            stressHex = fh5.root.NASTRAN.RESULT.ELEMENTAL.STRESS.HEXA_CPLX
            hasHex = True
        except  :
            hasHex = False
        domain1 = fh5.root.NASTRAN.RESULT.DOMAINS
        freq=[]
        doma=[]
        subc=[]
        outf.write('FREQ,SUBCASE ID')
        for row in domain1 :
           freq.append(row["TIME_FREQ_EIGR"])
           doma.append(int(row["ID"]))
           subc.append(int(row["SUBCASE"]))
        #
        # QUAD4
```

```
#
if hasQ4 == True :
    strDQ4={}
    elemQ4=[]
    for row in stressQ4 :
       strDQ4[int(row["DOMAIN_ID"]), int(row["EID"])]=[
                  float(row["X1R"]),   float(row["X1I"]),
                  float(row["Y1R"]),   float(row["Y1I"]),
                  float(row["XY1R"]),  float(row["XY1I"]),
                  float(row["X2R"]),   float(row["X2I"]),
                  float(row["Y2R"]),   float(row["Y2I"]),
                  float(row["XY2R"]),  float(row["XY2I"]) ]
       if int(row["EID"]) not in elemQ4:
          elemQ4.append(int(row["EID"]))
    for el1 in elemQ4 :
       tmp=',%i BOTTOM,%i TOP' %(el1,el1)
       outf.write(tmp)
#
# TRIA3
#
if hasT3 == True :
    strDT3={}
    elemT3=[]
    for row in stressT3 :
       strDT3[int(row["DOMAIN_ID"]), int(row["EID"])]=[
                  float(row["X1R"]),  float(row["X1I"]),
                  float(row["Y1R"]),  float(row["Y1I"]),
                  float(row["TXY1R"]),float(row["TXY1I"]),
                  float(row["X2R"]),  float(row["X2I"]),
                  float(row["Y2R"]),  float(row["Y2I"]),
                  float(row["TXY2R"]),float(row["TXY2I"]) ]
       if int(row["EID"]) not in elemT3:
          elemT3.append(int(row["EID"]))
    for el1 in elemT3 :
       tmp=',%i BOTTOM,%i TOP' %(el1,el1)
       outf.write(tmp)
#
# TETRA
#
if hasTet == True :
    elemTet=[]
    strDTet={}
    for row in stressTet :
       strDTet[int(row["DOMAIN_ID"]), int(row["EID"])]=[
               float(row["XR"][0]),   float(row["XI"][0]),
               float(row["YR"][0]),   float(row["YI"][0]),
               float(row["ZR"][0]),   float(row["ZI"][0]),
               float(row["TXYR"][0]), float(row["TXYI"][0]),
               float(row["TYZR"][0]), float(row["TYZI"][0]),
               float(row["TZXR"][0]), float(row["TZXI"][0]) ]
       if int(row["EID"]) not in elemTet:
```

```
            elemTet.append(int(row["EID"]))
        for el1 in elemTet :
           tmp=',%i CENTROID' %(el1)
           outf.write(tmp)
#
# PENTA
#
if hasPen == True :
    elemPen=[]
    strDPen={}
    for row in stressPen :
       strDPen[int(row["DOMAIN_ID"]), int(row["EID"])]=[
                float(row["XR"][0]),   float(row["XI"][0]),
                float(row["YR"][0]),   float(row["YI"][0]),
                float(row["ZR"][0]),   float(row["ZI"][0]),
                float(row["TXYR"][0]), float(row["TXYI"][0]),
                float(row["TYZR"][0]), float(row["TYZI"][0]),
                float(row["TZXR"][0]), float(row["TZXI"][0]) ]
       if int(row["EID"]) not in elemPen:
          elemPen.append(int(row["EID"]))
    for el1 in elemPen :
       tmp=',%i CENTROID' %(el1)
       outf.write(tmp)
#
# HEXA
#
if hasHex == True :
    elemHex=[]
    strDHex={}
    for row in stressHex :
       strDHex[int(row["DOMAIN_ID"]),int(row["EID"])]=[
                  float(row["XR"][0]),   float(row["XI"][0]),
                  float(row["YR"][0]),   float(row["YI"][0]),
                  float(row["ZR"][0]),   float(row["ZI"][0]),
                  float(row["TXYR"][0]), float(row["TXYI"][0]),
                  float(row["TYZR"][0]), float(row["TYZI"][0]),
                  float(row["TZXR"][0]), float(row["TZXI"][0]) ]
       if int(row["EID"]) not in elemHex:
          elemHex.append(int(row["EID"]))
    for el1 in elemHex :
       tmp=',%i CENTROID' %(el1)
       outf.write(tmp)
cnt=0
outf.write('\n')
#
while cnt < len(freq) :
   fr1=freq[cnt]
   dom1=doma[cnt]
   subc1=subc[cnt]
   cnt=cnt+1
   outf.write('%12.6f,%i,' %(fr1,subc1))
```

```
            stressVM=[]
        #
        # QUAD4
        #
            if hasQ4 == True :
               for el1 in elemQ4 :
                  cvm=vM2D(strDQ4[dom1,el1])
                  stressVM.append(cvm[0])
                  stressVM.append(cvm[1])
        #
        # TRIA3
        #
            if hasT3 == True :
               for el1 in elemT3 :
                  cvm=vM2D(strDT3[dom1,el1])
                  stressVM.append(cvm[0])
                  stressVM.append(cvm[1])
        #
        # TETRA
        #
            if hasTet == True :
               for el1 in elemTet :
                  cvm=vM3D(strDTet[dom1,el1])
                  stressVM.append(cvm)
        #
        # PENTA
        #
            if hasPen == True :
               for el1 in elemPen :
                  cvm=vM3D(strDPen[dom1,el1])
                  stressVM.append(cvm)
        #
        # HEXA
        #
            if hasHex == True :
               for el1 in elemHex :
                  cvm=vM3D(strDHex[dom1,el1])
                  stressVM.append(cvm)
            for st in stressVM :
               outf.write('%12.6f,' %(st))
            outf.write('\n')


    if __name__ == '__main__'   :
        import sys
        infile=sys.argv[1]
        if len(sys.argv) > 2 :
            outfile=sys.argv[2]
        else :
            outfile=infile[:-3] + '_stress.csv'
        main(infile,outfile)
```

Example 11: Print nodal displacement by module

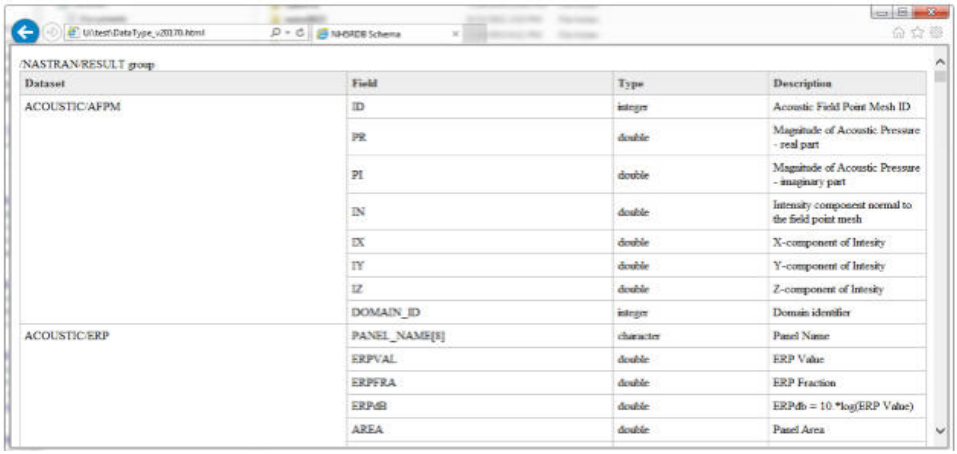
This example extracts nodal displacement in a multiple module model and prints out the displacement by each module.

```
     import os
     import sys
     import tables
# Usage: python print_module_disp.py test.h5

if __name__ == '__main__'   :
    if len(sys.argv) < 2:
        print('Please give the file name')
        sys.exit(1)

    try:
        # Open h5 file
        file = tables.open_file(sys.argv[1])

        # Get domain ID list for module number
        domains = file.root.NASTRAN.RESULT.DOMAINS
        d = {}
        for domain in domains:
            m = domain['MODULE']
            id = domain['ID']
            if m in d:
                d[m].append(id)
            else:
                d[m] = [id]

        # Print nodal displacement for each module
        disp = file.root.NASTRAN.RESULT.NODAL.DISPLACEMENT
        for m in sorted(d.keys()):
            l = d[m]
            # Print module number
            print('Module = {0}:'.format(str(m)))
            for id in l:
                s = "DOMAIN_ID == {0}".format(str(id))
                # Find nodal displacement of the module and print
                for row in disp.where(s):
```

```
                    gid = row['ID']
                    x = row['X']
                    y = row['Y']
                    z = row['Z']
                    # Print node id, X, Y, Z and domain
                    print('ID = {0}, {1}, {2}, {3}, {4}'.format(gid, x, y, z,
id))

        # Close file
        file.close()
    except Exception as e:
        print('Exception: {0}'.format(str(e)))
```

Schema document

The XML schema is distributed with Nastran. With XML schema, an HTML document is provided for dataset descriptions. See reference section for these documents.



/NASTRAN/RESULT group

Dataset	Field	Type	Description
ACOUSTIC/AFPM	ID	integer	Acoustic Field Point Mesh ID
	PR	double	Magnitude of Acoustic Pressure - real part
	PI	double	Magnitude of Acoustic Pressure - imaginary part
	IN	double	Intensity component normal to the field point mesh
	IX	double	X-component of Intesity
	IY	double	Y-component of Intesity
	IZ	double	Z-component of Intesity
	DOMAIN_ID	integer	Domain identifier
ACOUSTIC/ERP	PANEL_NAME[8]	character	Panel Name
	ERPVAL	double	ERP Value
	ERPFRA	double	ERP Fraction
	ERPdB	double	ERPdb = 10 *log(ERP Value)
	AREA	double	Panel Area

Figure 12-8 NH5RDB schema HTML document

Reference

MSC Nastran Quick Reference Guide

MSC Nastran DMAP Programmer's Guide

HDF5 Software Documentation

Examples Files

- DataType_v20191.html
- DataType_v20191.xml
- Example 1
- Example 2
- Example 3
- Example 4
- Example 5
- Example 6
- Example 7
- Example 8a
- Example 8b
- Example 9

- Example 10
- Utility Function

A Numerical Accuracy Considerations

- Introduction 575
- Linear Equation Solution 575
- Eigenvalue Analysis 575
- Matrix Conditioning 576
- Numerical Accuracy Issues 577
- Sources of Mechanisms 578
- Sources of Nonpositive Definite Matrices 579
- Detection and Avoidance of Numerical Problems 579

Introduction

MSC Nastran is an advanced finite element analysis (FEA) program that can solve models in excess of 50 million degrees of freedom. The numerical analysis capabilities of MSC Nastran are continually enhanced to provide the highest level of speed and accuracy.

This section provides a brief overview for detecting and avoiding numerical ill-conditioning problems, especially as they relate to dynamic analysis. For more information regarding the numerical analysis algorithms of MSC Nastran, see the *MSC Nastran Numerical Methods User's Guide*.

Linear Equation Solution

The basic statement of the linear equation solution is:

$$[A]\{x\} = \{b\}$$

where $[A]$ is a square matrix of known coefficients (and usually symmetric for structural models), $\{b\}$ is a known vector, and $\{x\}$ is the unknown vector to be determined.

MSC Nastran has two methods to solve the set of linear equations: the direct and the iterative method. For the remainder of this section, only the direct method will be considered.

The methods used for solution in MSC Nastran are based on a decomposition of $[A]$ to triangular matrices and are followed by forward-backward substitution to get $\{x\}$. The equations for this solution technique are:

$$[A] = [L][U]$$

where $[L]$ is a lower-triangular matrix and $[U]$ an upper-triangular matrix, and

$$[L]\{y\} = \{b\} \tag{1-1}$$

where $\{y\}$ is an intermediate vector. Equation (A-1) is called the forward pass because the solution starts with the first row where there is only one unknown due to the triangular form of $[L]$. The backward pass starts with the last row and provides the solution

$$[U]\{x\} = \{y\}$$

Eigenvalue Analysis

The general eigensolution equation is

$$[K + pB + p^2M]\{u\} = 0 \tag{1-2}$$

where p is the complex eigenvalue. This equation can always be transformed to a special eigenvalue problem for a matrix $[A]$

$$[A - \lambda I]\{\phi\} = 0 \tag{1-3}$$

where $[I]$ is the identity matrix. Equation (A-3) is the basis of all the transformation methods of MSC Nastran (HOU, GIV, etc.). The iterative methods (INV, SINV) work directly from equation (A-2). The Lanczos method uses both. If $[A]$ is a symmetric matrix, the eigenvectors are orthogonal, and they can be normalized such that

$$[\phi^T][\phi] = [I]$$

where $[\phi]$ is a square matrix whose columns contain the eigenvectors $\{\phi_i\}$. With this normalization convention, then

$$[\phi^T][A][\phi] = [\lambda]$$

and

$$[A] = [\phi][\lambda][\phi^T]$$

where $[\lambda]$ is the eigenvalue diagonal matrix.

Matrix Conditioning

Reordering the previous equations, any matrix $[A]$ can be expressed as a sum of its eigenvalues multiplied by dyadic eigenvector products

$$[A] = \lambda_1\{\phi_1\}\{\phi_1\}^T + \lambda_2\{\phi_2\}\{\phi_2\}^T + \ldots + \lambda_n\{\phi_n\}\{\phi_n^T\}$$

Defining $[B_i] = \{\phi_i\}\{\phi_i^T\}$, which is a full rank 1 matrix, then

$$[A] = \sum_{k=1}^{n} \lambda_k[B_k]$$

where n is the dimension of $[A]$. On the average, an element of $[B_k]$ has the same magnitude as the corresponding element of $[B_{k+1}]$. Let B_{max} be the magnitude of the largest coefficient of all $[B_k]$ matrices. Then

$$|a_{ij}| \le B_{max}(\lambda_1 + \lambda_2 + \ldots + \lambda_n)$$

This equation shows that the terms of $[A]$ are dominated by the largest eigenvalues. Unfortunately, the smallest eigenvalues are those of greatest interest for structural models. These small eigenvalues must be calculated by taking the differences of coefficients that are dominated by the largest eigenvalue. For this reason, the ratio λ_n/λ_1 is called a numerical conditioning number. If this number is too large, numerical truncation causes a loss of accuracy when computing the lowest eigenvalues of a system.

The assumptions that allow this simple analysis are often pessimistic in practice, that is, the bounds predicted by the error analysis are conservative. However, the effects it predicts do occur eventually so that models that produce acceptable results for one mesh size may produce unacceptable results with a finer mesh size due to the higher eigenvalues included in the larger-sized matrices occurring from the finer mesh.

Definiteness of Matrices

A matrix whose eigenvalues are all greater than zero is said to be positive definite. If some eigenvalues are zero but none are less than zero, the matrix is positive semi-definite. An elastic stiffness matrix assembled from elements is at least positive semi-definite. If all of the structure's rigid-body modes are constrained, the stiffness matrix is positive definite. Another category is the indefinite matrices category. These matrices have zeroes or blocks of zeroes on the diagonal.

Although definiteness is most concisely defined in terms of eigenvalues, it is not a practical test for large matrices because of the computational cost of extracting all of the eigenvalues. However, other operations, particularly linear equation solution and dynamic reduction, may detect nonpositive definite matrices and provide diagnostics using these terms as described later in this appendix.

Numerical Accuracy Issues

The numerical operations of MSC Nastran are executed in a finite 64-bit floating point arithmetic. Depending on the specific computer's word structure (number of bits for mantissa versus exponent), different roundoff errors may occur. To attain the most numerical accuracy possible, the following strategies are used in MSC Nastran.

In the decomposition of positive definite matrices, the Gaussian elimination process (which is the basis of all decomposition algorithms in MSC Nastran) does not require numerical pivots. Since some of the matrices are not positive definite, sparse decomposition (both symmetric and unsymmetric) employs a numerical pivoting (row/column interchange) algorithm. These methods consider a pivot term suitable if

$$|a_{kk}| \geq 10^{\mathrm{THRESH}} \cdot \underset{i}{\mathrm{MAX}} |a_{ki}|$$

which means that a diagonal term is accepted as a pivot if it is greater than the maximum term in that row multiplied by 10^{THRESH} (the default for THRESH is –6).

To ensure numerical accuracy in eigenvalue calculations, most MSC Nastran methods use a spectral transformation of

$$\mu = \frac{1}{\lambda - \lambda_s}$$

where λ_s is an eigenvalue shift.

This transformation is shown in Figure A-1.

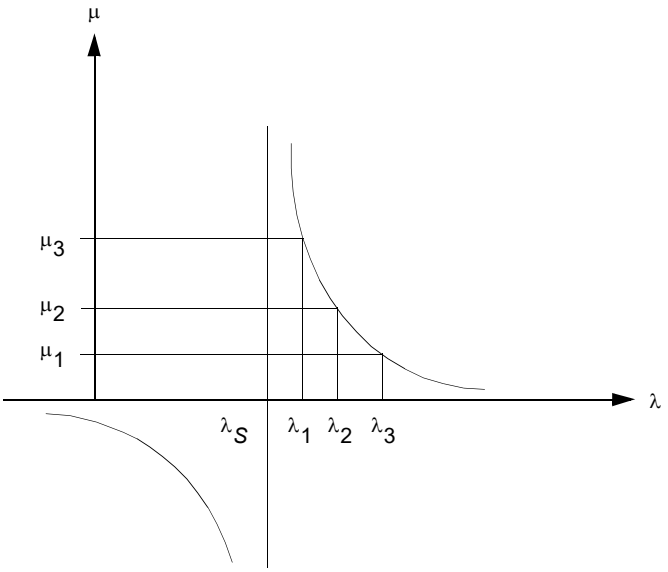


Figure 1-1 Spectral Transformation

The spectral transformation ensures uniform accuracy throughout the frequency by *shifting* to the area of interest. Another effect of this transformation is the welcomed μ space separation of closely-spaced eigenvalues. When λ_s is close to an eigenvalue λ_i, the decomposition of the shifted matrix

$$[K - \lambda_s M] = [L][D][L^T]$$

may produce high MAXRATIO messages. Automatic logic to perturb the λ_s value in this case is implemented in MSC Nastran.

Sources of Mechanisms

In all of the decomposition methods, a null row or column in $[A]$ causes a fatal error message. All other causes of singularity are not distinguishable from near-singularity because of the effects of numerical truncation. Only warning messages are usually provided for these cases.

In standard decomposition,

$$[A] = [L][D][L^T]$$

the process starts to compute the first term of $[D]$ with the first internal degree of freedom and then processes each additional degree of freedom and its associated terms. It can be shown that when processing the k-th row, the k-th row and all rows above it are in effect free, and all rows below it are constrained to ground.

The term of $[D]$ at the k-th row is proportional to the amount of coupling between that degree of freedom and the degree of freedom with a higher value of k. If the terms of the k-th row and above are not connected to the remaining rows, the k-th term of $[D]$ goes to zero. Because of numerical truncation, the term may be a small positive or negative number instead. If a term of $[D]$ is calculated to be identically zero, it is reset to a small number because of the indeterminacy of its calculation.

The existence of such a small term defines a mechanism in static analysis in MSC Nastran. A mechanism is a group of degrees-of-freedom that may move independently from the rest of the structure as a rigid body without causing internal loads in the structure. A hinged door, for example, is a mechanism with one rigid-body freedom. If the hinges are disconnected, the door mechanism has six rigid-body freedoms.

Mechanisms are characterized by non-dimensional numbers derived by dividing the terms of $[D]$ into the corresponding diagonal term of $[A]$. If these *matrix diagonal to factor diagonal* ratios are large numbers, various warning and fatal messages are produced, depending on the context.

Sources of Nonpositive Definite Matrices

A negative semi-definite element stiffness matrix, whose eigenvalues are all negative or zero, implies that the element has an energy source within it. MSC Nastran discourages using such elements by giving fatal messages when negative element thicknesses or section properties are input. However, there are applications where such elements are useful, such as when using an element with negative stiffness in parallel with a passive element to model a damaged or thinned-down element. For this reason, negative material property coefficients and negative stiffnesses for scalar elements are allowed. Also, some incorrect modeling techniques, such as the misuse of the membrane-bending coupling term on PSHELL entries (MID4), can lead to negative eigenvalues of the element stiffness matrix.

Stiffness matrices with negative eigenvalues cause negative terms in $[D]$. The number of such terms is automatically output by the standard symmetric decomposition subroutine. Their existence again causes various warning and fatal messages, depending on the context. The most common cause of negative terms in $[D]$ is true mechanisms, whose terms are small negative numbers and are actually computational zeroes.

Indefinite matrices occur when using the Lagrange multiplier method for constraint processing. In this method, the constraints are not eliminated from but are concatenated to the g-size system of equations. The constraint equations have zeroes on the diagonal, resulting in indefinite matrices. These systems can be solved using the block pivoting scheme of the sparse decomposition.

Detection and Avoidance of Numerical Problems

Static Analysis

Models used in static analysis must be constrained to ground in at least a statically determinate manner even for unloaded directions. For example, a model intended for only gravity loading must be constrained in horizontal directions as well as vertical directions. The evidence of unconstrained directions is that the entire model is a mechanism, that is, the large ratio occurs in the following scenarios:

- At the last grid point in the internal sequence.
- Connecting soft elements to stiff elements. Local stiffness is a function of element thickness (`moment` per unit rotation) through element thickness cubed (force per unit deflection) and is inversely proportional to mesh spacing, again in linear through cubic ratios. Some relief is possible by sequencing the soft degrees-of-freedom first in the internal sequence, although this is difficult to control in the presence of automatic re-sequencing. Reliable corrections are:
  - Replace the very stiff elements with rigid elements.
  - Place the soft and stiff elements in different superelements.
- The elements omitted through oversight. The corrective action here is to start with the grid points listed in the diagnostics and track back through the elements connected to them through the upstream grid points. The missing elements may be anywhere upstream. `PARAM,GPECT,1` output and undeformed structure plots all provide useful data for detecting missing elements.

There are two major methods of identifying large ratios and non-positive definite matrices. In some solutions, the largest matrix diagonal to factor diagonal ratio greater than 10^5 (`MAXRATIO` default) is identified by its internal sequence number, and the number of negative factor diagonal terms is output. The best method to identify mechanisms here is to apply checkout loads that cause internal loads in all of the elements. Then inspect the displacement output for groups of grid points that move together with implausibly large displacements and common values of grid point rotation. The only condition that causes a fatal error is a true null column, and `NASTRAN SYSTEM(69)=16` avoids this fatal error by placing a unit spring coefficient on the degrees of freedom with null columns. This option is recommended only for diagnostic runs because it may mask modeling errors.

In other solution sequences, all ratios greater than 10^5 are printed in a matrix format named the MECH table. The external sequence number of each large ratio is also printed, which is the grid point and degree of freedom number. If any such ratios exist, the action taken depends on the value of `PARAM,BAILOUT`. In the conventional solution sequences, its default value causes the program to continue after printing the MECH matrix. In the superelement solution sequences, a different default causes a fatal error exit after printing the MECH matrix. For both types of solution sequences, the opposite action may be requested by setting the value of `PARAM,BAILOUT` explicitly. Also, the criterion used for identifying large ratios may be changed using `PARAM,MAXRATIO`.

For static analysis, values between 10^3 and 10^6 are almost always acceptable. Values between 10^7 and 10^8 are questionable. When investigating structures after finding these values, some types of structures may be found to be properly modeled. It is still worthwhile to investigate the structures with questionable values.

The solutions with differential stiffness effects offer another method to obtain nonpositive definite stiffness matrices. For example, a column undergoing compressive gravity loading has a potential energy source in the gravity load. A lateral load that is stabilizing in the absence of gravity (that is, a decreased load causes a decreased deflection) is destabilizing when applied in a postbuckled state.

Eigensolutions Using the Inverse Iteration and Lanczos Methods

The matrix $[K - \lambda_s M]$ is decomposed where λ_s is an eigenvalue shift and $[K]$ and $[M]$ are the stiffness and mass matrices, respectively. This condition allows the solution of models with rigid-body modes since the singularities in $[K]$ are suppressed if there are compensating terms in $[M]$. The only conditions that should cause fatal messages due to singularity are

- The same null column in both $[K]$ and $[M]$. These columns and rows are given an uncoupled unit stiffness by the auto-omit operation if the default value (0) for PARAM,ASING is used. If this value is set to -1, a null column in both matrices is regarded as an undefined degree of freedom and causes a fatal error exit.
- A massless mechanism. One commonly encountered example is a straight-line, pinned rod structure made from CBAR elements with no torsional stiffness (J) defined. The structural mass generated for CBAR elements does not include inertia for the torsion degrees-of-freedom. The natural frequency for this torsion mode approaches the limit of zero torsional stiffness divided by zero mass, which is an undefined quantity. If the elements lie along a global coordinate axis, the mass term is identically zero, which leads to very large negative or positive eigenvalues and is usually beyond any reasonable search region. If the elements are skewed from the global axes, the eigenvalues may be computed at any value (including negative) because of the indeterminacy caused by numerical truncation.

The `negative terms on factor diagonal` message generally occurs for every decomposition performed in the iteration. It can be shown from Sturm sequence theory that the number of negative terms is exactly equal to the number of eigenvalues below λ_s. This condition is a means of determining if all roots in the range have truly been found.

Eigensolutions Using Transformation Methods

If the GIV or HOU methods (and their variations) are selected, one of the first operations performed is a Choleski decomposition. It is used to reduce the problem to a special eigenvalue problem. The mass matrix is given this operation for the straight GIV or HOU method. Identically null columns are eliminated by the auto-omit operation. Poor conditioning can result from several sources. An example is a point mass input on a CONM2 entry that uses offsets in three directions. The grid point to which it is attached has nonzero mass coefficients for all six degrees-of-freedom. However, only three independent mass degrees-of-freedom exist, not six. Another example results in a superelement analysis when most of the elements in a superelement do not have mass and all interior masses are restricted to only a few structural or mass elements. The boundary matrix produced for the superelement is generally full, no matter what its rank; that is, regardless of the number of independent mass degrees of freedom that it contains.

The presence of a rank-deficient mass matrix when using the GIV or HOU method produces fatal messages due to the singularity of the mass matrix or produces solutions with poor numerical stability. Poor stability is most commonly detected when making small changes to a model and then observing large changes in the solution. Either the MGIV and MHOU methods or the AGIV and AHOU methods are the preferred methods to use when nearly singular mass matrices are expected because these methods decompose the matrix $[K - \lambda_s M]$ instead of the mass matrix.

The shift parameter λ_s is automatically set to be large enough to control rigid-body modes. Better modeling practices also reduce the costs and increase the reliability for the two examples cited above. If an offset point mass is significant in any mode, it is better to attach it to an extra grid point at its center of gravity and model the offset with a rigid element. The auto-omit feature then eliminates the rotational degrees of freedom. Similarly, if only a few interior points of a superelement

have mass, it may be more economical to convert them to exterior points, which also eliminates the singular boundary mass matrix.

It is possible to input negative terms into a mass matrix directly with DMIG terms or scalar mass elements. This class of problem causes fatal errors due to non-positive definite mass matrices for the transformation methods, fatal errors with the Lanczos method, and wrong answers with the INV method. The complex eigenvalue methods should be used for this type of problem, which infers that modal or dynamic reduction methods may not be used.

A similar but different problem arises because Cholesky decomposition is used on the generalized mass matrix (named MI in the diagnostics) when orthogonalizing the eigenvectors with respect to the mass matrix. The existence of negative terms here indicates that poor eigenvectors were computed. The row number where the negative term occurs is printed in the diagnostic. This row number does not refer to a physical degree of freedom but refers instead to an eigenvector number. The usual cause is computing eigenvectors for computational infinite roots in the modified transformation methods. This problem can be avoided by setting ND on the EIGR entry to a value less than the row number that appears in the diagnostics or before the fact by setting F2 on the EIGR entry to a realistic value instead.

Frequency Response Analysis

Negative terms on the factor and high factor to matrix diagonal ratios can be expected when using coupled methods and may often be safely ignored. These messages are merely an indication that the excitation frequency is above one or some of the natural frequencies of the system (negative terms), or is near a natural frequency (high ratio).

Transient Response Analysis

A well-conditioned model should have neither negative terms nor high ratios on its factor terms. The causes of such messages include all of the effects described above. Negative mass terms can be detected by rapidly diverging oscillations in the transient solution.

The Large Mass Method

The large mass method is one of the methods for enforcing motion. It can be used with direct methods and with all the reduction methods. The basis of the method is to attach artificial masses to the structure at the degrees-of-freedom where the motion is to be enforced. These large masses should be orders of magnitude larger than the total mass or the moment of inertia of the structure. A survey of the literature shows recommendations for mass ratios ranging from 10^3 to 10^8 with a value of 10^6 as the most common recommendation.

The mass ratio affects both the accuracy and numerical conditioning, and must be adjusted in a compromise that meets both criteria. With regard to load accuracy, the error in the approximation is inversely proportional to the ratio. A ratio of 10^3 causes an error of ten percent at resonance for a mode with one-half percent of damping—it represents an extreme case. Off-resonance excitation or higher damping ratios result in lower errors. Numerical conditioning problems are much more difficult to predict.

The Lagrange Multiplier Method

Another method to enforce motion is the Lagrange multiplier method. In this method, the input motion function is described by a constraint equation. This method provides better accuracy than the large mass method where the numerical error introduced is proportional to the large mass.

The formulation introduces an indefinite system of linear equations where some numerical problems may arise from the fact that the system matrix contains terms that have dimensions of stiffness as well as nondimensional (constraint) terms. The decomposition of this matrix with the sparse decomposition methods pivoting strategy is stable.

You may control the pivoting strategy with the `THRESH DMAP` parameter. The default value (–6) is adequate for most Lagrange multiplier solutions. In fact, higher (ranging up to –2) values provide better accuracy, while increasing the number of pivots may result in performance degradation.

B Theory of Eigenvalue Extraction

- Sturm Sequence Theory 585
- Eigensolution Methods 585

Sturm Sequence Theory

Methods of Computation

A numerical technique which is very helpful in a modal extraction is Sturm Sequence. In this approach, a frequency is selected and computations are performed to determine the number of roots which exist below that frequency. In some cases this frequency is chosen manually by user input, other times it is selected automatically by MSC Nastran. This choice depends upon which modal extraction technique is being used.

Starting with the basic characteristic equation

$$[K + \lambda_s M] = 0 \tag{2-1}$$

a frequency, $\omega^2 = \lambda_s$ is substituted. This frequency is called the shift point. The determinant $K - \lambda_s M$ is evaluated for the frequency λ_s. The resulting matrix is factored into it $[L][D][L]^T$ components. The diagonal matrix, $[D]$, is known as the Factor Diagonal Matrix. The number of negative terms in the Factor Diagonal Matrix, known as the Sturm Number, are the number of frequencies that exist below the shift point frequency, λ_s.

Sturm Sequence logic is a very important part of the many of the modal extraction and dynamic reduction techniques. When it is utilized, the Sturm Sequence logic issues a number of messages which indicate the various shifts within the frequency range of interest. These messages help the various techniques to determine how many roots are in specific ranges and also if all modes have been found in the range.

Eigensolution Methods

Methods of Eigensolution

Seven methods of real eigenvalue extraction are provided in MSC Nastran. These methods are numerical approaches to solving the equation of motion for its natural frequencies and modes shapes. The reason for seven different numerical techniques is related to the fact that no one method is the best for all problems. While most of the methods can be applied to all problems, the difference between the methods many times reduces to the efficiency of the solution process.

Most methods of algebraic eigenvalue extraction belong to one of two groups:

- Transformation methods
- Tracking methods.

In a transformation method, the eigenvalue equation is first transformed into a special form (such as tridiagonal) from which eigenvalues may easily be extracted. In a “tracking” method, the roots are extracted from the original dynamic matrix one at a time using an iterative procedure.

Four of the real eigenvalue extraction methods available in MSC Nastran are transformation methods. These are:

- Givens Methods
- Householder Method
- Modified Givens Method

- Modified Householder Method.

These transformation methods use a “tridiagonal” solution method.

Two of the real eigenvalue extraction methods available in MSC Nastran are classified as tracking methods. These are:

- Inverse Power Method
- Sturm Modified Inverse Power Method.

The final real eigenvalue extraction method available in MSC Nastran is the Lanczos method. The Lanczos method combines the best characteristics of the tracking and transformation methods.

The Standard Tridiagonal Methods (GIV and HOU)

The Givens(GIV) and Householder (HOU) modal extraction methods require a positive definite mass matrix. There are no restrictions on the stiffness matrix other than that it must be symmetric. These matrices will always result in real (positive) roots. The tridiagonal methods are the most efficient methods for small problems and problems with dense matrices when a large portion of the eigenvectors are needed. These methods find all of the eigenvalues in the structure as a result of the transformation process. The user must make a specific request for the required eigenvectors (mode shapes). The tridiagonal methods do not take advantage of the sparse matrices, but they are efficient with the dense matrices sometimes created using dynamic reduction. The steps used in the tridiagonal methods are as follows:

1. Perform a Cholesky decomposition of the mass matrix:

   $$[M] = [L][L]^T \tag{B-2}$$

   where $[L]$ is a lower triangular matrix.

2. The symmetric eigenvalue equation can be written as:

   $$[J - \lambda I]\{w\} = 0 \tag{B-3}$$

   where $[J]$ is the symmetric matrix and $[I]$ is the identity matrix. To do this, premultiply equation (2-1) by $[L]^{-1}$ and substitute for $[M]$ from equation (B-2) to set the following:

   $$[L]^{-1}[K]\{u\} - \lambda[L]^{-1}[L][L]^T\{u\} = 0 \tag{B-4}$$

   Then make the transformation:

   $$\{w\} = [L]\{u\} \tag{B-5}$$

   which reduces equation (B-4) to equation (B-3) using the following:

   $$[J] = [L]^{-1}[K][L]^{-1,T} \tag{B-6}$$

3. The $[J]$ matrix is converted to a tridiagonal matrix by a transformation method according to the Givens or Householder method. (A tridiagonal matrix is a matrix where the only nonzero terms in the i-th column are the $i - 1$, i, $i + 1$ row.)
4. All the eigenvalues of the tridiagonal matrix are extracted using a modified QR algorithm.
5. The eigenvectors are computed over a given frequency range or, at user option, for a given number of eigenvalues using inverse iteration. Since the roots are known to high accuracy at this point, the vectors converge rapidly.

6. Physical eigenvectors are recovered by performing the inverse transformation to that made in Step 3 and then solving equation (B-5) as follows:

   $$\{u\} = [L]^{T,-1}\{w\}$$

7. If a SUPORT entry is used, the first N_r eigenvectors computed are discarded. The rigid body modes from the SUPORT computation are put in their place, and the corresponding eigenvalues are set to zero. The SUPORT operation does cause some redundant operations to be performed as the rigid body modes are computed by two methods and then one set is discarded. This is generally a minor issue.

8. The back-transformed vectors $[PHI]$ are given a final stage of orthogonalization with respect to the mass matrix. The modal mass matrix is calculated as follows:

   $$[M_{modal}] = [\phi]^T[M][\phi]$$

   then decomposed,

   $$[M_{modal}] = [L]^T[L]$$

The refined eigenvectors, $[\bar{\phi}]^T$ are found by a forward pass on the equation:

$$[L][\bar{\phi}]^T = [\phi]^T$$

Note that the tridiagonal methods fail in Step 1 if the mass matrix is not positive definite. In order to minimize this eventuality, degrees of freedom with null columns are removed from (u) by the automatic application of static condensation. This is called AUTO-OMIT. (This procedure can be bypassed by using PARAM,ASING). The application of the AUTO-OMIT process is a precaution and may not remove all possible causes of mass matrix singularity such as, a point mass offset from a grid point, but it greatly improves the reliability and convenience of the standard tridiagonal methods.

The Modified Tridiagonal Methods (MGIV, MHOU, AGIV, AHOU)

The modified tridiagonal methods, Modified Givens (MGIV) and Modified Householder (MHOU), are similar to the standard tridiagonal methods (GIV, HOU) with the exception that the mass matrix can be singular.

The steps used in the modified methods are as follows:

1. Perform a Cholesky decomposition of the positive definite matrix

   $$[K + \lambda_s M] = [L][L]^T \tag{B-7}$$

   where λ_s is a positive number automatically selected by the program to optimize the reliability and accuracy of eigenvalue extraction. It is large enough to stabilize all rigid body modes but also be as small as possible to reduce the numerical contamination. The goal of the algorithm is to find a value near the first flexible frequency of the system. It uses the diagonal terms of the mass and stiffness matrices as follows:

$$\lambda_s = \frac{1}{N^{1/2}\sum_{i=1}^{N}\frac{M_{ii}}{K_{ii}}}$$

where N is the dimension of the matrices. If M_{ii} or $X_{ii} = 0$ (an under defined degree-of-freedom) or $M_{ii}/X_{ii} > 10{**}8$ (a likely "very large" mass term used to approximate a rigid body mode), the corresponding term is omitted from the summation. If the value calculated by this approximate method does not result in a stable Cholesky decomposition, λ_s is increased by adding the arbitrary amount of 0.01. If this value also results in failure, the original value is increased by 0.02. If this value also fails, a fatal error is issued.

2. Write the eigen-equation standard form as:

$$[\bar{J} - \bar{\lambda} I]\{\bar{w}\} = 0 \tag{B-8}$$

where:

$$\bar{\lambda} = \frac{1}{\lambda + \lambda_s} \tag{B-9}$$

and:

$$[\bar{J}] = [L]^{-1}[M][L]^{-1,T} \tag{B-10}$$

by rearranging the terms to read as follows:

$$[K + \lambda_s M - (\lambda + \lambda_s)M]\{u\} = 0 \tag{B-11}$$

Then premultiply equation (B-11) by $-1/(\lambda + \lambda_s)$ and substitute for $[K + \lambda_s M]$ from equation (B-7) to obtain:

$$\left[M - \frac{LL^T}{\lambda + \lambda_s}\right]\{u\} = 0$$

Premultiply by $[L]^{-1}$ and then substitute:

$$\{\bar{w}\} = [L]^T\{u\}$$

resulting in equations (B-8), (B-9), and (B-10).

3. The remaining steps are identical to those for the unmodified methods where:

$[J] = [\bar{J}]$ and $\{w\} = \{\bar{w}\}$

Although the mass matrix is not required to be nonsingular in the modified methods, a singular mass matrix can produce one or more zero eigenvalues, λ_s, which are equivalent to infinite eigenvalues, λ_i. Due to roundoff error, these will appear in the output as very large positive or negative eigenvalues. To reduce the incidence of such meaningless results, degrees of freedom with null masses are eliminated by the AUTO-OMIT static condensation as in the case of the unmodified methods. This step can be bypassed by using PARAM,ASING.

Many times the user is not conversant in the status of the mass matrix prior to the actual matrix assembly. To assist the user in choosing the appropriate method, two options, Automatic Givens (AGIV) and Automatic Householder (AHOU) are available. The automatic methods, AHOU and AGIV, initially use the standard methods. In the first step of the method, if the mass matrix is not well-conditioned for decomposition, the method shifts to the corresponding modified method.

The modified methods are more expensive and introduce numerical "noise" due to the shift, but they resolve most of the numerical problems of the ill-conditioned mass matrix.

Operations involved in the computation of the natural frequencies and mode shapes is proportional to the number of degrees of freedom (N). The time-consuming operations for the tridiagonal methods are mass matrix decomposition and reduction to standard form ($5/2\ N^3$ for a full mass matrix, negligible for a diagonal matrix) tridiagonalization ($2/3\ N^3$ for HOU, $4/3\ N^3$ for GIV), QR root iteration ($9\ N^2$), inverse iteration for vectors (N^2 per vector), and inverse transformation of vectors (N^2 per vector for HOU, $2\ N^2$ for GIV). As N becomes large (greater than 100, perhaps), terms in N^3 dominate.

The Inverse Power Methods (INV, SINV)

The inverse power method with shifts is particularly effective when the mass and stiffness matrices are sparse and when only a few of the eigenvalues are required. In MSC Nastran, the method is used as a stand-alone method to find all of the eigenvalues within a user-specified domain.

If we define the eigenvalue problem as:

$$[K - \lambda M]\{u\} = 0 \tag{2-12}$$

Let:

$$\lambda = \lambda_o + \Lambda$$

where λ_o is called the shift point.

The iteration algorithm is:

$$[K - \lambda_o M]\{w_n\} = [M]\{u_{n-1}\} \tag{2-13}$$

$$\{u_n\} = \frac{1}{C_n}\{w_n\} \tag{2-14}$$

where C_n is equal to the value of the element of $\{w_n\}$ with the largest absolute value. $\{u_n\}$ is then placed on the right side of equation (2-13) and iteration continues. It is easy to prove that $1/C_n$ converges to λ_1, the shifted eigenvalue nearest to the shift point, and that $\{u_n\}$ converges to the corresponding eigenvector of equation (2-12). The following observations are pertinent.

- A triangular decomposition of the matrix $[K - \lambda_o M]$ is required in order to evaluate $\{w_n\}$ from (2-13). The effort required to perform the triangular decomposition is greatly reduced if $[K - \lambda_o M]$ is a narrow band matrix. Structural analysis is characterized by narrowly banded stiffness and mass matrices. From Sturm's theorem, the number of negative terms on the factor diagonal is exactly equal to the number of roots below λ_o. This number is printed out at every shift and is identified as the Sturm sequence number.
- It is unnecessary for $[K]$ to be nonsingular, so that rigid body modes cause no special difficulty. Similarly, null rows in $[M]$ cause no problems. If they are removed by static condensation, they will increase the cost of the solution unnecessarily. This occurs primarily due to the increase in bandwidth inherent in static condensation

- The shift point λ_o may be changed (at the cost of an additional triangular decomposition) in order to improve the rate of convergence toward a particular eigenvalue or to improve the accuracy and convergence rate after several roots have been extracted from a given shift point. The next shift point is computed from the residual terms remaining after computing the present root by a root estimation algorithm. See the MSC Nastran Numerical Methods User's Guide for details.
- λ_o can be placed to obtain the eigenvalues within a desired frequency band and not just those that have the smallest absolute value.
- The Sturm sequence data can be used to ensure that all eigenvalues within the requested frequency band have been found. This can be done manually with DIAG 16 for the INV method or automatically with the SINV method (see the MSC Nastran Numerical Methods User's Guide for details). In the SINV method, the requested band is broken into subregions, and the Sturm number is determined at each boundary of the subregion. If the root estimation algorithm predicts that the next root is out of the subregion but the Sturm sequence data states that more roots remain in the subregion, a bisection method is used to iterate to the missing roots. All roots for a subregion must be found before the method is allowed to enter the next subregion. The INV method tends to compute shifts that wander more about the requested frequency band and computes more roots outside the band than the SINV method does.

In the practical application of the tracking methods, most of the numerical efficiency issues (trial eigenvectors, convergence criteria, etc.) are automatically handled by MSC Nastran. The shift points must be selected so that the numerical algorithms can operate in an optimum manner. Generally, this is handled by MSC Nastran, but the user can assist MSC Nastran through a careful choice of frequency range.

If the SUPORT entry is used, the rigid body modes calculated are used directly as the first set of eigenvectors with their eigenvalues set to zero. Iteration then proceeds for the nonzero roots. Use of the SUPORT feature reduces costs of the INV and SINV methods considerably besides giving the aesthetic benefits of exact zero-frequency rigid body modes. The SUPORT entry is recommended for use with these methods.

The Shifted Block Lanczos Method

The Lanczos method (LAN) available in MSC Nastran is an extremely reliable and accurate method for solving the large sparse eigenvalue problems typical of many structures.

A shifted block Lanczos algorithm is implemented in MSC Nastran. A summary of that method is given here with special emphasis on the effect of user inputs.

For the standard eigenvalue problem written in the form:

$$[A]\{x\} = \lambda\{x\}$$

the Lanczos method computes a sequence of vectors q called Lanczos vectors and scalars Q_j, A. The basic recurrence relation for the j-th step of the Lanczos algorithm can be written as follows:

$$\{r_{j+1}\} = [A]\{q_j\} - \alpha_j\{q_j\} - \beta_j\{q_{j-1}\}$$

$$\alpha_j = \{q_i\}^T[A]\{q_j\}$$

$$\beta_{j+1} = \|r_{j+1}\|$$

$$\{q_{j+1}\} = \frac{\{r_{j+1}\}}{B_{j+1}}$$

The first j-steps of the method can be combined and written in matrix form:

$$[A][Q_j] - [Q][T_j] = \beta_{j+1}\{q_{j+1}\}\{e_j\}^T$$

where $\{e_j\}$ = unit vector with the j-th component equal to one and all other components equal to zero

$\{Q_j\} = (\{q_1\}, \{q_2\}, ..., \{q_j\})$ with $[Q_j^T][Q_j] = [I]$

$$[T_j] = \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \beta_3 & \alpha_3 & \beta_4 & \\ & & \ddots & \ddots & \ddots \\ & & & \beta_j & \alpha_j \end{bmatrix}$$

The eigenvalues of the tridiagonal matrix $[T_j]$ are called Ritz values. Some of these Ritz values are good approximations to eigenvalues of $[A]$. The corresponding Ritz vectors (x_i) are computed from the eigenvectors of $[T_j]$ and the Lanczos vectors. An eigenpair is defined as an eigenvalue (s_i) and its corresponding eigenvector $\{\theta_{ik}\}$. Let $(\{\theta_{ik}\}, s_i)$ be an eigenpair of the tridiagonal matrix $[T_j]$. Then an approximate eigenpair for the original problem is $(\lambda_i, \{x_i\})$ where:

$$\lambda_i = \theta_i$$

$$\{x_i\} = \{Q_j\}\{s_i\}$$

As implied by the formulation of the algorithm, the method computes a tridiagonal matrix and a set of Lanczos vectors that become larger with each iteration. The algorithm in exact arithmetic can be terminated when $B_{j+1} = 0$ or when sufficiently many eigenvalues of $[A]$ have been computed. Lanczos also handles the effects of roundoff.

The algorithm that has been incorporated in MSC Nastran is a shifted block version of the basic Lanczos recurrence relation. The blocking approach to the Lanczos algorithm (working with blocks of p Lanczos vectors simultaneously) enables the method to handle eigenvalues of high multiplicity including shifting in the algorithm which helps to speed the convergence of the method.

In this method, the A matrix is replaced by the sum of the stiffness matrix plus shifted mass matrix, similar to the approach in the modified tridiagonal methods. This allows solution of problems with rigid body modes and generally faster convergence. The shifts (trial roots) are influenced by user input. The user is required to input either a frequency band of interest, a desired number of roots, or a combination of this data.

As an example, suppose that both the lower bound (V1) and upper bound (V2) of the frequency band are defined by the user as well as the number of roots desired (ND). The main objective for the shift strategy is to encompass the region containing the eigenvalues of interest in which all of the eigenvalues have been computed and the number of eigenvalues has been verified with a Sturm sequence check.

The first shift is at V1 to determine the Sturm sequence number. The factor matrix is stored for possible later use. If V1 is not defined by the user, the first shift is made at λ_s. The second shift point is chosen at V2.

After a Lanczos iteration based on the selected shift and the computation of the corresponding Ritz values, some values will be found to be acceptable eigenvalues while other eigenpairs will fail convergence tests. A decision is made as to whether it is more efficient to re-shift or continue with further iteration. The re-shift point is chosen from the existing Ritz values (smaller than V2) or from the stored factor from the shift at V2. At this point, the number of accepted eigenvalues and the Sturm sequence numbers at each end of the region are known.

If iteration with the factor from V2 does not produce all of the roots desired, two more types of shift will be used. If it appears that the missing roots are near the lowest shift, an approximate first mode eigenvalue is used. This is either a nonzero SHFSCL field from the EIGRL entry or the a similar shift term used in the modified tridiagonal methods. If the missing roots are elsewhere, bisection between prior shifts is used. If all requested eigenpairs still cannot be found, a user information message is printed, and all accepted eigenpairs are output. If some roots have been found, the module exits normally. A fatal error is issued if no roots have been found. If accepted Ritz values are found outside the requested region, they are discarded.

The strategy for selecting block size is to first find the minimum of N/2, 2E, p and MAXSET. N is the problem size, E the number of roots desired, p the number of Lanczos vectors that will fit into memory, and MAXSET the user selection for block size (which has a default of 7).

Another efficiency feature unique to the Lanczos method is selective orthogonalization of the trial eigenvectors. All of those contained in the same block are made orthogonal to one another. Tests are made, at several steps, to determine if further orthogonalization between vectors in different blocks is required. This avoids the cost term cubic on E (the number of eigenvectors calculated) which becomes the dominant term for GDR and the SINV methods as E becomes large.

If the SUPORT entry is used, the $[D_{ir}]$ matrix is merged with an identity matrix $[I_{rr}]$ to form raw rigid body vectors $[D_{ar}]$. These raw vectors are used as initial vectors. If the r-set has been well-chosen, the rigid body modes will converge rapidly. If they have been chosen poorly, the Lanczos iterations will continue until a good set of eigenvectors have been found. The corresponding roots are set to the aesthetically pleasing value of zero rather than to the small numerical zeros likely to be produced by the actual computations.

The Lanczos method is more forgiving than the other methods since it repairs the effects of a poor r-set selection. However, the repeatability of rigid body modes computed by Lanczos with the SUPORT entry is not the same as with other extraction methods. Rigid body modes computed using the SUPORT entry and a non-Lanczos method will produce identical modes, irrespective of the method. If the rigid body modes produced by the Lanczos method with SUPORT entries are compared with those produced by any of the other methods, the Lanczos modes will be a proper linear combination of the modes from the other methods and may be markedly different in appearance. Furthermore, if small changes are made to the stiffness matrix, the Lanczos modes again may be markedly different, although they are again a proper linear combination of the set produced by other methods. This lack of repeatability makes the Lanczos method unusable with the scaled response spectra capability of MSC Nastran.

The cost for the LAN method is dominated by the decomposition, inverse iteration, and orthogonalization operations at the size of the input matrices N. The number of roots per decomposition varies by problem and is difficult to determine *a priori*.

User Interface for Real Eigenvalue Analysis

The EIGR and EIGRL Bulk Data entries are the heart of the user interface because they define a method and select parameters that control the eigenvalue extraction procedure. The EIGRL entry is used for the LANCZOS method, and the EIGR entries are used for all of the other methods.

The data used for the EIGR entry depend on the method selected. The basic data required is related to the eigenvalue extraction method chosen and to the frequency range or number of required roots. The basic format of the Bulk Data entry is as follows:

1	2	3	4	5	6	7	8	9	10
EIGR	SID	METHOD	F1	F2	NE	ND			+BC
+BC	NORM	GID	CID						

The SID (Field 2) is the Set ID associated with this EIGR entry. A particular EIGR entry is activated in an analysis using the METHOD = SID entry in the Case Control Section. If the Set ID on the METHOD and EIGR match, the eigenvalues will be extracted using the method referenced on that entry.

The METHOD Entry LANzos (Field 3) selects the eigenvalue method from the following list:

1.	INV	Inverse Power
2.	AHOU	Automatic Householder
3.	SINV	Sturm Modified Inverse Power
4.	GIV	Givens Method
5.	HOU	Householder Method
6.	MGIV	Modified Givens Method
7.	MHOU	Modified Householder Method
8.	AGIV	Automatic selection of Givens or Modified Givens

The F1 Entry (Field 4) is used to specify the lowest frequency of interest in the eigenvalue extraction.

The F2 Entry (Field 5) is used to specify the highest frequency of interest in the eigenvalue extraction.

The NE entry (Field 6) is used by the INV method only. It defines the estimated number of roots in the range. A good estimate will result in a more efficient solution.

The ND entry (Field 7) is used to specify the desired number of roots in the range of interest.

The NORM (Field 2 – continuation) entry on the continuation card is used to specify the method of eigenvector normalization. The choices are:

1. MASSMass normalization (Default – if used continuation entry not required)
2. MAXNormalization to maximum A-set component
3. POINTNormalization to user-defined degrees of freedom

The POINT entry (Field 3 – continuation) is used to specify the point of interest from the analysis set for POINT normalization only.

The CID entry (Field 4 – continuation) is used to specify the component of displacement for POINT normalization only.

There is an interrelationship between the F1, F2, and ND field on the EIGR entry.

Table 2-1 EIGR Input Parameters

	When METHOD = INV or SINV	When METHOD = GIV, MGIV, HOU, or MHOU
F1,F2	Frequency range of interest (Real ≥ 0.0). F1 must be input. If METHOD = SINV and ND is blank, then F2 must be input.	Frequency range of interest (Real ≥ 0.0; F1 < F2. If ND is blank, F1 and F2 are ignored. If ND is blank, eigenvectors are found whose natural frequencies lie in the range between F1and F2.
NE	Estimate of number of roots in range (Required for METHOD = "INV" (Integer > 0). Not used by SINV method.	Not used.
ND	Desired number of roots (Default is 3*NE, INV only) (Integer > 0). If this field is blank and METHOD = SINV, then all roots between F1 and F2 are searched and the limit is 600 roots.	Desired number of eigenvectors (Integer ≥ 0). If ND is zero, the number of eigenvectors is determined from F1 and F2 (Default = 0). If all three are blank, then ND is automatically set to one more than the number of degrees of freedom listed on SUPORT entries.

The rules for METHOD = GIV, HOU, MGIV, MHOU, AGIV and AHOU are identical. If any of these methods are selected, MSC Nastran will find all of the eigenvalues but will compute only the eigenvectors specified by F1 and F2 or those specified by ND (the "desired" number). F1 and F2 specify the lower and upper bounds of the frequency range in which eigenvectors are computed, and ND specifies the number of eigenvectors, starting with the lowest (or the first rigid body mode, if present). If F1, F2, and ND entries are present, ND takes precedence. Note that the default for F1 is 0.0.

The following examples demonstrate the use of the EIGR data entry.

1	2	3	4	5	6	7	8	9	10
EIGR	1	AHOU			10				

In the first example (SID = 1), the automatic Householder method is selected with 10 roots requested. Since the default "MASS" eigenvector normalization is requested, no continuation entry is needed.

EIGR	2	AHOU		100.0					+2
+2	MAX								

For the second example (SID = 2), the same method is requested, but all the roots below 100 cycles per unit time are requested with “MAX” vector normalization.

1	2	3	4	5	6	7	8	9	10
EIGR	3	SINV	0.0	100.0		6			+3
+3	POINT	32	4						

For the third example (SID = 3), the inverse iteration method with Sturm sequence checks is requested for the first six roots in the range specified. The POINT normalization is requested for grid point 32 in the R1 direction.

If METHOD = SINV, the values of F1, F2, and ND determine both the number of eigenvalues and the eigenvectors that will be computed. These entries also provide hints to help MSC Nastran find the eigenvalues. F1 and F2 specify the frequency range of interest within which MSC Nastran will search for modes. MSC Nastran will attempt to find all of the modes in the range F1, F2 or the number specified by ND, whichever is less. If searching stops because ND modes are found, there is no guarantee that they will be the lowest eigenvalues. Iterative methods only guarantee the solution for modes nearest the shift point. If ND modes are not found in the range of interest, SINV will usually find one mode outside the range F1, F2 (or possibly more) before stopping the search.

The inverse power method is particularly efficient when only a small number of eigenvalues and eigenvectors are wanted. Very often only the lowest mode is of interest. The following example illustrates an EIGR entry which will extract only the lowest nonzero eigenvalue.

EIGR	SID	METHOD	F1	F2	NE	ND			
EIGR	13	INV	0.0	.011	1				

It is assumed in the example that the frequency of the lowest mode is greater than 0.01 cycles per unit time. The program will find one eigenvalue outside the range F1, F2 and stop the search. The eigenvalue found is guaranteed to be the lowest nonzero eigenvalue (or a member of the lowest closely spaced cluster of eigenvalues in cases with pathologically close roots) provided that there are no negative eigenvalues and that the SUPORT entry has been used to specify the correct number of zero eigenvalues. The NE entry (field 6) is used only by the INV method.

The fields of the EIGRL entry are designed to select and set parameters for the Lanczos method as well as request diagnostics.

EIGRL	SID	V1	V2	ND	MSGL	VL	MAXSET	SHFSCL	
EIGRL	1	0.1	3.2		10				

The V1 field defines the lower frequency bound, the V2 field defines the upper frequency bound, and the ND field defines the number of eigenvalues and eigenvectors desired in the region. The V1 and V2 are expressed in units of cycles per unit time. Examples of the results of using explicit or default values for the V1, V2, and ND fields are shown in Table 2-2. The defaults on the EIGRL entry are designed to provide the minimum number of roots in cases where the input is ambiguous.

Table 2-2 Model Outputs from EIGRL Input Options

Case	V1	V2	ND	Number and Type of Roots Found
1	V1	V2	ND	Lowest ND or all in range; whichever is smaller
2	V1	V2		All in range
3	V1		ND	Lowest ND in range [V1, +00]
4	V1			Lowest root in range [V1, +00]
5			ND	Lowest ND roots in [-00, +00]
6				Lowest root
7		V2	ND	Lowest ND roots below V2
8		V2		All below V2

The MSGLVL field of the EIGRL entry is used to control the amount of diagnostic output provided by Lanczos. The default value of 0 produces no diagnostic output. The values 1, 2, or 3 provide more output with the higher values providing increasingly more output. Typically, the default value is sufficient. In some cases, higher diagnostics levels may be in order to help resolve difficulties with special modeling problems.

The MAXSET field is used to control the block size, the number of appropriate eigenvectors computed in the outermost iteration loop. The default value of 7 is recommended for most applications. There may be special cases where a larger value may result in quicker convergence of highly multiple roots or a lower value may result in more efficiency when the structure is lightly coupled. However, the default value has been chosen after reviewing results from a wide range of problem types on several different computer types with the goal of minimizing the sum of CPU and I/O cost.

One common occurrence is for the block size to be reset by MSC Nastran during the run because there is insufficient memory for a block size of 7. Computational efficiency tends to degrade as the block size decreases. It is, therefore, important to examine the EIGENVALUE ANALYSIS SUMMARY output to determine that the program has sufficient memory to use an efficient block size. A smaller block size may be more efficient when only a few roots are requested. The minimum recommended block size is 2.

The SHFSCL field allows a user-designated shift to be used when all other shifting strategies fail. This entry is only used under special circumstances. However, when used, its value should be set to the expected first natural frequency.

The Lanczos method normalizes the computed eigenvectors using the MASS method. The other normalization methods are not available.

Solution Control For Normal Modes

When used as an independent solution, normal modes analysis is available in SOL 103. The Executive Control Section can also contain diagnostic DIAG 16 which will print the iteration information used in the INV or SINV methods.

One of the most important entries in the Case Control Section is the METHOD entry. This entry is required. The Set ID referenced by the METHOD entry refers to the Set ID of an EIGR entry or an EIGRL entry in the Bulk Data.

When a modal extraction is performed, the MSC Nastran output file contains various diagnostic messages and an Eigenvalue Table. Optional grid and element output are available using standard Case Control output requests. These requests are summarized in Table 2-3.

Table 2-3 Eigenvalue Extraction Output Requests

Grid Output	
DISPLACEMENT (or VECTOR)	Requests the eigenvector (mode shape) for a set of grid points
GPFORCE	Requests the Grid Point Force Balance Table to be computed for each mode
GPSTRESS	Requests grid point stresses to be computed for a set of grid points. This request must be accompanied by the ELSTRESS Case Control request and the definition of Stress Surfaces and/or Stress Volumes in the OUTPUT(POST) section of the Case Control. This request also requires the use of Rigid Format Alter RF3D81 when used in SOL 3.
SPCFORCE	Requests Forces of Single Point Constraint to be computed for a set of grid points for each mode.
Grid Output	
Element Output	
ELSTRESS (or STRESS)	Requests the computation of modal stresses for a set of elements for each mode
ESE	Requests the computation of modal element strain energies for a set of elements for each mode
ELFORCE (or FORCE)	Requests the computation of modal element forces for a set of elements for each mode
STRAIN	Requests the computation of the modal element strains for a set of elements
MODES	A special Case Control request which permits selective output requests to be processed on selective modes

Note: For large problems, if GPSTRESS is requested for all eigenvalues, it is likely that a large amount of time will be spent on stress recovery.

In addition to Bulk Data entries required to define the structural model, the only required Bulk Data entry is the eigenvalue selection entry, EIGR or EIGRL. The EIGR entry is used to select the modal extraction parameters for the Givens, Householder, Modified Givens, Modified Householder, Automatic Givens, and Automatic Householder methods. The EIGRL entry is used to select the modal extraction parameters for the Lanczos method.

C Nonlinear Analysis

- General Description of Nonlinear Capabilities 599
- Nonlinear Modal Analysis 687

General Description of Nonlinear Capabilities

This section discusses the procedures in SOL 106, SOL 129, SOL 153, and SOL 159. Nonlinear effects in structures occur mainly due to nonlinear materials and large deformations (rotations and/or strains). Contact problems exhibit nonlinear effects due to changes in boundary conditions. All of these attributes may be represented by MSC Nastran nonlinear elements.

The nonlinear element library consists of:

- CBUSH and CBUSH1D for generalized nonlinear springs and dampers.
- CROD, CONROD, and CTUBE for unidirectional truss members.
- CBEAM for axially and laterally deforming line members.
- CQUAD4 and CTRIA3 for membrane, plate and shell modeling and hyperelastic plane strain.
- CQUAD8, CQUAD, and CTRIA6 for hyperelastic plane strain.
- CQUADX and CTRIAX for hyperelastic axisymmetric modeling.
- CHEXA, CPENTA, and CTETRA for solid modeling.
- CGAP and slideline contact (BCONP, BLSEG, BFRIC, BWIDTH) for contact and friction modeling.

For advanced nonlinear analysis, we recommend you use SOL 400 (see MSC Nastran Nonlinear User's Guide.)

Nonlinear elements may be combined with linear elements for computational efficiency if the nonlinear effects can be localized. Primary operations for nonlinear elements are updating element coordinates and applying loads for large displacements, and the internal relaxation iteration for material nonlinearity. Refer to Table 3-1 for a summary of the nonlinear element properties.

The geometric nonlinearity becomes discernible when the structure is subjected to large displacement and rotation. Geometric nonlinear effects are prominent in two different aspects: geometric stiffening due to initial displacements and stresses, and follower forces due to a change in loads as a function of displacements. These effects are included, but the large deformation effect resulting in large strains is implemented only for hyperelastic materials. Large strain effects are not available for elastic-plastic, creep, or any other material types.

Material nonlinearity is an inherent property of any engineering material. Material nonlinear effects may be classified into many categories and those included in MSC Nastran are plasticity, nonlinear elasticity, hyperelasticity, creep, and viscoelasticity. Creep and viscoelasticity, implemented as a generalized nonlinear viscoelastic capability, may be coupled with plasticity. Many options are available for yield criteria and hardening behavior in plasticity.

The primary solution operations are gradual load or time increments, iterations with convergence tests for acceptable equilibrium error, and stiffness matrix updates. The iterative process is based on the modified-Newton's method combined with optional expeditious methods such as the quasi-Newton (BFGS) update and the line search. The stiffness matrix updates are performed occasionally to improve the computational efficiency, and may be overridden at the user's discretion. A number of options for arc-length methods are also available for snap-through or post-buckling analysis of static problems. For transient response analysis, a number of options are available for implicit direct time integration, combined with adaptive and expedient iteration strategies similar to those implemented for static analysis.

Solution Sequences 106, 129, 153, and 159 consolidate all the nonlinear features described above. SOLs 106 and 153 are applicable to static, quasi-static, and nonlinear buckling analyses. SOLs 129 and 159 are primarily applicable to dynamic

transient response analysis with some limited static analysis capability. SOLs 153 and 159 may also be used to perform a nonlinear heat transfer analysis. See the *MSC Nastran Thermal Analysis User's Guide*. These solution sequences can accommodate superelements and provide easy restarts from the database.

Table 3-1 Summary of Properties of the Nonlinear Elements

Connectivity	Properties	No. of GRID Points	No. of Gauss Points	Geometric Nonlinearity	Material Property			Static Loads			DATA Recovery				
					Nonlinear Isotropic	Anisotropic	Orthotropic	Thermal	Pressure	Gravity	Stress	Grid Point Stress	Force	Structure Plot	Contour Plot
BCONP BLSEG	BWIDTH BFRIC	variable		X							X		X		
CBEAM	PBEAM	3	16*	X	1†			X		X	X		X	X	
	PBCOMP														
CBUSH	PBUSH	1 or 2			X	X					X		X	X	X
	PBUSHT														
CBUSH1D	PBUSH1D	1 or 2		X	X						X		X	X	X
CGAP	PGAP	3	0								X		X		
CHEXA	PSOLID	8	8	X	1	9		X	X	X	X	X		X	
	PLSOLID	8	8	X	MATHP			X	X	X	X			X	
		20‡	27												
CONROD	---	2	1	X	1			X		X	X	X		X	X
CPENTA	PSOLID	6	6	X	1	9		X	X	X	X	X		X	
	PLSOLID	6	6	X	MATHP			X	X	X	X			X	
		15‡	21												
CQUAD4	PSHELL	4	5*	X**	1	2	8	X	X	X	X	X	X	X	X
	PCOMP														
	PLPLANE	4	4	X	MATHP			X		X	X			X	
CQUAD8	PLPLANE	8‡	9	X	MATHP			X		X	X			X	
CQUAD	PLPLANE	9‡	9	X	MATHP			X		X	X			X	

Table 3-1 Summary of Properties of the Nonlinear Elements (continued)

Connectivity	Properties	No. of GRID Points	No. of Gauss Points	Geometric Nonlinearity	Material Property			Static Loads			DATA Recovery				
					Nonlinear Isotropic	Anisotropic	Orthotropic	Thermal	Pressure	Gravity	Stress	Grid Point Stress	Force	Structure Plot	Contour Plot
CROD	PROD	2	1	X	1			X		X	X		X	X	
CTETRA	PSOLID	4,10	1	X	1	9		X	X	X	X	X		X	
	PLSOLID	4	1	X	MATHP			X	X	X	X			X	
		10‡	5												
CTRIA3	PSHELL	3	5*	X**	1	2	8	X	X	X	X	X	X	X	X
	PCOMP														
	PLPLANE	3	1	X	MATHP			X		X	X			X	
CTRIA6	PLPLANE	6‡	3	X	MATHP			X		X	X			X	
CTUBE	PTUBE	2	1	X	1			X		X	X		X	X	
CQUADX	PLPLANE	4	4	X	MATHP			X	X	X	X			X	
		9‡	9												
CTRIAX	PLPLANE	3	1	X	MATHP			X	X	X	X			X	
		6‡	3												

*May be changed by the user.

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

‡Center grid, if applicable, and midside grids may be omitted. For hyperelastic elements it is recommended that either all or no midside grids be omitted.

**Always in equilibrium in updated system.

Note:	1. Integers listed under Material properties identify (i) on MATi Bulk Data entries.
	2. MATS1 and CREEP may be attached to MAT1, MAT2, and MAT9.
	3. Edge nodes are not applicable to nonlinear elements, except hyperelastic elements and the TET10 element.

Rudiments of User Interface

The input file consists of an optional NASTRAN statement, Executive Control Section, Case Control Section, and Bulk Data Section. All the features and principles for the user interface are common in MSC Nastran; thus, all the features of the user interface for nonlinear analysis are compatible with those for the linear analysis. Any exceptions for nonlinear analysis will be explained in the relevant sections that follow.

Mechanical design is dictated by the strength, dynamic, and stability characteristics of the structure. MSC Nastran provides the analysis capabilities of these characteristics with solution sequences, each of which is designed for specific applications. The type of desired analysis is specified in the Executive Control Section by using a solution sequence identification. SOLs 106 and 153 are designed for static, quasi-static, and buckling analyses. SOLs 129 and 159 provide nonlinear transient response analysis.

The basic input data required for a finite element analysis may be classified as follows:

- Geometric data
- Element data
- Material data
- Boundary conditions and constraints
- Loads and enforced motions
- Solution methods

The first three classes of data may not be changed during the course of an analysis whereas the last three classes of data may be changed during the analysis as defined in the Case Control Section. Examples of input data are given in Listing 3-1 and Listing 3-2 for nonlinear static and dynamic analysis, respectively.

Listing 3-1 Example Input Data for Nonlinear Static Analysis

```
SOL 106                 $ NONLINEAR STATIC ANALYSIS
DIAG 8,50               $ DIAGNOSTIC PRINTOUT
CEND                    $ END OF EXECUTIVE CONTROL DATA
TITLE = TEST OF CTETRA ELEMENT (CUBE SUBJECT TO UNIAXIAL LOADING)
       DISP = ALL
     STRESS = ALL
  SPC = 100
SUBCASE 1
  SUBTITLE = ELASTIC -- LOAD TO 850. PSI
  LABEL = LOAD TO YIELD
  LOAD = 50
  NLPARM = 50
SUBCASE 2
  SUBTITLE = PLASTIC -- LOAD TO 1000. PSI
  LABEL = LOAD BEYOND YIELD
  LOAD = 100
  NLPARM = 100
SUBCASE 4
  SUBTITLE = ELASTIC -- UNLOAD COMPLETELY TO 0. PSI
  LABEL = FULL UNLOAD
  LOAD = 200
  NLPARM = 200
$OUTPUT(PLOT) $ PLOT DATA ADDED FOLLOWING THIS LINE IF NEEDED
$ END OF CASE CONTROL DATA
BEGIN BULK
$ PARAMETERS FOR NONLINEAR ITERATION
NLPARM      50     1     AUTO     UPW     NO
NLPARM     100     8     SEMI     UPW     NO
NLPARM     200     2     AUTO     UPW     NO
  .
  .
  .

ENDDATA
```

Listing 3-2 Example Input Data for Nonlinear Dynamic Analysis

```
SOL 129              $ NONLINEAR TRANSIENT ANALYSIS
DIAG 8,50            $ DIAGNOSTIC PRINTOUT
CEND                 $ END OF EXECUTIVE CONTROL DATA
TITLE = GAP VERIFICATION BY NONLINEAR TRANSIENT ANALYSIS SEALL = ALL $ FOR ALL MATRICIES
  SET 1 = 201        $ ,202,200
  SET 3 = 22         $ ,23
  SET 4 = 20         $ CONROD
    DISP = 1
    STRESS = 3       $ FOR GAP FORCES
    ELFORCE = 4
  LOADSET = 20
  `SPC = 100
SUBCASE 1            $ UP TO 0.5 SECONDS
  DLOAD = 210
  TSTEPNL = 22
SUBCASE 2            $ UP TO 1 SECOND
  DLOAD.= 210
  TSTEPNL = 21
OUTPUT(XYPLOT)
  CSCALE = 1.3
    XTITLE = TIME IN SECONDS
    YTITLE = DISPLACEMENT IN INCHES
    YMIN = 0.
    YMAX = 15.
  XYPLOT DISP RESP/201(T1),202(T1),200(T1)
$ END OF CASE CONTROL DATA
BEGIN BULK
TSTEPNL     21     200     .0025     2     AUTO     10
  .
  .
  .

ENDDATA
```

Case Control Section

The Case Control Section is used to define input data sets, output features, and subcases. The subcase structure in MSC Nastran provides a unique means of changing loads, boundary conditions, and solution methods by making selections from the Bulk Data. Confining the discussion to SOL 106 and SOL 129, loads and solution methods may change from subcase to subcase on an incremental basis. However, constraints can be changed from subcase to subcase only in the static solution sequence. As a result, the subcase structure determines a sequence of loading and constraint paths in a nonlinear analysis. The subcase structure also allows the user to select and change output requests for printout, plot, etc., by specifying set numbers with keywords. Any selections made above the subcase specifications are applicable to all the subcases. Selections made in an individual subcase supersede the selections made above the subcases. Table 3-2 summarizes the Case Control commands for nonlinear analysis.

Table 3-2 Summary of Nonlinear Case Control Commands

Load Selection	
LOAD	Selects static loading condition.
CLOAD	Selects static load combination for superelements.
DLOAD	Selects dynamic loading conditions.
LOADSET	Selects static load sets defined on the Bulk data LSEQ.
NONLINEAR	Selects nonlinear loading (NOLINi) for transient response.
IC	Selects initial conditions for transient response.
Solution Method Selection	
METHOD	Selects methods for eigenvalue analysis.
NLPARM	Selects iteration methods for nonlinear static analysis.
TSTEPNL	Selects iteration methods for nonlinear transient analysis.
Output Requests	
DISPLACEMENT	Requests output for displacements of physical points.
VELOCITY	Requests output for velocities of physical points.
ACCELERATION	Requests output for acceleration of physical points.
ELFORCE	Requests output for element forces.
STRESS	Requests output for element stresses.
BOUTPUT	Requests output for slideline contact.
SPCFORCES	Requests output for constraint forces of SPC points.
NLLOAD	Requests output for NOLINi in transient response.
OUTPUT(PLOT)	Requests the beginning of the plotter output.
Superelement Control	

Table 3-2 Summary of Nonlinear Case Control Commands (continued)

SUPER	Specifies the superelement identification number and the load sequence number.
SEALL	Combines the functions of SEMG, SELG, SEKR, SEMR, and SELR.
SEKR	Specifies the superelement identification numbers for which stiffness matrices will be assembled and reduced.
SELG	Specifies the superelement identification numbers for which load vectors will be generated.
SELR	Specifies the superelement identification numbers for which the static load matrices will be assembled and reduced.
SEMG	Specifies the superelement identification numbers for which stiffness, mass, and damping matrices will be generated.
SEMR	Specifies the superelement identification numbers for which the mass and damping matrices will be assembled and reduced.

Bulk Data Section

All details of the input data are specified in the Bulk Data Section. Constraint, load, and solution method Bulk Data entries are not used unless they are selected in the Case Control Section. The nonlinear Bulk Data entries are summarized in Table 3-3. All the input data designed specially for nonlinear analysis will be described in the following sections.

Table 3-3 Summary of Nonlinear Bulk Data Entries

Element Connectivity	
CBEAM	Defines connection for beam element.
CBUSH	Defines connections for generalized spring and damper.
CBUSH1D	Defines connections for rod type spring and damper.
CGAP	Defines connection for gap or frictional element.
CHEXA	Defines connection for six-sided solid element.
CONROD	Defines connection and properties for rod.
CPENTA	Defines connection for five-sided solid element.
CQUAD4	Defines connection for quadrilateral element with plane strain or bending and membrane stiffness.
CQUAD8 and CQUAD	Defines connection for plane strain hyperelastic quadrilateral element.
CROD	Defines connection for rod with axial and torsional stiffness.
CTETRA	Defines connection for four-sided solid element.

Table 3-3 Summary of Nonlinear Bulk Data Entries (continued)

CTRIA3	Defines connection for triangular element with plane strain or bending and membrane stiffness.
CTRIA6	Defines connection for hyperelastic plane strain triangular element.
CQUADX	Defines connection for axisymmetric hyperelastic quadrilateral element.
CTRIAX	Defines connection for axisymmetric hyperelastic triangular element.
CTUBE	Defines connection for a tube.
Element Properties	
PBCOMP	Defines properties for composite CBEAM.
PBEAM	Defines properties for CBEAM.
PBUSH and PBUSHT	Defines properties for CBUSH.
PBUSH1D	Defines properties for CBUSH1D.
PCOMP	Defines properties for composite material laminate.
PGAP	Defines properties for CGAP.
PROD	Defines properties for CROD.
PSHELL	Defines properties for CTRIA3 and CQUAD4.
PSOLID	Defines properties for CHEXA, CPENTA and CTETRA.
PTUBE	Defines properties for CTUBE.
PLPLANE	Defines properties for hyperelastic plane elements CTRIA3, CTRIA6, CQUAD4, CQUAD8, CQUAD, CQUADX and CTRIAX.
PLSOLID	Defines properties for hyperelastic solid elements CHEXA, CPENTA and CTETRA.
Material Properties	
CREEP	Defines creep material properties.
MAT2	Defines anisotropic material properties for shell elements.
MAT8	Defines orthotropic material properties for shell elements.
MAT9	Defines anisotropic material properties for solid elements.
MATS1	Defines properties for plastic and nonlinear elastic material.
MATHP	Defines properties for hyperelastic material.
TABLEDi	Defines a function for internal load versus disp or velocity for bush element.
TABLES1	Defines a function for stress-dependent material properties or experimental data for hyperelastic material.
TABLEST	Combines many TABLES1 entries for temperature dependent material properties.

Table 3-3 Summary of Nonlinear Bulk Data Entries (continued)

Slideline Contact	
BCONP	Defines contact parameters.
BLSEG	Defines boundary line segments.
BFRIC	Defines frictional properties for slideline contact.
BWIDTH	Defines boundary line segments width/thickness.
BOUTPUT	Defines grid points where contact output is requested.
Constraints	
SPC	Defines single-point constraints and enforced displacements.
SPC1	Defines single-point constraints.
MPC	Defines a linear relationship for two or more degrees of freedom.
Loads	
CLOAD	Defines a static load combination for superelement loads.
FORCEi	Defines concentrated load at grid point.
LSEQ	Defines static load sets for dynamic analysis.
MOMENTi	Defines moment at a grid point.
NOLINi	Defines nonlinear transient load.
PLOAD	Defines pressure loads on CQUAD4, CTRIA3, CHEXA, CPENTA, and CTETRA. Should not be used for hyperelastic plane elements CQUAD4, CQUAD8, CQUAD, CTRIA3, and CTRIA6 or for hyperelastic CHEXA, CPENTA, CTETRA with midside nodes.
PLOAD2	Defines pressure loads on shell elements, CQUAD4 and CTRIA3. Not available for hyperelastic elements.
PLOAD4	Defines pressure loads on surfaces of CHEXA, CPENTA, CTETRA, CTRIA3 and CQUAD4 elements. Not available for hyperelastic plane elements CQUAD4, CQUAD8, CQUAD, CTRIA3, and CTRIA6.
PLOADX1	Defines pressure loads on axisymmetric elements `CQUADX` and `CTRIAX`.
RFORCE	Defines load due to centrifugal force field.
TEMP	Defines temperature at grid points.
TEMPPi	Defines temperature field for surface elements.
TEMPRB	Defines temperature field for line elements.
TIC	Specifies initial values for displacement and velocity.
TLOADi	Defines loads as a function of time.
Solution Methods	

Table 3-3 Summary of Nonlinear Bulk Data Entries (continued)

NLPARM	Defines iteration methods for nonlinear static analysis.
NLPCI	Defines arc-length methods for nonlinear static analysis.
TSTEPNL	Specifies integration and iteration methods for nonlinear transient analysis.
EIGB or EIGRL	Defines eigenvalue extraction method for buckling analysis.

Parameters

Parameters constitute an important part of the input data because they maintain communications among the user, DMAP, and modules. Parameters are single-valued input data used for requesting special features or specifying miscellaneous data.

Users can change the initial values of parameters by specifying the PARAM entry in the Bulk Data Section or in the Case Control Section. Many options available in MSC Nastran are controlled by user-specified parameters. Table 3-4 lists the parameters used in nonlinear Solution Sequences 106 and 129.

Table 3-4 PARAMeters Used in Nonlinear Solution Sequences

B = Must be specified in the Bulk Data Section only.
E = May be specified in either the Bulk Data or Case Control Section.

Parameter Name	Solution Sequence		Default	Description
	106	129		
ADPCON	E	E	1.0	Adjusts automatic penalty values for slideline contact regions.
AUTOSPC	E	E	NO	Specifies automatic single-point-constraints.
AUTOSPCR	E	B	NO	Specifies automatic SPC's for residual structure.
BAILOUT	E	E	0	Terminates superelement program with singularities.
BETA		B	0.33333	Specifies control factor for Newmark-Beta method.
BUCKLE			-1	Selects nonlinear buckling analysis for restarts.
CB1,2	E		1.0	Selects superelement damping matrix coefficients.
CK1,2,3	E		1.0	Selects superelement stiffness matrix coefficients.
CM1,2	E		1.0	Selects superelement mass matrix coefficients.
CNSTRT			1	Controls restarts for constraint change.
COUPMASS	E	E	-1	Generates coupled rather than lumped mass matrices.
CURV			-1	Computes stress data in a material coordinate system.
CURVPLOT			-1	Requests x-y (curve) plots.
DBDICT	B	B	-1	Prints database dictionary at the end and beginning of the run.

Table 3-4 PARAMeters Used in Nonlinear Solution Sequences (continued)

B = Must be specified in the Bulk Data Section only.
E = May be specified in either the Bulk Data or Case Control Section.

Parameter Name	Solution Sequence		Default	Description
	106	129		
DBDRNL	E	E	0	Prints database dictionary in the nonlinear loop (SOLs 106 and 129 only).
DLOAD	B	B	0	Indicates no other changes except for loads (SOLs 106 and 129 only).
DOPT			0	Controls x spacing of curves for CURVPLOT.
ERROR	E	E	-1	Terminates run for superelement errors.
EST	E	E	2	Requests element measure and volume computation.
FACTOR	B	B	10,000	Generates SEQID on SEQGP entry.
FKSYMFAC	B		1.0	Controls symmetrization of stiffness.
FOLLOWK	E	E	YES	Include follower force stiffness.
GPECT	E	E	-1	Prints all elements connected to each grid point.
GRDPNT	E	E	-1	Executes Grid Point Weight Generator.
G		E	0.0	Specifies uniform structural damping coefficient.
INRLM		E	0	Requests appending of inertia relief modes.
KDIAG			1.0	Applies value of a spring constant to nonlinear stiffness matrix diagonals.
K6ROT	E	E	100.0	Assigns stiffness to normal rotation of CQUAD4 and CTRIA3 elements.
LANGLE	E	E	1	Selects gimbal angle approach for large rotations if 1 and rotation vector approach if 2.
LGDISP	E	E	-1	Selects large displacement effects.
LOOPID	E	E	0	Specifies loop ID in the database for restarts.
MAXLP		E	5	Maximum number of iterations for internal loop.
MAXRATIO	E	E	1 .0E + 7	Determines the singularity of the stiffness matrix.
MODEL	B	B	0	Stores several models in the database.
MPCX	B	B	0	Controls MPC and rigid element processing on restarts.
NDAMP		E	0.025	Specifies numerical damping in ADAPT method.
NINTPTS			10	Requests CURV to interpolate over all elements.
NLAYERS	B	B	5	Specifies number of integration points through thickness for CQUAD4 and CTRIA3.
NLTOL	B		2	Sets accuracy tolerances

Table 3-4 PARAMeters Used in Nonlinear Solution Sequences (continued)

B = Must be specified in the Bulk Data Section only.
E = May be specified in either the Bulk Data or Case Control Section.

Parameter Name	Solution Sequence		Default	Description
	106	129		
NMLOOP	E		0	Saves tangent stiffness in SOL 106 for normal modes and other linear solution sequences.
NOCOMPS	E		1	Suppresses composite stress output.
NODATA	B	B	0	Indicates no changes in Bulk Data (SOLs 106 and 129 only).
NOTRED		E	0	Controls t-set reduction (SOL 129 only).
OG			0	Calculates stress/strain data at grid points for CURV.
OLDELM			1	Computes element forces using old method.
OUTOPT			0	Sets format for output quantities computed by CURV.
OUTPUT			0	Jumps to data recovery.
PDRMSG		E	1	Suppresses data recovery messages.
PLOT	E	E	1	Jumps to undeformed plot.
PLOTSUP	E	E	0	Makes undeformed plots for superelements.
PLTMSG	E	E	1	Suppresses undeformed plot messages.
POST	E	E	1	Stores post-processing data blocks in database.
PRGPST	E	E	YES	Suppresses singularity printout.
PROUT	E	E	-1	Suppresses execution and printout from ELTPRT.
PRPHIVZ		E	1.0E + 37	Prints eigenvector matrix for general dynamic reduction.
RESDUAL	E	E	1	Skips to nonlinear operations (Phase 2) (SOLs 106 and 9129 only).
SDATA		E	1	Obtains additional solution set output (SOL 129 only).
SEFINAL	B	B	-1	Indicates no data changes if change SEFINAL.
SEMAP	B	B	SEMAP	Determines partitioning of superelements.
SEMAPPRT	B	B	3	Prints SEMAP table.
SENAME	E	E	SENAME	Identifies superelement by name.
SEQOUT	B	B	0	Controls output options for OLDSEQ.
SKPLOAD			1	Skips load vector calculation.
SKPMTRX			1	Skips matrix reduction and decomposition.
SLOOPID		B	0	Specifies loop ID from SOL129 database for restart.

Table 3-4 PARAMeters Used in Nonlinear Solution Sequences (continued)

B = Must be specified in the Bulk Data Section only.
E = May be specified in either the Bulk Data or Case Control Section.

Parameter Name	Solution Sequence		Default	Description
	106	129		
SOLID	B	B	0	Stores several solutions in same database.
SPCGEN	E	E	0	Places automatic SPC's in SPC1 format.
SSG3			0	Skips operations prior to FBS.
START	B	B	0	Defines number of grid points for OLDSEQ.
STIME		E	0.0	Specifies time step for restarts.
SUBID	E		1	Specifies subcase ID for restarts.
SUBSKP			0	Skips a number of subcases for restarts.
SUPER	B	B	0	Deletes ungrouped grid points.
TABID		E	2	Controls punch for response spectra.
TABS	E		0.0	Converts units of temperature input to absolute.
TESTNEG	E		-1/-2/1	Tests for negative terms on factor diagonal of matrix.
TESTSE			1.0E + 36	Tests for strain energy for iteration control.
TSTATIC		E	-1	Selects static solution with TSTATIC=1 using ADAPT method in TSTEPNL.
USETPRT	E	E	-1	Selects output type for degrees of freedom.
USETSEL	E	E	0	Controls sets for row list output option.
WTMASS	E	E	1.0	Multiplies terms of structural mass matrix by value.
W3,W4		E	0.0	Selects frequency for conversion of damping.

Nonlinear Element Output Codes for Plotting

Nonlinear elements (which reference nonlinear geometry and/or material) have different plot codes than linear elements. The plot codes be found in Item Codes (p. 1047) in the .

Nonlinear Characteristics and General Recommendations

Modeling for nonlinear analysis should follow the guidelines for good modeling practice pertaining to linear analysis, which are summarized as follows:

- The analyst should have some insight into the behavior of the structure to be modeled; otherwise, a simple model should be the starting point.

- Substructuring should be considered for the modularity of the model and/or synergism between projects and agencies involved.
- The size of the model should be determined based on the purpose of the analysis, the trade-offs between accuracy and efficiency, and the scheduled deadline.
- Prior contemplation of the geometric modeling will increase efficiency in the long run. Factors to be considered include selection of coordinate systems, symmetric considerations for simplification, and systematic numbering of nodal points and elements for easy classification of locality.
- Discretization should be based on the anticipated stress gradient; that is, a finer mesh in the area of stress concentrations.
- Element types and the mesh size should be carefully chosen. For example, avoid highly distorted and/or stretched elements (with high aspect ratio); use `CTRIA3` and `CTETRA` only for geometric or topological reasons.
- The model should be verified prior to the analysis by some visual means, such as plots and graphic displays.

Nonlinear analysis requires better insight into structural behavior. First of all, the type of nonlinearities involved must be determined. If there is a change in constraints due to contact during loading, the problem may be classified as a boundary nonlinear problem and would require CGAP elements or slide line contact algorithm. The material nonlinearity is characterized by material properties. However, the material nonlinear effects may or may not be significant depending on the magnitude and duration of the loading, and occasionally on environmental conditions. The anticipated stress level would be a key to this issue. The geometric nonlinearity is characterized by large rotations which usually cause large displacements. Intuitively, geometric nonlinear effects should be significant if the deformed shape of the structure appears distinctive from the original geometry without amplifying the displacements. There is no distinct limit for large displacements because geometric nonlinear effects are related to the dimensions of the structure and the boundary conditions. The key to this issue is to know where the loading point is in the load-deflection curve of the critical area.

SOL 106 or SOL 129 must be used if the model contains any of the following:

- A parameter LGDISP (for geometric nonlinearity).
- Any hyperelastic element (PLPLANE or PLSOLID property entry, pointing to a MATHP material).
- Any CGAP element (CGAP and PGAP).
- Slideline contact (BCONP, BLSEG, and/or BFRIC).
- Any active nonlinear material data (specified on MATS1 and/or CREEP).
- Any combination of the above.

The model may consist of superelements, but only the residual structure (superelement 0) may consist of nonlinear elements mixed with any type of linear elements. All the gaps, slideline contact regions, and hyperelastic elements are always actively nonlinear if included in the residual structure. However, other potentially nonlinear elements in the residual structure become actively nonlinear only if a parameter, LGDISP, is used and/or if they use the nonlinear material data specified on the MATS1 and/or CREEP data entries.

The subcase structure of MSC Nastran may be used in SOLs 106 and 129 to characterize a nonproportional loading path. Cyclic loading is one example. Some nonlinear problems, such as plastic deformation, creep deformation, and contact problems with frictional forces, are path-dependent; others, such as hyperelastic or purely geometric nonlinear, are path-

independent. An incremental process may not be required for a purely geometric nonlinear problem. For computational efficiency, however, the analysis of such a problem does frequently need incremental solutions.

With these points in mind, additional recommendations are imperative for nonlinear analysis:

Identify the type of nonlinearity and localize the nonlinear region for computational efficiency. If unsure, perform a linear analysis by SOL 101 or SOL 109 prior to the nonlinear analysis by SOL 106 or SOL 129, respectively.

Segregate the linear region by using superelements and/or linear elements if possible. Notice that the potentially nonlinear elements can be used as linear elements.

The nonlinear region usually requires a finer mesh. Use a finer mesh if severe element distortions or stress concentrations are anticipated.

Be prepared for restarts with the database properly stored in cases of divergence and changing constraints or loading paths via subcases.

The subcase structure should be utilized properly to divide the load or time history for convenience in restarts, data recovery, and database storage control, not to mention changing constraints and loading paths.

- The load or time for the subcase should then be further divided into increments, not to exceed 20 load steps (for SOL 106) or 200 time steps (for SOL 129) in each subcase.
- Many options are available in solution methods to be specified on NLPARM (for SOL 106) or TSTEPNL (for SOL 129) data entries. The defaults should be used on all options before gaining experience.
- Caution should be exercised in specifying CGAP element properties. In particular, the closed gap stiffness should not exceed the stiffness of the adjacent degree of freedom by 1000 times.
- Normal rotation for `CQUAD4` and `CTRIA3` elements should be restrained by a parameter K6ROT when the geometric nonlinearity is involved. Default value for parameter K6ROT is 100.
- Understand the basic theory of plasticity or creep before using these capabilities.
- Caution should be exercised in preparing input data for creep, because it is unit dependent.
- For the transient analysis by SOL 129, some damping is desirable and the massless degree of freedom should be avoided.
- The time step size for a transient response analysis should be carefully determined based on the highest natural frequency of interest because it has significant effects on the efficiency as well as the accuracy.

For any anomalies, refer to the MSC Nastran error list and general limitations list.

Material Nonlinearity

Material nonlinearity becomes an issue in the analysis if the stress-strain relationship is nonlinear. Constitutive equations characterize the macroscopic behavior of the material, which stems from its microscopic constitution. All engineering materials are inherently nonlinear. It is, however, not feasible to characterize a nonlinear material by a single set of equations for the entire range of environmental conditions, such as loading, temperature, and rate of deformation. Depending on the application or the situation, the material behavior may be idealized or simplified to account for certain effects, which are important in the analysis.

The linear elastic (Hookean) material is the simplest case of all. As the strain exceeds the infinitesimal range, the material exhibits nonlinear behavior: nonlinear elastic if it is recoverable and plastic if it is irrecoverable. If the temperature effects on the material properties become an important issue, coupling between thermal and mechanical behavior should be properly taken into consideration, which is the realm of thermo-elasticity and thermo-plasticity. If the strain-rate has significant effects on the material response, we have to resort to the theories of visco-elasticity and visco-plasticity. Anisotropic effects add complexity to the material model.

Development of material constitutive models has been prolific in recent years. Detailed discussion in this section will be confined to the capabilities provided in MSC Nastran. The following nonlinear materials are available:

- Nonlinear elastic
- Hyperelastic
- Elastic-plastic
- Visco-elastic
- Temperature dependent

Nonlinear Elastic Material

Typical stress-strain relationship for a nonlinear elastic material is shown in Figure 3-1. Notice that both the loading and unloading occur along the same path.

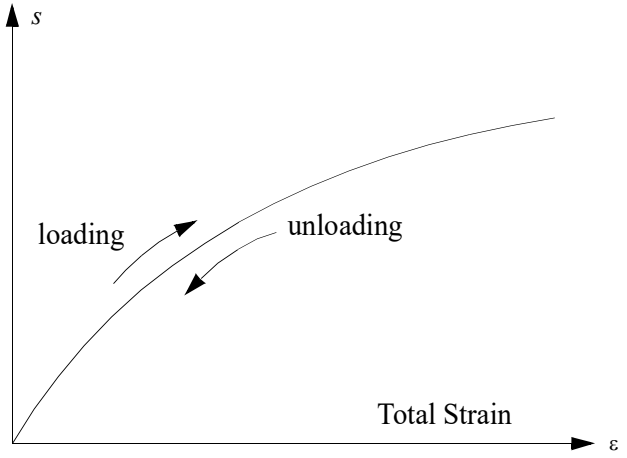


Figure 3-1 Nonlinear Elastic Material

The nonlinear elastic theory implemented is useful to predict multiaxial stress state for materials for which only the tension and compression stress-strain data is available. The stress-strain data in tension may be different from that in compression. The theory implemented in MSC Nastran is adequate only for small strains (less than 20%) in multiaxial stress state as it

is not based on the classical theory of finite elasticity. For large strains in multiaxial state, use of hyperelastic material is strongly recommended.

Hyperelastic Material

The hyperelastic model is used to analyze elastomers up to a large strain. Elastomers are natural or synthetic rubberlike materials that have the ability to undergo deformation under the influence of a force and regain their original shape once the force is removed (unless damage occurs). They are used extensively in many industries, because of their wide availability and low cost. Some examples of "real world" analyses in the automotive industry include o-rings, bushings, gaskets, seals, rubber boots, and tires.

The hyperelastic constitutive model available in MSC Nastran is based on a generalized Rivlin strain energy function expressed as follows:

$$U = \sum_{i+j=1}^{N_a} A_{ij}(\bar{I}_1 - 3)(\bar{I}_2 - 3)^j + \sum_{i=1}^{N_d} D_i(J - 1 - \alpha_v(T - T_0))^{2i}$$

where A_{ij} are the material constants related to shear deformation (distortion) and D_i are the material constants related to volumetric deformation. The strain energy function is a potential for the stresses. For example, the Cauchy stresses; i.e., components of the stress vector $\boldsymbol{t} = d\boldsymbol{P}/dA$ where dA is an infinitesimal area in the deformed configuration and $\boldsymbol{P}$ is the force, are obtained as follows:

$$\sigma = \frac{2}{J}\boldsymbol{F}\frac{\partial U}{\partial \boldsymbol{C}}\boldsymbol{F}^T$$

where $\boldsymbol{F} = d\boldsymbol{x}/d\boldsymbol{X}$ is the deformation gradient; $\boldsymbol{x}$ and $\boldsymbol{X}$ are the positions of a material point in the deformed and the undeformed configurations, respectively; $\boldsymbol{C} = \boldsymbol{F}^T\boldsymbol{F}$ is the right Cauchy-Green deformation tensor; $J = det\ \boldsymbol{F}$ is the volume ratio dV/dV_0; and $\bar{I}_1$ and $\bar{I}_2$ are the first and second distortional strain invariants,

$$\bar{I}_1 = \bar{\lambda}_1^2 + \bar{\lambda}_2^2 + \bar{\lambda}_3^2 \qquad \bar{I}_2 = \bar{\lambda}_1^{-2} + \bar{\lambda}_2^{-2} + \bar{\lambda}_3^{-2}$$

where $\bar{\lambda}_I$ are principal stretches l/l_0 of the distortional deformation; l is the current and l_0 is the original length along a principal direction; $\bar{\lambda}_1^2$ are principal values of $\bar{\boldsymbol{C}} = J^{-2/3}\boldsymbol{C}$; α_v is the volumetric coefficient of thermal expansion; $\alpha_v = 3\alpha$, where α is the coefficient of linear thermal expansion of an isotropic material; T is the current and T_0 is the initial temperature.

The use of a strain energy function implies isotropy in the undeformed configuration. The model can have up to 25 material constants.

Special cases of the material law are:

Mooney-Rivlin Material, $N_a = 1$

Neo-Hookean Material, $N_a = 1$, $A_{01} = 0.0$

A curve-fitting algorithm based on least squares fitting is available to determine the material constants A_{ij} and D_i based on experimental data up to the fifth-order polynomial form (N_a and/or N_d up to 5) for the following deformations of the material:

- Simple tension/compression

  The simple tension/compression test is characterized by a prescribed axial force or displacement. X values in the TABLES1 entry which contains experimental data must be stretch ratios l/l_0. Y values must be values of the engineering stress F/A_0.

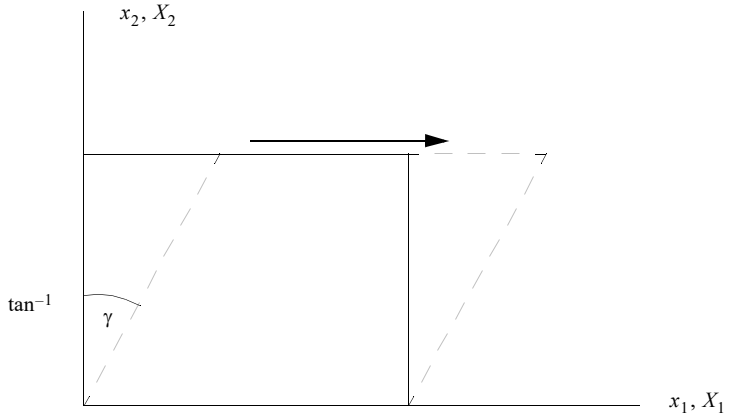
- Equibiaxial tension

  The equibiaxial tension test is characterized by prescribed equal tensile stresses or elongations in two directions and is typically produced by the pressure of a spherical membrane. X values in the TABLES1 entry which contains experimental data must be stretch ratios l/l_0. Y values must be values of the engineering stress F/A_0, l = current length, F = current value of the force, l_0 and A_0 = initial length and cross-sectional area, respectively. In the case of pressure of a spherical membrane, the engineering stress is given by $Pr_0\lambda^2/2t_0$ where P = current value of the pressure and r_0, t_0 = initial radius and thickness.

- Simple shear

  The simple shear test is obtained by a prescribed shearing force or displacement of a cube. This is not a homogeneous deformation. It is approximated by one that is homogeneous isochoric as follows:

$x_1 = X_1 + \gamma X_2 \qquad x_2 = X_2 \qquad x_3 = X_3$



where x_1, x_2, x_3 are the coordinates of the deformed body and X_1, X_2, X_3 are the coordinates of the body in its undeformed configuration. The user should be aware of this approximation. X values in the TABLES1 entry which contains experimental data must be values of the shear tangent γ. Y values must be values of the engineering shear stress F/A_0. It is recommended that simple shear be used along with another test case; otherwise, the coefficients cannot be determined uniquely.

- Pure shear

  The pure shear test is a more direct way of imposing a state of shear deformation and is obtained by a prescribed uniform stretch λ_1 of a specimen, which is long in the 2-direction leaving the 3-direction stress free, or by holding the 2-direction fixed while applying strains in the 1-direction of a thin rectangular sheet, as in the classical experiments of Rivlin and Saunders. If the material is incompressible, this results in $\lambda_3 = 1/\lambda_1$, which is characteristic of a state of shear deformation. It is recommended that pure shear be used along with another test case; otherwise, the coefficients cannot be determined uniquely. X and Y values, respectively, in the TABLES1 entry which contains experimental data must be stretch ratios $\lambda_1 = l/l_0$ and values of the nominal stress F/A_0 l = current length, F = current value of the force, l_0 and A_0 = initial length and cross-sectional area, respectively in the 1-direction.

- Volumetric compression

  Pure volumetric compression data may be used to obtain fitted values of the material constants related to volumetric deformation. X values in the TABLES1 entry which contains experimental data must be values of the volume ratio $J = \lambda^3$, where $\lambda = l/l_0$ is the stretch ratio in all three directions; Y values must be values of the pressure, assumed positive in compression.

The first four test cases (experiments) are used to obtain the material constants A_{ij}. The last test case (volumetric compression) is used to obtain the material constants D_i. In obtaining the material constants, compressibility is taken into account, in the case of simple tension/compression and equibiaxial tension, when the volumetric part of the strain energy function is of the order of one.

User Interface

The MATHP material entry defines the hyperelastic material. The MATHP entry consists of five continuation entries that define the A_{ij} and D_i material constants of the strain energy function. The continuation entries are not required if a Neo-Hookean or a Mooney-Rivlin material is modeled.

If the material constants are not known, the curve-fitting algorithm is invoked if the sixth continuation entry of the MATHP entry contains nonblank fields, TAB1, TAB2, TAB3, TAB4, and TABD. The TABi fields reference experimental data that is contained in TABLES1 entries. The order of the strain energy polynomial function is specified using the NA and ND fields of the MATHP entry for the distortional and volumetric parts, respectively. The curve-fitting algorithm computes the material constants from the supplied data, and the analysis continues using the produced constants.

Output Description

The curve-fitting algorithm computes the material constants and prints the results in an updated MATHP entry image in the output file. The image is printed using a 16-character field and therefore is convenient to directly input the produced MATHP entry into a data file in large-field format. An example of the produced output is shown in Listing 3-3.

Guidelines

If the strains are small, the results of the hyperelastic elements should compare with those of the linear elastic elements where the shear modulus and the bulk modulus $G \approx 2(A_{10} + A_{01})$ and the bulk modulus $K \approx 2D_1$.

Conventional Mooney-Rivlin and Neo-Hookean materials are fully incompressible. Full incompressibility is not presently available but may be simulated with a large enough value of D_1. However, a value of D_1 higher than $10^3 \cdot (A_{10} + A_{01})$ is not recommended.

When determining the material constants from experimental data, it is best to use data from more than one test involving different kinds of deformation. A rank deficiency warning may occur in the least squares fitting algorithm if there is insufficient experimental data. It is best to try to avoid rank deficiency by providing more experimental data or by lowering the order of the strain energy polynomial.

Limitations

It is recommended that Poisson's ratio for the hyperelastic material be no greater than .4995, or, equivalently, that $D_1/(A_{10} + A_{01})$ does not exceed 1000. As incompressibility increases, convergence difficulties and/or ill conditioning of the stiffness matrix occurs.

The hyperelastic material cannot be defined for a small strain element; that is, it cannot be referenced by a PSHELL or PSOLID Bulk Data entry.

Temperature dependence of the hyperelastic material constants is not supported.

Example: Curve-Fitting Algorithm

An example created to solve a fully nonlinear (finite deformation) CQUAD4 element under pure shear with material constants obtained from the simultaneous fitting of experimental data in simple tension, equibiaxial tension, and pure shear. Annotated Bulk Data input is shown in the following listing that illustrates the MATHP and TABLES1 entries, which contain the experimental data for curve fitting.

Listing 3-3 Example Using Material Curve Fitting

```
BEGIN BULK
$ PARAMETERS
PARAMDBDROPT0
PARAMLGDISP1
$
NLPARM1040AUTO1
$
$ DEFINE GEOMETRY
GRID10.0.0.123456
GRID21.0.0.23456
GRID31.0.1.2456
GRID40.0.1.12456
$ DEFINE ELEMENTS
CQUAD4111234
$ CONSTRAINTS AND LOADING
MPC10211.31-1.
MPC10431.33-1.
SPC1031
SPCD100315.
$ DEFINE PROPERTIES
$ LARGE STRAIN QUAD4 ELEMENT
PLPLANE11ISOP2
$ HYPERELASTIC MATERIAL
$ D1 SPECIFIED
$ AIJ MATERIAL CONSTANTS TO BE DETERMINED FROM EXPERIMENTAL DATA
MATHP11500.+MA101
+MA10131+MA102
+MA102+MA103
+MA103+MA104
+MA104+MA105
+MA105+MA106
$TAB1TAB2TAB3TAB4TABD
+MA106100200400
$
$ TAB1 DATA
$

$ Treloar [1944] data in simple tension.  Nominal stresses in kgcm-2
$
TABLES1 100                                                             +TA101
+TA101  1.     0.     1.125   1.     1.25    2.     1.5     3.      +TA102
+TA102  1.525  4.     1.875   5.     2.      6.     2.25    7.      +TA103
+TA103  2.5    8.     3.      9.     3.625   10.    4.      12.5    +TA104
+TA104  4.75   16.    5.25    20.    5.75    23.    6.      27.     +TA105
+TA105  6.25   31.    6.5     34.    6.75    38.75  7.      42.5    +TA106
+TA106  ENDT
$
$ TAB2 DATA
$
$ Treloar [1944] data in equibiaxial tension.  Nominal stresses in kg-cm-2
$
TABLES1 200                                                             +TA201
+TA201  1.016  0.83   1.07    1.56    1.15    2.55    1.21    3.25    +TA202
+TA202  1.335  4.31   1.44    5.28    1.66    6.65    1.93    7.88    +TA203
+TA203  2.46   9.74   3.      12.64   3.4     14.61   3.77    17.33   +TA204
+TA204  4.1    20.11  4.32    22.40   4.54    24.41   ENDT
$
$ TAB4 DATA
$
$ Treloar* [1944] data in pure shear.  Units of the nominal stress are kgcm-2


$
TABLES1 400                                                             +TA401
+TA401  1.01   0.24   1.1     1.12    1.2     1.92    1.31    2.87    +TA402
+TA402  1.49   3.62   1.86    5.61    2.36    7.2     2.97    9.045   +TA403
+TA403  3.45   10.8   3.93    12.45   4.4     14.23   4.714   15.826  +TA404
+TA404  4.96   17.46  ENDT
ENDDATA
```

*See Experimental Data Fitting in the *MSC Nastran Nonlinear User's Guide*.

The following listing illustrates the produced MATHP entry image that is printed in the output just after the ENDDATA entry image.

Listing 3-4 Effective Stress-Strain Law

```
*** SYSTEM INFORMATION MESSAGE 6410 (IFP8).  BEGIN PROCESSING OF MATHP ENTRY ID =    1
    DISTORTIONAL PARAMETER FITTING.
    SQUARE ROOT OF SUM OF THE ERRORS SQUARED =    0.25629E+01
    UPDATED MATHP ENTRY
            1       1.54082        0.24046     1500.00000        0.00000        0.00000        0.00000        0.00000
                          3              1
     -0.00375      -0.01743        0.00104        0.00000
      0.00029       0.00033       -0.00004        0.00000        0.00000
      0.00000       0.00000        0.00000        0.00000        0.00000        0.00000
      0.00000       0.00000        0.00000        0.00000        0.00000        0.00000        0.00000
```

Figure 3-2(a) through Figure 3-2(c) illustrate the curves obtained from this and similar files containing a large strain CHEXA in simple and equibiaxial tension with the material constants generated for the first-, second-, and third-order strain energy polynomial functions from simultaneous fitting of experimental data in simple tension, equibiaxial tension, and pure shear.

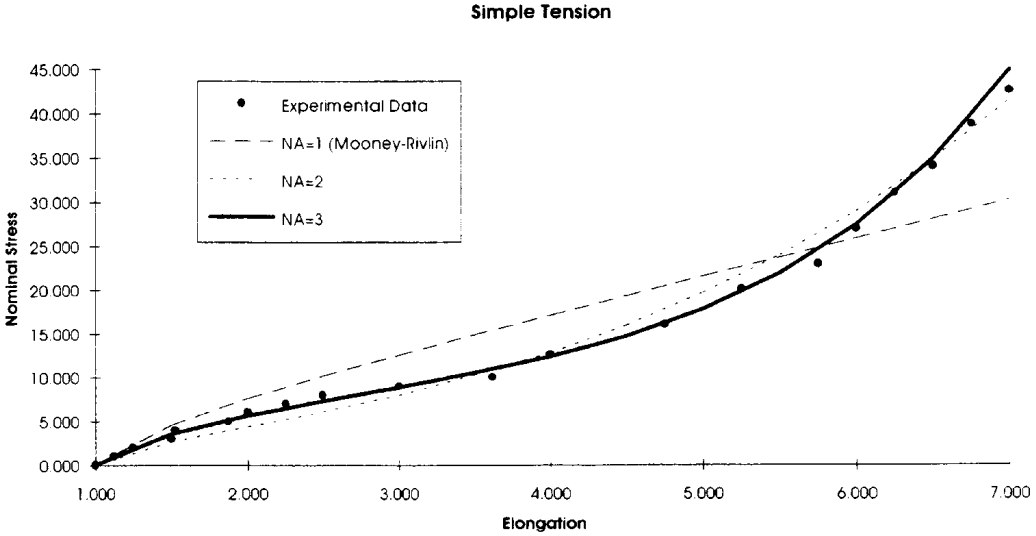


Figure 3-2 MSC Nastran Curve-Fitting Algorithm for Hyperelastic Materials (a)

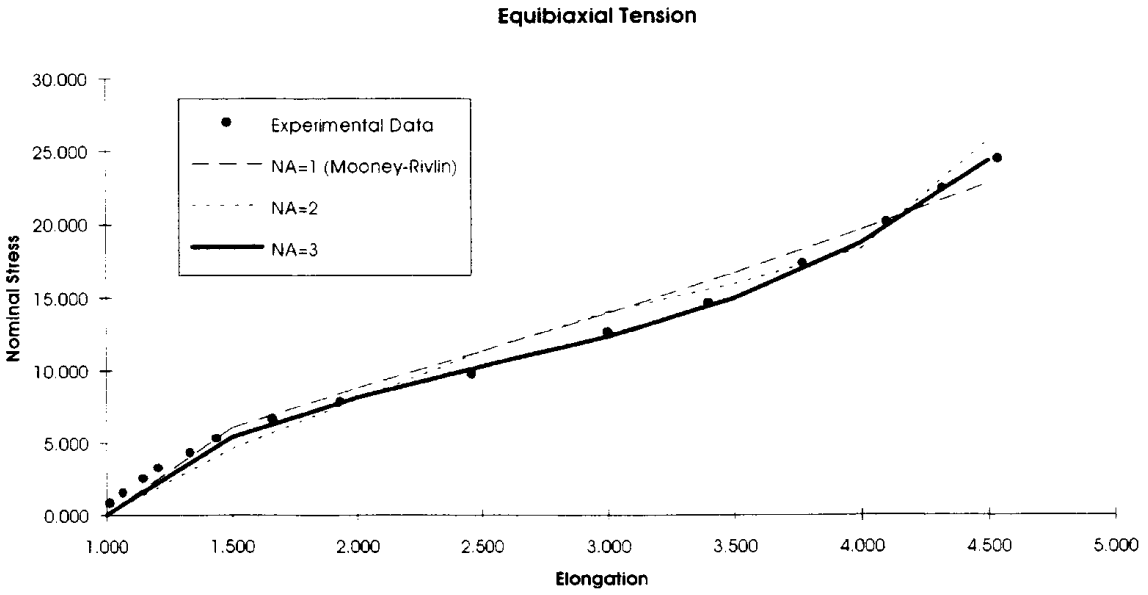


Figure 3-3 MSC Nastran Curve-Fitting Algorithm for Hyperelastic Materials (b)

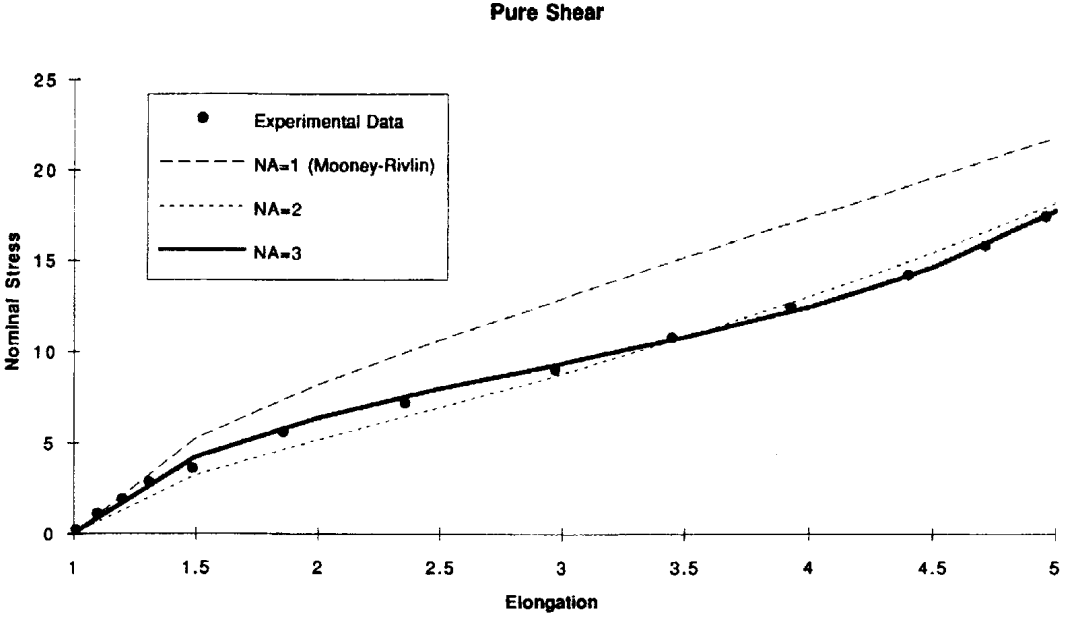


Figure 3-4 MSC Nastran Curve-Fitting Algorithm for Hyperelastic Materials (c)

Elastic-Plastic Material

Before discussing the computational aspects associated with the various models, the definitions of some of the commonly used terms in plasticity are given below.

Yield Stress	Yield Stress is usually measured as the value of stress which produces the smallest measurable permanent strain.
Bauschinger Effect	When a plastically deformed specimen is unloaded, residual stresses on a microscopic scale remain and influence the plastic yielding for the different loading. If the previous strain was a uniform extension and the specimen is then reloaded in compression in the opposite direction, it is observed that yielding occurs at a much reduced stress. This is known as the Bauschinger Effect.
	The phenomenon of Bauschinger effect can best be described with reference to Figure 3-5 which shows an idealized stress-strain curve of a metal first deformed by uniform tension, the load removed and then reloaded in compression. According to one model (which is one extreme viewpoint), it is assumed that the elastic unloading range will be double the initial yield stress. If the initial yield stress in tension is σ_y and the specimen is loaded up to a stress σ_1 and unloaded, the plastic yielding will begin in compression at a stress σ_2 given by

$$\sigma_2 = \sigma_1 - 2\sigma_y$$

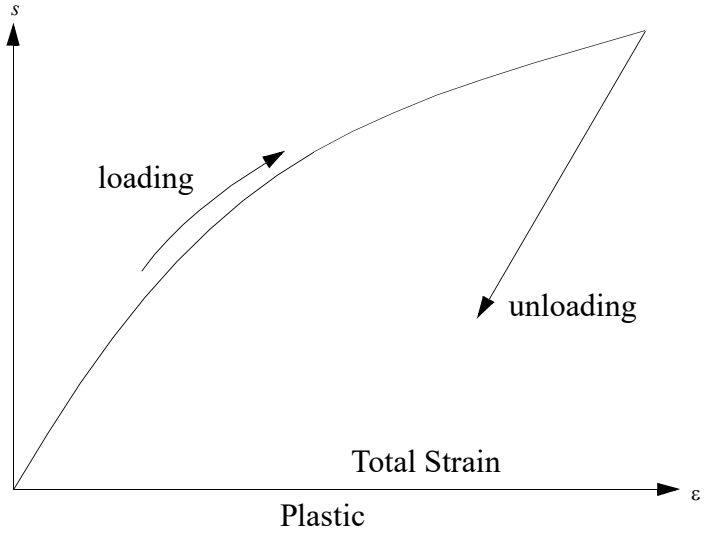


Figure 3-5 Material Nonlinear Models

Yield Function

Associated with the yield criteria, there exist yield functions in the form of

$$F(\sigma,\bar{\varepsilon}^p) = f(\sigma) - Y(\bar{\varepsilon}^p) = 0$$

where:

$f(\sigma)$ = an effective stress, a function of the stress state

$Y(\bar{\varepsilon}^p)$ = a tensile yield stress, a function of the stress hardening.

Geometric representation of the yield function is a surface in the stress space ($\sigma_1 - \sigma_2 - \sigma_3$ coordinates) which is called a yield surface. The von Mises yield surface is a circular cylinder and Tresca's is a hexagonal cylinder. The yield surface is reduced to the yield locus in the case of plane stress, as shown in Figure 3-6 and Figure 3-7 for the von Mises and Tresca criteria, respectively. The Mohr-Coulomb and Drucker-Prager yield functions represent conical surfaces as shown in Figure 3-8 and Figure 3-9.

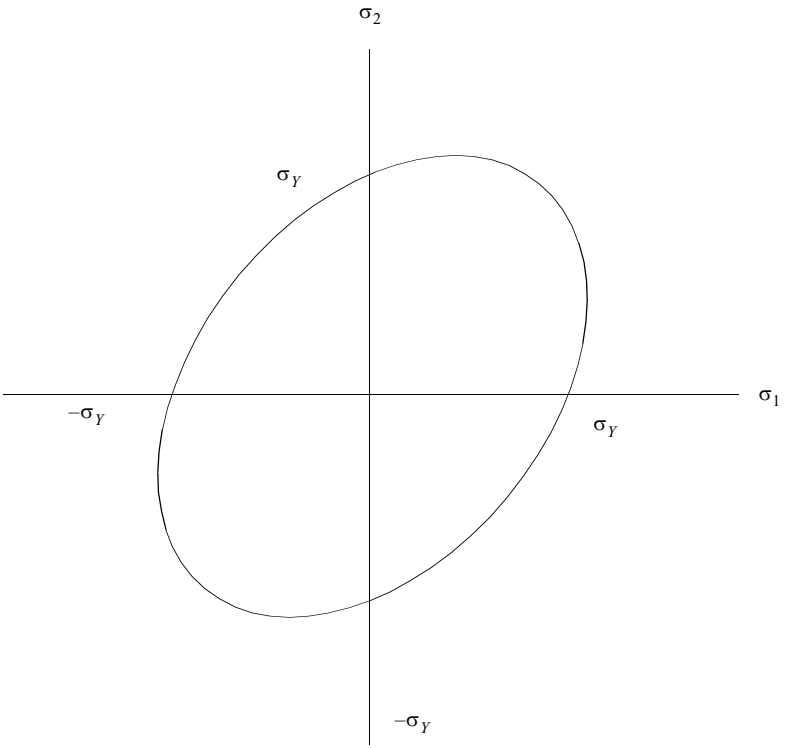


Figure 3-6 von Mises Yield Locus for Plane Stress

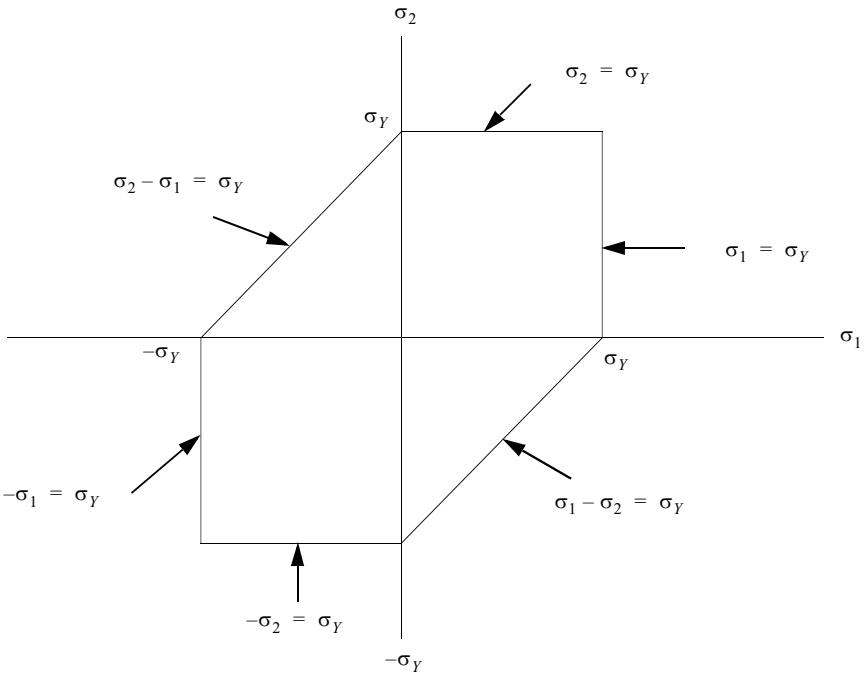


Figure 3-7 Tresca's Yield Locus for Plane Stress

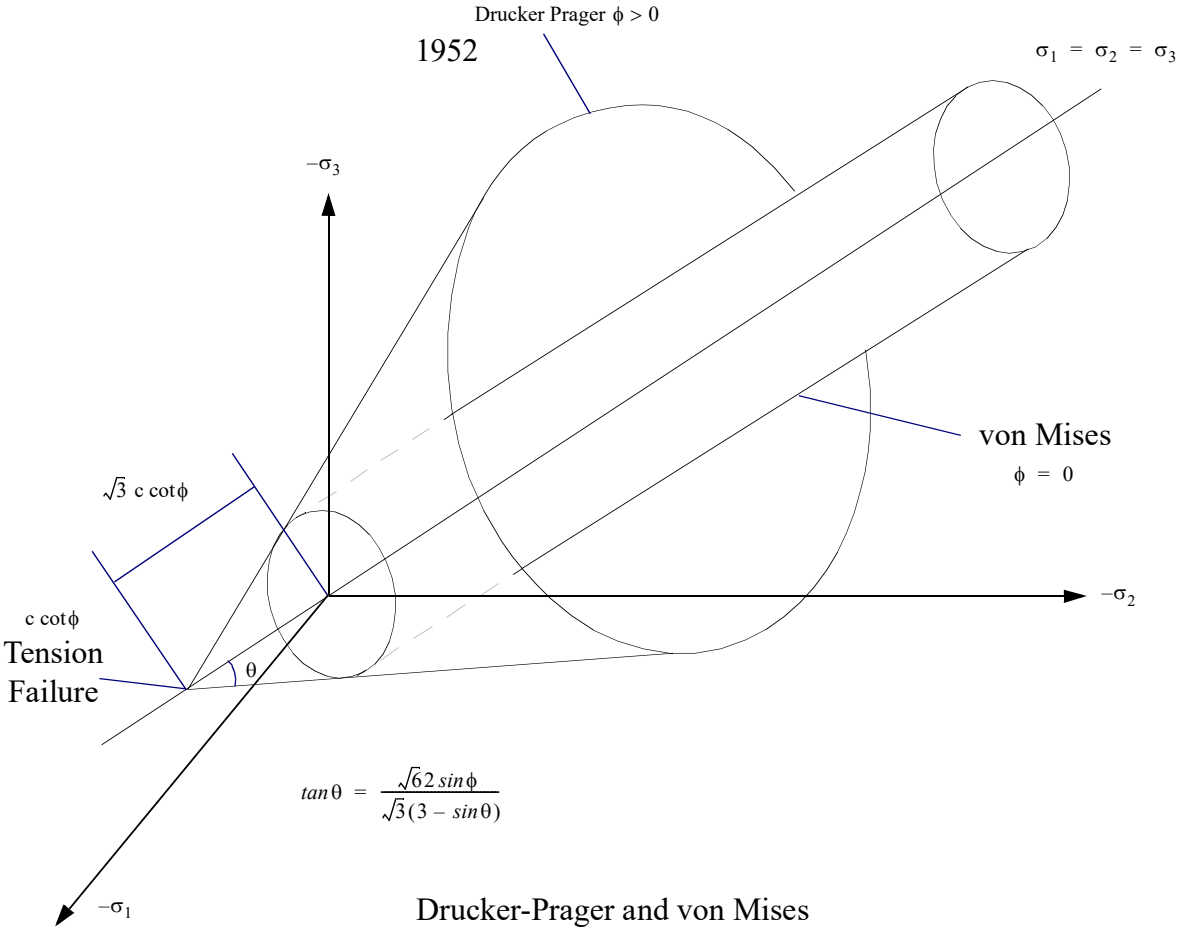
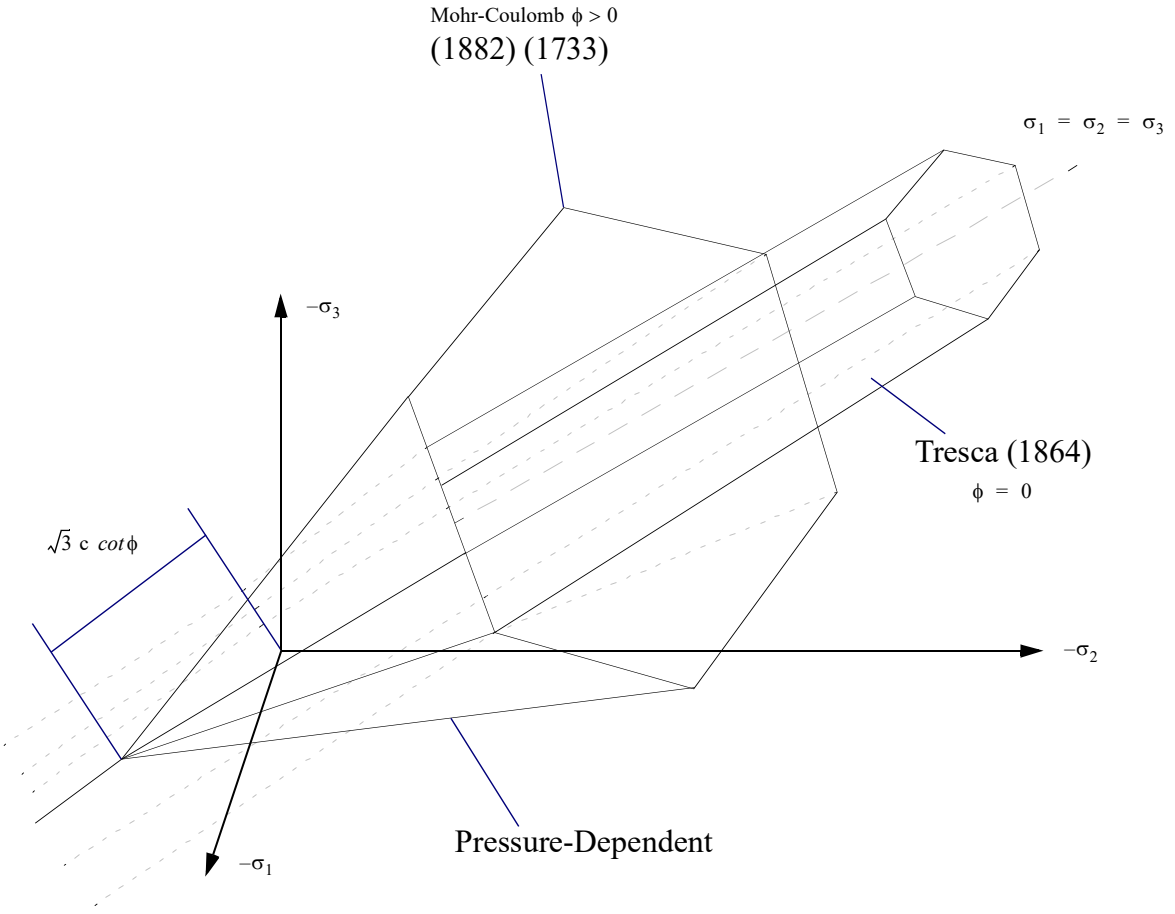


Figure 3-8 Some Isotropic Yield Surfaces in Principal Stress Space



Mohr-Coulomb and Tresca

Figure 3-9 Some Isotropic Yield Surfaces in Principal Stress Space

Viscoelastic Material in Nonlinear Analysis

The quasi-static behavior of the viscoelastic material may be analyzed using the creep capability of MSC Nastran. Viscoelastic properties (linear or nonlinear) can be defined in terms of rheological model parameters (K_p, C_p, and C_s) of the Maxwell-Kelvin model in the Bulk Data entries CREEP and TABLES1. Then the quasi-static analysis of the viscoelastic materials can be performed by specifying time increment in the Bulk Data entry NLPARM (selected by the subcase) in SOL 106.

Temperature-Dependent Material

Currently only linear and nonlinear elastic materials can be temperature dependent. Temperature-dependent properties for elastic and nonlinear elastic materials are specified by using the MATTi, TABLEST, and TABLES1 Bulk Data entries. Elastic-plastic and hyperelastic materials cannot be temperature dependent. A limited temperature dependence for creep materials is available based on the Arrhenius equation.

Geometric Nonlinearity

Geometric nonlinearities are manifested in problems involving large rotations and large deformation. The characteristics are follower forces due to large rotations, geometric stiffening due to initial stress effect (as a result of large rotations), and large strains due to large deformation.

Geometric nonlinear effects should be significant if the deformed shape of the structure appears distinctive from the original geometry by a visual inspection. A more rigorous and quantitative definition for the large displacements can be derived from the plate theory of Kirchhoff and Love: the small deflection theory is valid for a maximum deflection of less than 20% of the plate thickness or 2% of the small span length. However, this definition seems to be a little conservative for numerical analysis, and there is no distinct limit for large displacements because geometric nonlinear effects are related to the boundary conditions as well as the dimensions of the structure. If the load-deflection curve of the critical point can be estimated, the loading point should be in the nonlinear portion of the curve.

Geometric nonlinear effects in the structure involving large rotations, whether rigid body rotations or deformation induced rotations, are self-evident. Stiffening of a membrane, stiffness in a pendulum or snap-through of an arch belong to this category. The motion of a pendulum under gravity is caused by geometric (differential) stiffness. Follower forces are manifested when the applied loads are displacement dependent, such as pressure load and thermal load applied on the surface that rotates. Centrifugal force is another example of follower forces. Large strain effects are pronounced in metal forming, rubber and elastomer applications. In such applications the strains exceed 100%. Finite strain formulation is required to treat the problems in this category. MSC Nastran currently supports the large strain capability for rubbers and elastomers. This formulation contains the effect of large strain as well as geometric nonlinearity. However, it does not support large strain formulation for metal forming applications. In most structural applications, however, moderately large strains (20 to 30%) appear in local areas if there is any large deformation

Other geometric nonlinear effects are treated by updated element coordinates, gimbal angles (or rotation vector), and the differential stiffness $[K^d]$.

Nonlinear Elements

The nonlinear elements can be divided into two categories: physical and contact elements. Physical elements have material properties associated with them. Contact elements do not have any material properties associated with them.

Physical Elements

The nonlinear physical elements are listed in Table 3-5 and Table 3-6.

The CBUSH element may have nonlinear force-deflection properties defined on the PBUSH and PBUSHT property entries. The CBUSH element is geometric linear; the orientation is not updated in case of large deformation. The CBUSH1D element is both geometric and material nonlinear.

The elements CROD, CONROD, and CTUBE may have material nonlinear extensional properties, with linear torsion. The user may supply plastic or nonlinear elastic material properties. Since the stress-strain curve for compression need not be the same as for tension, this element can, for example, be used to model cables which cannot carry tension.

The CBEAM element has been modified to provide plastic hinges at the ends of an otherwise elastic element.This element is intended for collapse analysis in frameworks with loads at the joints, and materials with small work hardening. The user need not specify the cross-section axis about which the yielding occurs, since the implementation allows for combinations of bending moments in two directions plus an axial load. The flexibility of the plastic hinge is based upon eight idealized rods at each end, chosen to match the total area, center of gravity and moments of inertia of the cross section. As is shown in Table 3-7, the calculated ultimate moment agrees reasonably well with the theoretical value for many cross-sections. The material specified on a MATS1 Bulk Data entry should be elastic-perfectly plastic. Use of any nonlinear material other than elastic-perfectly plastic will give inaccurate results.

The CQUAD4 and CTRIA3 plate elements are available for combined bending and membrane strains found in shell structures as well as for plane strain applications. For shell type problems the state of stress is assumed to be constant in each of a variable number of equal layers. One layer should be used for plane strain. The user input is the one-dimensional stress-strain curve for tension. Higher order CQUAD8, CQUAD, and CTRIA6 elements exist for fully nonlinear (i.e., large strain and large rotation) hyperelastic plane strain analysis.

The CHEXA, CPENTA, and CTETRA elements are used to model three dimensional solid structures. No midside nodes are allowed for nonlinear analysis, except for the hyperelastic 20-noded CHEXA, 15-noded CPENTA, and 10-noded CTETRA elements.

Axisymmetric elements CQUADX and CTRIAX are available for fully nonlinear analysis with the hyperelastic material.

Table 3-5 Physical Elements for Nonlinear Analysis -- Small Strain Elements

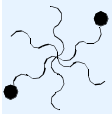
Name	Characteristics	Nonlinear Elastic	Elasto-Plastic	Geometric Only Nonlinear
CBUSH 	Generalized 3D spring damper with 6 stiffnesses for 3 translations and 3 rotations.	Yes	No	No

Table 3-5 Physical Elements for Nonlinear Analysis -- Small Strain Elements (continued)

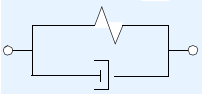
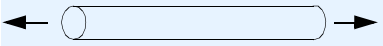
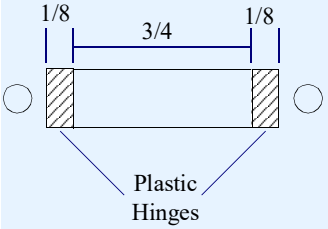
Name	Characteristics	Nonlinear Elastic	Elasto-Plastic	Geometric Only Nonlinear
CBUSH1D 	Rod-like spring and damper with generalized force function $f(u,\dot{u})$ for axial displacement u and velocity $\dot{u}$.	Yes	No	Yes
CROD, CONROD, CTUBE 	Extension may be plastic or nonlinear elastic. Torsion is elastic (linear).	Yes	Yes	Yes
CBEAM 	Plastic hinge at each end, which couples axial motion and rotations. Linear material for center section, traverse shear, and torsion.	No	Yes (only elastic-perfectly plastic)	Yes

Table 3-5 Physical Elements for Nonlinear Analysis -- Small Strain Elements (continued)

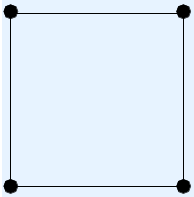
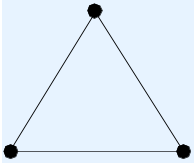
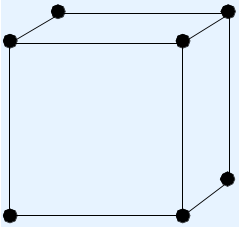
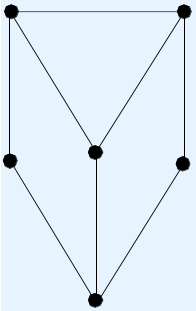
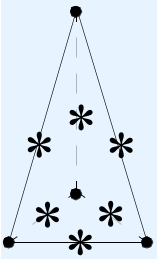
Name	Characteristics	Nonlinear Elastic	Elasto-Plastic	Geometric Only Nonlinear
CQUAD4 	Shell and Plate Two-dimensional plasticity with FIVE layers for membrane and bending. 12 layers maximum; 1 layer minimum. Transverse shear is linear.	Yes	Yes	Yes
	Plane Strain PSHELL with MID2=-1 and PARAM,NLAYER=1	Yes	Yes	Yes
	Plane Stress MID2=0 NLAYER=1	Yes	Yes	Yes
CTRIA3 	Shell and Plate Two-dimensional plasticity with FIVE layers for membrane and bending. 12 layers maximum; 1 layer minimum. Transverse shear is linear.	Yes	Yes	Yes
	Plane Strain PSHELL with MID2=-1 and PARAM,NLAYER=1	Yes	Yes	Yes
	Plane Stress PSHELL with MID2=0 NLAYER=1	Yes	Yes	Yes

Table 3-5 Physical Elements for Nonlinear Analysis -- Small Strain Elements (continued)

Name	Characteristics	Nonlinear Elastic	Elasto-Plastic	Geometric Only Nonlinear
CHEXA 	Eight Gauss points for nonlinear analysis. Reduced shear integration. Strain (bubble) function. No mid side nodes allowed.	Yes	Yes	Yes
CPENTA 	Six Gauss points for nonlinear analysis. Reduced shear integration. Strain (bubble) function.	Yes No	Yes No	Yes No
CTETRA 	For four-noded tetrahedron a single Gauss point for nonlinear analysis. For 10-noded tetrahedron four Gauss points for nonlinear analysis.	Yes	Yes	Yes

- One-dimensional stress-strain curves use the MAT1 entry.
- All other elements may be used for nonlinear analysis as long as they remain linear.

Table 3-6 Physical Elements for Nonlinear Analysis -- Fully Nonlinear (Finite Deformation) Hyperelastic Elements

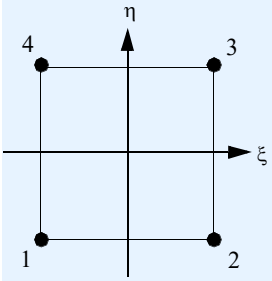
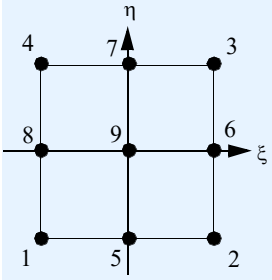
Name	Characteristics	Gauss Point	ξ	η	ζ
CQUAD, CQUAD4, CQUAD8, CQUADX 	Plane strain or axisymmetric analysis. Linear isoparametric quadrilateral with constant pressure. Compressible and nearly incompressible applications. Four Gauss points.	1 2 3 4 $\alpha = 1/\sqrt{3}$	$-\alpha$ α $-\alpha$ α	$-\alpha$ $-\alpha$ α α	--
CQUAD, CQAUD8, CQUADX 	Plane strain or axisymmetric analysis. Quadratic isoparametric quadrilateral with linear pressure. Compressible and nearly incompressible applications. Center node not available on the CQUAD8 connectivity. Nine Gauss points.	1 2 3 4 5 6 7 8 9 $\alpha = \sqrt{15}/5$	$-\alpha$ 0 α $-\alpha$ 0 α $-\alpha$ 0 α	$-\alpha$ $-\alpha$ $-\alpha$ 0 0 0 0 α α	--

Table 3-6 Physical Elements for Nonlinear Analysis -- Fully Nonlinear (Finite Deformation) Hyperelastic Elements (continued)

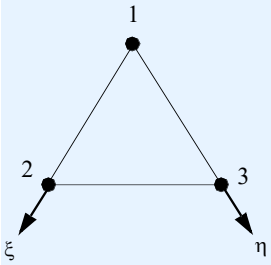
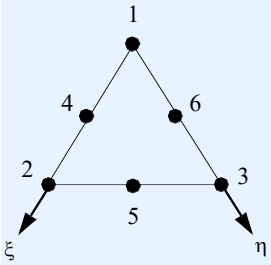
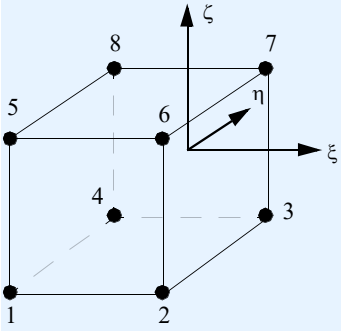
Name	Characteristics	Gauss Point	ξ	η	ζ
CTRIA3, CTRIA6, CTRIAX 	Plane strain or axisymmetric analysis. Linear isoparametric triangle with constant pressure. Compressible and nearly incompressible applications. One Gauss point.	1	1/3	1/3	--
CTRIA6, CTRIAX 	Plane strain or axisymmetric analysis. Quadratic isoparametric triangle with constant pressure. Compressible and nearly incompressible applications. Three Gauss points.	1 2 3	1/6 2/3 1/6	1/6 1/6 2/3	--
CHEXA 	Linear isoparametric hexahedron with constant pressure. Compressible and nearly incompressible applications. Eight Gauss points.	1 2 3 4 5 6 7 8 $\alpha = 1/\sqrt{3}$	α $-\alpha$ α $-\alpha$ α $-\alpha$ $+\alpha$ $-\alpha$	α α $-\alpha$ $-\alpha$ α α $-\alpha$ $-\alpha$	α α α α $-\alpha$ $-\alpha$ $-\alpha$ $-\alpha$

Table 3-6 Physical Elements for Nonlinear Analysis -- Fully Nonlinear (Finite Deformation) Hyperelastic Elements (continued)

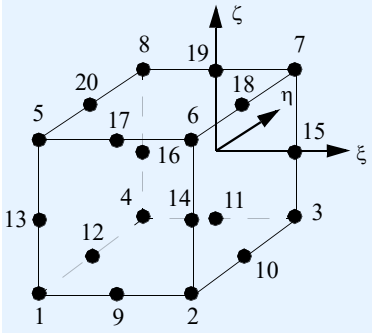
Name	Characteristics	Gauss Point	ξ	η	ζ
CHEXA 	Quadratic ispoarametric hexahedron with linear pressure. Compressible and nearly incompressible applications. Twenty-seven Gauss points.	1	α	α	α
		2	0	α	α
		3	−α	α	α
		4	α	0	α
		5	0	0	α
		6	−α	0	α
		7	α	−α	α
		8	0	−α	α
		9	−α	−α	α
		10	α	α	0
		11	0	α	0
		12	−α	α	0
		13	α	0	0
		14	0	0	0
		15	−α	0	0
		16	α	−α	0
		17	0	−α	0
		18	−α	−α	0
		19	α	α	−α
		20	0	α	−α
		21	−α	α	−α
		22	α	0	−α
		23	0	0	−α
		24	−α	0	−α
		25	α	−α	−α
		26	0	−α	−α
		27	−α	−α	−α
		$\alpha = (\sqrt{15})/5$			

Table 3-6 Physical Elements for Nonlinear Analysis -- Fully Nonlinear (Finite Deformation) Hyperelastic Elements (continued)

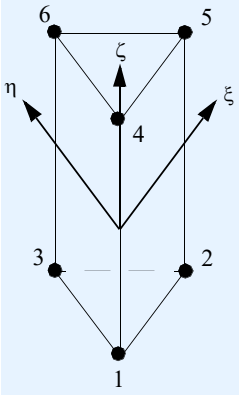
Name	Characteristics	Gauss Point	ξ	η	ζ
CPENTA 	Linear isoparametric pentahedron with constant pressure. Compressible and nearly incompressible applications. Six Gauss points.	1	1/2	0	$-1/\sqrt{3}$
		2	1/2	1/2	$-1/\sqrt{3}$
		3	0	1/2	$-1/\sqrt{3}$
		4	1/2	0	$1/\sqrt{3}$
		5	1/2	1/2	$1/\sqrt{3}$
		6	0	1/2	$1/\sqrt{3}$

Table 3-6 Physical Elements for Nonlinear Analysis -- Fully Nonlinear (Finite Deformation) Hyperelastic Elements (continued)

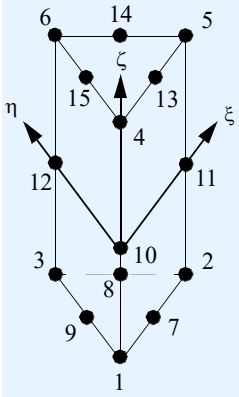
Name	Characteristics	Gauss Point	ξ	η	ζ
CPENTA	Quadratic isoparametric pentahedron with linear pressure. Compressible and nearly incompressible applications.	1	1/3	1/3	$-\sqrt{15}/5$
	Twenty-one Gauss points.	2	α_1	β_1	$-\sqrt{15}/5$
		3	β_1	α_1	$-\sqrt{15}/5$
		4	β_1	β_1	$-\sqrt{15}/5$
		5	α_2	β_2	$-\sqrt{15}/5$
		6	β_2	α_2	$-\sqrt{15}/5$
	$\alpha_1 = 0.0597158717$	7	β_2	β_2	$-\sqrt{15}/5$
	$\beta_1 = 0.4701420641$	8	1/3	1/3	0
	$\alpha_2 = 0.7974269853$	9	α_1	β_1	0
	$\beta_2 = 0.1012865073$	10	β_1	α_1	0
		11	β_1	β_1	0
		12	α_2	β_2	0
		13	β_2	α_2	0
		14	β_2	β_2	0
		15	1/3	1/3	$+\sqrt{15}/5$
		16	α_1	β_1	$+\sqrt{15}/5$
		17	β_1	α_1	$+\sqrt{15}/5$
		18	β_1	β_1	$+\sqrt{15}/5$
		19	α_2	β_2	$+\sqrt{15}/5$
		20	β_2	α_2	$+\sqrt{15}/5$
		21	β_2	β_2	$+\sqrt{15}/5$

Table 3-6 Physical Elements for Nonlinear Analysis -- Fully Nonlinear (Finite Deformation) Hyperelastic Elements (continued)

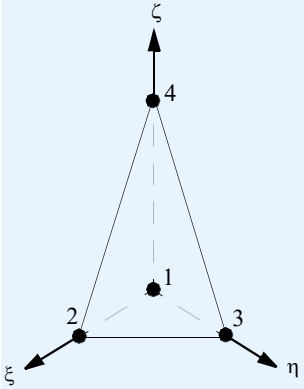
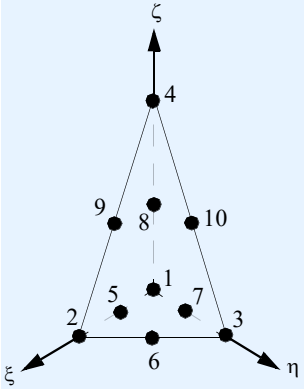
Name	Characteristics	Gauss Point	ξ	η	ζ
CTETRA 	Linear isoparametric tetrahedron with constant pressure. Compressible and nearly incompressible applications. One Gauss point.	1	1/4	1/4	1/4
CTETRA 	Quadratic isoparametric tetrahedron with linear pressure. Compressible and nearly incompressible applications. Five Gauss points.	1 2 3 4 5	1/6 1/2 1/6 1/6 1/4	1/6 1/6 1/2 1/6 1/4	1/6 1/6 1/6 1/2 1/4

Table 3-7 Accuracy in Calculation of Ultimate `Moment` in Yielded State

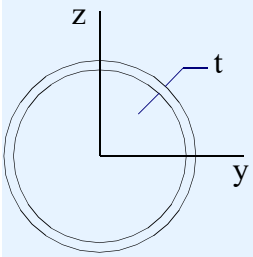
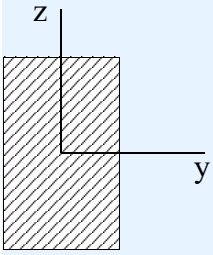
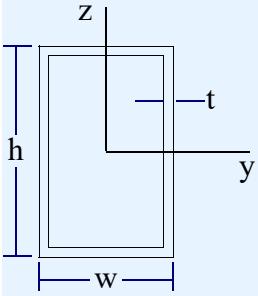
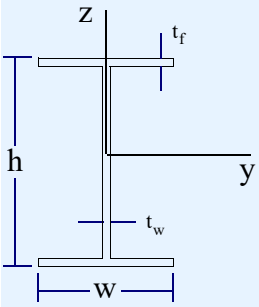
Section			
	`Moment` Axis	η	
	Any multiple of 45° from y-axis	.9481	
	`Moment` Axis	η	
	y or z	.9856	
 $t \ll h, w$	h/w	η for y-axis	η for z-axis
	0	.8536	.9856
	.5	.9031	.9543
	1.0	.9295	.9295
	2.0	.9543	.9031
	∞	.9856	.8536

Table 3-7 Accuracy in Calculation of Ultimate Moment in Yielded State (continued)

Section			
 t_f, t_w « h, w	ht_w / wt_f	η for y-axis	η for z-axis
	0	.8536	.9856
	.5	.8826	1.102
	1.0	.9031	1.207
	2.0	.9295	1.394
	∞	.9856	

Contact Elements

The contact (interface) elements are listed in Table 3-8.

The CGAP element is intended to model surfaces which may come into contact. When positive pressure exists, the gap can carry any transverse shear load which is less than the coefficient of friction times the normal load. The CGAP element connects two grid points which may be initially coincident. There is no geometric nonlinear behavior, which implies that the orientation of the contact plane does not change during deflection. (The physical shape of the two contact surface would have to be specified and it would require the solution of a difficult analytic problem to determine the location of the actual contact point or points.)

Due to the requirements of the solution algorithm, there must be a finite compression stiffness for the CGAP element. The value must be carefully chosen, since a very large value may lead to numerical problems. A finite extensional stiffness may be used to prevent the drift of structures that are supported only by CGAP elements. The user may also supply an initial opening of the gap. An orientation vector is required (see CGAP entry for details) to define the transverse axes, even if no shear is to be carried.

The slideline contact is capable of modeling nonlinear contact geometry and inelastic material behavior including large deformation. Slideline contact is useful for two-dimensional geometries, for example plane stress, plane strain and axisymmetric. They can also be used for three-dimensional geometries provided the contact between the two bodies can be defined in terms of parallel planes called the slideline planes. The sliding and separation of the two bodies is restricted to the slideline planes. The bodies can have large relative motions within the slideline plane. However, relative motions outside the slideline planes are ignored; therefore, they must be small compared to a typical slideline element.

Table 3-8 Contact Elements for Nonlinear Analysis

Name	Characteristics	Friction Law	Geometric Nonlinearity
CGAP	Contact between two points. Extension has different springs for compression and tension. Shear force is less than the friction coefficient times the compression. Transverse finite compression stiffness is required.	Coulomb's Law with: kinetic friction 0 static friction	No
Slideline	Contact between a point and a line.	Coulomb's Law with: kinetic friction = static friction	Yes

Small Strain Elements

A subset of MSC Nastran's elements for linear analysis is available for nonlinear structural analysis (see Table 3-9). A more general overview for all types of elements is presented in Structural Elements, 41. The nonlinear elements for small strain analysis can handle material or geometric nonlinearities or both. In geometric nonlinear analysis, the elements may undergo large total displacements and rotations but the net deformation of each element has to remain small, therefore these elements are called “small strain” elements. For linear material and infinitesimal deformation, a nonlinear analysis with small strain elements gives results which are identical to a linear analysis. The small strain elements may be combined with any other type of elements, e.g., hyperelastic elements and/or linear elements.

Table 3-9 Small Strain Elements in Nonlinear Structural Analysis

Element	Property	Material[4]
Line Elements		
CROD	PROD	MAT1 (MATT1), MATS1
CONROD	--	MAT1 (MATT1)
CTUBE	PTUBE	MAT1 (MATT1)
CBEAM	PBEAM PBCOMP[1] (Composite Beam)	MAT1 (MATT1), MATS1 (only elastic-perfectly plastic)
2-D Elements -- Plane Strain, Plane Stress, Shells		
CQUAD4	PSHELL	MAT1 (MATT1), MAT2 (MATT2), MAT8[3], MATS1, CREEP

Table 3-9 Small Strain Elements in Nonlinear Structural Analysis (continued)

Element	Property	Material[4]
CTRIA3	PCOMP (Composite Shell)[2]	
Solid Elements		
CHEXA 8 Node only	PSOLID	MAT1 (MATT1), MAT9 (MATT9), MATS1, CREEP
CPENTA 6 Node only		
CTETRA 4 & 10 Node		

For all elements, geometric nonlinearity is turned on with PARAM, LGDISP,1.

- PBCOMP may refer to MAT1 only (linear material).
- PCOMP must refer to MAT1, MAT2, MAT8 (only linear material).
- MAT8 must not be referred by a MATS1 entry (only linear orthotropic material).
- MAT1linear elastic isotropic material.
  MAT2linear elastic anisotropic material for plates.
  MAT8linear elastic orthotropic material for plates.
  MAT9linear elastic anisotropic material for solids.
  MATTitemperature dependency for MAT1, MAT2, MAT9.
  MATS1nonlinear material (piecewise linear, elasto-plastic) referred by MAT1, MAT2, MAT9.
  CREEPcreep characteristics, referred by MAT1, MAT2, MAT9.

User Interface

Element-related user input is summarized in Table 3-9. The nonlinear element calculations are activated if a nonlinear material exists (MATS1 or CREEP Bulk Data entries) or if a geometric nonlinear analysis of the whole model is requested with PARAM,LGDISP,1. The default is geometric linear analysis, PARAM,LGDISP,-1.

For elements with rotational degrees of freedom (CBEAM, CTRIA3, CQUAD4), two different definitions for large rotations are available, gimbal angles and rotation vector. The definitions are chosen with PARAM,LANGLE,i. The default is the gimbal angle definition, i = 1. The three gimbal angle components are the three subsequent rotations around the global axes. The second rotation component refers to the second global axis which has been rotated once by the first rotation. The third rotation refers to the third global axis which has been rotated twice by the first and second rotation. The rotation vector definition is chosen with i = 2. The components of the rotation vector refer to the fixed global axes. The direction of the rotation vector is the principal direction of the 3-D rotation. The magnitude of the rotation vector is the principal value of the 3-D rotation. The difference of the two angle definitions in the output of the elements appears only for large 3-D rotations.

Output

Element stress and strain output is requested with the STRESS Case Control command. The stress and strain output for all nonlinear elements is divided into a nonlinear and a linear format. The nonlinear format outputs stresses together with

strains. Compared to the linear format, the nonlinear format provides more information with regard to nonlinear material laws (effective strain, equivalent stress etc.). In the linear format, strain output is not available. The stresses and strains of the small strain elements refer to the undeformed area and length, respectively. The components of the stresses and strains are in the deformed element coordinate system which correlates with the elements' rigid body deformation.

Averaged grid point stresses are requested with the GPSTRESS Case Control command. Averaged grid point stresses are available in nonlinear static analysis but not in nonlinear transient analysis. Element forces are requested with the FORCE or ELFORCE Case Control command. Element forces per unit length refer to the undeformed length. The components of the element forces are in the deformed element system. Element forces are available in nonlinear static analysis but not in nonlinear transient analysis.

Grid point forces are requested with the GPFORCE Case Control command. Grid point forces are not available in nonlinear structural analysis.

Elforces are not available for solid elements.

Deformations are requested with the DISP Case Control command. The output of displacements and rotations is available for all types of nonlinear structural analysis.

Corotational Formulation in Geometric Nonlinear Analysis

For small strain elements, a large deformation is split into element rigid body deformation and element net distortion. The rigid body rotations of the points in the element are approximated by the rotation of the local element triad which follows the element deformation, as shown in Figure 3-10. For the remaining element net distortions, a linear strain measure is used. The method is called corotational formulation. The formulation is capable to simulate large total deformations if a fine mesh is used and the stretches remain small.

In geometric nonlinear analysis with MSC Nastran, the tangent stiffness matrix is calculated in two steps. First, the linear stiffness or nonlinear material stiffness is calculated. The linear stiffness or nonlinear material stiffness is constructed from the variation of the stresses in the virtual work. Second, the differential stiffness is calculated. The differential stiffness is constructed from the variation of the strains in the virtual work. The differential stiffness is also called geometric stiffness.

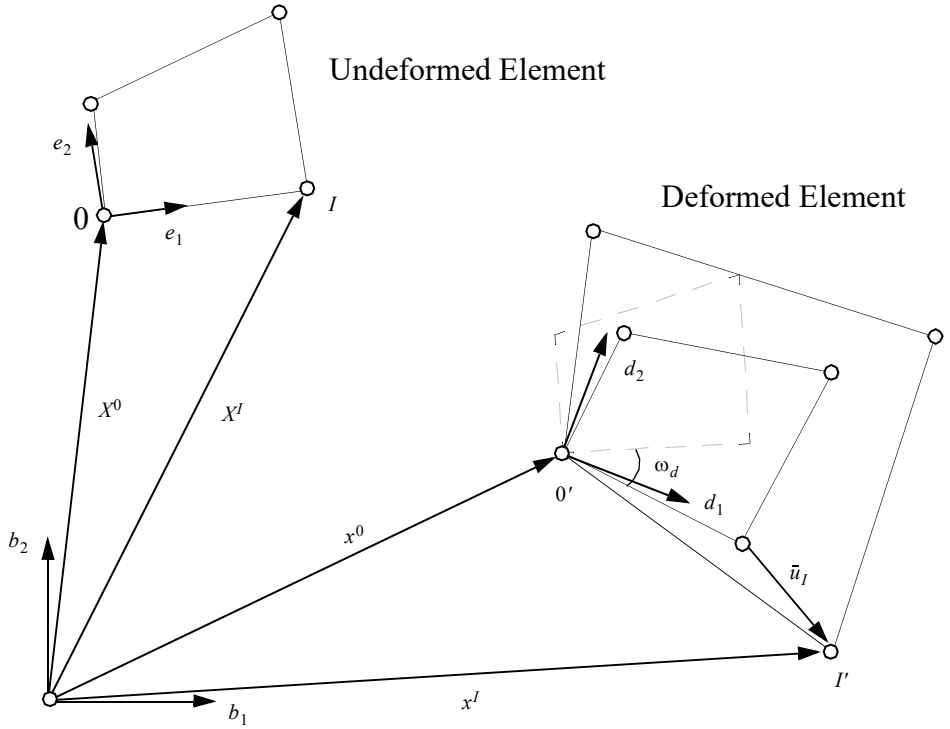


Figure 3-10 Corotational Concept for Small Strain Elements in Geometric Nonlinear Analysis

Guidelines

Large element net deformations should be avoided. In areas of the structure where large total deformations are expected, the mesh must be fine enough to keep the element net deformations small. The element net rotation should not exceed 20 degrees and the element should not be stretched by more than 10%. If stretches exceed 20%, it is recommended to use hyperelastic elements if applicable.

Limitations

- The element net distortions have to remain small.
- Rigid body elements (RBEi, RBAR, RROD entries, etc.) do not rotate in geometric nonlinear analysis.
- Multipoint constraints (MPCs) remain linear, the user defined constraint equations do not change automatically in geometric nonlinear analysis. In nonlinear static analysis, the user may change the MPCs from subcase to subcase. Then, the changes in the MPCs are accounted for incrementally.
- Offsets in the `CBEAM`, `CTRIA3` and `CQUAD4` elements are not allowed in combination with nonlinear material.
- The results in linear and nonlinear buckling with offsets may be incorrect.
- A temperature variation of the nonlinear material constants in plasticity is not available.

- For the CBEAM, the only nonlinear material available is elastic-perfectly plastic.

Fully Nonlinear Hyperelastic Elements

The hyperelastic elements are fully nonlinear finite deformation elements, including large strain and large rotation. The following elements are available: plane strain (CQUAD4, CQUAD8, CQUAD, CTRIA3, and CTRIA6), solid (CHEXA, CPENTA, and CTETRA) and axisymmetric elements (CQUADX and CTRIAX). See Nonlinear Elements, 627, Table 3-6 for more details. The element formulation is total Lagrangean in updated coordinates. See Figure 3-11 for an illustration of the total Lagrangean concept. A nonlinear strain measure and point wise rotation computed at each Gauss point are employed. Equilibrium is satisfied in the deformed configuration.

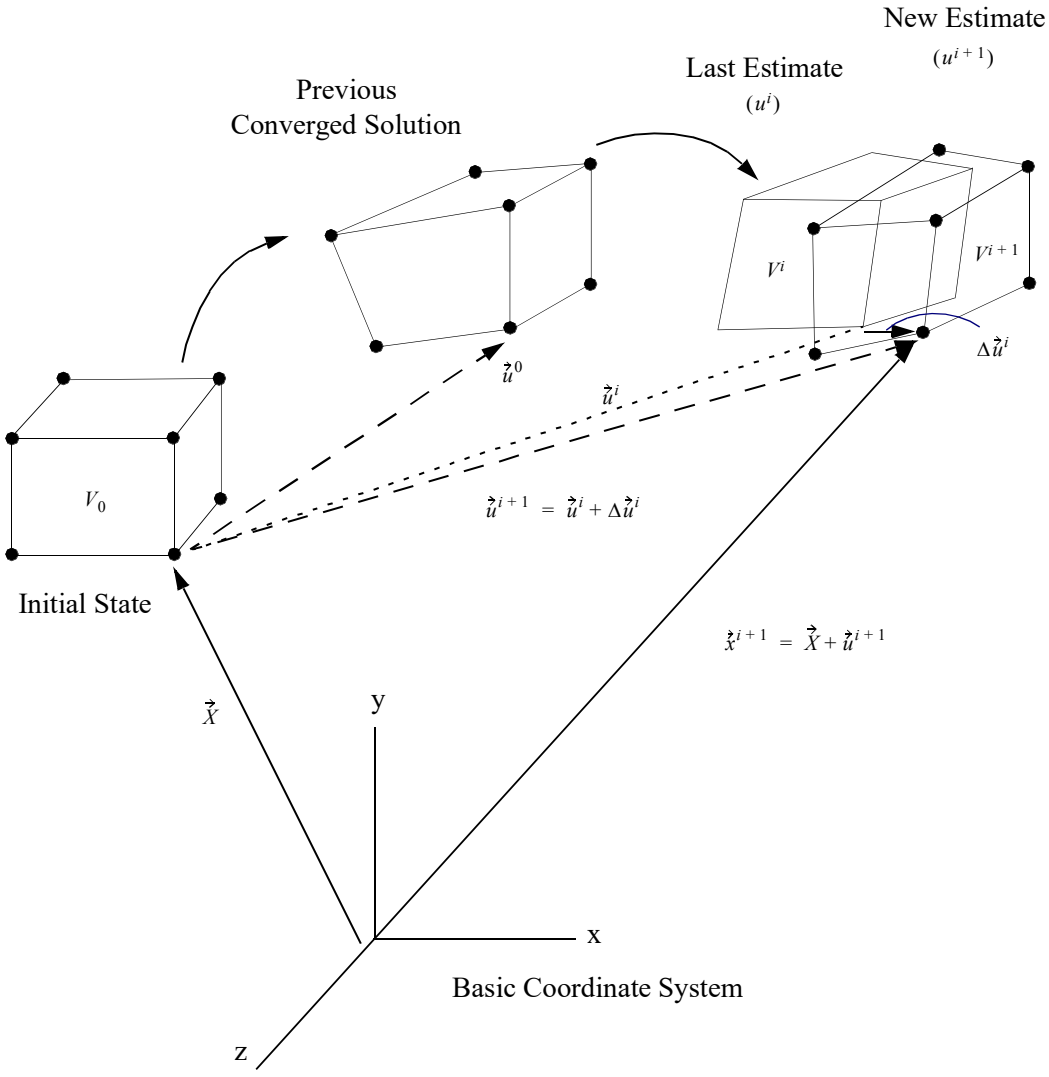


Figure 3-11 Total Lagrangian Concept: Deformations are Measured from an Initial Undeformed State

The elements are especially designed to address volumetric locking and other related problems such as ill-conditioning of the stiffness matrix, incorrect stresses etc., due to near incompressibility. To this end, a mixed formulation based on a three field variational principle in which the pressure and the volume ratio are interpolated independently from the displacements is employed. At the incompressible limit the bulk modulus becomes infinite and the volumetric strain $J-1$ becomes zero so that the pressure may no longer be obtained from the displacements through the constitutive equation. The pressure and volume ratio interpolations are discontinuous between the elements.

The hyperelastic elements are supported in the nonlinear solutions, SOLs 106 and 129. They can be run through SOLs 101 and 105 for model checkout in linear analysis, the results, however, will generally not be correct. For buckling analysis, PARAM,BUCKLE in SOL 106 should be used instead.

User Interface

Because of no difference in geometry, the hyperelastic elements are defined using the current CQUAD4, CQUAD8, CTRIA3, CTRIA6 CHEXA, CPENTA, and CTETRA connectivity entries. Additional CQUAD, CQUADX, and CTRIAX connectivity entries are available for the hyperelastic elements. The PID field of the connectivity entries for the plane or axisymmetric and for the solid hyperelastic elements references the PLPLANE and PLSOLID property entries, respectively. The MID field references a MATHP material entry that defines the hyperelastic material. This is the only material model available for fully nonlinear, finite deformation analysis.

Output Description

The nonlinear stress output for the hyperelastic elements differs from other material nonlinear stress output. The hyperelastic stress output contains the following quantities, at each Gauss point, in the basic coordinate system for solid and axisymmetric elements and in the CID coordinate system, defined on a PLPLANE entry, for plane strain elements:

Cauchy stresses, i.e., components of the stress vector $\boldsymbol{t} = d\boldsymbol{P}/dA$ where dA is an infinitesimal area in the deformed configuration, and $\boldsymbol{P}$ is the force. See Hyperelastic Material, 615 for a description of the constitutive relation.

- Pressure

  $$p = \frac{1}{3}(\sigma_x + \sigma_y + \sigma_z)$$

- Logarithmic strains, i.e., the components of

  $$\sum_{l=1}^{3} ln\lambda_l \boldsymbol{N}_l \boldsymbol{N}_l^T$$

  where λ_l are principal stretches l/l_0 in the principal directions $\boldsymbol{N}_l$. See Figure 3-12. For compatibility with other MSC Nastran elements, twice the shear components of the logarithmic strain tensor are printed in the output.

$\boldsymbol{N}_l$ = principal directions in undeformed configuration; $\boldsymbol{n}_l = \boldsymbol{R}\boldsymbol{N}_l$ = principal directions in deformed configuration, obtained from $\boldsymbol{N}_l$ by a rigid body rotation. The deformation along $\boldsymbol{N}_l$ is a pure stretch, after the rigid body motion has been factored out.

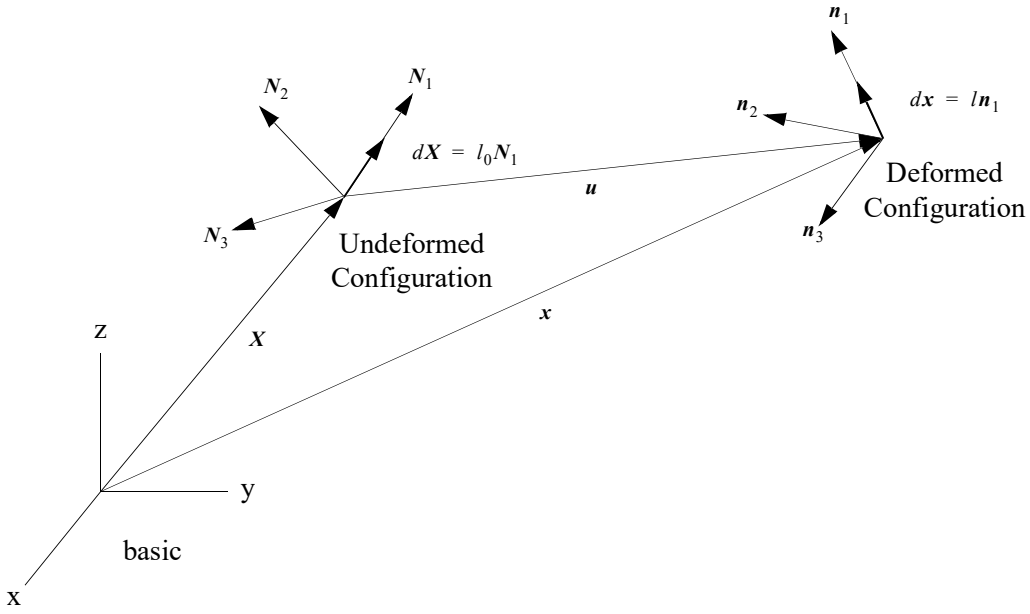


Figure 3-12 Deformation Configurations

- Volumetric strain

$$J - 1 = \frac{dV - dV_0}{dV_0}$$

where J is an approximation to the determinant of the deformation gradient $d\boldsymbol{x}/d\boldsymbol{X}$.

This output is obtained with the STRESS (or ELSTRESS) Case Control command. Examples of the new output for the plane, axisymmetric and solid elements are shown in Listing 3-5.

The linear output contains principal Cauchy stresses and directions with respect to the basic (or CID) coordinate system. See Listing 3-6.

Listing 3-5 Output for Continuum Elements

N O N L I N E A R S T R E S S E S I N H Y P E R E L A S T I C Q U A D R I L A T E R A L E L E M E N T S (Q U A D F D)

ELEMENT ID	GRID/ GAUSS	POINT ID	CAUCHY STRESSES/ LOG STRAINS X	Y	Z	XY	PRESSURE	VOL. STRAIN
1	GAUS	1	1.678579E+02	-1.188351E-03	5.456201E+00	1.560323E-16	5.777097E+01	1.925699E-02
			1.791759E+00	-1.772686E+00	.0	.0		
		2	1.678579E+02	-1.188351E-03	5.456201E+00	-1.191492E-15	5.777097E+01	1.925699E-02
			1.791759E+00	-1.772686E+00	.0	.0		
		3	1.678579E+02	-1.188351E-03	5.456201E+00	-3.368891E-16	5.777097E+01	1.925699E-02
			1.791759E+00	-1.772686E+00	.0	.0		
		4	1.678579E+02	-1.188351E-03	5.456201E+00	-7.083336E-16	5.777097E+01	1.925699E-02
			1.791759E+00	-1.772686E+00	.0	.0		

N O N L I N E A R S T R E S S E S I N H Y P E R E L A S T I C A X I S Y M M. Q U A D R I L A T E R A L E L E M E N T S (QUADXFD)

ELEMENT ID	GRID/ GAUSS	POINT ID	CAUCHY STRESSES/ LOG STRAINS RADIAL(X)	AXIAL(Y)	THETA(Z)	XY	PRESSURE	VOL. STRAIN
1	GAUS	1	1.808360E-09	2.917973E+02	1.808360E-09	4.298727E-15	9.726578E+01	3.242192E-02
			-9.420009E-01	1.930910E+00	-9.570014E-01	1.188717E-01		
		2	1.808132E-09	2.917973E+02	1.808132E-09	-2.820565E-17	9.726578E+01	3.242192E-02
			-9.420009E-01	1.930910E+00	-9.570014E-01	1.188717E-01		
		3	1.808019E-09	2.917973E+02	1.808019E-09	-2.820565E-17	9.726578E+01	3.242192E-02
			-9.420009E-01	1.930910E+00	-9.570014E-01	1.188717E-01		
		4	1.808218E-09	2.917973E+02	1.808218E-09	1.004509E-16	9.726578E+01	3.242192E-02
			-9.420009E-01	1.930910E+00	-9.570014E-01	1.188717E-01		

N O N L I N E A R S T R E S S E S I N H Y P E R E L A S T I C H E X A H E D R O N E L E M E N T S (H E X A F D)

ELEMENT ID	GRID/ GAUSS	POINT ID	CAUCHY STRESSES/ LOG STRAINS X	Y	Z	XY	YZ	ZX	PRESSURE	VOL. STRAIN
1	GAUS	1	2.9180E+02	1.8047E-09	1.8047E-09	-7.8413E-15	-7.7424E-17	2.3330E-15	9.7266E+01	3.2422E-02
			1.9459E+00	-9.5700E-01	-9.5700E-01	.0	.0	.0		
		2	2.9180E+02	1.8047E-09	1.8047E-09	-3.1028E-16	-1.4850E-16	9.6370E-16	9.7266E+01	3.2422E-02
			1.9459E+00	-9.5700E-01	-9.5700E-01	.0	.0	.0		
		3	2.9180E+02	1.8047E-09	1.8047E-09	2.8235E-15	6.2553E-17	-6.4668E-15	9.7266E+01	3.2422E-02
			1.9459E+00	-9.5700E-01	-9.5700E-01	.0	.0	.0		
		4	2.9180E+02	1.8047E-09	1.8047E-09	-5.9973E-16	9.1812E-17	-2.1878E-15	9.7266E+01	3.2422E-02
			1.9459E+00	-9.5700E-01	-9.5700E-01	.0	.0	.0		
		5	2.9180E+02	1.8047E-09	1.8047E-09	2.3338E-15	-1.2646E-16	2.8235E-15	9.7266E+01	3.2422E-02
			1.9459E+00	-9.5700E-01	-9.5700E-01	.0	.0	.0		
		6	2.9180E+02	1.8047E-09	1.8047E-09	-2.3359E-16	-1.1296E-16	2.1388E-15	9.7266E+01	3.2422E-02
			1.9459E+00	-9.5700E-01	-9.5700E-01	.0	.0	.0		
		7	2.9180E+02	1.8047E-09	1.8047E-09	2.1626E-15	2.1405E-17	-6.2480E-15	9.7266E+01	3.2422E-02
			1.9459E+00	-9.5700E-01	-9.5700E-01	.0	.0	.0		
		8	2.9180E+02	1.8047E-09	1.8047E-09	-4.0477E-16	2.4964E-17	-1.9690E-15	9.7266E+01	3.2422E-02
			1.9459E+00	-9.5700E-01	-9.5700E-01	.0	.0	.0		

Main Index

Listing 3-6 Output for Continuum Elements in Basic Coordinate System

```
                                                                                                                                NONLINEAR
    LOAD STEP =  1.00000E+00
          S T R E S S E S   I N   H Y P E R E L A S T I C   Q U A D R I L A T E R A L   E L E M E N T S   ( QUADFD )
 ELEMENT      GRID/    POINT        ---------CAUCHY STRESSES--------              PRINCIPAL STRESSES (ZERO SHEAR)
    ID        GAUSS      ID       NORMAL-X       NORMAL-Y       SHEAR-XY       ANGLE          MAJOR           MINOR
       1      GAUS         1   1.678579E+02  -1.188351E-03   1.560323E-16       .0000    1.678579E+02   -1.188351E-03
                           2   1.678579E+02  -1.188351E-03  -1.191492E-15       .0000    1.678579E+02   -1.188351E-03
                           3   1.678579E+02  -1.188351E-03  -3.368891E-16       .0000    1.678579E+02   -1.188351E-03
                           4   1.678579E+02  -1.188351E-03  -7.083336E-16       .0000    1.678579E+02   -1.188351E-03
S T R E S S E S  I N  H Y P E R E L A S T I C   A X I S Y M M.  Q U A D R I L A T E R A L   E L E M E N T S (QUADXFD)
 ELEMENT      GRID/    POINT        ---------CAUCHY STRESSES--------              PRINCIPAL STRESSES (ZERO SHEAR)
    ID        GAUSS      ID     RADIAL(X)       AXIAL(Y)       SHEAR-XY       ANGLE          MAJOR           MINOR
       1      GAUS         1   1.808360E-09   2.917973E+02   4.298727E-15    90.0000    2.917973E+02    1.808360E-09
                           2   1.808132E-09   2.917973E+02  -2.820565E-17   -90.0000    2.917973E+02    1.808132E-09
                           3   1.808019E-09   2.917973E+02  -2.820565E-17   -90.0000    2.917973E+02    1.808019E-09
                           4   1.808218E-09   2.917973E+02   1.004509E-16    90.0000    2.917973E+02    1.808189E-09
                                                                                                  NONLINEAR    SUBCASE 100 $ UNIAXI
    LOAD STEP =  1.00000E+00
          S T R E S S E S   I N   H Y P E R E L A S T I C   H E X A H E D R O N   E L E M E N T S   ( H E X A F D )
          GRID/    POINT                 --------CAUCHY STRESSES---------                 DIR.  COSINES       MEAN
 ELEMENT-ID GAUSS      ID          NORMAL              SHEAR              PRINCIPAL       -A-  -B-  -C-     PRESSURE
       1    GAUS
                           1  X   2.917973E+02  XY  -7.841313E-15   A   2.917973E+02  LX 1.00  .0   .0    9.726578E+01
                              Y   1.804707E-09  YZ  -7.742389E-17   B   1.804693E-09  LY  .00  .0   .0
                              Z   1.804707E-09  ZX   2.332978E-15   C   1.804693E-09  LZ  .00  .0   .0
                           2  X   2.917973E+02  XY  -3.102808E-16   A   2.917973E+02  LX 1.00  .0   .0    9.726578E+01
                              Y   1.804707E-09  YZ  -1.485038E-16   B   1.804693E-09  LY  .00  .0   .0
                              Z   1.804707E-09  ZX   9.636988E-16   C   1.804693E-09  LZ  .00  .0   .0
                           3  X   2.917973E+02  XY   2.823466E-15   A   2.917973E+02  LX 1.00  .0   .0    9.726578E+01
                              Y   1.804707E-09  YZ   6.255328E-17   B   1.804707E-09  LY  .00  .0   .0
                              Z   1.804707E-09  ZX  -6.466815E-15   C   1.804707E-09  LZ  .00  .0   .0
                           4  X   2.917973E+02  XY  -5.997311E-16   A   2.917973E+02  LX 1.00  .0   .0    9.726578E+01
                              Y   1.804707E-09  YZ   9.181191E-17   B   1.804707E-09  LY  .00  .0   .0
                              Z   1.804707E-09  ZX  -2.187819E-15   C   1.804707E-09  LZ  .00  .0   .0
                           5  X   2.917973E+02  XY   2.333811E-15   A   2.917973E+02  LX 1.00  .0   .0    9.726578E+01
                              Y   1.804707E-09  YZ  -1.264558E-16   B   1.804707E-09  LY  .00  .0   .0
                              Z   1.804707E-09  ZX   2.823466E-15   C   1.804707E-09  LZ  .00  .0   .0
                           6  X   2.917973E+02  XY  -2.335867E-16   A   2.917973E+02  LX 1.00  .0   .0    9.726578E+01
                              Y   1.804693E-09  YZ  -1.129638E-16   B   1.804694E-09  LY  .00  .0   .0
                              Z   1.804693E-09  ZX   2.138826E-15   C   1.804694E-09  LZ  .00  .0   .0
                           7  X   2.917973E+02  XY   2.162630E-15   A   2.917973E+02  LX 1.00  .0   .0    9.726578E+01
                              Y   1.804665E-09  YZ   2.140466E-17   B   1.804636E-09  LY  .00  .0   .0
                              Z   1.804665E-09  ZX  -6.248027E-15   C   1.804636E-09  LZ  .00  .0   .0
                           8  X   2.917973E+02  XY  -4.047673E-16   A   2.917973E+02  LX 1.00  .0   .0    9.726578E+01
                              Y   1.804651E-09  YZ   2.496398E-17   B   1.804622E-09  LY  .00  .0   .0
                              Z   1.804651E-09  ZX  -1.969031E-15   C   1.804622E-09  LZ  .00  .0   .0
    LOAD STEP =  1.00000E+00
```

Guidelines

MSC Nastran performs a fully nonlinear analysis, including the effect of large strain and large rotation, if a PLPLANE and/or PLSOLID Bulk Data entry are encountered. The hyperelastic material is defined by using the MATHP Bulk Data entry. A nonlinear analysis is always performed except in linear solutions when a hyperelastic material is defined.

It is best that the differential stiffness PARAM,LGDISP,1 be used in all hyperelastic runs, but it is not the default. If the strains and the rotations are small, then the hyperelastic analysis should compare to the linear elastic analysis. However, for hyperelastic analysis, it is always best to use PARAM,LGDISP,1.

Due to the independent pressure interpolations, particularly at very high compressive values of the pressure (of the order of the bulk modulus), indefiniteness of the stiffness matrix may occur, caused by an apparent material instability. By default in such instances MSC Nastran will ignore the so-called differential stiffness. Although indefiniteness of the stiffness matrix is probably not associated with a geometric instability, this may indeed help, as many of the terms, which heavily depend on the pressure, are part of the differential stiffness.

Limitations

The fully nonlinear hyperelastic CQUAD, CQUAD4, CQUAD8, CTRIA3, and CTRIA6 are plane strain elements only; they do not contain bending properties, nor can they be used in plane stress analysis. Therefore, they cannot be subject to out-of-plane loading conditions and do not sustain a thickness. The elements assume a thickness of 1.0 for mass calculation in dynamic analyses.

The lower-order fully nonlinear hyperelastic elements CQUAD4, CTRIA3, 8-noded CHEXA, 6-noded CPENTA, and 4-noded CTETRA are not immune to shear locking and therefore are not suitable for bending type loading. The higher order elements CQUAD8, 9-noded CQUAD, CTRIA6, 20-noded CHEXA, 15-noded CPENTA, and 10-noded CTETRA should be used in such situations. See Example 1: Rubber Bushing Problem, 651.

Even though the fully nonlinear elements are defined using the current CQUAD4, CQUAD8, CTRIA3, CTRIA6, CHEXA, CPENTA, and CTETRA Bulk Data entries, internally, MSC Nastran designates them as the separate elements, CQUAD4FD, CHEXAFD (for Finite Deformation) etc., because of the different element formulations. This designation is printed in all the output in which the elements are referenced by name, such as PARAM,EST, PARAM,GPECT, the sequence processor, and the element stress output.

GPSTRESS and FORCE output is not available for the fully nonlinear hyperelastic elements.

The element types for the fully nonlinear hyperelastic elements are as follows:

Element	Element Number	Element Number in OESNL Data Block
CQUAD4	139	201
CQUAD8 (Reverts to CQUAD4 when all midside grids are omitted)	164	208
CQUAD (Reverts to CQUAD4 when center and all midside grids are omitted)	164	208
CTRIA3	162	206
CTRIA6 (Reverts to CTRIA3 when all midside grids are omitted)	167	211
CHEXA with 8 nodes	140	202
CHEXA (Reverts to 8-noded CHEXA when all midside grids are omitted)	163	207
CPENTA with 6 grids	160	204
CPENTA (Reverts to 6-noded CPENTA when all midside grids are omitted)	165	209
CTETRA	161	205
CTETRA (Reverts to 4-noded CTETRA when all midside grids are omitted)	166	210
CQUADX	170	214
CQUADX (Reverts to 4-noded CQUADX when all midside grids are omitted)	171	215
CTRIAX	168	212
CTRIAX (Reverts to 3-noded CTRIAX when all midside grids are omitted)	169	213

Example 1: Rubber Bushing Problem

The cross section of a rubber bushing is shown in Figure 3-13. It is assumed that the frame and internal shaft are rigid and the rubber bushing is perfectly bonded to these components. The rubber material is assumed to be of the Mooney-Rivlin type with $A_{10} = 0.177\ N/mm^2$, $A_{01} = 0.045\ N/mm^2$, and $D_1 = 333\ N/mm^2$. Considering symmetric conditions, only one-half of the rubber bushing is modeled with 72 finite deformation 4-noded CQUAD elements. The same problem is also modeled with 72 hyperelastic 9-noded CQUAD elements. The grid points on the outer boundary are fully constrained to simulate the rubber/frame interface. For the grid points on the inner boundary, only the horizontal degrees of freedom were constrained, while the vertical degrees of freedom were tied together with MPCs. Force is applied to the top grid point on the inner wall in the vertical direction.

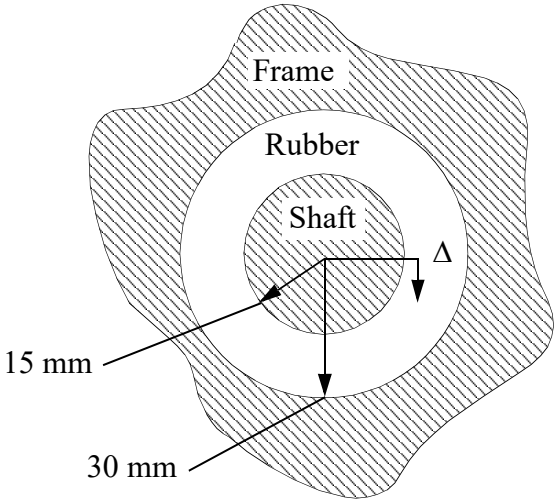


Figure 3-13 Rubber Bushing

The goal of the analysis is to determine the force-displacement curve of the unit and compare it with the solution in the reference. (Reference: Sussman, T., and Bathe, K. J., "A Finite Element Formulation for Nonlinear Incompressible Elastic and Inelastic Analysis," *Computers and Structures* (26), pages 357-409, 1987.)

Annotated Bulk Data input of interest is shown in Listing 3-7.

Listing 3-7 Input for Rubber Bushing Example

```
ID MSC,BUSHING $
SOL 106
TIME 180
CEND
TITLE = RUBBER BUSHING                                          BUSHING
SUBTITLE = HYPERELASTIC MODEL TEST
ECHO=UNSORT
DISP(PLOT)=ALL
STRESS(PLOT)=ALL
SUBCASE 100
  MPC    = 100
  NLPARM = 100
  LOAD   = 100  $
.
.
.
SUBCASE 800
  MPC    = 100
  NLPARM = 800
  LOAD   = 800  $
BEGIN BULK
$ PARAMETERS
PARAM,DBDROPT,0
PARAM,POST,0
PARAM,LGDISP,1
$
NLPARM1005ITER1NO+NL101
+NL101-3+NL102
.
.
.
NLPARM8005ITER1NO+NL801
+NL801-3+NL802
$ DEFINE GEOMETRY
CORD2C10.0.0.0.0.0.1.+COR1
+CORD1.0.0.
GRDSET3456
GRID101 115.0-90.0.13456
GRID102 117.5-90.0.13456
GRID103 120.0 -90.0.13456
.
.
.
$ DEFINE ELEMENTS
CQUAD41011101102202201
CQUAD41021102103203202
CQUAD41031103104204203
.
.
.
$ PROPERTY ENTRY FOR 4-NODE PLANE STRAIN ELEMENT WITH MIXED ELEMENT FORMULATION
PLPLANE 1       1
$ HYPERELASTIC MATERIAL - MOONEY RIVLIN TYPE
MATHP   1       0.177   0.045   333.3
$ DEFINE LOADING CONDITIONS
FORCE1001301-50.0.1.0.
.
.
FORCE8001301-400.0.1.0.
$ DEFINE CONSTRAINTS
MPC10010121.13012-1.
.
.
.
ENDDATA
```

The resulting force-displacement curve at grid point 1301 is shown in Figure 3-14 and the original and deformed models at a force of 800N are shown in Figure 3-15. The solution obtained with 4-noded CQUAD elements is overly stiff for a force greater than 200N (shear locking). The shear locking problem is addressed with the higher order 9-noded CQUAD elements.

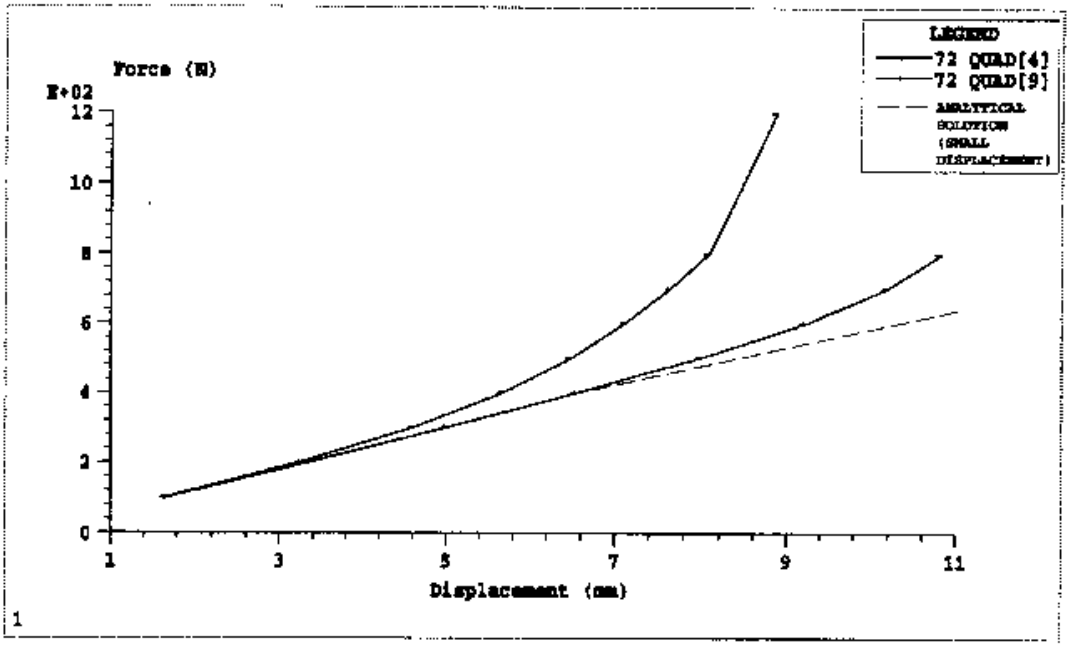


Figure 3-14 Force-Displacement Curve of a Rubber Bushing

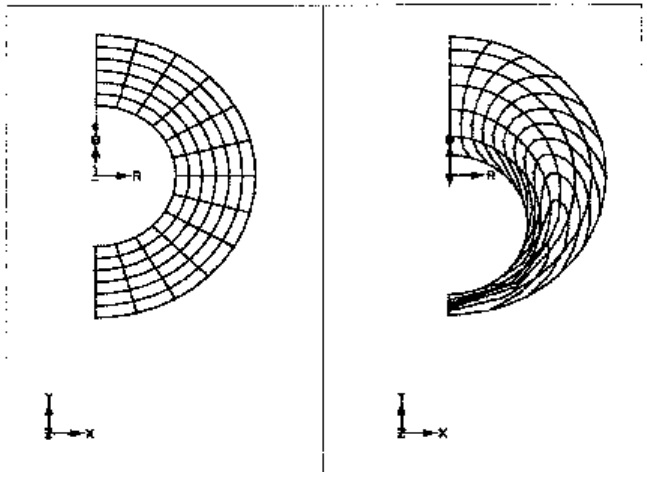


Figure 3-15 Initial and Deformed Shapes of a Rubber Bushing

The model file for this example is bushing.dat, which can be found in the tpl directory.

Example 2: Pressure of a Circular Plate

A circular plate is 15 in. in diameter and 0.5 in. thick. It is simply supported along the edge and is subjected to a uniform follower pressure. The plate is modeled with solid (CHEXA and CPENTA) and with axisymmetric elements. The solid model represents a 10° segment. Material constants are $A_{10} = 80$ psi , $A_{01} = 20$ psi , $D_1 = 50{,}000$ psi .

Annotated Bulk Data input of interest for the wedge model is shown in Listing 3-8 and for the axisymmetric model is shown in Listing 3-9.

Listing 3-8 Input of Circular Plate Wedge Model

```
BEGIN BULK
$ DEFINE PARAMETERS
PARAM,LGDISP,1
PARAM,DBDRNL,-1
PARAM,POST,0
NLPARM  10      10              ITER    1       100                     +NL10
+NL10                           -3
.
.
.
$ DEFINE GEOMETRY
CORD2C  1               0.      0.      0.      0.      0.      1.      +COR1
+COR1   1.      0.      0.
GRDSET                                                  2456
GRID    101     1       0.      -5.     -0.25   1       12456
GRID    102     1       0.75    -5.     -0.25   1
GRID    103     1       1.50    -5.     -0.25   1
.
.
.
$ DEFINE ELEMENTS
CHEXA   102     1       102     103     1103    1102    202     203     +HX102
+HX102  1203    1202
CHEXA   103     1       103     104     1104    1103    203     204     +HX103
+HX103  1204    1203
CHEXA   104     1       104     105     1105    1104    204     205     +HX104
+HX104  1205    1204
.
.
.
CPENTA  101     1       101     102     1102    201     202     1202
CPENTA  201     1       201     202     1202    301     302     1302
CPENTA  301     1       301     302     1302    401     402     1402
CPENTA  401     1       401     402     1402    501     502     1502
.
.
.
$ DEFINE ELEMENT AND MATERIAL PROPERTIES
PLSOLID 1       1
MATHP   1       80.     20.     5.+4
$ DEFINE LOADING CONDITIONS
LOAD    3       1.      3.      1
LOAD    15      1.      15.     1
.
.
.
LOAD    45      1.      45.     1
PLOAD4  1       101     1.                              101
PLOAD4  1       102     1.                              102     1103
.
.
.
$ DEFINE CONSTRAINTS
MPC     10      1102    1       1.      102     1       -1.
MPC     10      1102    3       1.      102     3       -1.
.
.
.
ENDDATA
```

Listing 3-9 Input for Axisymmetric Model

```
$ INFLATION OF A CIRCULAR PLATE
$ AXISYMMETRIC QUAD4 ELEMENTS
BEGIN BULK
$ DEFINE PARAMETERS
PARAM,LGDISP,1
PARAM,POST,0
NLPARM  10      10              AUTO1                                   +NL10
+NL10                           -3
.
.
.
$ DEFINE GEOMETRY
GRDSET                                                  3456
GRID    101             0.      -0.25   0.              13456
GRID    102             0.75    -0.25   0.
GRID    103             1.50    -0.25   0.
.
.
.
$ DEFINE ELEMENTS
CQUADX  101     1       101     102     202     201
CQUADX  102     1       102     103     203     202
CQUADX  103     1       103     104     204     203
.
.
.
$ DEFINE ELEMENT AND MATERIAL PROPERTIES
PLPLANE1        1
MATHP   1       80.     20.     5.+4
$ DEFINE LOADING CONDITIONS
PLOADX1 3       101     3.              101     102
PLOADX1 3       102     3.              102     103
.
.
.
PLOADX1 45      110     45.             110     111
ENDDATA
```

Original and deformed models are shown in Figure 3-16 and Figure 3-17.

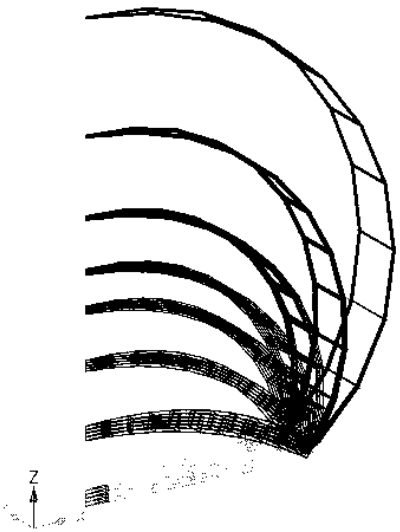


Figure 3-16 Deformed Shape of Circular Plate Under Pressure — CHEXA and CPENTA Hyperelastic Elements

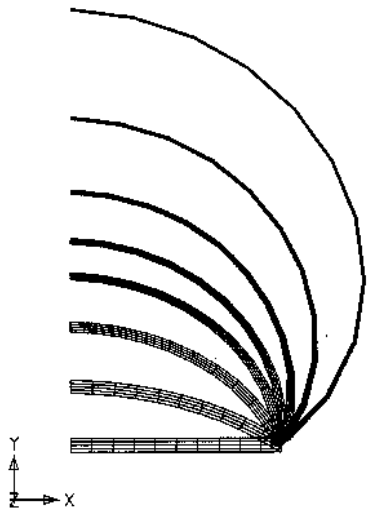


Figure 3-17 Deformed Shape of Circular Plate Under Pressure — Axisymmetric Hyperelastic Elements

The model files for this example are prplth.dat and prpltg.dat, which can be found in the examples directory.

Bushing Elements

The CBUSH element is a generalized spring and damper. Stiffness and damping can be specified for six degrees of freedom, three translations and three rotations. In SOL 106 and 129, the user can specify six nonlinear force-deflection curves instead of combined stiffness values. The properties of the CBUSH element are defined on the PBUSH and PBUSHi entries. The PBUSHT entry refers to TABLEDi entries for force-deflection curves. The CBUSH element is geometric linear only. The circulation is not updated.

The CBUSH1D element is a rod-type spring and damper, it is a 1D version of the CBUSH element. The CBUSH1D element can be geometric and material nonlinear. General nonlinear functions of the axial forces versus axial displacement and for velocity are defined with the PBUSH1D entry. The CBUSH1D element is the only element which can model rotation damping. The effect is demonstrated in the following example.

The CBUSH and CBUSH1D elements are recommended over the CELASi, CVISC, and CDAMP elements. The bushing elements always insure rigid body invariance and they offer nonlinear capabilities.

Swinging Pendulum

The solution to a swinging pendulum is calculated using SOL 129. The pendulum has a length of 1.0 [m]. and has a concentrated mass of M = 1,000.0 [kg] at the free end, see Figure 3-18. We start the analysis with the pendulum at rest in horizontal position. The free end is loaded with a gravity induced load of G = 10,000.0 [N]. The leg of the pendulum is

very stiff, K = 1.e+7 [N/m], so that the relative axial deformation is small compared to the overall motion of the pendulum. Large deformation effects are turned on with PARAM,LGDISP,1.

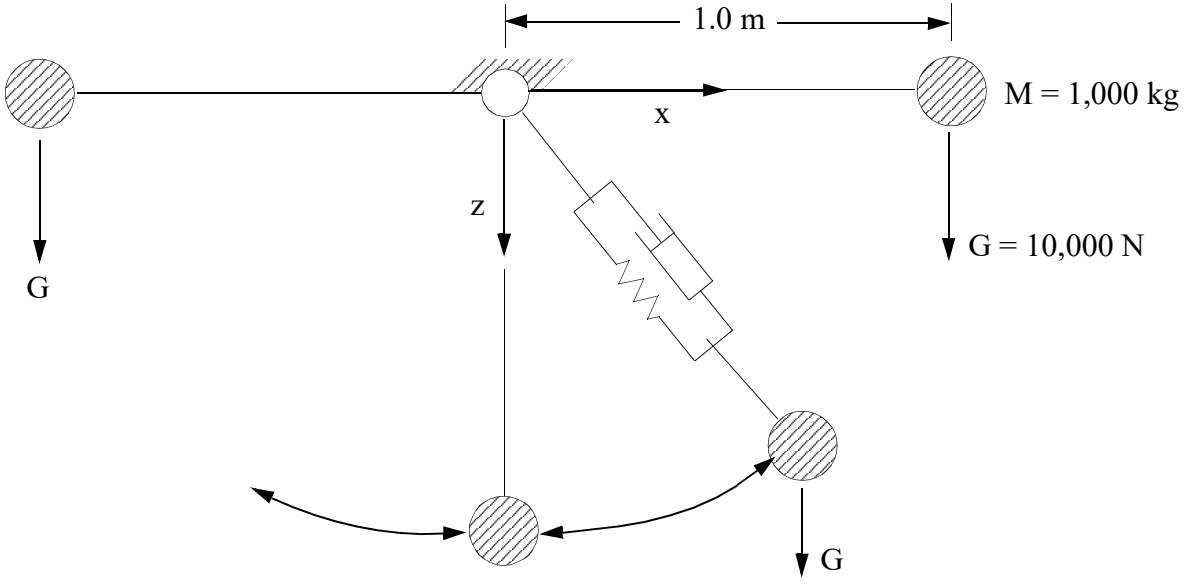


Figure 3-18 Swinging Pendulum

Listing 3-10 MSC Nastran Input File for the Swinging Pendulum

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

We investigate two different models of the pendulum in one run. In the first model, we use a Rod element for the stiffness of the leg, K = 1.e+7 [N/m], and a Visc element for 5% equivalent viscous damping, C = 1,000.0 [Ns/m]. he x- and z-displacements of the free end are shown in Figure 3-19. The Rod rotates while the viscous damper Visc stays fixed in its initial horizontal position. The Visc element is linear and does not follow large deformations. The displacements are damped out because the viscous damper is acting in the x-direction during the whole motion.

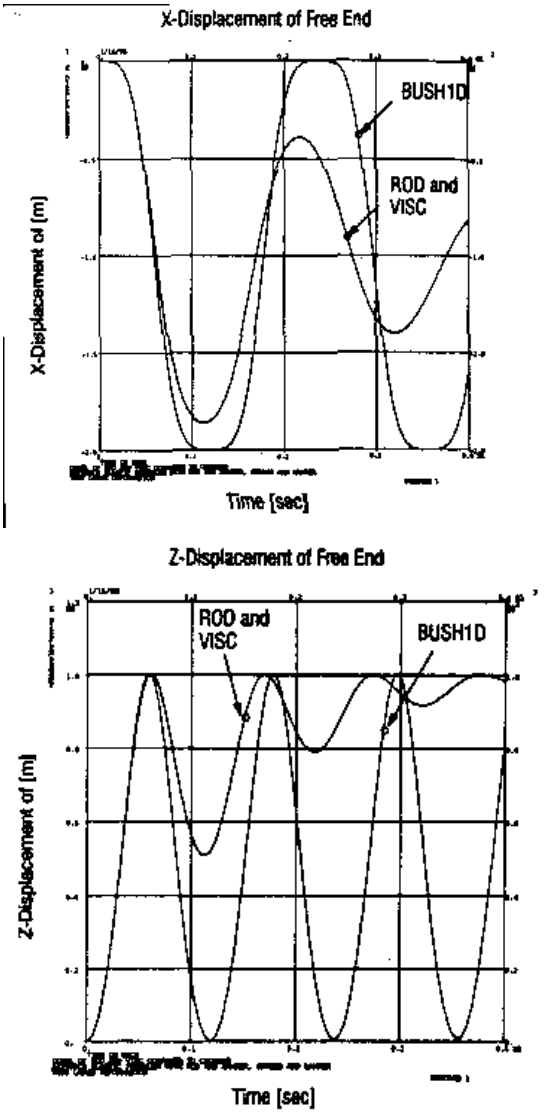


Figure 3-19 Displacement of Swinging Pendulum, Model with Rod/Visc Element Versus Model with BUSH1D Element

In the second model, we use a BUSH1D element. The element has a linear stiffness (K) and a viscous damper (C) with the same values as in the first model. In the BUSH1D element, the spring and damper rotate. The relative axial displacements and velocities are small because of the high axial stiffness of the element. The rotating damper has no noticeable effect on the overall motion of the pendulum because it damps only the small relative deformations. The x-and z-displacements of the free end are not damped, see Figure 3-19.

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

Adaptive Gap Element

The gap element is used in MSC Nastran to model surface contact problems. The adaptivity is based on the subincremental scheme with the capability to update the stiffness at the proper timing, bisect the increment when it is excessive, and adjust the penalty stiffness to the proper values. Generally, in an non-adaptive gap element, the axial stiffness has to be properly chosen in order to reach convergence. However, the main goal of the self-adaptive method is to relieve users from this trial and error method and to achieve convergence on the first try. Due to the adaptive nature of the algorithm, the adaptive gap method costs more in CPU time when compared to the nonadaptive gap element. However, the adaptive gap method allows a user to choose a wider latitude of axial stiffness and yet still achieve convergence. Therefore, with more experience, the user can decrease CPU time by fine-tuning this axial stiffness.

Usage

The gap element is used in nonlinear static analysis (SOL 106) and nonlinear transient analysis (SOL 129). The gap element provides point-to-point contact. When the gap element is open, there is no contact and no friction. When the gap element is closed, there are three different conditions. The first case is when the gap is sliding (no friction). The second case occurs when the gap element is sticking (static friction). The third case occurs when the gap element is slipping (kinetic friction). Sometimes, it may be difficult to converge because of the switching among these conditions.

Limitations

One of the contact surfaces should not rotate by a large angle because the gap element orientation is not updated for large rotations.

Input

How to Use the CGAP and PGAP Bulk Data Entries

Often a mistake is made in the orientation of the gap element, which is defined through G0 or X1, X2, and X3. If the grid points GA and GB are not coincident, then line AB is the element x-axis and the orientation vector lies in the X-Y plane. The use of coincident grid points is not recommended.

On the PGAP Bulk Data entry, U0 and F0 are the initial gap opening and the preload, respectively. U0 is the separation of the gap element. If you are unsure about what preload to use, use the default value of zero. KA, KB, and KT are called the penalty values. KA is the axial stiffness when the gap is closed, KB is the axial stiffness when the gap is open, and the KT is the transverse gap stiffness. All you need to input is the KA value. KB is defaulted to be zero, and KT is defaulted as a function of KA. The selection of KA or a penalty value is critical to the convergence of the problem. The nonadaptive gap does not update this KA value. The new adaptive gap updates this KA value if you specify a positive value for TMAX. The default option for the new adaptive gap is NO adaptive stiffness update. This is when the TMAX value is equal to 0.0. If TMAX is a negative number, then MSC Nastran uses the nonadaptive gap formulation. A good initial KA selection is roughly 1000 times the Young's modulus of the material that is in the neighborhood of the gap element.

Fields 8 and 9 also require some explanation. For the new adaptive value this is MU1, which is the static coefficient of friction, and MU2, which is the dynamic coefficient of friction. The nonadaptive gap elements use MUY and MUZ in the same fields instead of MU1 and MU2 values. Therefore, to compare the results using adaptive and nonadaptive gap elements, set both of these entries to the same values. On most problems, you will not know how to estimate the penetration depth or field 2 on the continuation entry. Therefore, if you skip the continuation entry, the default new adaptive gap is obtained, which is everything adaptive except the KA update.

The recommended allowable penetration of TMAX is 10 percent of the thickness for the shell element, and for the solid element it is roughly 1/10,000 of the characteristic length for the problem. A small positive value will turn on the adaptive stiffness update, which will then converge at a faster rate.

The maximum adjustment ratio or MAR in field 3 of the continuation entry is used only for the penalty value adjustment of the adaptive gap element. The default value of 100 is sufficient for most problems.

Output

The gap element output is obtained by the STRESS output request in the Case Control. An example of the gap output format is shown in Listing 3-11. The load step is 1.95 on the top left-hand corner. The digit on the left-hand side of the decimal indicates that subcase 1 is completed. The digits on the right-hand side of the decimal point indicate that 95 percent of subcase 2 is completed. The far left column lists the gap element numbers. The second column provides the forces in the axial direction of the gap element. The third and the fourth columns represent forces in the Y and Z direction. These forces should be zero if the gap is in the open position. Columns 5 to 9 represent the displacements of the gap in its element coordinate system. The last column indicates the status of the gap. This status can be “open”, “stick”, or “slip”. Sticking is when the ratio of (shear force)/coefficient of friction * normal force) = SR is such that 0.<SR<1. Slip occurs when SR = 1.0, open is when the gap is not in contact; hence SR = 0.

Listing 3-11 Sample of CGAP Output

```
 NONLINEAR STATIC CONTACT OF A SPHERE ON A RIGID          NG6603       NOVEMBER  10, 1992  MSC/NASTRAN  11/ 9/92   PAGE   177
   USE TMAX=0.0016,FULLY ADAPTIVE FEATURE
   KA=1.0E6                                                                                                  SUBCASE 2
  LOAD STEP =  1.92500E+00
                    S T R E S S E S   ( F O R C E S )   I N   G A P   E L E M E N T S     ( C G A P )

 ELEMENT    - F O R C E S   I N   E L E M   S Y S T -       - D I S P L A C E M E N T S   I N   E L E M   S Y S T -
    ID        COMP-X       SHEAR-Y       SHEAR-Z       AXIAL-U       TOTAL-V       TOTAL-W       SLIP-V        SLIP-W    STATUS
    2001   3.60382E+00  -4.47466E-01   0.0           2.23604E-02  -8.30369E-04   0.0          -6.81213E-04   0.0         STICK
    2003   5.04829E+00  -1.10082E+00   0.0           8.85048E-02  -7.80763E-03   0.0          -7.44069E-03   0.0         STICK
    2005   0.0           0.0           0.0           1.95640E-01  -1.70540E-02   0.0          -1.70540E-02   0.0         OPEN
    2007   0.0           0.0           0.0           2.23109E-01  -9.91112E-03   0.0          -9.91112E-03   0.0         OPEN
    2009   0.0           0.0           0.0           2.40853E-01  -4.95583E-03   0.0          -4.95583E-03   0.0         OPEN
 NONLINEAR STATIC CONTACT OF A SPHERE ON A RIGID          NG6603       NOVEMBER  10, 1992  MSC/NASTRAN  11/ 9/92   PAGE   178
   USE TMAX=0.0016,FULLY ADAPTIVE FEATURE
   KA=1.0E6                                                                                                  SUBCASE 2
  LOAD STEP =  1.95000E+00
                    S T R E S S E S   ( F O R C E S )   I N   G A P   E L E M E N T S     ( C G A P )

 ELEMENT    - F O R C E S   I N   E L E M   S Y S T -       - D I S P L A C E M E N T S   I N   E L E M   S Y S T -
    ID        COMP-X       SHEAR-Y       SHEAR-Z       AXIAL-U       TOTAL-V       TOTAL-W       SLIP-V        SLIP-W    STATUS
    2001   3.64258E+00  -4.47838E-01   0.0           2.23643E-02  -8.30493E-04   0.0          -6.81213E-04   0.0         STICK
    2003   5.15982E+00  -1.12016E+00   0.0           8.85160E-02  -7.81408E-03   0.0          -7.44069E-03   0.0         STICK
    2005   8.83100E-02  -5.82848E-03   0.0           1.97000E-01  -1.73532E-02   0.0          -1.73532E-02   0.0         STICK
    2007   0.0           0.0           0.0           2.25831E-01  -1.01834E-02   0.0          -1.01834E-02   0.0         OPEN
    2009   0.0           0.0           0.0           2.44078E-01  -5.09924E-03   0.0          -5.09924E-03   0.0         OPEN
 NONLINEAR STATIC CONTACT OF A SPHERE ON A RIGID          NG6603       NOVEMBER  10, 1992  MSC/NASTRAN  11/ 9/92   PAGE   179
   USE TMAX=0.0016,FULLY ADAPTIVE FEATURE
   KA=1.0E6                                                                                                  SUBCASE 2
  LOAD STEP =  1.97500E+00
                    S T R E S S E S   ( F O R C E S )   I N   G A P   E L E M E N T S     ( C G A P )

 ELEMENT    - F O R C E S   I N   E L E M   S Y S T -       - D I S P L A C E M E N T S   I N   E L E M   S Y S T -
    ID        COMP-X       SHEAR-Y       SHEAR-Z       AXIAL-U       TOTAL-V       TOTAL-W       SLIP-V        SLIP-W    STATUS
    2001   3.66578E+00  -4.47205E-01   0.0           2.23666E-02  -8.30282E-04   0.0          -6.81213E-04   0.0         STICK
    2003   5.22790E+00  -1.12097E+00   0.0           8.85228E-02  -7.81435E-03   0.0          -7.44069E-03   0.0         STICK
    2005   2.37324E-01  -3.59185E-02   0.0           1.97024E-01  -1.73652E-02   0.0          -1.73532E-02   0.0         STICK
    2007   0.0           0.0           0.0           2.27630E-01  -1.03924E-02   0.0          -1.03924E-02   0.0         OPEN
    2009   0.0           0.0           0.0           2.46395E-01  -5.21857E-03   0.0          -5.21857E-03   0.0         OPEN
 NONLINEAR STATIC CONTACT OF A SPHERE ON A RIGID          NG6603       NOVEMBER  10, 1992  MSC/NASTRAN  11/ 9/92   PAGE   180
   USE TMAX=0.0016,FULLY ADAPTIVE FEATURE
   KA=1.0E6                                                                                                  SUBCASE 2
  LOAD STEP =  2.00000E+00
                    S T R E S S E S   ( F O R C E S )   I N   G A P   E L E M E N T S     ( C G A P )

 ELEMENT    - F O R C E S   I N   E L E M   S Y S T -       - D I S P L A C E M E N T S   I N   E L E M   S Y S T -
    ID        COMP-X       SHEAR-Y       SHEAR-Z       AXIAL-U       TOTAL-V       TOTAL-W       SLIP-V        SLIP-W    STATUS
    2001   3.68888E+00  -4.46565E-01   0.0           2.23689E-02  -8.30068E-04   0.0          -6.81213E-04   0.0         STICK
    2003   5.29569E+00  -1.12167E+00   0.0           8.85296E-02  -7.81458E-03   0.0          -7.44069E-03   0.0         STICK
    2005   3.86731E-01  -6.51561E-02   0.0           1.97039E-01  -1.73749E-02   0.0          -1.73532E-02   0.0         STICK
    2007   0.0           0.0           0.0           2.29423E-01  -1.06010E-02   0.0          -1.06010E-02   0.0         OPEN
    2009   0.0           0.0           0.0           2.48707E-01  -5.33770E-03   0.0          -5.33770E-03   0.0         OPEN
```

Example: Hertz Contact Stress

Hertz contact stress is a well known problem arising in the contact of a wheel and a rail. To guard against the possibility of surface failure, it is necessary to calculate the stress that results from loading one body against another. The gap element is used to model the point-to-point contact phenomenon. In this case, the gap element represents the contact surface. The Hertz contact stress problem considered here is a nonlinear static analysis using SOL 106.

Modeling Techniques

For the static analysis, a symmetric half of a 10-degree sector of the sphere is modeled by CHEXA and CPENTA elements with axisymmetric boundary conditions as shown in Figure 3-20. Contact with a rigid plane is simulated by adaptive gap elements.

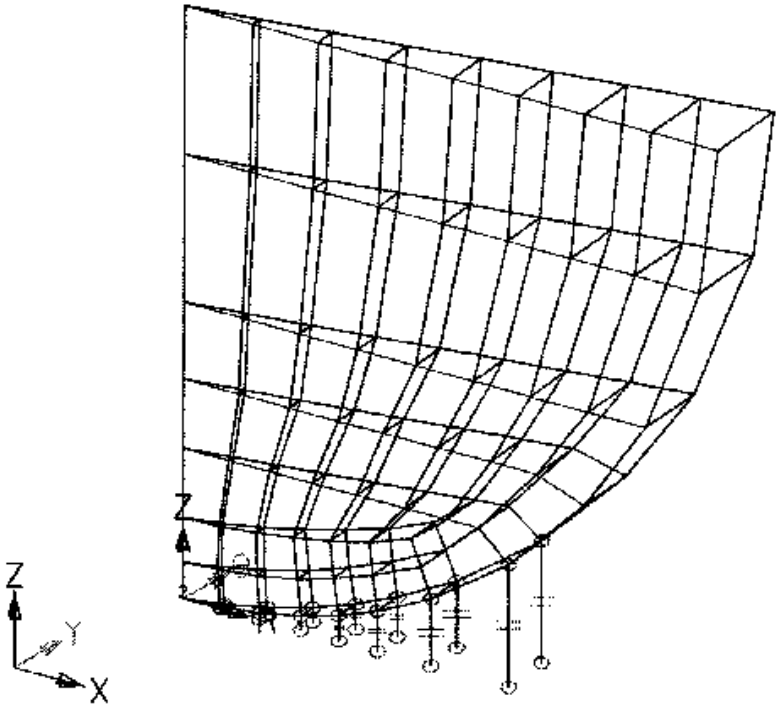


Figure 3-20 Hertz Contact Stress

Uniform vertical displacement of the symmetric plane is maintained by constraining the vertical movement of all the nodes on the top plane by MPC relations at the center line. The radius of the sphere is 8 inches, Young's modulus is 1000 psi, and Poisson's ratio is 0.3. The static analysis is performed in 17 increments up to the maximum loading. The characteristic length of the problem is the diameter of the sphere, i.e., 16 inches. Therefore, the TMAX value is 16/10000, or 0.0016. The initial KA or axial gap stiffness is selected to be 1000 times Young's modulus. These values are inserted in the PGAP entry as shown in Figure 3-21.

1	2	3	4	5	6	7	8	9	10
$	EID	PID	GA	GB	X1	X2	X3		
CGAP	2001	1	105	301	1.	0.	0.		

$	PID	U0	F0	KA	KB	KT	MU1	MU2	
PGAP	1	.022		1.+6			.3	.3	+
$	TMAX	MAR	TRMIN						
+	0.0016								

Figure 3-21 Partial Input Data

Discussion of Results

The objective of this example is to demonstrate that the adaptive features of the gap element allows the user to use a wider range of KA values, and still get good results. Figure 3-22 shows that the if the adaptive features of gap element are used, good results are obtained for KA values ranging from 1.0E1 to 1.0E6. If adaptive features are not used, good results are obtained for KA values between1.0E3 and 1.0E5. For KA values higher than 1.0E5 nonadaptive gap element failed to converge, and below 1.0E3, the accuracy decreased.

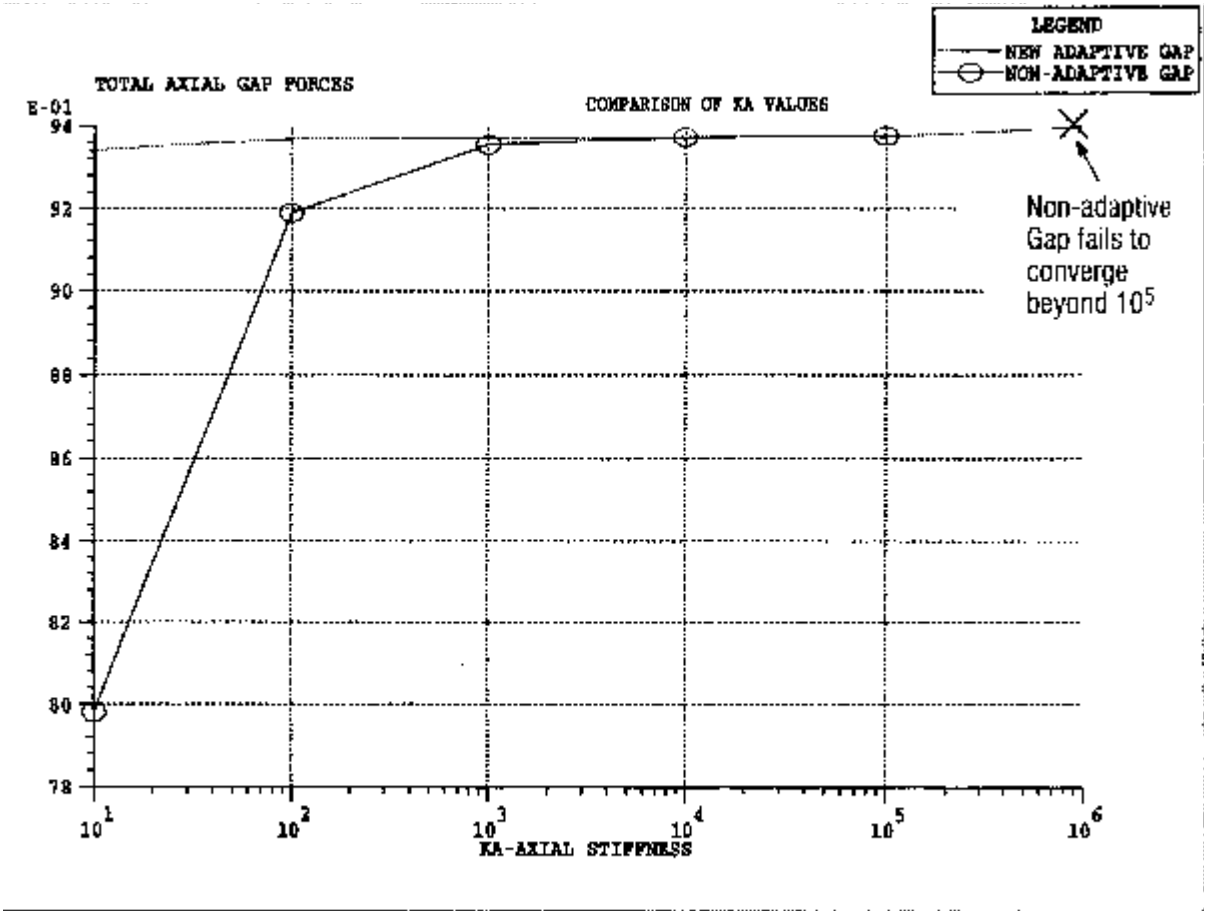
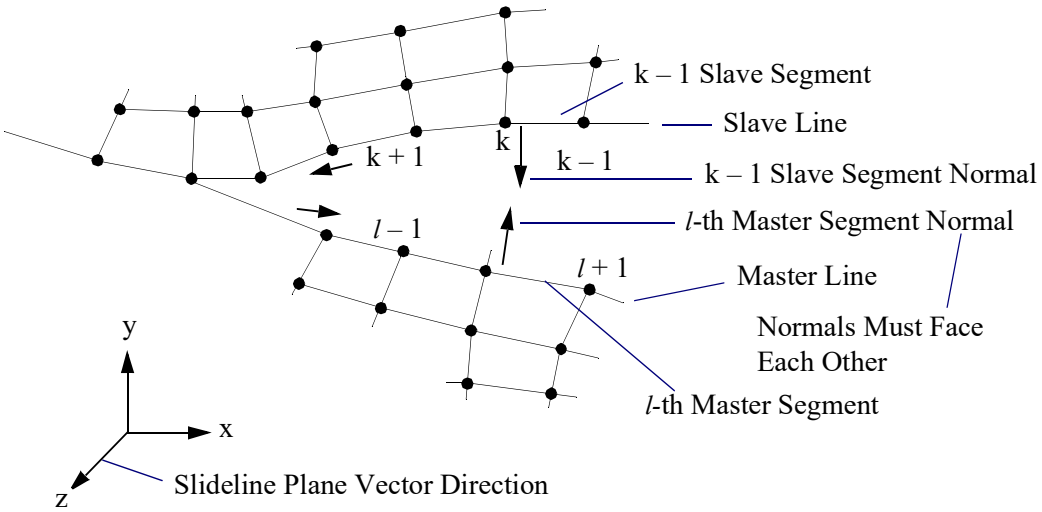


Figure 3-22 Comparison of Axial Stiffness Values

3-D Slideline Contact Capability

Slide line contact allows the user to model interactions between two deformable bodies. One of the deformable bodies is called the master and the other is called the slave. The modeling of interaction requires the user to define contact regions in terms of slide lines. The user may specify as many slide lines as desired.

A slide line contact region consists of a master line and a slave line as shown in Figure 3-23. A master line is a list of grid points in the topological order on the master body. A slave line is a list of grid points in the topological order on the slave body. The grid points on the master line are called the master nodes, and on the slave line are called the slave nodes. A line segment joining two consecutive master nodes is called a master segment. Thus a master line, in general, consists of number of master segments. A minimum of one master segment consisting of two master nodes are required for the master line. Similar to master line, a slave line, in general, consist of a number of slave segments. However, a slave line may not have any slave segments; it may have just one slave node.



- Arrows show positive direction for ordering nodes. Counterclockwise from master line to slave line.
- X-Y plane is the slideline plane. Unit normal in the Z-direction is the slideline plane vector.
- Slave and master segment normals must face each other; i.e., the normals must be in the opposite directions.

Figure 3-23 Typical Slideline Contact Region

The master and the slave nodes in a contact region must lie in a plane called the slide line plane. The slide line plane is defined by a vector normal to the plane called the slide line plane vector. The master and slave can have large relative motions within the slide line plane. However, relative motion outside the slide line plane are ignored, therefore, must be small compared to a typical master segment. Thus, slide line contact is ideal for modelling interactions between two bodies which may come in contact in specific planes. MSC Nastran checks to make sure that in the initial geometry all the nodes of a contact region lie in the slide line plane. However, no such check is made during analysis.

In general, contact is determined only for slave nodes and the master line. This may result in master nodes penetrating the slave line. However, the error involved depends only on the mesh discretization. At the expense of increase in CPU cost, a user may wish to determine contact between master nodes and the slave line in addition to determining contact between

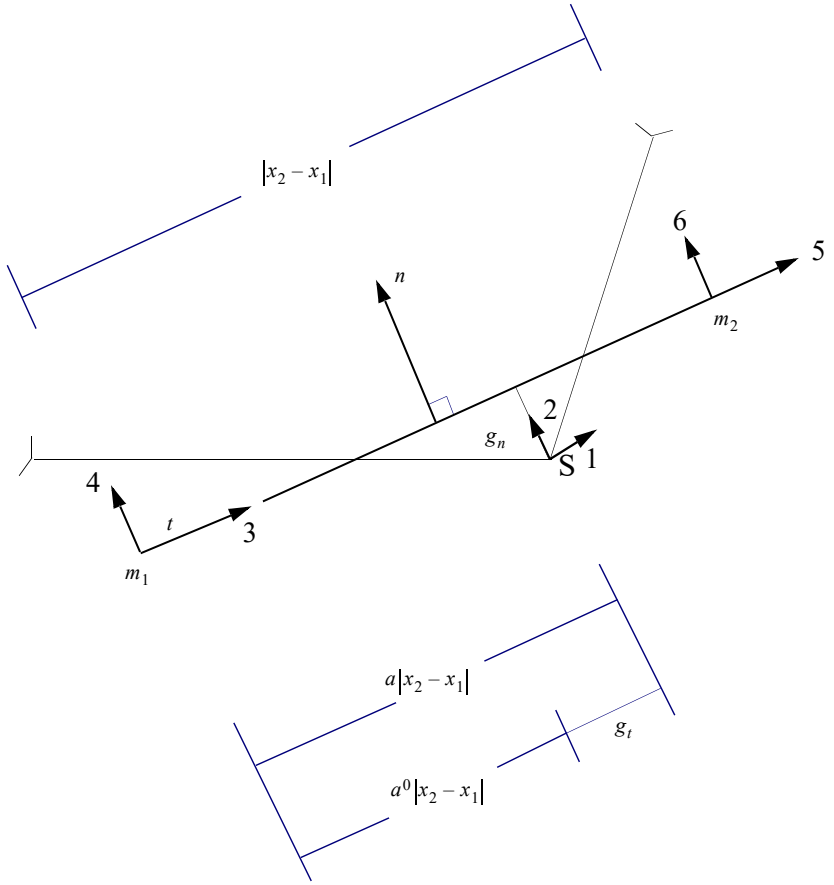
slave nodes and the master line. This is called symmetric penetration. Thus, for symmetric penetration, no distinction is made between slave and master.

To determine contact between the slave nodes and the master line, a local coordinate system is created for each master segment. The tangent direction for the master segment is given by the vector from its first node to the second node. Remember that the master nodes are specified in the topological order. The normal direction to the master segment is determined by the cross product between the slide line plane vector and the tangent vector. This normal direction must point from master line to the slave line. This is because MSC Nastran determines the contact between slave nodes and the master line by measuring how close the slave nodes are to the master segment in this normal direction.

For symmetric penetration, similar to the master segments, tangent and normal directions are also defined for slave segments. Now, the normals of master segments and slave segments must face each other as shown in Figure 3-23. This is generally accomplished by traversing form master line to slave line in a counter-clockwise fashion, or clockwise fashion depending on whether the slideline plane vector forms a right hand or a left hand coordinate system with the slide line plane. It is the user's responsibility to make sure that the normals face each other. MSC Nastran checks indirectly to make sure that the normals face each other by calculating the initial gap between the two bodies. However, this is not a foolproof check, especially when there is no initial clearance.

Initial penetration between the two bodies is not allowed. The initial gaps between the slave nodes and the master line are calculated based on the coordinates specified for the slave and the master nodes. Thus, if there is no initial gap, MSC Nastran may calculate initial penetration if the coordinates are not specified accurately. To avoid having the user to calculate the coordinates very accurately, MSC Nastran adjusts the coordinates automatically and issues a warning message if the initial penetration of a slave node is less than ten percent of the length of the master segment to which it projects. However, if the initial penetration is larger then ten percent, the analysis is terminated and a fatal message is issued. Sometimes, the analysis may be terminated with this message even though there is a large initial gap between the two bodies.This implies that the normals of master or slave segments are not pointing in the correct direction.

Once a contact between a node and a body is detected, a three node slide line element as shown in Figure 3-24 is automatically created. The first node of this element is the slave node and the last two nodes are the master nodes of the master segment with which the slave node is in contact. The slave node is forced to slide on the master line after contact and must remain on the master line until a tensile force develops.



- S, m_1, , m_2 = slave, master node 1 and master node 2, respectively
- a, a_0 = current and previous surface coordinate
- g_n = penetration of slave node into the master segment
- g_t = sliding of the slave node on the master segment
- n = normal direction for the master segment

Figure 3-24 Geometry and Definition of a Typical Three Node Slide Line Element

In MSC Nastran the contact and traction forces/stresses are always associated with slave nodes. In order to compute the contact and traction stresses, an area is associated with each slave node, based on the contributory length and the width/thickness from the adjacent slave segments. Therefore, the user needs to specify widths associated with slave line in

a three dimensional model or thickness associated with slave line in a two dimensional model. A user can request to output results for any number of slave nodes in any number of slide line contact regions. The output for a slave node consists of (a) the slide line contact region identification number, (b) the master segment to which it projects, (c) the parametric surface coordinate to identify the exact projection of slave node relative to the two master nodes of the master segment, (d) contact and tangential forces and stresses in the element coordinate system, and (e) a slip ratio to indicate whether the slave node is sticking, slipping (or sliding). The slip ratio is defined as the (shear force)/(coefficient of friction * normal force). Thus, a ratio of 0. indicates no shear or no contact. A slip ratio of 1. indicates slipping. A value between 0. and 1. is a normalized value which indicates how close the value of the shear force is to impending slipping.

The slide line contact formulation is based on the penalty method which uses an artificial stiffness value. However, the user need not provide penalty values. MSC Nastran will automatically calculate the appropriate numbers. The user has the option to scale the calculated penalty values. It is possible that the automatically calculated penalty values, may sometimes, cause convergence difficulties. However, a user can overcome the convergence problems by restarting the analysis from any previous converged solution and scaling down the penalty values. Generally, it is recommended to scale down the penalty values by one order of magnitude at a time.

Diagnostic Output

Diagnostic output related to 3-D slide line contact can be obtained by setting the executive control statement DIAG to 35.

DIAG 35	Prints current coordinates of contact node and other information associated with slave nodes to debug slide line contact algorithm for each iteration for slide line contact.

Limitations

The slide line capability is only available for SOL 106 and SOL 129. For all other solution sequences all the Bulk Data entries and Case Control commands related to slide line contact are ignored.

Contact stress or force for any slave node could be plotted using XYPLOT BOUT RESP command. The grid ID is the slave node external grid ID. However, the slave node requested for plotting must also be requested for printout via the BOUTPUT Case Control command.

The user interface for slideline contact consists of one Case Control command, five Bulk Data entries and one PARAMeter. The Case Control command is BOUTPUT. The Bulk Data entries are BCONP, BLSEG, BWIDTH, BFRIC, and BOUTPUT. The parameter is ADPCON.

Example

The purpose of this example is to demonstrate how to use the features of slide line contact capability in MSC Nastran.

An elastic sphere in contact with a rigid plane is pressed down with a uniform pressure at the top. No friction is assumed between the sphere and the rigid plane. The analytical solution to this problem were first published by Timoshenko, S. P., and Goodier, J. N., *Theory of Elasticity*, McGraw-Hill Book Company, 1970, pgs. 409-420. For demonstration purposes, a symmetric half of a ten degree sector of the sphere is modeled by CHEXA and CPENTA elements with axisymmetric boundary conditions (see Figure 3-25). Uniform vertical displacements of the symmetry plane is maintained by tying the vertical displacements of all the grid points on the top plane (loading plane) by MPC relations to the one at the center line

(z-axis). A total load of 2160 is applied at the top of sphere. Because only a ten degree sector is modeled and because of the MPC relationships, a load of only 60 is applied at the top center grid point. This load is applied in three subcases with five load increments per subcase.

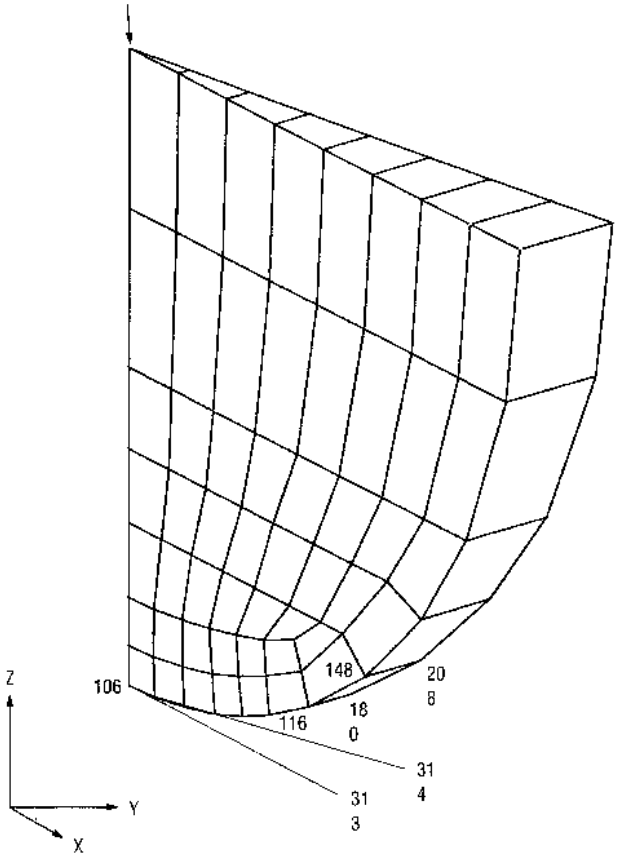


Figure 3-25 Model of a Sphere in Contact with Rigid Plane

To model the contact, two slide line regions are defined via the BCONP Bulk Data entry. The first slide line contact region models the contact between the outer grid points of the sphere at $\theta = 0°$ and the rigid plane as shown in Figure 3-26. The outer grid points of the sphere (148, 116, 114, 112, 110, 108, 105) are used to define the slave line. To define the master line a grid point 313 at $\theta = 0°$ is defined on the rigid plane. The master line is defined by the lower most center grid point of the sphere (grid point 106) and the grid point 313. Notice that the grid points on the master and slave are defined in topological order traversing in counter-clockwise form the master line to the slave line (see BCONP with ID = 10, BLSEG with ID's = 10 and 20 in the example input). This is because the slide line plane is the R-Z plane at $\theta = 0°$ and the slide line plane vector forms a right hand system with the slide line plane (see CORD2R with ID = 1 in the example listing).

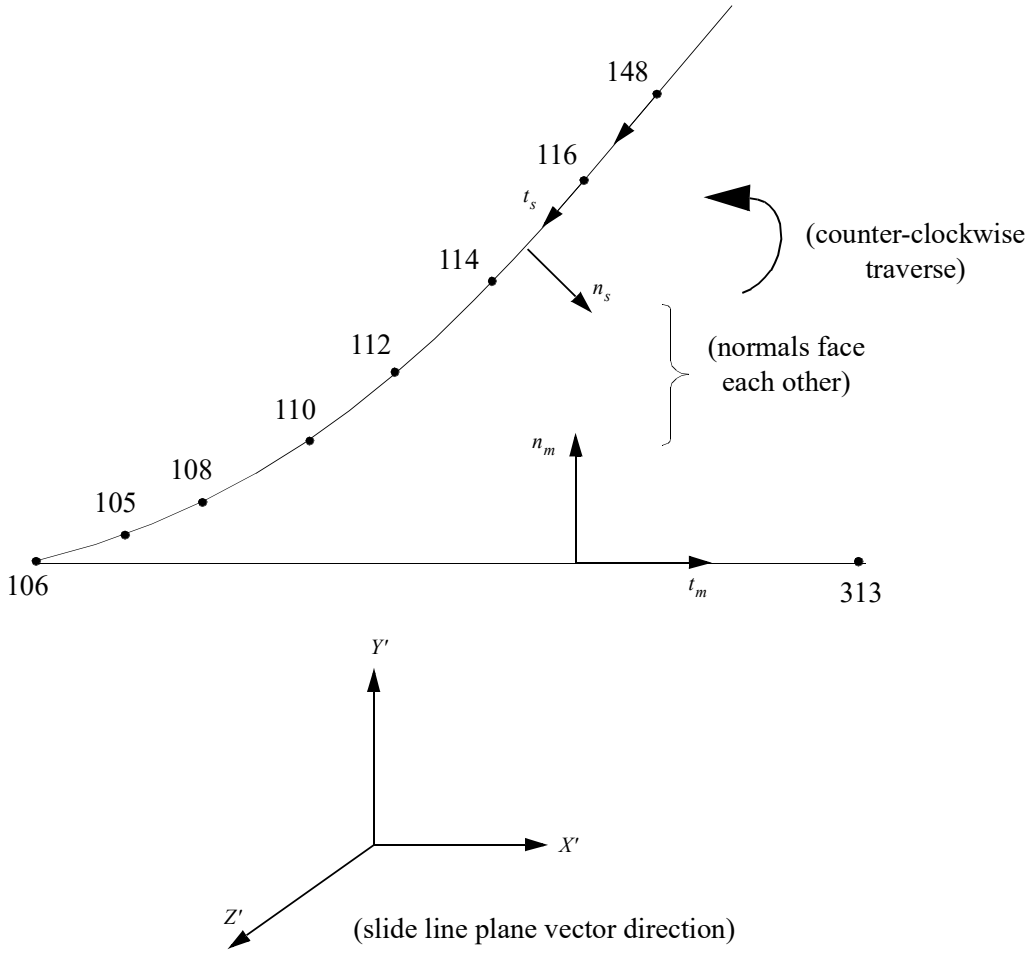


Figure 3-26 Slide Line Contact Region One

The second slide line contact region models the contact between the outer grid points of the sphere at $\theta = 10°$ and the rigid plane as shown in Figure 3-27. Another grid point 314 at $\theta = 10°$ on the rigid plane is defined to model the master line. This time we traverse clockwise form the master line to the slave line in topological order (see BCONP with ID = 20 and BLSEG with ID's = 30 and 40 in the example input) because the slide line plane vector forms a left hand system with the slide line plane (see CORD2R with ID = 3 in the example input).

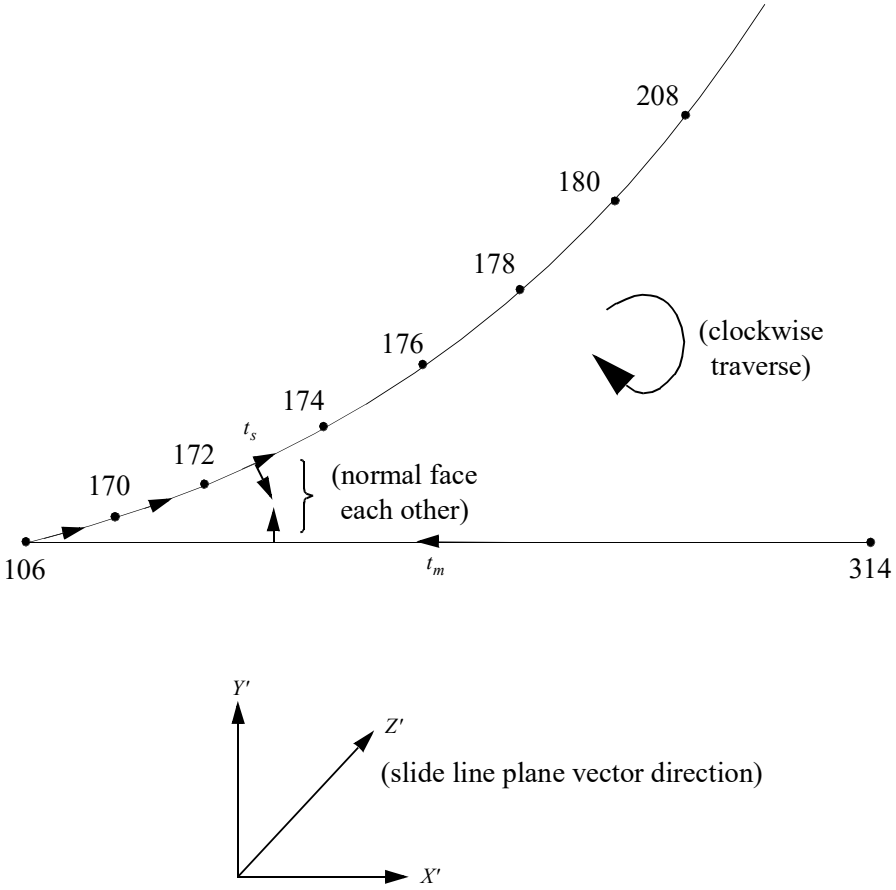


Figure 3-27 Slide Line Contact Region Two

The most important issue is to make sure that the normals of master and slave line point towards each other. The normal to a segment is obtained by $e_3 \times t$. e_3 is the unit vector in the z-direction of the coordinate system used in field CID of BCONP Bulk Data entry. t is the tangential vector form node 1 to node 2 of the line in the order specified in the BLSEG Bulk Data entry.

Listing 3-12 Slideline Example Problem of Sphere Contacting a Flat Plate

```
TIME 300          $
SOL 106           $
CEND
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$           HERTZIAN CONTACT OF A SPHERE ON A FLAT SURFACE      $
$           TWO UNSYMMETRICAL SLIDELINES                        $
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
TITLE=NONLINEAR STATIC CONTACT OF A SPHERE ON A RIGID PLANE WITHOUT FRICTION
  SET 333 = 103,117,124,131,138,151,152,
            105,108,110,112,114,116,148,149,150,168,167
  SET 444 = 106, 301 THRU 314
  SET 555 = 9001, 9002
$
$ use BOUTPUT to request output for contact regions
$
  BOUTPUT(SORT1) = ALL
$
    DISP   = 333
    SPCF   = 444
    STRESS = 555
  SPC = 4
  MPC = 100
SUBCASE 1                       $ 20 LBS
  NLPARM = 410
  LOAD = 100
SUBCASE 2                       $ 40 LBS
  NLPARM = 420
  LOAD = 200
SUBCASE 3                       $ 60 LBS
  NLPARM = 430
  LOAD = 300
OUTPUT(XYPLOT)
  XTITLE = LOAD AFCTOR
  YTITLE = CONTACT NORMAL FORCE
  XYPLOT BOUTPUT RESP/108(6)
  XYPLOT BOUTPUT RESP/105(6)
  YTITILE BOUTPUT ELEMENTS NORMAL STRESS - X
  XYPLOT BOUTPUT RESP/9001(6),9002(6)
BEGIN BULK
$
$ SLIDEllNE CONTACT related bulk data entries
$
BCONP,1,10,20,,     ,,1,1
BCONP,2,30,40,,     ,,1,3
BLSEG,10,148,116,114,112,110,108,105 $
BLSEG,20,106,313 $
BLSEG,30,170,172,174,176,178,180,208 $
BLSEG,40,314,106 $
$
$  COORDINATE SYSTEM FOR SLIDE LINE PLANE VECTOR
$
CORD2R,1,2,0.,.0,.0,1.,-90.,.0,+C2
+C2,1.,0.,0.0
CORD2R,3,2,0.,0.0,.0,1.,100.,.0,+C3
+C3,1.,0.,.0
```

Output Formats

SORT1 Output

```
NONLINEAR STATIC CONTACT OF A SPHERE ON A RIGID PLANE WITHOUT FRIC        MARCH   1, 1993  MSC/NASTRAN   2/17/93   PAGE   109

                                                                                                                SUBCASE 3

 LOAD STEP =  3.00000E+00
                          R E S U L T S   F O R   S L I D E   L I N E   E L E M E N T S   (IN ELEMENT SYSTEM)

SLAVE  CONTAC     MASTER      SURFACE      NORMAL      SHEAR       NORMAL      SHEAR       NORMAL      SLIP      SLIP     SLIP
GRID     ID    GRID1 GRID2  CORDINATE     FORCE       FORCE       STRESS      STRESS        GAP                    RATIO   CODE
  148     1   106   313   9.7604E-01    .0          .0          .0          .0          -1.1240E+00  1.5218E-02   .0      OPEN
  116     1   106   313   7.3917E-01    .0          .0          .0          .0          -4.1923E-01 -4.1647E-03   .0      OPEN
  114     1   106   313   5.7432E-01    .0          .0          .0          .0          -1.1861E-01 -2.3407E-02   .0      OPEN
  112     1   106   313   4.5907E-01   2.4318E+00   .0         4.1069E+00   .0           1.2792E-04 -3.9671E-02   .0     SLIDE
  110     1   106   313   3.4298E-01   1.1404E+01   .0         1.8499E+01   .0           5.9989E-04 -4.8093E-02   .0     SLIDE
  108     1   106   313   2.1905E-01   1.0069E+01   .0         1.6320E+01   .0           5.2968E-04 -3.5747E-02   .0     SLIDE
  105     1   106   313   1.0478E-01   5.2337E+00   .0         1.7661E+01   .0           2.7531E-04 -1.8078E-02   .0     SLIDE
  170     2   314   106   8.9522E-01   5.2335E+00   .0         1.7660E+01   .0           2.7530E-04  1.8078E-02   .0     SLIDE
  172     2   314   106   7.8095E-01   1.0069E+01   .0         1.6320E+01   .0           5.2968E-04  3.5747E-02   .0     SLIDE
  174     2   314   106   6.5702E-01   1.1404E+01   .0         1.8499E+01   .0           5.9989E-04  4.8093E-02   .0     SLIDE
  176     2   314   106   5.4093E-01   2.4318E+00   .0         4.1069E+00   .0           1.2792E-04  3.9671E-02   .0     SLIDE
  178     2   314   106   4.2568E-01    .0          .0          .0          .0          -1.1861E-01  2.3407E-02   .0      OPEN
  180     2   314   106   2.6083E-01    .0          .0          .0          .0          -4.1923E-01  4.1647E-03   .0      OPEN
  208     2   314   106   2.3956E-02    .0          .0          .0          .0          -1.1240E+00 -1.5218E-02   .0      OPEN
```

SORT2 Output

```
NONLINEAR STATIC CONTACT OF A SPHERE ON A RIGID PLANE WITHOUT FRIC        MARCH   1, 1993  MSC/NASTRAN   2/17/93   PAGE    99

                                                                                                                SUBCASE 1

  SLAVE GRID-ID =     110
                          R E S U L T S   F O R   S L I D E   L I N E   E L E M E N T S   (IN ELEMENT SYSTEM)

 STEP     CONTAC     MASTER      SURFACE      NORMAL      SHEAR       NORMAL      SHEAR       NORMAL      SLIP      SLIP     SLIP
            ID    GRID1 GRID2  CORDINATE     FORCE       FORCE       STRESS      STRESS        GAP                    RATIO   CODE
2.000E-01    1   106   313   3.5166E-01    .0          .0          .0          .0          -1.0336E-01 -4.6982E-03   .0      OPEN
4.000E-01    1   106   313   3.5067E-01    .0          .0          .0          .0          -7.1160E-02 -9.6436E-03   .0      OPEN
6.000E-01    1   106   313   3.4968E-01    .0          .0          .0          .0          -4.3237E-02 -1.4611E-02   .0      OPEN
8.000E-01    1   106   313   3.4868E-01    .0          .0          .0          .0          -1.5314E-02 -1.9579E-02   .0      OPEN
1.000E+00    1   106   313   3.4787E-01   5.6225E-01   .0         9.1206E-01   .0           2.9576E-05 -2.3653E-02   .0     SLIDE
1.200E+00    1   106   313   3.4727E-01   1.8074E+00   .0         2.9318E+00   .0           9.5072E-05 -2.6643E-02   .0     SLIDE
1.400E+00    1   106   313   3.4667E-01   3.0525E+00   .0         4.9516E+00   .0           1.6057E-04 -2.9632E-02   .0     SLIDE
1.600E+00    1   106   313   3.4608E-01   4.2976E+00   .0         6.9713E+00   .0           2.2606E-04 -3.2622E-02   .0     SLIDE
1.800E+00    1   106   313   3.4548E-01   5.5427E+00   .0         8.9911E+00   .0           2.9156E-04 -3.5612E-02   .0     SLIDE
2.000E+00    1   106   313   3.4488E-01   6.7878E+00   .0         1.1011E+01   .0           3.5706E-04 -3.8601E-02   .0     SLIDE
2.200E+00    1   106   313   3.4428E-01   8.0329E+00   .0         1.3031E+01   .0           4.2255E-04 -4.1591E-02   .0     SLIDE
2.400E+00    1   106   313   3.4368E-01   9.2780E+00   .0         1.5050E+01   .0           4.8805E-04 -4.4580E-02   .0     SLIDE
2.600E+00    1   106   313   3.4322E-01   1.0329E+01   .0         1.6756E+01   .0           5.4336E-04 -4.6913E-02   .0     SLIDE
2.800E+00    1   106   313   3.4310E-01   1.0867E+01   .0         1.7628E+01   .0           5.7162E-04 -4.7503E-02   .0     SLIDE
3.000E+00    1   106   313   3.4298E-01   1.1404E+01   .0         1.8499E+01   .0           5.9989E-04 -4.8093E-02   .0     SLIDE
```

Nonlinear Static Analysis

User Interface

Nonlinear static analysis can be performed using SOLs 106 or 153. Modeling options are compatible with linear analysis; that is, a SOL 101 model can be converted to SOL 106 analysis with a few additional data entries relevant to the nonlinear analysis. The nonlinear properties and/or effects are defined by nonlinear material data (for example, MATS1, MATHP, CREEP, and TABLES1), gap elements (GAP), slideline contact (BCONP, BLSEG and BFRIC) for nonlinear interface, and PARAMeter LGDISP for geometric nonlinearity.

Case Control

Each subcase defines a set of loading and boundary conditions which can be subdivided into a number of increments. Loading conditions are specified using the same loads as in linear static analysis; that is, FORCEi, MOMENTi, PLOADi, GRAV, TEMP, etc. Constraints can be specified using SPCs and MPCs, which can be changed from subcase to subcase. The enforced motion can be applied using SPCD or nonzero value in SPC. The specified value for SPCD or SPC is a measure of displacement with reference to the undeformed position. The enforced motion is also subdivided in the subcase like the incremental loads. The unique data required for SOL 106 is supplied on the NLPARM entry, which controls the incremental and iterative solution processes. The snap-through analysis or post-buckling analysis can be performed using arc-length methods which are controlled by parameters in the NLPCI Bulk Data entry.

The load step is labeled by the cumulative load factor. The load factor varies from 0 to 1 in each subcase. Then the load step will end with 1, 2, 3, etc. for the first, the second and the third subcase, respectively. There are advantages to divide the entire loading history into many subcases so that each subcase does not have excessive number of increments; for example, not to exceed 20 steps. The data blocks containing solutions can be generated at each increment or the end of each subcase depending on the intermediate output option specified in the INTOUT field of NLPARM data entry. Data blocks are stored in the database for output process and restarts. As such, the database size or the output quantities are better controlled with multiple subcases. Furthermore, the nonproportional loading can only be achieved by using multiple subcases.

Output requests for each subcase are processed independently. Requested output quantities for all the subcases are appended after the computational process for actual output operation. Available outputs are DISPLACEMENT, OLOAD, STRESS, FORCE, GPSTRESS, and SPCFORCE. However, FORCE and GPSTRESS output are available only for linear elements. It should be noted that the STRESS or STRESS(PLOT) command will cause the nonlinear stress to be printed. In other words, the STRESS(PLOT) command suppresses the combined linear/nonlinear stress output only.

Restarts are controlled by parameters (LOOPID and SUBID) which can be specified either in the Case Control Section or Bulk Data Section. Some optional parameters (ADPCON, BUCKLE, LANGLE, MAXLP, NMLOOP) are provided for additional control or capabilities. All the superelement model generation options and matrix reduction options are allowed for the linear portion of the structure.

Restriction and Limitations in SOL 106

- Only CROD, CONROD, CTUBE, CBEAM, CGAP, CQUAD4, CQUAD8, CQUADX, CTRIA3, CTRIA6, CTRIAX, CHEXA, CPENTA, and CTETRA elements may be defined with material or geometric nonlinear properties. All other elements will be treated as having small displacements and linear materials.
- The CBEAM is assumed to have its nonlinear region concentrated at the ends. Taper is allowed but pin joints are illegal for material nonlinearity. However, pin joints can be used if the only nonlinearity is geometric.
- Constraints apply only to the nonrotated displacements at a grid point. In particular, multipoint constraints and rigid elements may cause problems if the connected grid points undergo large motions. However, also note that replacement of the constraints with overly stiff elements can result in convergence problems.
- Large deformations of the elements may cause nonequilibrium loading effects. The elements are assumed to have constant length, area, and volume, except for hyperelastic elements.
- All restarts with grid point or element changes must start over at the first subcase.

- For large displacement analysis, there are two different approaches for the angular motions: gimbal angle and rotation vector approach. In Gimbal angle approach angular motions are treated as three ordered rotations about the z, y, and x axes. In rotation vector approach the three angular motions are treated as a vector. The rotation is about the rotation axis and the magnitude of rotation is equal to the amplitude of rotation vector.
- The CGAP element contact planes do not rotate as a function of displacement. The actual shapes of the contacting surfaces would be required for such calculations.
- Since the solution to a particular load involves a nonlinear search procedure, solution is not guaranteed. are must be used in selecting the search procedures on the NLPARM data. Nearly all iteration control restrictions may be overridden by the user.
- Follower force effects are calculated for loads which change direction with grid point motion.
- All cold-start runs are assumed to start in a stress-free condition with zero displacements and rotations. Prestressed and preloaded cases must be simulated by running them as the initial load case, followed by the desired solution subcase.
- Every subcase must define a new total load or enforced boundary displacement. This is necessary because error ratios are measured with respect to the load change.
- All superelements, except for the residual (SE = 0), are assumed to be linear. All existing superelement restrictions also apply here.
- Any changes to a nonresidual superelement requires the use of the appropriate SEMG, SEMA, SELA, etc., case control request for efficient restarts (see also Nonlinear Analysis With Superelements, 719).
- Only the residual superelement has automatic output data processing. Upstream superelements may be loaded only via `LSEQ` and CLOAD data.
- In postbuckling analysis with arc-length methods (NLPCI Bulk Data entry), the iteration algorithm may jump between different buckling modes when multiple bifurcation points are present in the structure. It is recommended to check all possible buckling modes with the linear buckling solution sequence SOL 105 before making a run with SOL 106. The user has to introduce initial imperfections (loads or enforced displacements) to keep the structure in the intended buckling mode.

Iteration Control: NLPARM Entry

The most crucial data for successful nonlinear static solutions are contained in the NLPARM entry, which defines strategies for the incremental and iterative solution processes. It is difficult to choose the optimal combination of all the options for a specific problem. However, based on a considerable number of numerical experiments, the default option was intended to provide the best workable method for a general class of problems. Therefore, users with little insight or experience in a specific application should start with the default option. However, users with some experience and insight in a specific problem may change the default values based on the following observations:

- Computing cost for each line search is comparable to that of an iteration.
- The SEMI method usually provides better convergence than the AUTO method at the expense of higher computing cost.
- Default tolerances for the convergence criteria may be somewhat conservative. However, loose tolerances may cause difficulties in the subsequent steps.

- The quasi-Newton method is effective in most problems. However, it seems to have adverse effects in some problems; for example, creep analysis.
- The line search method is effective to cope with difficulties in convergence in some problems. More extensive line searches may be exercised by a large value of MAXLS and/or a smaller value of LSTOL. On the other hand, line searches may have adverse effects in some problems; for example, plane stress plasticity.
- The arc-length method should be used if the problem involves snap-through or postbuckling deformation. Then the Bulk Data entry NLPCI must be attached.

The NLPARM entry format is shown below with default values wherever applicable:

1	2	3	4	5	6	7	8	9	10
$NLPARM	ID	NINC	DT	KMETHOD	KSTEP	MAXITER	CONV	INTOUT	
NLPARM			0.0	AUTO	5	25	PW	NO	

$	EPSU	EPSP	EPSW	MAXDIV	MAXQN	MAXLS	FSTRESS	LSTOL	

$	MAXBIS				MAXR		RTOLB		
	5				20.		20.		

1.0E-3	1.0E-3	1.0E-7	3	MAXITER	4	0.2	0.5

The ID field is referenced by the NLPARM Case Control command. The NINC field is an integer which specifies the number of increments to be processed in the subcase. The total load specified in the subcase minus the load specified in the preceding subcase is equally divided by this integer (NINC) to obtain the incremental load for the current subcase. Another subcase should be defined to change constraints or loading paths. However, multiple subcases may be required in the absence of any changes in constraints or loads to use a moderate value (for example, not to exceed 20) for NINC. Use of a moderate value has advantages in controlling database size, output size, and restarts.

The DT field requires a real number specifying the time increment for each load step in the case of creep analysis. The time unit must be consistent with the unit used in the CREEP data to define the creep characteristics. The creep time increment should be determined properly for efficiency and accuracy based on the creep rate represented by the creep law to be used. The larger the creep rate, the smaller the time increment should be. No creep deformation is considered with the default value of 0. It is noted that a creep analysis should be preceded by a static analysis with at least one subcase, because the creep deformation requires stresses in the material.

Stiffness matrix update strategies are determined by a combination of the data specified in the two fields KMETHOD and KSTEP. Options for KMETHOD are AUTO, SEMI, or ITER. The KSTEP field, which is an auxiliary to the KMETHOD field, should have an integer equal to or greater than 1. With the AUTO option, the program automatically determines when to reevaluate the stiffness matrix based on the rate of convergence. At each iteration, the computing time for convergence without the stiffness matrix update is estimated and compared with the computing time for the matrix update in order to determine whether the update is more efficient. This decision is deferred in the first two iterations after a new stiffness is obtained. If the solution tends to diverge, however, the update decision will be made effective immediately. The stiffness matrix will be updated upon convergence if the number of iterations required for convergence is greater than

KSTEP. The SEMI option is identical to the AUTO option except for one additional stiffness update after the first iteration which always occurs unless the solution converges in a single iteration. With the ITER option, the stiffness matrix is updated at every KSTEP iterations. Thus, the full Newton-Raphson iteration is exercised if KSTEP is 1. If KSTEP > MAXITER, the stiffness will never be updated.

The MAXITER field is an integer representing the number of iterations allowed for each load increment. If the number of iterations reaches MAXITER without convergence, the load increment is bisected and the analysis is repeated. If the load increment cannot be bisected (that is, MAXBIS is reached or MAXBIS = 0) and MAXDIV is positive, the best attainable solution is computed and the analysis is continued to the next load increment. If MAXDIV is negative, the analysis is terminated.

The convergence test is performed at every iteration with the criteria specified in the CONV field. Any combination of U (for displacement), P (for load), and W (for work) may be specified. All the specified criteria must be satisfied to achieve convergence, except for an absolute convergence condition, under which the solution is converged regardless of criteria. This subject will be elaborated in the following section. The convergence tolerances are specified in the fields EPSU, EPSP, and EPSW for U, P, and W criteria, respectively.

The defaults for the nonlinear convergence tolerances have been modified to improve run times for most large nonlinear static analyses without appreciably sacrificing accuracy. The new default tolerances also reflect values similar to those used in other popular nonlinear FEA programs such as Marc. Run times of large nonlinear static models with plasticity and/or large deformation models will usually improve significantly. Run times for models with gaps or line contact may also improve, but not as significantly. At this time, it appears that static nonlinear heat transfer models and most nonlinear dynamic analysis models continue to require the present tight default tolerances.

Default nonlinear tolerances in Solutions 106 and 153 will vary according to the model, analysis type, and user-defined precision. Although default tolerances can still be overridden by entering values of CONV, EPSU, EPSP, and EPSW on the NLPARM entry, default tolerances can now be easily invoked with a single parameter, leaving the above fields on the NLPARM entry blank. Depending on the desired level of accuracy and run times, various default tolerance sets may be selected. These tolerances are designated as “very high,” “high,” “engineering design,” and “preliminary design” and can now be controlled with the new parameter

PARAM,NLTOL,ITOL

where ITOL is an integer value ranging from 0 (very high) to 3 (preliminary design). The default is 2 (engineering accuracy) for nonlinear static analysis and 0 (very high) for nonlinear heat transfer and dynamic analyses. See Parameter Descriptions (p. 786) in the *MSC Nastran Quick Reference Guide*, for a complete description.

Tables 3-10, 3-11, and 3-12, illustrate the tolerances that will be set by the program if the CONV, EPSU, EPSP, and EPSW fields on the NLPARM entry are left blank.

Table 3-10 Default Tolerances for Static Nonlinear SOL 106 Models Without Gaps, Contact or Heat Transfer

NLTOL	Designation	CONV	EPSU	EPSP	EPSW
0	Very high	PW	_____	1.0E-3	1.0E-7
1	High	PW	_____	1.0E-2	1.0E-3
2	Engineering	PW	_____	1.0E-2	1.0E-2

Table 3-10 Default Tolerances for Static Nonlinear SOL 106 Models Without Gaps, Contact or Heat Transfer

NLTOL	Designation	CONV	EPSU	EPSP	EPSW
3	Prelim Design	PW	_____	1.0E-1	1.0E-1
none	Engineering	PW	_____	1.0E-2	1.0E-2

Table 3-11 Default Tolerances for Static Nonlinear SOL 106 Models With Gaps or Contact (Enter NLTOL Values of 0 or 2 Only or Omit the Parameter)

NLTOL	Designation	CONV	EPSU	EPSP	EPSW
0	Very high	PW	_____	1.0E-3	1.0E-7
2	Engineering	PW	_____	1.0E-3	1.0E-5
None	Engineering	PW	_____	1.0E-3	1.0E-5

Table 3-12 Default Tolerances for Static Nonlinear SOL 106 or 153 Models With Heat Transfer (Enter NLTOL Value of 0 Only or Omit the Parameter)

NLTOL	Designation	CONV	EPSU	EPSP	EPSW
0	Very high	PW	_____	1.0E-3	1.0E-7
None	Very high	PW	_____	1.0E-3	1.0E-7

To use the new default tolerances, the CONV, EPSU, EPSP, and EPSW fields of the NLPARM entry must be left blank. In addition, the NINC field must also either be blank.

As in the past, if the user has a good idea of what to use for the default tolerances and/or number of increments, then these values should be used. If the problem is nearly linear, a value of NINC=1 sometimes yields acceptable results.

Because nonlinear tolerances now depend on NLTOL, NINC (on the NLPARM entry), and other possible NLPARM overrides, convergence tolerances actually used by the code are now echoed in the output file (right after the Bulk Data echo, if any) as shown in the following example:

```
DEFAULT NONLINEAR TOLERANCES
   NLPARM ID     10
   NINC      10
   EPSU     1.00000E-03 (not normally used)
   EPSP     1.00000E-03
   EPSW     1.00000E-05
BASED ON
MODEL HAS GAPS AND/OR LINE CONTACT
  (VALUES YOU ENTER OVERRIDE THE ABOVE DEFAULTS)
```

Examples of Convergence Tolerance Changes

A very large nonlinear model consisting of approximately one million degrees of freedom was studied extensively to determine the effect of various tolerances on run time and accuracy. This particular model did not employ heat transfer, gaps, or contact. Results of the study showed that compared to the "very high accuracy" solution, acceptable accuracy can be achieved for both the "high accuracy" and "engineering accuracy" designations. The "preliminary design" accuracy tolerances produces results with the same general trends as other reasonable tolerances, but the answers were considered too inexact for analyses other than preliminary design trade off studies. The following table summarizes the effect of changing the tolerances for this large model.

Accuracy Type	Tolerances		Run time CPU hours	Accuracy for the Large Model
	EPSP	EPSW		
Very high	0.001	1.0E-7	163.2	Very Good
High	0.01	0.001	10.3	5% or less accurate
Engineering Design	0.01	0.01	7.0	10% or less accurate
Preliminary Design	0.1	0.1	2.9	10% to 20% less accurate

The INTOUT field requires a specification of YES, NO or ALL. If YES is specified, the output requests (DISPLACEMENT, FORCE, STRESS, etc.), specified in the Case Control Data, will be processed for every computed converged solution in the subcase. If NO is specified, the output requests will be processed only for the last load step of the subcase. If ALL is specified, the output requests are processed for every computed and user specified load increment. For Newton's iteration methods (that is, without NLPCI), the option ALL is equivalent to the option YES, since the computed load increment is equal to the user specified load increment. For arc-length methods (that is, NLPCI command is specified), the computed load increment in general is not the same as the user specified load increment, and is not known in advance. The option ALL allows the user to obtain solutions at the desired intermediate load increments. With a small database option, the database will retain only those data blocks corresponding to the incremental load steps for which output is requested by an INTOUT field. Then, the restarts will be restricted to those starting from these steps. However, if the job is terminated with an incomplete subcase for some reason (for example, diverging, time expiration, etc.), data blocks for all the interim steps of that subcase are stored in the database to allow flexible restarts.

The MAXDIV field requires an integer to specify a limit on the probable divergence conditions allowed for each iteration to continue. There are two classifications in the divergence condition: probable and absolute. The absolute divergence is treated as two occurrences of the probable divergence. When the probable divergence occurs MAXDIV times or more and if MAXDIV is positive, the current solution base (displacements) is retracted and the stiffness matrix is updated in reference to the preceding iteration step. If the solution diverges again in the same load increment while MAXDIV is positive, the best attainable solution is computed and the analysis is continued to the next load increment. If MAXDIV is negative, the analysis is terminated on second divergence.

The MAXQN field requires an integer to specify the maximum number of quasi-Newton vectors to be saved in the database. Quasi-Newton vectors are accumulated, if desired, until MAXQN is reached. If the number of accumulated QN vectors is greater than or close to MAXQN upon convergence, the stiffness matrix will be updated before the next increment is processed. All the QN vectors already accumulated are purged upon a stiffness update and the accumulation is reinitiated.

The quasi-Newton update may be suppressed with a value of 0 for MAXQN. Details of the quasi-Newton method will be presented in the following section.

The line search is controlled by the fields MAXLS and LSTOL. The maximum number of line searches allowed for each iteration is specified in the MAXLS field. The line search process may be suppressed with a value of 0 for MAXLS. The LSTOL field requires a real number between 0.01 and 0.9 to specify a tolerance for the line search operation. The line search operation will be conducted if the error defining the divergence rate is greater than LSTOL. If the line search convergence is not achieved, the line search continues until the number of line searches reaches MAXLS.

The FSTRESS field requires a real number (0.0 < FSTRESS < 1.0), representing a fraction of the effective stress ($\bar{\sigma}$) used to limit the subincrement size in the material routines. The number of subincrements in the material routines for elasto-plastic or creep deformation processes is determined such that a subincrement in the effective stress is approximately $\text{FSTRESS} \cdot \bar{\sigma}$. This data is also used to establish an error tolerance in the yield function to alleviate accumulation and propagation of the error. If the error in the yield function exceeds $\text{FSTRESS} \cdot \bar{\sigma}$ at the converging state, the program will terminate the job with a fatal error message “ERROR EXCEEDED 20% OF CURRENT YIELD STRESS.” If the bisection option is selected, bisection will be activated under this condition.

The MAXBIS field requires an integer to specify the number of bisections allowed for a load or arc-length increment. When the solution diverges, different actions are taken depending on the sign of MAXBIS. If MAXBIS is positive, the stiffness matrix is updated on the first divergence and the load increment is bisected on the second divergence. If MAXBIS is negative, the load increment is bisected every time the solution diverges until the limit on bisection is reached. If the solution does not converge after |MAXBIS| bisections, the analysis is continued or terminated depending on the sign of MAXDIV.

The MAXR field requires a real number to specify the bounds on the ratio of the adjusted arc-length increment relative to the initial value. In the adaptive load/arc-length increment, the overall upper and lower bounds on the load/arc-length increment in the subcase are defined as

$$\frac{1}{MAXR} \le \frac{\Delta l_n}{\Delta l_0} \le MAXR$$

where Δl_n is the arc-length at step n and Δl_0 is the original arc-length. The arc-length method for load increments is selected by the NLPCI Bulk Data entry, which must have the same ID as the NLPARM Bulk Data entry.

The RTOLB field requires a real number to specify the incremental rotation (in degrees) allowed per iteration. The bisection is activated if the incremental rotation for any degree of freedom ($\Delta\theta_x$, $\Delta\theta_y$ or $\Delta\theta_z$) exceeds the value specified for RTOLB. This bisection strategy based on the incremental rotation is controlled by the MAXBIS field.

User Interface for Arc-Length Methods: NLPCI Entry

Most of the parameters in the NLPARM Bulk Data entry are used to control the iteration strategy for arc-length methods. However, MAXLS is not applicable because the line search procedure is not coupled with arc-length methods currently. The parameters which are applicable only to the arc-length methods are specified in the Bulk Data entry NLPCI, in connection with the NLPARM data by the same ID. The NLPCI entry is shown below with default values, followed by brief description of the fields:

1	2	3	4	5	6	7	8	9	10
NLPCI	ID	TYPE	MINALR	MAXALR	SCALE		DESITER	MXINC	
NLPCI		CRIS	0.25	4.	0.0		12		

ID	Must be identical to ID of NLPARM to be active.
TYPE	Constraint type (CRIS for Crisfield, RIKS for Riks, or MRIKS for modified Riks method)
MINALR, MAXALR	For variable arc-length method, the arc-length is bounded by $\text{MINALR} \le \frac{\Delta l_{new}}{\Delta l_{old}} \le \text{MAXALR}$
	where $\text{MINALR} \le 1.$ and $\text{MAXALR} \ge 1$. The automatic adjustment is not activated if MAXALR = MINALR = 1.
SCALE	Scaling factor for the magnitude of the load term in arc length. Real > 0 Default = 0. The constraint equation has a disparity in the dimension by mixing the displacements with the load factor. The scaling factor (w) is introduced as user input so that the user can make constraint equation unit-independent by a proper scaling of the load factor, μ. As the value of w is increased, the constraint equation is gradually dominated by the load term. In the limiting case of infinite w, the arc-length method would be degenerated to the conventional Newton's method.
DESITER	Desired number of iterations for convergence to be used for adaptive arc-length adjustment. $\Delta l_{new} = \Delta l_{old} \sqrt{\frac{Id}{I_{max}}}$
MXINC	Maximum number of controlled increments allowed within the specified load step. There can be cases where the structure never snaps back and collapses to failure. In these cases, the analysis cannot be completed without bounding the maximum number of increments. For such cases, this value should be confined to a reasonable limit, e.g., less than 100, to prevent inadvertent depletion of the computing resources.

Iteration Related Output Data

At the end of every iteration, the relevant data from the iteration process are printed under the following heading. The section numbers in parentheses refer to sections in the *MSC Nastran Handbook for Nonlinear Analysis*.

ITERATION	iteration count i.
EUI	relative error in terms of displacements (see Section 3.6.3).

EPI	relative error in terms of loads (see Section 3.6.3).
EWI	relative error in terms of work (see Section 3.6.3).
LAMBDA	rate of convergence (defined in Section 3.3.1).
DLMAG	absolute norm of the load error vector.
FACTOR	final value of the line search parameter (see Section 3.4).
E-FIRST	divergence rate, initial error before line search (see Section 3.4.3).
E-FINAL	error at the end of line search (see Section 3.4.3).
N-QNV	number of quasi-Newton vectors appended (see Section 3.5.3).
N-LS	number of line searches performed during the iteration.
ENIC	expected number of iterations for convergence (see Section 3.3.1).
NDV	number of occurrences of probable divergence during the iteration (see Section 3.3.2).
MDV	number of occurrences of bisection conditions in the material routine (excessive stress increment) or in the rotation angle (excessive rotation) during the iteration.

Loads Overview

The static loads in nonlinear analysis consist of a subset of the total MSC Nastran static load set. Most of the relevant loads data applicable to the linear static analysis are also applicable to nonlinear static analysis except for the Bulk Data entry DEFORM. The enforced displacements can be applied as loads using the Bulk Data entry SPCD, which defines an enforced motion of a grid in a specific fixed direction. Each grid point with an enforced displacement must also appear on an SPC or SPC1 entry.

The loads are selected in each subcase by a Case Control command LOAD. The load for a subcase is subdivided into the number of increments specified for the subcase. The solution strategy in nonlinear is to apply the loads in an incremental fashion until the desired load level is reached. The algorithms “remember” the loads from one subcase to the next. If the load reaches the desired level in a subcase and if the load description is left out in the subsequent subcase inadvertently, the nonlinear solution algorithm will begin in an incremental fashion to remove the load. The loads described below make up the valid subset applicable for nonlinear static analysis.

Concentrated Loads

The concentrated loads consist of two categories: those which are stationary in direction through out the analysis and those which follow the grid motion. The first category consists of the following: FORCE, MOMENT, and SLOAD.

The second category consists of the following: FORCE1, FORCE2, MOMENT1, and MOMENT2. The three FORCE entries differ only in the manner in which the direction of the force is specified. FORCE uses the components of a vector. FORCE1 uses two grid points, not necessarily the same as the loaded grid points. FORCE2 defines the direction of the force as a vector which is the vector product of two other vectors. The distinctions between the three MOMENT entries are similar. These loads are follower loads in nonlinear and produce follower stiffness terms.

Distributed Loads

The distributed loads consist of two categories: those which are stationary in direction through out the analysis and those which follow the element motion. The first category consists of the PLOAD1.

The second category consists of the following: PLOAD, PLOAD2, and PLOAD4.

Loads may be applied to the interiors of finite elements. The PLOAD1 entry defines concentrated and linearly distributed forces and moments to the CBEAM element. However, the stress calculation for nonlinear CBEAM element ignores the distributed loads, but lumps them onto the nodal points. The PLOAD entry is exceptional in that it references three or four arbitrarily located grids rather than an element. The PLOAD2 entry defines a uniform static pressure load applied to the CQUAD4 or CTRIA3 elements. The PLOAD4 defines a pressure load on surfaces of CHEXA, CPENTA, CTRIA3, and CQUAD4 elements which need not be normal to the surface. When normal to the surface, the PLOAD4 is a follower load.

Mass Related Loads

The mass related loads are gravity and centrifugal loads, which require specification of mass data. The gravity loads which can be specified in the GRAV Bulk Data entry are stationary in direction through out the analysis. It is used to define the direction and magnitude of the gravity vector in some identified coordinate system. The components of the gravity vector are multiplied by the mass matrix to obtain the components of the gravity force at each grid point.

The centrifugal loads to be specified in the RFORCE Bulk Data entry is used to define the components of a spin vector which is used internally to compute centrifugal forces. Each component of the spin vector is multiplied by a scale factor.

Thermal Loads

Thermal loads are selected in each subcase by the Case Control command TEMP(LOAD). Thermal loading which is a follower loading is accomplished through the following entries: TEMP, TEMPD, TEMPP1, TEMPP3, and TEMPRB. The TEMP and TEMPD provide for grid point temperature specification which are then interpolated to points within elements. For the nonlinear CQUAD4 and CTRIA3 elements, the TEMPP1 provides for an average temperature and thermal gradient in the thickness direction and the TEMPP3 provides for temperature at points through the thickness. The TEMPRB provides for average temperature at ends A and B for the CROD and CBEAM element. Average temperature specified directly for an element will take precedence over the temperature interpolated from the elements connected grid points.

The Case Control TEMP(INIT) is required above the subcase level. For nonlinear static analysis TEMP(MAT) or TEMP(BOTH) are not applicable. The use of TEMP(MAT) or TEMP(BOTH) will cause a fatal error. Subcases that do not contain a TEMP(LOAD) will default to TEMP(INIT), which implies no thermal load. Thermal loads are often used to provide initial strains.

Loads Combination

Various load specifications can be combined using LOAD and (CLOAD, LSEQ).

The LOAD provides for the linear combination of static load sets. The static loads (LOAD, FORCE, etc.) applied to the upstream superelements cannot be referenced by a Case Control command LOAD. The CLOAD entry is designed to apply static loads to upstream superelements by combining loads defined in LSEQ entries. The (CLOAD, LSEQ) combination allows the nonlinear algorithms to apply in an incremental fashion on the upstream superelement loads to the boundary of the residual structure.

The CLOAD defines a static load combination for superelement loads acting on residual boundary and the LSEQ defines a static load combination for superelement upstream loads. The LSEQ assigns load vectors to the superelements and numerically labels them via the DAREA field entry value. The LID and TID field entries point to the appropriate load and temperature distribution entries. The CLOAD defines a static load as a linear combination of previously calculated superelement loads defined by the LSEQ. The IDVi fields correspond directly to the numeric labels defined via the DAREA fields of the LSEQ entries.

The CLOAD must be selected in the residual solution subcase via the case control CLOAD entry. Only one LOADSET may appear in Case Control. The hierarchy of the loads data is shown schematically in Figure 3-28. An example Case Control Section is shown below:

```
                :
SEALL=ALL $ Required in SOL 106 only

LOADSET = 1000 $ Selects LSEQ 1000 for upstream loads
SUPER = ALL $ Identify SEs to process
.
.
.
DISP = ALL
.
.
.
.
.
.
.
.
.
$ NONLINEAR SOLUTIONS FOR RESIDUAL SUPERELEMENTS
SUBCASE 10
CLOAD = 1001 $ Refers to CLOAD Bulk Data
NLPARM = 12 $ Iteration control
.
.
.
SUBCASE 20
CLOAD = 1002
LOAD = 10 $ Residual SE load
NLPARM = 22
.
.
.
```

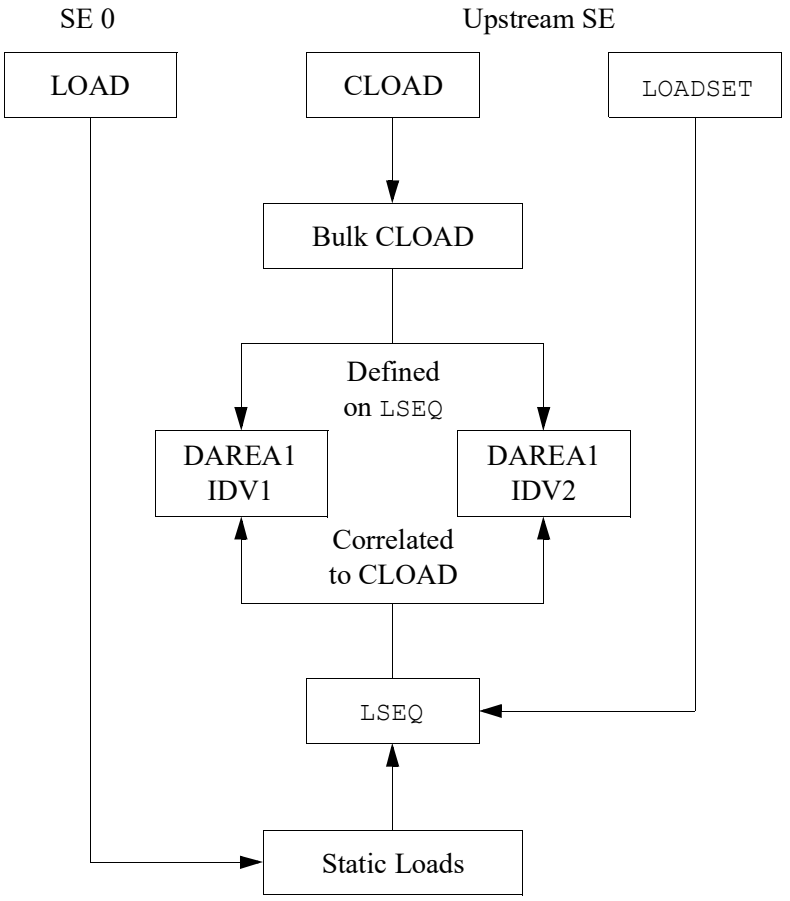


Figure 3-28 Hierarchy of Load Data

Restarts for Nonlinear Static Analysis

Any restart in SOL 106 must have the FMS command RESTART and the two parameters PARAM,SUBID,value and PARAM,LOOPID,value.

The SUBID is the sequential number of a subcase. It is recommended that the SUBID value always be incremented by one and point to a new subcase. The new Subcase should contain any standard subcase commands required.

The LOOPID is the identifier of the converged solution from which the restart is to take place.

Any converged and saved solution always gives the messages:

```
*** USER    INFORMATION MESSAGE 6186,
     *** SOLUTION HAS CONVERGED ***

      SUBID 7  LOADINC 5  LOOPID 67  LOAD STEP  7.000  LOAD FACTOR  1.00000
```

```
^^^ DMAP INFORMATION MESSAGE 9005 (NLSTATIC) -  THE SOLUTION FOR LOOPID=  67
    IS SAVED FOR RESTART
To restart from the above run, the restart might look like:
RESTART
TIME 10000
SOL 106
DIAG 50
CEND
TITLE = SAMPLE RESTART
PARAM,SUBID,8
PARAM,LOOPID,67
SPC = 200
MPC = 250
SET 1 = 1118
DISPLACEMENT = 1
SPCF=1
SUBCASE 1
LOAD = 101
NLPARM = 51
   .
   .
   .
SUBCASE 7
LOAD = 107
NLPARM = 57
SUBCASE 8
LOAD = 108
NLPARM = 58
BEGIN BULK
NLPARM  58      5               AUTO    20      20      PW      YES
        1-3     1-3     1-7     -3
SPCD    108     1118    3       0.025
FORCE   108     1118            0.      0.      0.      1.
ENDDATA
```

For a full data recovery restart after the completion of a complete analysis proceed as above but do not actually add the new Subcase in Case Control. For example, assume the analysis has ended in subcase 8 above with the following messages:

```
*** USER    INFORMATION MESSAGE 6186,
     *** SOLUTION HAS CONVERGED ***

      SUBID 8  LOADINC 5  LOOPID 72  LOAD STEP  8.000  LOAD FACTOR  1.00000

 ^^^ DMAP INFORMATION MESSAGE 9005 (NLSTATIC) -  THE SOLUTION FOR LOOPID= 72
     IS SAVED FOR RESTART
```

Perform a full data recovery restart as follows:

```
RESTART
TIME 10000
SOL 106
DIAG 50
CEND
TITLE = FULL DATA RECOVERY RESTART
MAXLINES = 10000000
$ POINT TO A NEW SUBCASE BUT DO NOT ACTUALLY ADD A NEW SUBCASE
```

```
PARAM,SUBID,9
PARAM,LOOPID,72
SPC = 200
MPC = 250
SET 1 = 1118
$DISPLACEMENT = 1
$ ADD DISPLACEMENT AND STRESS REQUESTS FOR DATA RECOVERY
DISPLACEMENT=ALL
STRESS=ALL
SPCF=1
SUBCASE 1
LOAD = 101
NLPARM = 51
    .
    .
    .
SUBCASE 7
LOAD = 107
NLPARM = 57
SUBCASE 8
LOAD = 108
NLPARM = 58
BEGIN BULK
ENDDATA
```

Note that in the above, there is no SUBCASE 9 added.

The result of this last restart will be the complete displacement and stress results output for each converged solution saved on the data base starting with the first converged solution and ending with the results of the converged solution at LOOPID = 72.

Nonlinear Modal Analysis

User Interface

Prestressed normal modes can be requested at the end of each subcase in SOL 106 by adding the following items to the input file.

- Add METHOD=SID in the subcase of interest. A prestressed normal mode analysis will then be performed at the end of this subcase. The METHOD command points to an EIGRL or EIGR Bulk Data entry which then selects an eigenvalue method.
- For multiple normal mode analyses in SOL 106, the METHOD command may appear in more than one subcase or above all subcases. In the latter case, a normal mode analysis is performed at the end of each subcase. The stiffness used for modal analysis corresponds to the last step of the subcase. Modal analysis cannot be performed at intermediate solution step.
- A PARAM,NMLOOP,loopid command must appear in the Case Control or Bulk Data Section to request a normal mode analysis at the end of those subcases with a METHOD command. The actual value of loopid is unimportant as long as it is a positive integer. This alleviates the cumbersome task of figuring out the exact loopid.
- Add the appropriate EIGRL or EIGR entry in the Bulk Data Section of the *MSC Nastran Quick Reference Guide*.

Example (TURBINE1.DAT)

The following example is a normal mode analysis of a prestressed turbine blade (see Figure 3-29). The blade is modeled with CHEXA elements and loaded with centrifugal forces (RFORCE). The default iteration strategy is selected on the NLPARM Bulk Data entry. MSC Nastran automatically updates the stiffness matrix at the end of the subcase if a METHOD Case Control command appears in the subcase.

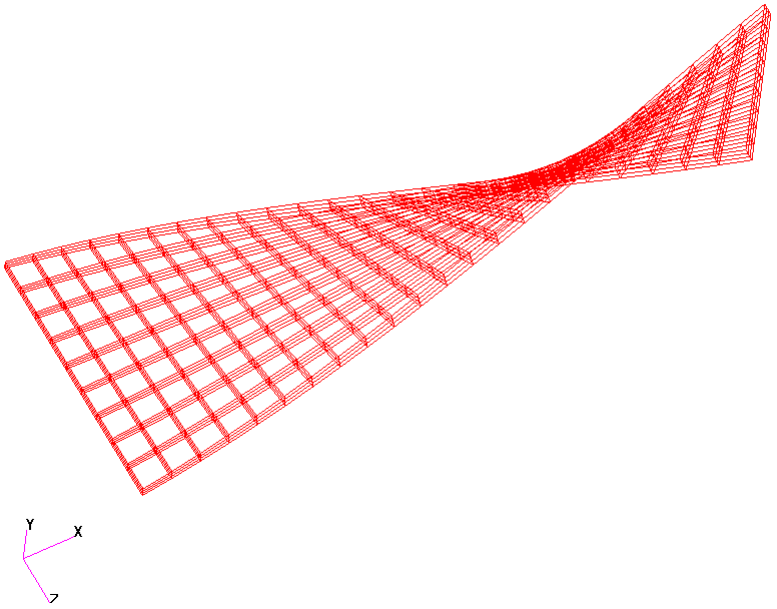


Figure 3-29 Prestressed Modes of a Turbine Blade

An excerpt of the input file is shown.

```
SOL106
DIAG8
CEND
TITLE = Turbine Blade
  ECHO  = SORT,PUNCH
  SPC= 200
  LOAD= 300
  NLPAR= 400
  METHO= 500
  DISP(PLOT) = ALL
  ELST(PLOT) = ALL
  NLSTRESS   = NONE
BEGIN BULK
PSOLID100100
MAT1100     1.+7            .3      1.
RFORCE  300                     .5                      1.      2
NLPARM400     1                             200                     +N
+N            1.-6    1.-12                                         +B
+B      0
EIGRL500                      6
PARAMCOUPMASS1
PARAMLGDISP1
PARAMNMLOOP1
PARAMPOST0
$
CHEXA   1       100     10000   10001   10101   10100   11000   11001   +M00000
+M00000 11101   11100
GRID    19324           44.     4.5     -.2             456
SPC1    200     123     10000   10100   10200   10300   11000   11100   +M00648
+M00648 11200   11300   12000   12100   12200   12300   13000   13100   +M00649
+M00649 13200   13300   14000   14100   14200   14300   15000   15100   +M00650
+M00650 15200   15300   16000   16100   16200   16300   17000   17100   +M00651
+M00651 17200   17300   18000   18100   18200   18300   19000   19100   +M00652
+M00652 19200   19300
$
$    PLUS THE REST OF THE MODEL
$
ENDDATA
```

Restart from SOL 106 into SOL 103 or into Another Linear Solution Sequence

For the purpose of a prestressed normal modes analysis, the old way of restarting from SOL 106 into SOL 103 is no longer necessary; the user can, instead, restart from a SOL 106 run into another SOL 106 run to perform the prestressed normal modes calculation.

Restarts from SOL 106 into linear solution sequences are not recommended to the novice user because of several limitations. The results of the linear restart are incremental values with respect to the preload, not total values. However, some experienced users restart from SOL 106 into SOL 103 to perform prestressed modal analysis with changing boundary conditions, or restart into another linear solution sequence to perform a perturbed linear solution on a preloaded structure. In the SOL 106 cold start run, KMETHOD='ITER' or 'AUTO' and KSTEP=1 must be specified on the NLPARM Bulk Data entry in order to restart from the end of the subcase. With these parameters, all nonlinear information is stored on the database so that the nonlinear tangent stiffness can be recovered in the linear restart run. The following command must be included in the linear restart run:

- PARAM,NMLOOP,loopid in the Case Control or Bulk Data Section, where loopid is the load step id at the end of a subcase from the SOL 106 cold start run.

Restarts are also allowed at intermediate steps within a subcase. The parameter INTOUT on the NLPARM entry must be set to 'YES' or 'ALL' in the SOL 106 cold start run to store intermediate steps on the database for restart purposes. However, the follower force stiffness may not be updated at intermediate steps. Therefore, the user should restart from the end of a subcase if the effect of follower force stiffness is important.

Differential Stiffness and Follower Force Stiffness in Linear and Nonlinear Solution Sequences

For prestressed normal mode analysis, it is useful to know which stiffness contributions are included. The table below summarizes the action of the parameters LGDISP and FOLLOWK in SOL 106 as compared to the action of FOLLOWK in linear solution sequences. In some cases, the default settings for linear and nonlinear solution sequences are not the same. In nonlinear static analysis (SOL 106), for the default of LGDISP=-1 (geometric linear), the parameter FOLLOWK (follower force stiffness) is ignored. For LGDISP=1, FOLLOWK is set to YES by default. In the linear solution sequences, FOLLOWK=YES is the default, see Case No. 3 in Table 3-13.

In theory, the follower force stiffness is unsymmetric. The linear solution sequences always symmetrize the follower force stiffness. By default, SOL 106 uses the symmetrized follower force stiffness to improve performance. If the user wants to use the unsymmetric follower force stiffness, the parameter FKSYMFAC can be specified for this purpose. See FKSYMFAC (p. 829) in the *MSC Nastran Quick Reference Guide* for further details. The use of the symmetrized follower force stiffness is recommended because it gives sufficient accurate results with substantial performance improvement.

In SOL 129, follower forces are included, however, the follower force stiffness is never calculated.

Table 3-13 Differential Stiffness and Follower Force Stiffness Contribution

	Parameters		SOL 106 Cold Start or Restart		Restart from SOL 106 into SOL 103	SOL 103 Cold Start Using STATSUB
	LGDISP	FOLLOWK	Nonlinear Element Force	Stiffness	Stiffness	Stiffness
1	1	YES	L+M+D+F	L+M+D+F	L+M+D+F	L+D'+F'
2	1	NO	L+M+D+F	L+M+D	L+M+D	L+D'
3	-1	YES	L+M	L+M	L+M+D'+F'	L+D'+F'
4	-1	NO	L+M	L+M	L+M+D'	L+D'
5	2	YES	L+M+D	L+M+D+F	L+M+D+F	L+D'+F'
6	2	NO	L+M+D	L+M+D	L+M+D	L+D'

where:

L	=	linear part of element force or stiffness
M	=	material nonlinear part of element force or stiffness
D	=	differential or geometric nonlinear part of element force or stiffness
D'	=	differential part of stiffness for small deformations

F	=	follower force part of element force or stiffness
F'	=	follower force part of stiffness for small deformations

All other linear solution sequences (for example, SOL 105, 108, etc.) have the same default as SOL 103.

Buckling Analysis Enhancement

An accurate solution to a buckling problem requires more meticulous efforts than just following a numerical procedure. There are a number of factors to consider before a buckling solution can be accepted with confidence. Such points along with a few examples are discussed by comparing solutions obtained from linear buckling analysis with preloads and nonlinear buckling with/without unsymmetric stiffness matrix.

A starting step for a nonlinear buckling analysis should be a linear buckling analysis.

Theory

Let us start with a linear equation of motion for a preloaded structure, i.e.,

$$M\ddot{u} + C\dot{u} + Ku + K^d u = P(t)$$

where M, C, K and K^d are mass, viscous damping, material and differential stiffness matrices, and $P(t)$ is a forcing function in time domain. The differential stiffness is created by the initial stress due to preloads, and it may include the follower stiffness if applicable. Ignoring the damping term to avoid complex arithmetic, an eigenvalue problem may be formulated as

$$[(K + K^d) - \omega^2 M]\{\phi\} = \{0\}$$

which is a governing equation for normal mode analysis with a preload, but it could be recast for a dynamic buckling analysis at a constant frequency. For a static buckling, that constant frequency of vibration is zero and the inertia term drops out, i.e.,

$$[K + K^d]\{\phi\} = \{0\}$$

in which $\{\phi\}$ represents virtual displacements, or buckling modes shapes. A non-trivial solution exists for an eigenvalue that makes the determinant of $[K + K^d]$ vanish, which leads to an eigenvalue problem

$$[K + \lambda K^d]\{\phi\} = \{0\}$$

where λ is an eigenvalue which is a multiplier to the applied load to attain a critical buckling load. If there exists constant preloads other than the buckling load in question, the above equation should include additional differential stiffness; that is,

$$[K + K^d_{preload} + \lambda K_{buckle}]\{\phi\} = \{0\}$$

in which differential stiffness is distinguished for constant preload and variable buckling load. Notice that no eigenvalue solutions are meaningful if the preload makes the structure buckle, i.e., $[K + K^d_{preload}]$ should be positive definite. The buckling analysis with an excessive preload can render wrong solutions, unless the positive definiteness is ensured.

Example - prld_buck.dat

An example of input data for a linear buckling analysis with a preload is shown below:

```
ID plate105, v2001 $    SHL 2/23/01
DIAG 8, 15 $
SOL 105
TIME 60
CEND
TITLE = MSC Nastran job created on 20-Feb-01
   SPC = 2   $ clamp the bottom edge
   DISP=ALL
SUBCASE 10
   SUBTITLE= preload tensile edge force 300 lb/in
   LOAD = 2
SUBCASE 20
   Subtitle= top pressure 100 lb/in.
   LOAD = 7
SUBCASE 30
  SUBTITLE=buckling analysis
  STATSUB(preload)=10
  STATSUB(buckle)=20
  METHOD=1
BEGIN BULK
PARAM    POST    0
EIGRL  1       0.      6.     3     0              1.
  .
  .
  .
ENDDATA
```

This is an example of a plate model (5 in. by 10 in., 0.1 in thick, steel, clamped on the bottom edge) with lateral tension by an edge load (in x) of 300 lb/in for a preload and vertical edge load (in y) of 100 lb/in for a buckling load. Alternatively, the preload and the buckling load may be switched to perform a similar analysis. The first buckling mode is a bending mode at vertical load of -64.55 lb/in, as shown in Figure 3-30. The second buckling mode in bending at -593 lb/in is also shown in Figure 3-30, but it is of no physical significance and shown here for verification purposes only. If the lateral load is in compression instead of tension, the plate could buckle in a twist mode at -123.29 lb/in in the absence of a vertical load, as shown in Figure 3-31. A parametric study, conducted with various combinations of lateral and vertical loads, resulted in a buckling envelope (or locus) on a loading plane as shown in Figure 3-32. The buckling shape is a bending mode above the curve and a twist mode in the left side of the curve. Bifurcation behavior can be observed in the transitional region near the corner point, where either the twist or bending mode could appear as the first and the second mode. Lower-right side of the curve is the safety zone.

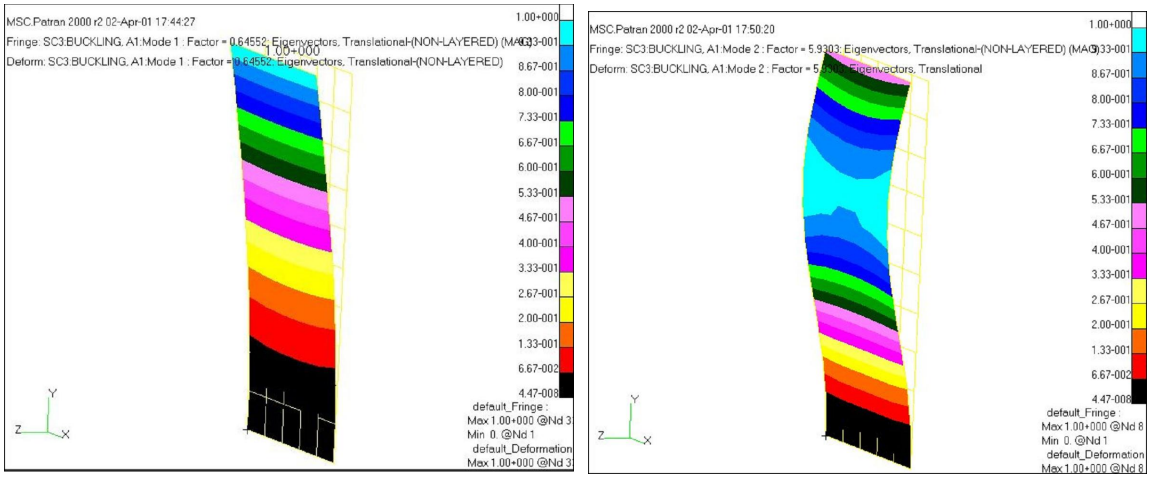


Figure 3-30 Panel Buckling Shapes for the First Two Bending Modes

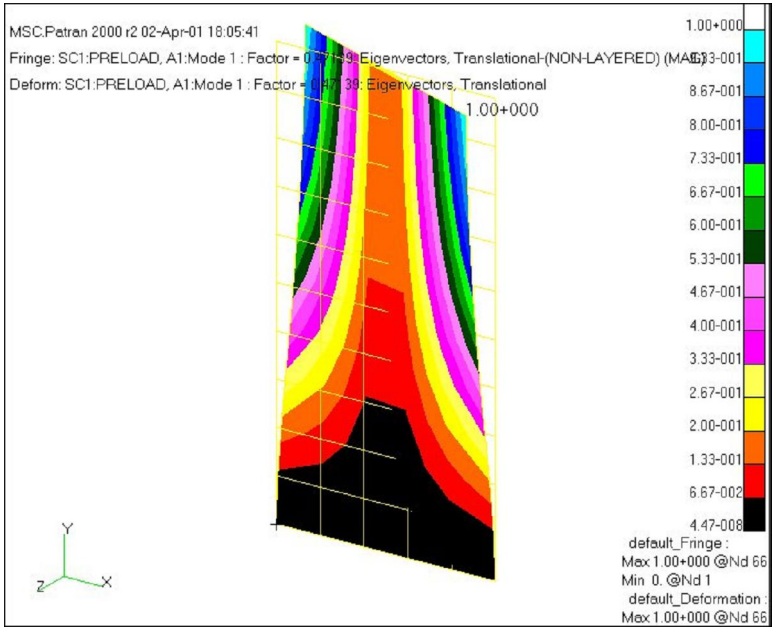


Figure 3-31 Panel Buckling in Twist Mode Under Lateral Compression

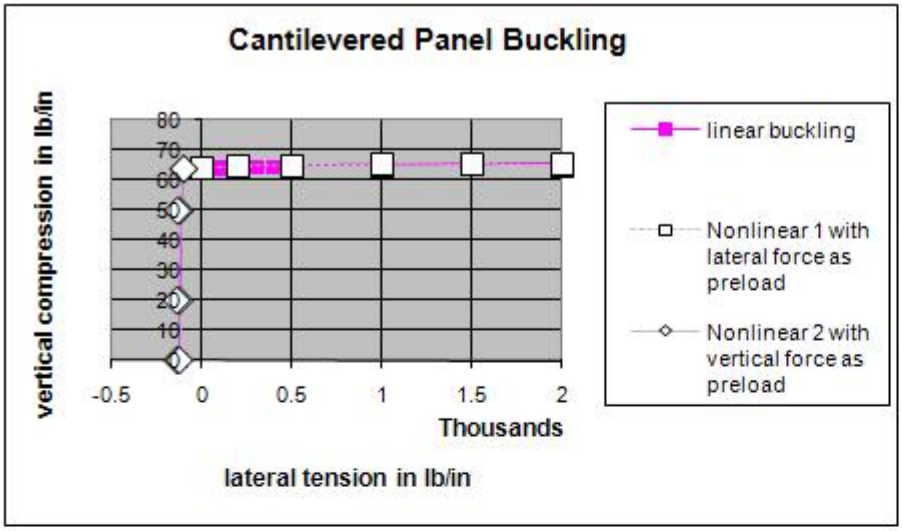


Figure 3-32 Cantilevered Panel Buckling Envelope

Nonlinear Buckling Analysis with Symmetric Stiffness Matrix

Nonlinear buckling analysis capability has been available in SOL 106 by restarts using PARAM, BUCKLE, 1. While serving the purpose of a nonlinear buckling analysis following a static nonlinear analysis, this buckling analysis procedure is cumbersome to the user because it requires a restart. To provide a more convenient and versatile capability for nonlinear buckling analysis, an option for a buckling analysis in a cold-start run with PARAM, BUCKLE, 2 was introduced in MSC Nastran Version 70.7. This option allows a buckling analysis in any Subcase that has the METHOD command specified for an eigenvalue analysis.

There are two more ways in nonlinear analysis to estimate the critical buckling load. One method is the arc-length method that can provide solutions past the critical buckling load into the post-buckling state. Using the arc-length method, the nodal displacement of a point with the most noticeable motion should be traced to illustrate the peak point. The applied load at this peak resembles the critical buckling load. Another method is to use Newton's method until the solution cannot be obtained due to divergence, in which case the adaptive bisection method is activated in the vicinity of the critical buckling load and stops at the limit load, very close to the critical buckling point. A similar method was applied to the nonlinear dynamic stability analysis using adaptive bisection algorithm in time domain for buckling or bifurcation prediction in the implicit direct-time integration.

The nonlinear buckling analysis procedure in MSC Nastran allows buckling analysis with preloads. However, the nonlinear procedure follows a different numerical procedure. Therefore, it could provide a means for validation of linear solutions. A parametric study using nonlinear buckling analysis rendered data points for the panel buckling problem, labeled “Nonlinear 1 and 2” in Figure 3-32, using the lateral force and vertical force as preloads, respectively. Although two different methods have been used in the nonlinear buckling prediction: namely, (1) eigenvalue estimation by extrapolation and (2) detection of the critical load by consecutive bisections with no recourse due to divergence, all the nonlinear solution points fell, by

and large, on the same curve. This figure also shows that the problem is basically linear, because the "Large Displacement" option introduced no visible effects.

Nonlinear Buckling Analysis with Unsymmetric Stiffness Matrix

Follower forces usually create unsymmetric follower stiffness unless the follower forces are completely enclosed, exerting zero resultant forces. This asymmetry condition should be addressed with caution. The follower stiffness in SOL 106 was introduced in Version 70.7, where symmetrization was applied conservatively so that it is activated in rare occasions. The logic was modified in MSC Nastran 2001 to always symmetrize the follower stiffness by default. Some problems exhibited convergence difficulties as a drawback of symmetrizing follower stiffness. The symmetrization does not show adverse effects until the follower stiffness effect becomes dominant under large displacements. In MSC Nastran 2004, the symmetrization is modified to be activated adaptively, detecting the severity of the asymmetry.

For adaptive symmetrization of the follower stiffness, we need a criterion to determine the extent of asymmetry. This can be achieved by normalizing the norm of unsymmetric part of the follower stiffness by a stiffness matrix that is independent from follower stiffness. The differential stiffness was chosen for this purpose after some numerical experiments. Then, the ratio of asymmetry is compared with a tolerance, which is represented by a user parameter FKSYMFAC so that the user can control the process. The default value for FKSYMFAC is determined to be 0.024 by a parametric study, allowing symmetrization of the follower stiffness during the stiffness update process if the iterative process can render converged solutions. If the asymmetry ratio is greater than FKSYMFAC, symmetrization process will not be activated. The asymmetry ratio increases as the geometric nonlinearity intensifies, which is confirmed by numerous test runs.

When the follower stiffness becomes significantly asymmetric, the real eigenvalue analysis would be invalid or erroneous. The eigenvalue analysis with unsymmetric matrices requires complex arithmetic with left- and right-handed eigenvectors. To use the complex eigenvalue analysis, you must specify CMETHOD instead of METHOD in the Subcase and the CMETHOD should reference EIGC Bulk Data entry. The eigenvalues obtained from the complex eigenvalue analysis for buckling with asymmetric stiffness are complex numbers, of which imaginary numbers become zeros. All the eigenvectors are also real numbers. By comparing results from both real and complex eigenvalue analysis, it is confirmed that the real eigenvalue analysis renders wrong solutions to unsymmetric stiffness for nonlinear buckling. Particularly, the mode shapes are erroneous with real eigenvalue analysis from unsymmetric problem, whereas the discrepancy in the critical buckling load is not pronounced.

The buckling shape should be examined graphically to fully understand the physical phenomenon. Fortunately, the complex eigenvalue analysis produces real eigenvectors from real and unsymmetric matrices. However, the buckling shapes displayed with eigenvectors currently are with reference to the undeformed shape, and do not visualize the real buckling shape if the deformed shape at the onset of the buckling has very large displacements.

Example

An example problem is a cantilevered beam model subject to an end force that is a follower force; that is, `FORCE1`. The eigenvalue analysis is performed at load factor 4, where the asymmetry ratio is 0.069, which produced buckling shapes shown in Figure 3-33 (obtained by complex method) and Figure 3-34 (obtained by real eigenvalue method). They are different and the real eigenvalue method is likely to be in error. The display tool should be improved so that the buckling shapes be plotted with an eigenvector as a perturbation with reference to the deformed shape to fully perceive the physics.

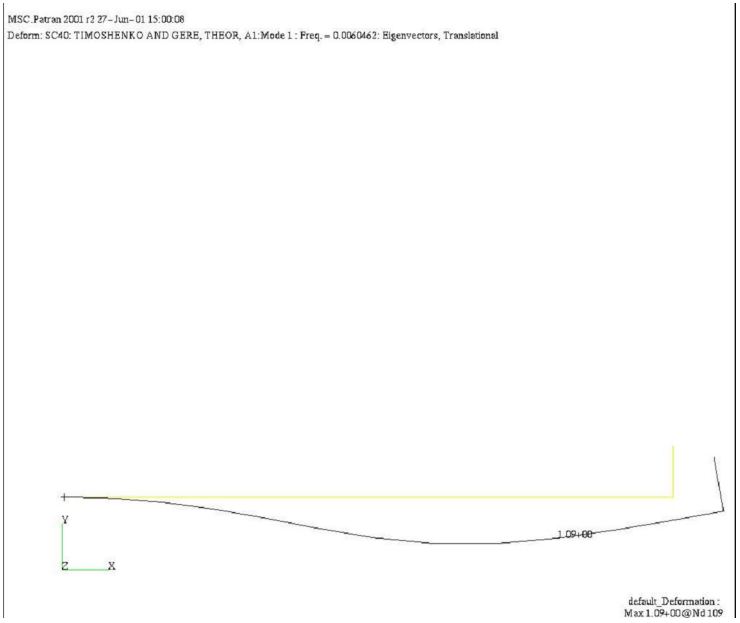


Figure 3-33 Buckling Shape by Complex Method at L.F.=4

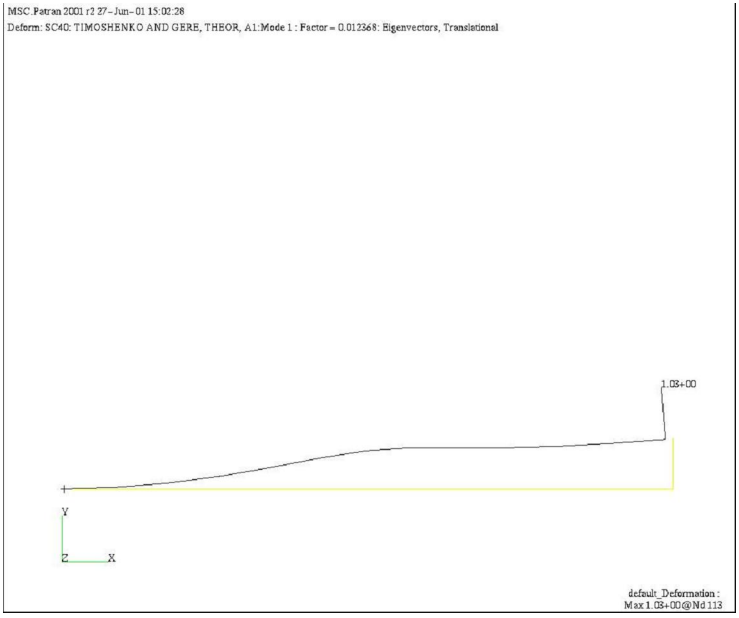


Figure 3-34 Buckling Shape by Real Eigenvalue Analysis at L.F.=4

An Example of Cylinder Buckling

A hollow cylinder is used to demonstrate an interesting behavior in buckling. Its geometric attributes include: 50 in. in diameter, 100 in. in height and 0.125 in. wall thickness with closed ends of 0.5 in. in thickness. The cylinder, made of steel, is subjected to a combined load of external pressure on the cylindrical surface and vertical edge force on the top, while the bottom is fixed in z.

The cylinder could buckle in two different modes depending on the combination of loads as shown in Figure 3-35 and Figure 3-36, where the first mode with vertical creases appear when the external pressure is predominant (showing a star-shaped cross-section at the mid-span) and the second mode with horizontal creases appear when the vertical edge force is more dominant. A failure curve (or buckling envelope) can be constructed by a parametric study using various magnitudes of the preloads with external pressure or vertical edge load, as shown in Figure 3-37. The first buckling shape appears above the curve and the second buckling in the right side of the curve. A bifurcation occurs at the corner, where either buckling mode may occur. The cylinder is free from buckling in the lower-left region of the curve. It is noted that the buckling resistance against the vertical load does not increase even if the pressure direction is reversed to an internal pressure (creating a tensile hoop stress), although the buckling load increases against the external pressure as the vertical compression decreases.

Nonlinear buckling solution points are added to the curve rendered by linear buckling solutions, where three categories of nonlinear solutions are presented. Solution points in square symbol are from the eigenvalue analysis using an extrapolation scheme, and those in diamond symbol are from the limit process using consecutive bisections upon divergence during the incremental/iterative processes. Although the second buckling loads in the right side of the curve predicted by nonlinear analyses are almost identical to the linear solutions, nonlinear solutions for the first buckling shape above the curve are somewhat dispersed. In the absence of an exact solution for a reference, these nonlinear solutions seem plausible to indicate that the geometric nonlinearity has significant effects on the buckling of the first kind with vertical creases, while the geometric nonlinear effect is negligible in the second kind. These solutions, however, were obtained with symmetrized stiffness matrix. The solution points in triangular symbol show tremendously different trend in the upper side of the curve, where the buckling is caused by the external pressure.

The pressure load creates follower stiffness, which is unsymmetric in this problem. The solutions in triangular symbol were obtained with unsymmetric matrices. By preserving asymmetric stiffness matrices, the solutions converged further and the predicted buckling loads were higher. The buckling analysis with asymmetric matrices (but real-valued) requires complex eigenvalue analysis, although the solution turns out to be real-valued. Furthermore, an ideal cylinder without the end condition does not buckle under pressure because of the follower stiffness effect. That is why the solutions with asymmetric stiffness converge to much higher external pressure, and the complex eigenvalue analyses with extrapolation at the last converged steps render solution points shown by triangles. Solutions may continue for increased pressure and it has been observed that the predicted buckling point increases as the eigenvalue analysis is performed at higher pressure. Two of the buckling modes of these solutions are shown in Figures 3-38 and 3-39, which are rather spurious and fortuitous. However, the deformed shape of the second kind of buckling as shown in Figure 3-40 was obtained by a limit process using adaptive bisections upon divergence, which depicts a genuine buckling shape. It also signifies that the nonlinear eigenvalue analysis based on extrapolation is an approximation and it can vary appreciably depending on where the extrapolation is applied.

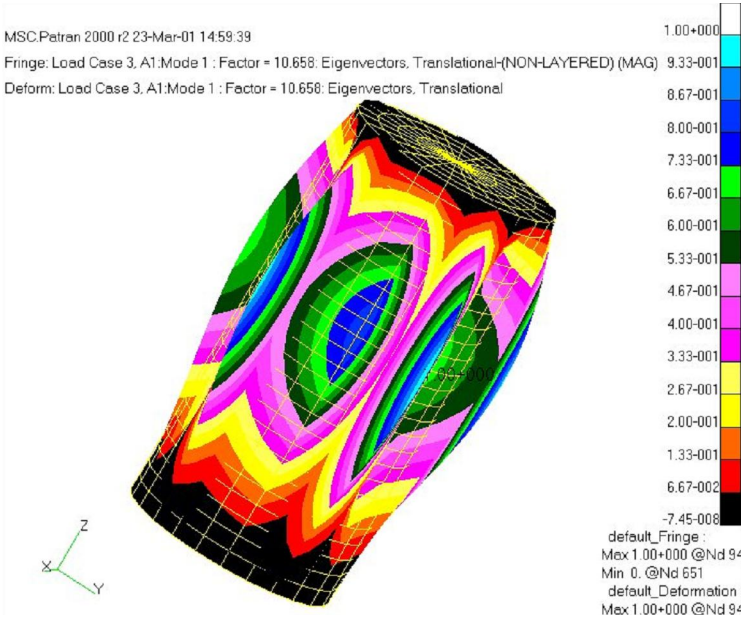


Figure 3-35 Cylinder Buckling Shape 1 With Vertical Creases

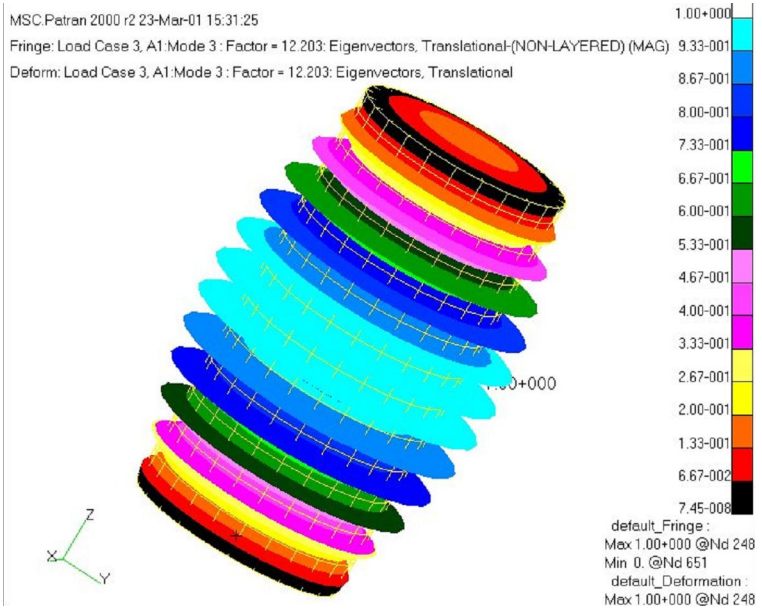


Figure 3-36 Cylinder Buckling Shape 2 with Horizontal Creases

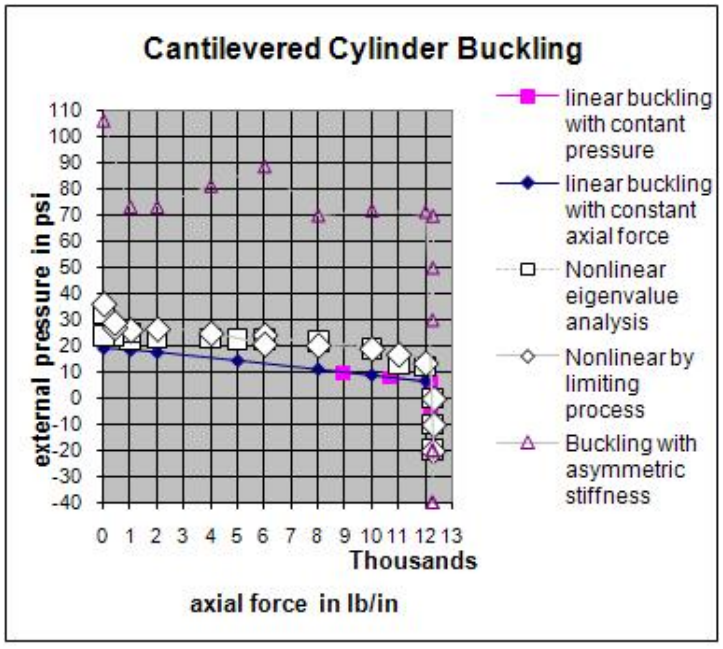


Figure 3-37 Cylinder Buckling Envelope (With Nonlinear Solutions)

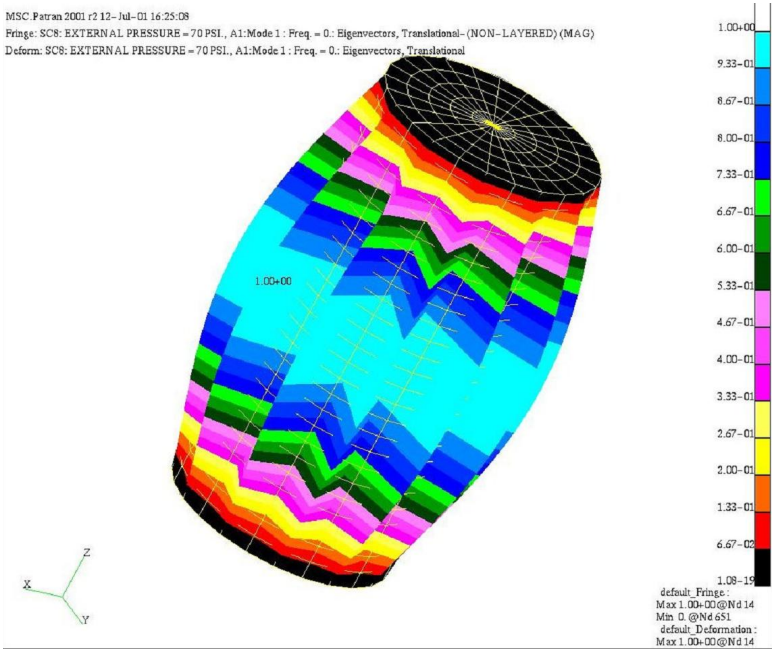


Figure 3-38 Buckling Shape at Paxial=2000 and External Pressure=73.44

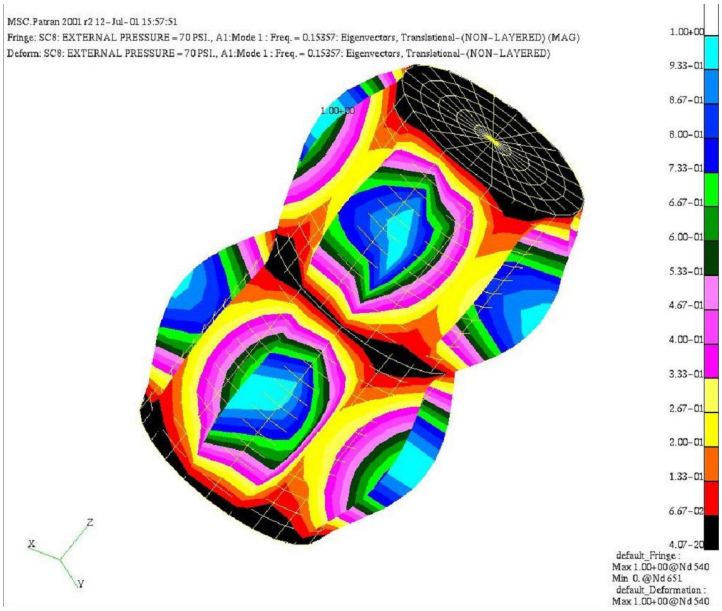


Figure 3-39 Buckling Shape at Paxial=10000 and External Pressure=72.02

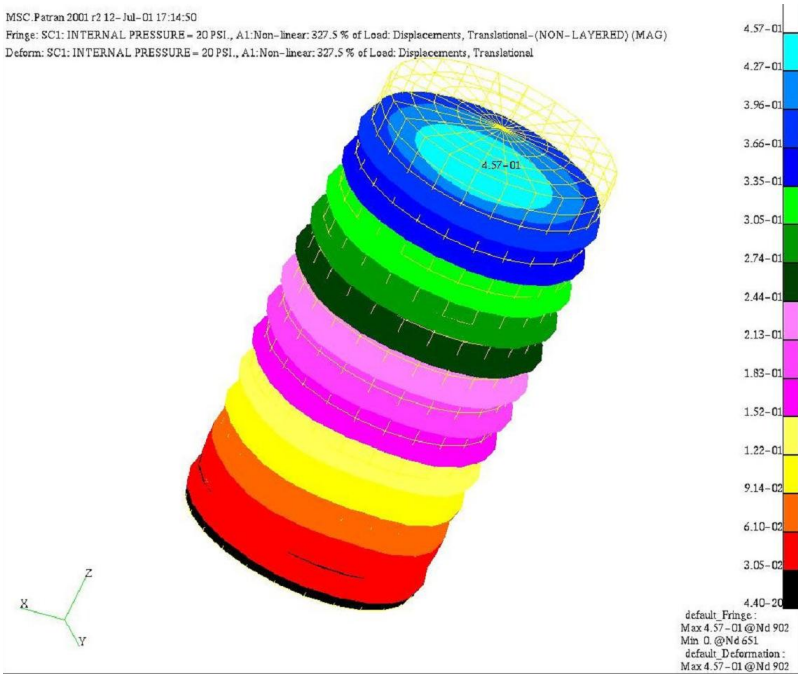


Figure 3-40 Deformed Shape on Buckling at Paxial=12243.4 and Internal Pressure=20

Nonlinear Transient Response Analysis

User Interface

The input data required for SOL 129 is a combination of direct time integration control data, similar to SOL 109 (for direct linear transient with superelements), and nonlinear modeling data similar to SOL 106 (for nonlinear statics). The nonlinear properties and/or effects are defined by nonlinear material data (MATS1, MATHP, and TABLES1), gap elements (GAP) or slideline contact (BCONP, BLSEG, and BFRIC) for nonlinear interface, and PARAMeter LGDISP for geometric nonlinearity. The transient effects are produced by time-dependent loading functions (TLOADi, DAREA, LSEQ, etc.), damping (parameters, elements and material data), and mass properties. Massless degree of freedom should be avoided for numerical stability and the small damping is also recommended.

The unique data required for SOL 129 is supplied on the TSTEPNL Bulk Data entry. The TSTEPNL entry in itself is a combination of the TSTEP entry for direct time integration and the NLPARM entry for nonlinear iteration control. Restarts are controlled by parameters (LOOPID, STIME, SLOOPID, and SDATA) which can be specified either in the Case Control Section or Bulk Data Section. Some optional parameters (TSTATIC, NDAMP) are provided for additional control or capabilities.

Case Control

Each subcase defines a time interval starting from the last time step of the previous subcase and the interval is subdivided into small time steps. The output time is labeled by the cumulative time, including all previous subcases. There are advantages to divide the total duration of analysis into many subcases so that each subcase does not have excessive number of time steps; for example, not to exceed 200 steps. The data blocks containing solutions are generated at the end of each subcase to store in the database for output process and restarts. As such, converged solutions are apt to be saved at many intermediate steps in case of divergence and more flexible control becomes possible with multiple subcases.

The input loading functions may be changed for each subcase or continued by repeating the same DLOAD command. However, it is recommended to use the same TLOADi Bulk Data for all the subcases in order to maintain the continuity between subcases, because TLOADi entries define the loading history as a function of cumulative time. Static loads (PLOADi, FORCEi, MOMENTi) may be associated with time-dependent functions by the LSEQ Bulk Data entry that can be selected by a Case Control command LOADSET. However, thermal loads or enforced displacements (SPCD) are not allowed in the nonlinear transient analysis. Nonlinear forces as function of displacements or velocities (NOLINi) may be selected and printed by Case Control commands NONLINEAR and NLLOAD, respectively. Each subcase may have a different time step size, time interval, and iteration control selected by the TSTEPNL request. The Case Control commands which may not be changed after the first subcase are: SPC, MPC, DMIG, and TF.

Output requests for STRESS, SDISPLACEMENT, SVELOCITY, SACCELERATION, and NONLINEAR may be specified within a subcase. DISPLACEMENT, VELOCITY, ACCELERATION, OLOAD, FORCE, and SPCFORCE must be specified above the subcase level. However, FORCE output and GPSTRESS output are not available for nonlinear elements.

Initial conditions (displacement or velocity) can be specified by the Bulk Data TIC selectable by the Case Control command IC. If initial conditions were given, all of the nonlinear element forces and stresses must be computed to satisfy equilibrium with the prescribed initial conditions. On the other hand, initial conditions could be a part of the nonlinear analysis by applying static analysis for the preload using PARAM, TSTATIC in the first subcase. Then the transient analysis can be performed in the ensuing subcases. Associated with the adaptive time stepping method, the PARAMeter NDAMP is used to control the stability in the ADAPT method. The parameter NDAMP represents the numerical damping (recommended value for usual case is 0.01) which often required to improve the stability and convergence in the contact problems.

All the superelement model generation options and matrix reduction options are allowed for the linear portion of the structure. The generalized dynamic reduction, component mode synthesis, and Guyan reduction may be performed for upstream superelements. The residual superelement may contain scalar degree of freedom representing linear modal formulations.

Limitations

- No constraint set charges are allowed between subcases or in restarts.
- No initial preloads such as closed gaps or terminal loads are allowed in a cold start. See PARAM,LOOPID under “Restarts” in this section.
- Overall structure damping (PARAM,G) is not provided in nonlinear elements. Use the damping field in the material (MATi) data entry and the PARAMeter W4 to supply damping based on the linear properties.

- Element forces are not computed for nonlinear elements. Output requests will be ignored.
- When multiple subcases are specified, it is recommended that a single time function (see the TLOADi Bulk Data entry description) be assigned to an applied load throughout the entire time span of the solution. This time function can be used for all subcases. If different time functions are to be used between subcases, ensure that the time functions are continuous between subcases.
- Element stresses for nonlinear elements are computed and printed during the nonlinear iteration phase only. They are not merged with those of the linear elements.

Direct Time Integration

If the effects of inertia and damping forces are to be included in the nonlinear analysis, the nonlinear transient response is analyzed by a step-by-step integration of an equation of motion, i.e.,

$$M\ddot{u} + C\dot{u} + F(u) = P(t) \tag{3-1}$$

By direct numerical integration, the equilibrium of equation (3-1) is satisfied at discrete time steps with an interval of Δt. The equilibrium is ensured by iterations until the solution converges to the preset error tolerance.

Procedures, such as the quasi-Newton update and line search process, provide users with additional options to adjust incremental and iterative processes in search of more efficient and effective strategy for obtaining solutions. A self-adaptive time stepping method that adjusts time step automatically during analysis is available.

Newmark's direct time integration method is implemented using the two-point recurrence (or one-step) formula as a foundation of this self-adaptive time stepping algorithm. The optimal time step size, which is required for accuracy and efficiency, changes continuously in the transient dynamic environment. The primary concept of automatic time step adjustment is that the proper size of the time step can be predicted based on the dominant frequency in the incremental deformation pattern at the previous time step. This concept presents a deficiency of time lag involved in the prediction process. Furthermore, changes in nonlinearity cannot be predicted from the deformation pattern at the previous time step. This deficiency is overcome by the bisection process, which is activated when any difficulties arise in terms of convergence during the iteration.

Frequent decomposition of the dynamic stiffness matrix is inevitable in adaptive time stepping using the implicit integration method. Efforts have been made to minimize the thrashing in the algorithm (for example, too frequent or repetitive adjustment of the time step size) and yet to provide a solution reliably to any type of problem. The iteration process for the equilibrium employs expeditious methods such as BFGS updates and line searches as well as an adaptive stiffness matrix update strategy. The present self-adaptive time stepping algorithm for the nonlinear transient analysis is proving to be a robust and practical method, aimed at efficiency, effectiveness, and user friendliness.

The transient response analysis capability is provided in the self-contained solution sequences, SOL 129. Although this solution sequences are developed for nonlinear transient response analysis, linear transient response analysis can be performed in these solution sequences in order to take advantage of the automatic time stepping and restart capabilities. The core parts of SOL 129 are the nonlinear modules (NLTRD2 and NLTRD) which perform the incremental and iterative processes for implicit direct time integration.

Implicit Integration Control: TSTEPNL Data Entry

The input fields of the TSTEPNL Bulk Data entry specify the time step size, the number of steps, and the output interval as well as the nonlinear iteration options. The TSTEPNL Bulk Data is selectable by the Case Control command TSTEPNL. Although the same TSTEPNL Bulk Data may be selected by more than one subcase, it is recommended to select different TSTEPNL entry for each subcase in preparation for changes in the restarts.

The choice of time step size is determined primarily by the frequency content of the input load and the highest frequency mode-of-interest. A general guideline is that seven or more steps per cycle be provided for reasonable accuracy. Modes with shorter periods (higher frequency) will be attenuated by the numerical process. Highly nonlinear problems may require smaller step size. However, the adaptive time stepping capability is the vital part of 129 and is recommended for any problem (linear or nonlinear). Nevertheless, the initial time step size should be estimated by the user according to the aforementioned principles.

A caution is necessary in using the automatic time step adjustment if the forcing function is a short duration pulse. Since the automatic time step adjustment algorithm does not consider the loading history, short pulses could be skipped if the time step is automatically adjusted to a larger value than the pulse duration. It is advised that a drastic change in the time step size between subcases be avoided. The drastic change (for example, a ratio exceeding 1000) could cause a loss of accuracy upon subcase switch. In such case, an intermediate subcase should be provided for a transition period of short interval to reduce the ratio.

The parameters for controlling the accuracy and stability of the incremental and iterative process can be specified in the TSTEPNL Bulk Data entry. The controls are applicable to the automatic time step adjustment and bisection process in addition to stiffness matrix updates, BFGS updates and line searches similar to those on the NLPARM Bulk Data entry. Since default values are resulted from numerous test runs, the analysis should be started with the default setting and changed if necessary. The TSTEPNL data format is shown below with default values:

Format:

1	2	3	4	5	6	7	8	9	10
TSTEPNL	ID	NDT	DT	NO		KSTEP	MAXITER	CONV	
	EPSU	EPSP	EPSW	MAXDIV	MAXQN	MAXLS	FSTRESS		
	MAXBIS	ADJUST	MSTEP	RB	MAXR	UTOL	RTOLB		

Example With Defaults:

1	2	3	4	5	6	7	8	9	10
TSTEPNL	250	100	.01	1		2	10	PW	
	1.E-2	1.E-3	1.E-6	2	10	2	0.2		
	5	5		0.75	16.0	0.1	20.		

The TSTEPNL Bulk Data entry is selected using ID by the Case Control command TSTEPNL. Each subcase (residual superelement solutions only) requires a TSTEPNL entry. Multiple subcases are assumed to occur sequentially in time. Therefore, the initial conditions of each subcase are defined by the end conditions of the previous subcase.

The NDT field specifies the number of time steps with DT as the size of each time step. The total duration for the subcase can be assessed by multiplying NDT with DT (that is, $\text{NDT} \cdot \text{DT}$). The time increment ($\Delta t$) remains constant during the analysis in AUTO and TSTEP options, and is equal to DT. However, the time increment (Δt) changes during the analysis in the ADAPT option and the actual number of time steps will not be equal to NDT. In the ADAPT option, DT is used as an initial value for Δt.

The NO field specifies the time step interval for output, i.e., every NOth step solution is saved for output. The data will be output at steps 0, NO, ..., etc., and the last converged step for printing and plotting purposes. The Case Control command OTIME may also be used to control the output points.

The program automatically adjusts the incremental time and uses the bisection algorithm in case of divergence. During the bisection process the stiffness is updated at every KSTEPth successful bisection. The stiffness matrix is always updated for a new subcase or restart, irrespective of the option selected.

The number of iterations for a time step is limited to MAXITER. If the solution does not converge in MAXITER iterations, the process is treated as a divergent process; that is, either a bisection or stiffness matrix update takes place based on the value of MAXBIS. The sign of MAXITER provides a control over the ultimate recourse (reiteration) in case of failure in convergence or bisection. If the MAXITER is negative, the analysis is terminated when the divergence condition is encountered twice during the same time step or the solution diverges for five consecutive time steps. If MAXITER is positive, the program computes the best attainable solution and continues the analysis on second divergence.

The convergence test is controlled by convergence test flags (U for displacement error test, P for load equilibrium error test, W for work error test) and the error tolerances (EPSU, EPSP and EPSW) which define the convergence criteria. All requested criteria (combination of U, P, and/or W) are satisfied upon convergence. It should be noted that at least two iterations are necessary to check the displacement convergence criterion.

The MAXDIV field provides control over diverging solutions. Depending on the rate of divergence, the number of diverging solutions (NDIV) is incremented by 1 or 2. The solution is assumed to be divergent when NDIV reaches MAXDIV during the iteration. If the bisection option is used, the time step is bisected upon divergence. Otherwise, the solution for the time step is repeated with a new stiffness based on the converged state at the beginning of the time step. If NDIV reaches MAXDIV twice within the same time step, the analysis is terminated with a fatal message.

The BEGS quasi-Newton updates and the line search process work in the way as in static analysis except for the default setting. The MAXQN field defines the maximum number of quasi-Newton vectors to be saved on the database, and the MAXLS defines the number of line searches allowed per iteration. Nonzero values of MAXQN and MAXLS activate the quasi-Newton update and the line search process, respectively.

The FSTRESS field defines a fraction of the effective stress ($\bar{\sigma}$) which is used to limit the subincrement size in the material routine. The number of subincrements in the material routines is determined such that the subincrement size is approximately $\text{FSTRESS} \cdot \sigma$ (equivalent stress). FSTRESS is also used to establish a tolerance for error correction in the elasto-plastic material, i.e.,

$$Error \text{ in yield function} < \text{FSTRESS} \cdot \bar{\sigma}$$

If the limit is exceeded at the converging state, the program will exit with a fatal error message. Otherwise, the stress state is adjusted to the current yield surface.

MAXBIS is the maximum number of bisections allowed for each time step ($-9 \le \text{MAXBIS} \le 9$). The bisection process is activated when divergence occurs and $\text{MAXBIS} \ne 0$. The number of bisections for a time increment is limited to |MAXBIS|. If MAXBIS is positive and the solution does not converge after MAXBIS bisections, the best solution is computed and the analysis is continued to the next time step. If MAXBIS is negative and the solution does not converge in |MAXBIS| bisection, the analysis is terminated.

The parameter ADJUST allows the user to control the automatic time stepping in the ADAPT option. A value of zero for ADJUST turns off the automatic adjustment completely. If ADJUST is positive, the time increment is continually adjusted for the first few steps until a good value of Δt is obtained. After this initial adjustment, the time increment is adjusted every ADJUST time steps only. A value of ADJUST an order greater than NDT will turn off adjustment after the initial adjustment. Since the automatic time step adjustment is based on the mode of response and not on the loading pattern, it may be necessary to limit the adjustable step size when the period of the forcing function is much shorter than the period of dominant response frequency of the structure. It is the user's responsibility to ensure that the loading history is properly traced with ADJUST option. The ADJUST option should be suppressed for the duration of short pulse loading. If unsure, start with DT which is much smaller than the pulse duration in order to properly represent the loading pattern.

MSTEP defines the desired number of time steps to obtain the dominant period response accurately ($10 \le \text{Integer} \le 200$). RB defines bounds for maintaining the same time step for the stepping function in the automatic time step adjustment method ($0.1 \le \text{Real} \le 1.0$). Parameters MSTEP and RB are used to adjust the time increment during the analysis in the ADAPT option. The adjustment is based on the number of time steps desired to capture the dominant frequency response accurately. The time increment is adjusted as follows:

$$\Delta t_{n+1} = f(r)\Delta t_n$$

where:

$$r = \frac{1}{\text{MSTEP}}\left(\frac{2\pi}{w_n}\right)\left(\frac{1}{\Delta t_n}\right)$$

with:

$$f = 0.25 \text{ for } r < 0.5 \cdot RB$$

$$f = 0.5 \text{ for } 0.5 \cdot RB < r < RB$$

$$f = 1.0 \text{ for } RB < r < 2$$

$$f = 2.0 \text{ for } 2. \le r < 3./RB$$

$$f = 4.0 \text{ for } r > 3./RB$$

The recommended value of MSTEP for nearly linear problems is 20. A larger value (for example, 40) is required for highly nonlinear problems. In the default option, the program automatically computes the value of MSTEP based on the changes in the stiffness.

The MAXR field defines the maximum ratio for the adjusted incremental time relative to DT allowed for time step adjustment (1.0 ≤ Real ≤ 32.0). MAXR is used to define the upper and lower bounds for adjusted time step size; that is,

$$MIN\left(\frac{DT}{2^{MAXBIS}}, \frac{DT}{MAXR}\right) \le \Delta T \le MAXR \cdot DT$$

The UTOL field defines the tolerance on displacement increment below which there is no time step adjustment (0.001 < Real ≤ 1.0). UTOL is used to filter undesirable time step adjustment; that is, no time step adjustment is performed if

$$\frac{\|\dot{U}_n\|}{\|\dot{U}_n\|_{max}} < UTOL$$

The RTOLB field defines the maximum value of incremental rotation (in degrees) allowed per iteration to activate bisection (Real > 2.0). The bisection is activated if the incremental rotation for any degree of freedom ($\Delta\theta_x$, $\Delta\theta_y$ or $\Delta\theta_z$) exceeds the value specified for RTOLB. This bisection strategy based on the incremental rotation is controlled by MAXBIS field.

Iteration Related Output Data

During the incremental and iterative computation, the process information consisting of iteration data is printed at the end of each iteration or time step. They are printed under the following heading:

TIME	Cumulative time for the duration of the analysis.
ITER	Iteration count for each time step.
DISP	Relative error in terms of displacements defined as $E_U^i = \lambda^i \frac{\|u^i - u^{i-1}\|}{(1-\lambda^i)u_{max}}$ for ADAPT method $E_U^i = \frac{\|u^i - u^{i-1}\|}{(1-\lambda^i)u_{max}}$ for AUTO or TSTEP where $u_{max} = max(\|u_1\|, \|u_2\|, \ldots, \|u_n\|)$
LOAD	Relative error in terms of load vectors defined as $E_p^i = \frac{\|R^i\|}{max(\|F_n\|, \|P_n\|)}$ for ADAPT method $E_p^i = \frac{\|R^i\|}{max(\|\bar{F}\|), \|\bar{P}\|}$ for AUTO or TSTEP where $\bar{F}$ and $\bar{P}$ are three-point average values for internal and external forces, respectively.
WORK	Relative error in terms of work defined as $E_w^i = \frac{\{u^i - u^{i-1}\}^T R^i}{max(\{u_n\}^T\{F_n\}, \{u_n\}^T\{P_n\})}$ (3-2)

LAMBDA(I)	Rate of Convergence in iteration where $\lambda^i = \frac{E_p^i}{E_p^{i-1}}$ (3-3)
LAMBDA(T)	Ratio of the load error for two consecutive time steps computed only for AUTO or TSTEP method $\lambda_n^n = \frac{E_p^n}{E_p^{n-1}}$ (3-4)
LAMBDA-BAR	Average of LAMBDA(T) over the last 3 steps, computed only for AUTO or TSTEP method.
DLMAG	Absolute norm of the load vector, $\|R\|$. The absolute convergence is defined using DLMAG by $\|R\| < 10^{-12}$.
FACTOR	Final value of the line search parameter.
E-FIRST	Divergence rate, error before line search.
E-FINAL	Error at the end of line search.
NQNV	Number of quasi-Newton vectors appended.
NLS	Number of line searches performed during the iteration.
ITR DIV	Number of occurrences of divergence detected during the adaptive iteration by the iteration module NLTRD2.
MAT DIV	Number of occurrences of bisection conditions in the material routine (excessive stress increment) or in the rotation angle (excessive rotation) during the iteration using the ADAPT method.
NO. BIS	Number of bisections executed for the current time interval.
ADJUST	Ratio of time step adjustment relative to DT within a subcase.

Restarts

Since SOL 106 and SOL 129 share the same database storage for nonlinear tables and matrices, the restart system for transient analysis can use either a previous static or transient nonlinear analysis as its initial conditions. A restart from SOL 129 into SOL 106 is not available.

Restarting from SOL 106 into SOL 129

The options for a restart from SOL 106 into SOL 129 are static to static and static to dynamic analysis. For a restart from a previous static analysis, only the first subcase is affected. Simply provide a database created in SOL 106 (or 129) and specify the parameter PARAM,SLOOPID,N where N is the printed value of LOOPID for the desired static solution. The initial transient load should be identical to static loads at the restart state. Constraint sets, direct input matrices, mass, and damping may be changed.

Restarting within SOL 129

Restarting within SOL 129 allows static to static, static to dynamic, and dynamic to dynamic analysis. Restarts from a previous nonlinear transient execution are available for a number of cases. If the same model is to be re-executed, only the residual superelement needs to be reassembled. If the final results from the previous transient run are to be used as the initial conditions at t = 0, add N dummy SUBCASE commands to start the residual Case Control execution and STIME = 0.

The normal restart for a transient run is to be continued from the last step of a previous subcase with different loads and/or TSTEPNL data. For the normal restart provide the following parameters:

LOOPID = N:	Start from the Nth subcase
STIME = t:	Start from time t

Note that constraint sets should not be changed to avoid incompatible matrix sizes. The values of LOOPID and STIME, which are printed with the iteration information for each subcase, can be directly read from the printout of the previous run. If the adaptive time stepping algorithm is not activated, the value of t may also be calculated by the following equation:

$$t = \left(\sum_{i=1}^{N} NDT_i \right) \cdot DT_i$$

where NDTi and DTi are the number of time steps and the time increment of the i-th subcase, respectively.

If a SOL 129 run is terminated abnormally in the middle of a subcase, it may or may not be restartable depending upon the cause of the abnormal exit. If the job is stopped due to a diverging solution, it can be restarted either from the end of a previous subcase or from the last saved solution step. The restart procedure for the former is identical to that for the normal restart as described in the preceding paragraph. The latter case also requires parameters LOOPID and STIME, however, the input value for STIME differs depending on the value of METHOD specified on the TSTEPNL entry. If METHOD = AUTO or TSTEP (the NLTRD module), STIME is the time corresponding to the last output step which may be calculated based on the output skip factor (that is, the NO on the TSTEPNL entry). If METHOD = ADAPT (NLTRD2 module) the last converged solution is treated as an output step and is always saved for the restart so that STIME can be the time of the last converged step. The values of STIME and LOOPID can also be found in the printout, if the ADAPT method is used. Once STIME and LOOPID are known, determine the number of remaining time steps in the subcase and create a new TSTEPNL entry for the remaining time. Insert a new subcase which references the new TSTEPNL entry prior to the remaining subcases in the Case Control Section.

A solution may be terminated in the middle of a subcase due to insufficient CPU time: (1) the CPU time specified in the Executive Control Section is insufficient so the run is forced to exit by the program, or (2) the CPU time exceeds the limit specified in the computer system which leads to a sudden job abortion by the system. In the first case, MSC Nastran is able to detect the specified CPU time in the Executive Control Section and automatically activate the wrapping-up procedure before time expiration. When completed, the solution can be restarted from the termination point as in the solution diverging case. In the second case the solution can only be restarted from the end of a subcase.

Restarts may also be performed solely for data recovery by providing the following parameters:

SDATA = –1 : Recover data without running the solution module LOOPID = N : from the 1st through the N-th subcases.

Note that solution sets DISP, VELO, ACCE, `OLOAD`, SPCF (printout and plotting) and `NLLOAD` (plotting only) are recoverable while STRESS, SDISP, SVELO and SACCE sets are not.

Example

A simply supported elastic beam is subjected to a single cycle of a sinusoidal forcing function at the quarter span as illustrated in Figure 3-41. The beam is modeled using 20 linear CBEAM elements. A stopper (snubber) is present underneath the center of the beam with a clearance of 0.02 inches. This gap is simulated by a displacement dependent force which is active only when the beam is in contact with the snubber. The problem is linear except for the effect of the stopper. The input data for this problem is given in Listing 3-13.

One subcase with 100 time steps and an initial time step size of 0.1 msec was analyzed in the cold start run. Since the adaptive time stepping algorithm was activated, the job ran to completion at 9.7 msec instead of the specified time span, 100 x 0.1 = 10 msec. A new subcase with 600 time steps was added in the first restart run to extend the time span for another 30 msec. The adaptive time stepping method was turned off and the execution time was specified to be 0.8 min. As expected, the run was aborted at 36 msec due to insufficient time. In the second restart run, the execution time was changed to 5.0 min. and another subcase with 164 time steps and a time increment of 0.1 msec was added to extend the analysis time span up to 40 msec. The third restart was simply to recover the DISP, VELO, ACCE, OLOAD, NLLOAD and SPC output. The response at the loading point of the beam is shown in Figure 3-42.

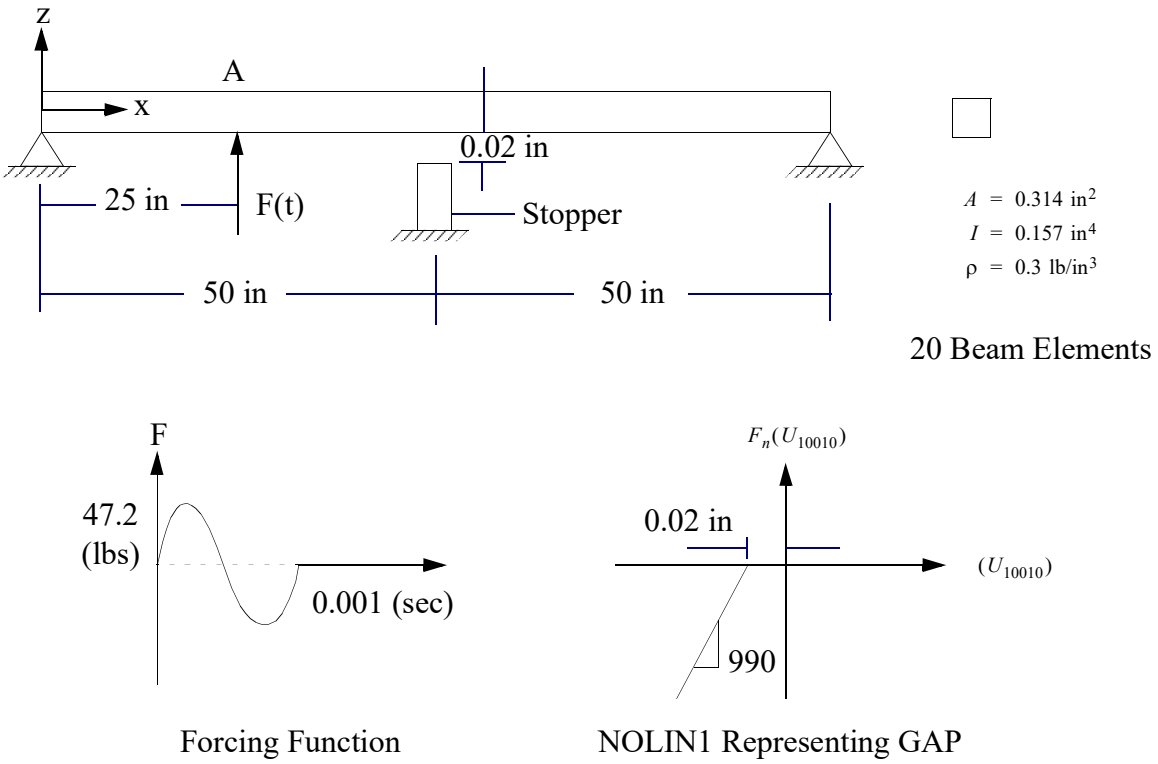


Figure 3-41 Simply Supported Beam With a Restrained Motion

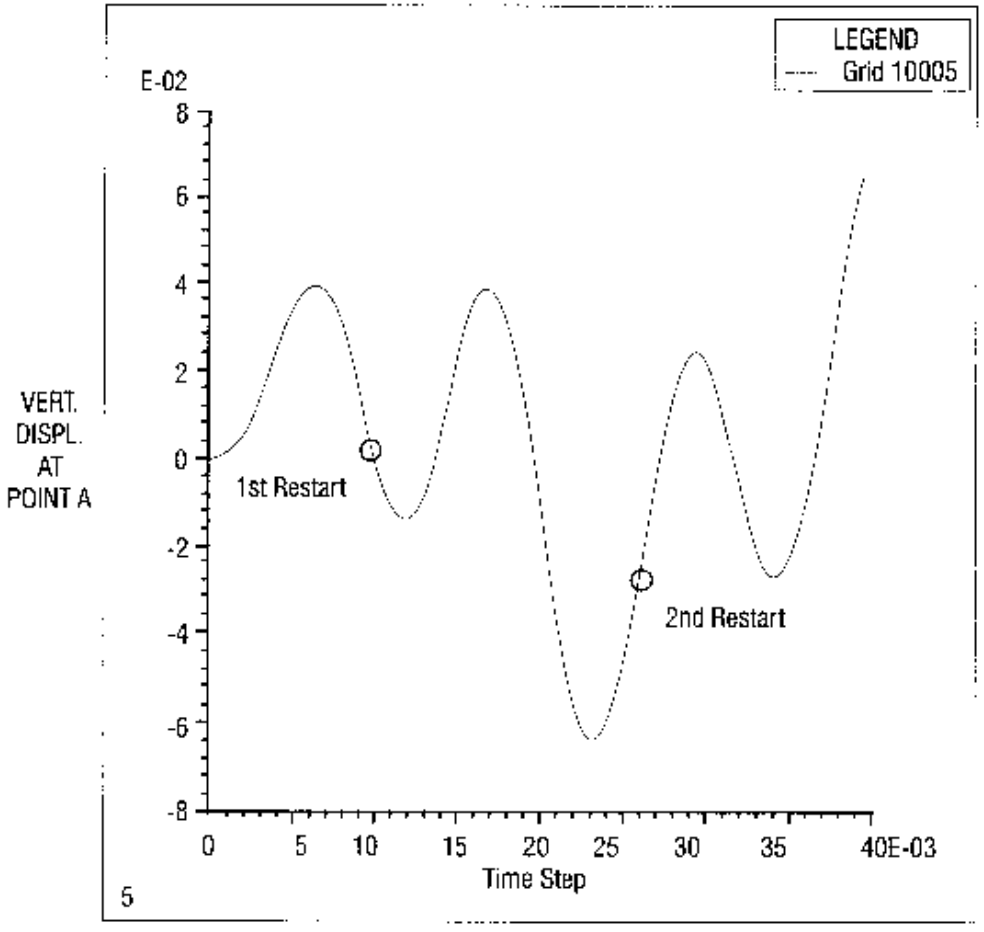


Figure 3-42 Simply Supported Beam: Loading Point Response

Listing 3-13 Input Data Listing for Simply Supported Beam

```
$RESTART$ ***added for the restart runs
ID TNOLIN, V66$ SSH 7/10/91
TIME $$
SOL 129$ NONLINEAR TRANSIENT ANALYSIS
DIAG 8,50$ PRINT MATRIX TRAILERS AND ITERATIONS
CEND
TITLE = DYNAMICS HANDBOOK DEMONSTRATION PROBLEM                    D9918
SUBTITLE = DIRECT TRANSIENT RESPONSE, NONLINEAR FORCE
LABEL = NOLIN IN SOL 129
SEALL = ALL
ECHO = UNSORT
  SET 1 = 10005
  SET 2 = 10010
  SET 3 = 10000
  SPC = 1002
$-----------------------
$PARAM,LOOPID,1*** used for the 1st restart
$PARAM,STIME,0.0093
$-----------------------
$PARAM,LOOPID,2*** used for the 2nd restart
$PARAM,STIME,0.0236
$-----------------------
$  DISPL=2
$  VELO=2*** used for the 3rd restart
$  ACCEL=2
$  OLOAD=1
$  NLLOAD=2
$  SPCF=3
$PARAM,LOOPID,3
$PARAM,SDATA,-1
$-----------------------
SUBCASE 1
  DLOAD= 30
  TSTEPNL= 10
  NONLINEAR= 13$ SELECT NONLINEAR FORCE
SUBCASE 2
  DLOAD= 30
  TSTEPNL= 20
  NONLINEAR= 13$ SELECT NONLINEAR FORCE
SUBCASE 3
  DLOAD= 30
  TSTEPNL= 30
  NONLINEAR= 13$ SELECT NONLINEAR FORCE
BEGIN BULK
$ MODELING INFORMATION FOR BEAM ONLY
CBAR10110010000100010.0.1.
CBAR10210010001100020.0.1.
CBAR10310010002100030.0.1.
CBAR10410010003100040.0.1.
CBAR10510010004100050.0.1:
CBAR10610010005100060.0.1.
CBAR10710010006100070.0.1.
CBAR10810010007100080.0.1.
CBAR10910010008100090.0.1.
CBAR11010010009100100.0.1.
CBAR11110010010100110.0.1.
CBAR11210010011100120.0.1.
CBAR11310010012100130.0.1.
CBAR11410010013100140.0.1.
CBAR11510010014100150.0.1.
```

Listing 3-13 Input Data Listing for Simply Supported Beam (continued)

```
CBAR116100100151001600.0.1.
CBAR117100100161001700.0.1.
CBAR118100100171001800.0.1.
CBAR119100100181001900.0.1.
CBAR120100100191002000.0.1.
CONM21210010.1
GRID1050.-1.
GRID100000.0.0.1246
GRID100015.0.0.1246
GRID1000210.0.0.1246
GRID1000315.0.0.1246
GRID1000420.0.0.1246
GRID1000525.0.0.1246
GRID1000630.0.0.1246
GRID1000735.0.0.1246
GRID1000840.0.0.1246
GRID1000945.0.0.1246
GRID1001050.0.0.1246
GRID1001155.0.0.1246
GRID1001260.0.0.1246
GRID1001365.0.0.1246
GRID1001470.0.0.1246
GRID1001575.0.0.1246
GRID1001680.0.0.1246
GRID1001785.0.0.1246
GRID1001890.0.0.1246
GRID1001995.0.0.1246
GRID10020100.0.0.1246
MAT110003.+7.3.3
PARAMGRDPNT10010
PARAMWTMASS.002588
PBAR1001000.31416.15708 +PB1
+PB1 1.    0.
SPC100210123456
SPC1002100203100003
$
$ MODELING INFORMATION FOR CENTER SPRING
CROD10101010010
MAT11010.0.
PROD10101.
MATS110PLASTIC 0.113.+8
$ LOADING AND SOLUTION INFORMATION
TLOAD230330..01145187.33-90.
DAREA3310005347.2
TSTEPNL10100.00011ADAPT
TSTEPNL20600.000051ADAPT+T521
+T521+T522
+T5220
TSTEPNL30164.000101ADAPT+T531
+T531+T532
+T5320
$
$ MODELING INFORMATION FOR NONLINEAR SPRING
NOLIN1  131001031.10010313
TABLED1 13+TAB
+TAB -2.5-24.95-2.0-20.0.0.ENDT
$
ENDDATA
```

Performance in SOL 129

Nonlinear transient response analysis in SOL 129 has become more robust and reliable. It can be used for linear analysis to take advantage of the automatic time stepping. If the problem is truly linear (that is, without NOLIN) SOL 129 should converge in one iteration at every time step. The computing speed is comparable to SOL 109 in such cases if the same number of time steps are used. The automatic time stepping is based on the dominant response frequency which is based on the system characteristics. This strategy is not effective if the loading changes faster than the system response. In case of an impact loading, an indiscriminate use of automatic time stepping would cause the impulse represented inadequately or unrecognized by the algorithm. This could be avoided by turning off the automatic time stepping. But it relies on the user's intervention and therefore is not user-friendly.

Benefits

SOL 129 will activate time step adjustment unless it is suppressed. As a result, the solution will be more accurate and the solution time is reduced, especially the engineering time will be significantly saved because accurate solution may be attained in the first trial. That is the goal of the automatic and adaptive time stepping method. However, if the model includes NOLIN entries, SOL 129 is likely to activate iteration for equilibrium, while SOL 109 takes one-step solution for each time step. That is because SOL 109 uses explicit approach for the NOLIN process, taking NOLIN forces at the previous time step for the equilibrium of the current time step. The iteration makes SOL 129 slower than SOL 109 when NOLIN entries are present, but the solution is more accurate. The iteration could be avoided by using large tolerances for convergence criteria.

Saving engineering time is by far more efficient even if the computation takes more time. The automatic time stepping strives for a one-shot solution for this reason, and the algorithm takes a rather conservative approach; that is, taking a smaller stride to avoid divergence or convergence difficulties. Particularly, bisection algorithms are devised for accuracy and effectiveness, not for efficiency. If the accuracy can be compromised for efficiency, the adaptive bisections should be turned off. Also larger strides can be achieved by specifying a smaller value for MSTEP field (for example, 10) in the TSTEPNL entry.

The following performance comparison of the analyses of a model with a lot of NOLIN entries shows the contrast in efficiency versus strategy:

Table 3-14

Version	method	tsteps	usertime	systime	elapsed	Remarks
2001	fixed	3200	1787.69	1208.59	50:52	no bisections
2001	auto		1662.74	1187.90	49:20	no bisections
2004	fixed	3200	1520.04	1147.16	45:20	nolin=0, tzeromax=0
2004	fixed	37039	17567.10	10535.30	9:48:34	bisections allowed
2004	auto	4540	1434.06	1228.18	47:42	nolin=0, tzeromax=0
2004	auto	14652	11854.37	5724.78	5:02:58	nolin=1., tzeromax=1
2004	auto	1320	829.84	570.50	23:38	nolin=0., tzeromax=1, mstep=10

Theory

Follower Force Generation

First, the load generation should be performed in the iteration module for efficiency. Direct time integration (in NLTRD2) is to solve the equation

$$M\ddot{u} + C\dot{u} + F(u) = P(t)$$

at every incremental time step. This process involves an integration in time domain combined with an iterative process at each time step to satisfy equilibrium. The iteration is based on an incremental process using

$$\left[\frac{4}{\Delta t^2}M + \frac{2(1-\eta)}{\Delta t}C + \tilde{K}\right]\{\Delta U^{i+1}\} = \{R^i_{n+1}\}$$

with

$$\{R^i_{n+1}\} = \{P^i_{n+1} - F^i_{n+1}\} + \frac{4(1-\eta)}{\Delta t}M\{\dot{U}_n\} + (1-2\eta)\{P_n - F_n\}$$
$$-\left[\frac{4}{\Delta t^2}M + \frac{2(1-\eta)}{\Delta t}C\right]\{U^i_{n+1} - U_n\}$$

where η denotes numerical damping (PARAM, NDAMP).

The current design in MSC Nastran is to generate load vectors in NLTRD2 one column at each time step, so that no update or interpolation is necessary; that is,

$$P^i_t(u) = f_{nf}(t)\cdot\{P_{nf}\} + f_f(t)\cdot\{P^i_f\} + N^i_t$$

where the first term represents non-follower forces, which is to be generated in NLTRD2. Notice that the nonfollower force vectors are formed once per time step, but the follower force vector (the second and the third terms) must be computed at every iteration. It improves not only load-dependent adaptive time stepping but also the memory usage and reduces the I/O operations.

Load-Dependent Time Step Adjustment

The second step is to make the time step adjustment algorithm adaptive to the loading history based on the TLOADi data. The adaptive time step algorithm for `TLOAD1` is implemented by considering time steps specified in the TABLEDi entries. A new time step size is compared against time interval in the TABLED(1-4) at every time step. When adjusted step exceeds the user-specified load increment interval or causes an abrupt changes in load magnitude, the adjusted value is bisected to be consistent with the binary-based time adjustment algorithm. This process will continue sweeping through the time duration for these subcase.

The transient loads specified by `TLOAD2` are analytical functions, continuous and smooth, but the excitation frequency should be considered for adaptive time stepping so that the period (field F on `TLOAD2`) is less than 8 times the incremental time step; that is, bisect if the time step size is larger than 1/8 of a period.

This algorithm activates an adaptive bisection upon load spikes to account for impulsive loads with adequate number of time steps. This modifications work for both structural analysis and heat transfer analysis.

Adaptive Time Stepping for NOLIN and NLRGAP

The third phase makes the time step adjustment algorithm sensitive to the NOLIN data. The NOLIN forces, denoted by $N_t(u, \dot{u})$, are implicit functions of time (except for NLRGAP that could be explicit function of time), which may vary abruptly as a function of time. Then the solution may not render a correct solution with a large time step. The NOLIN entries consists of

- NOLIN1 with TABLEDi: $N1_i = s \cdot T_j(u, \dot{u})$
- NOLIN2: $N2_i = s \cdot X_j \cdot X_k$
- NOLIN3: $N3_i = s \cdot X_j^A$
- NOLIN4: $N4_i = -s \cdot [-X_j]^A$
- NLRGAP: simulates a radial forces on a journal bearing joint

The criteria for the abruptness can be as follows:

$$\Delta t_{n+1} = 0.5 \cdot \Delta t \quad \text{if} \quad |N_{t+\Delta t} - N_t| > F_n \cdot Max(|N_t|, 10^{-4})$$

where the factor F_n is determined based on the numerical experiments (defaulted to 1000). This factor F_n is controlled by system cell 386 (=nolin), i.e.,

$$F_n = 1000./nolin$$

where *nolin* is the alias name of the real-valued sys(386) as follows:

nolin	=	0	bisection is suppressed (same as MSC Nastran 2001)
nolin	=	0.001	increase accuracy slightly
nolin	=	0.01	increase accuracy a little more
nolin	=	1.0	allow full adaptive bisection (default)

Consecutive bisections are allowed (KSTEP+1) times, in which KSTEP is defaulted to 2 in the TSTENL entry.

In order to make a full implementation of NOLIN forces as internal forces, they have to be subtracted from element forces, and thus NOLINs are elements having stiffness and damping. To remedy the automatic time stepping without full implementation as elements, NOLINs need to be considered like an internal force in computing a dominant frequency for a time stepping criterion. The dominant frequency should then be computed by

$$\omega_n^2 = \frac{\{\Delta u_n\}^T \{\Delta F_n - \Delta N_n\}}{\{\Delta u_n\}^T [M] \{\Delta u_n\}}$$

where:

$$\Delta N_n = \{N_n\} - \{N_{n-1}\}$$

The effectiveness for NOLIN-dependent time step adjustment is demonstrated in the two examples below: a vibrating beam with a stopper simulated by NOLIN1 and a thermostat model with a NOLIN1. The result is the accuracy improvement, not efficiency, as shown in the figures. The vibrating beam experiences abrupt changes in the NOLIN1 force (contact) when the GAP opens or closes. The adaptive bisection algorithm detects the transition from open to closed or vice versa. The thermostat model has similar characteristics as a vibrating beam.

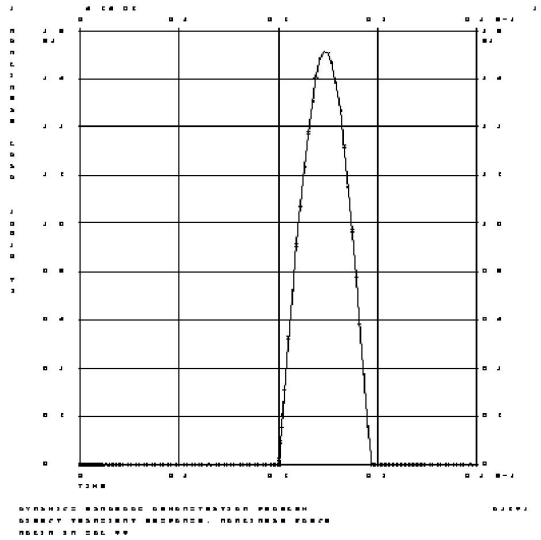


Figure 3-43 Vibrating with a Stopper

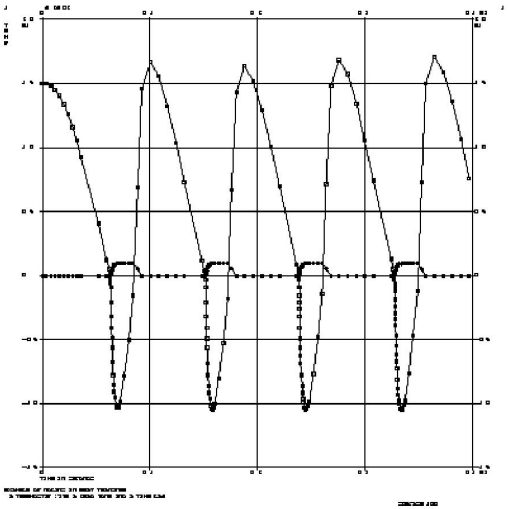


Figure 3-44 Thermal System with a Thermostat

Initial Time Step Adjustment

The adaptive time stepping algorithm in the NLTRD2 module starts with a user-specified value (DT field in the TSTEPNL entry). The algorithm estimates a dominant frequency in the system, from which an appropriate time step size is computed.

However, this process requires two converged time steps. If the process continues without reverting back to time zero, the solution could be erroneous for the first two steps with a large DT. This is particularly true if you specify excessively (two orders of magnitude) large DT as demonstrated in some heat transfer problems.

An initial quadri-section algorithm has been implemented in MSC Nastran to remedy this deficiency. This initial quadri-section is repeated until a right time step size is attained, which is controlled by an integer system cell 373 called TZEROMAX, i.e.,

TZEROMAX > 0	:	maximum number of times to return to time zero
TZEROMAX = 0	:	no initial time step adjustment
TZEROMAX < 0	:	no limit on DT adjustment

with a default value of TZEROMAX=4. When the initial time step adjustment is activated, a UWM (7600) is issued as:

INITIAL TIME STEP SIZE IS REDUCED TO X.XXXE-X.

This algorithm is applicable to both structural and heat transfer analyses, but especially effective for heat transfer analysis. It allows the adjustment of user-specified value of DT by almost three orders of magnitude by default. This initial time step adjustment algorithm accommodates the existing automatic time stepping capability as it currently works. However, the parameter MAXR (in TSTEPNL entry) has been modified to accept any real number without a bound, with a default value of 32.

An example of a thermal conduction process is analyzed with/without initial time step adjustment starting with DT=1.0e-4. The solution without initial time step adjustment shows gross error at the beginning as shown in Figure 3-45. The initial time step adjustment algorithm adjust DT to 1.56e-6 to render the second solution.

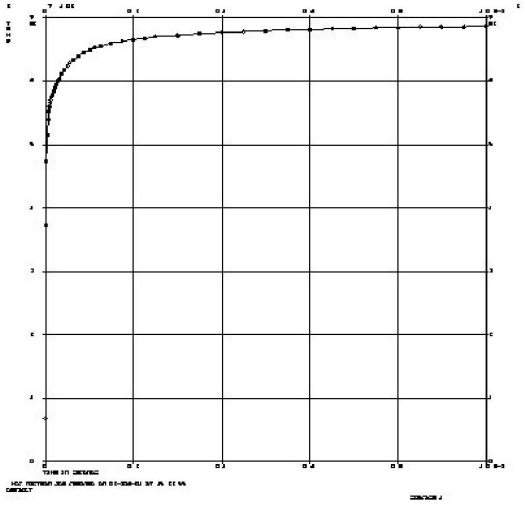


Figure 3-45 Temperature History

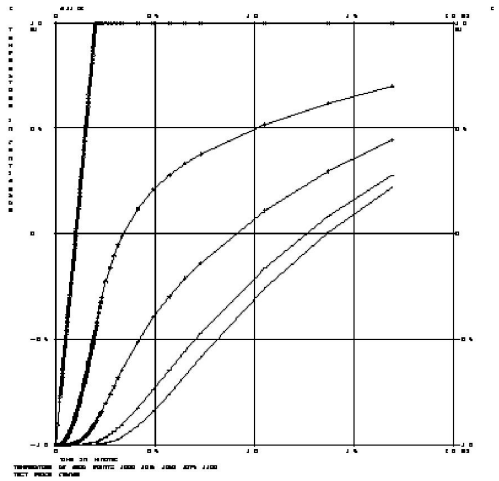


Figure 3-46 Phase Transformation

Nonlinear Analysis With Superelements

In nonlinear analysis with superelements, the nonlinear model must be in the residual model (Superelement = 0). All basic superelement rules apply with the exception that a nonlinear element can not have any of its degrees of freedom in the "o-set".

The unique feature pertaining to load application for superelements are listed in 3-D Slideline Contact Capability, 664 under "Load Combination".

Grid Point Force Balance and Element Strain Energy in Nonlinear Static Analysis

In nonlinear analysis, the strain energy, E, for an element is defined by integrating the specific energy rate, the inner product of strain rate and stress, over element volume and time

$$E = \int_0^t \int_V \dot{\varepsilon}^T \sigma \, dV \, d\tau \tag{3-5}$$

where:

σ = stress tensor

$\dot{\varepsilon}$ = strain rate

V	=	element volume
t	=	actual time in the load history

The integration over time leads to the following recursive formula using the trapezoidal rule

$$E_{n+1} = E_n + \Delta E_{n+1} = E_n + \frac{1}{2}\int \Delta\varepsilon_{n+1}^T(\sigma_n + \sigma_{n+1})dV \tag{3-6}$$

where:

$n+1$	=	current load step
n	=	previous load step
ΔE	=	strain energy increment
$\Delta\varepsilon$	=	strain-increment

The steps n and $n+1$ are converged solution steps.

By integrating equation (3-6) over the element volume, we get

$$E_{n+1} = E_n + \frac{1}{2}\Delta u_{n+1}^T(f_n + f_{n+1}) \tag{3-7}$$

where:

f	=	internal element forces
Δu	=	displacement increment from n to $n+1$

For computational convenience, MSC Nastran uses equation (3-7) to calculate the element strain energy. The internal element forces are readily available in every step because they are needed for the force equilibrium. Note that temperature effects are included in the internal element forces.

When loads from temperature differences or element deformation are present, the default definition of element strain energy for linear elements differ from the definition for nonlinear elements. For linear elements, the element strain energy is defined as

$$E = \frac{1}{2}u^T K_e u - u^T P_{et} \tag{3-8}$$

where P_{et} is the element load vector for temperature loads and element deformation. Equation (3-8) assumes that the temperatures are constant within a subcase. For nonlinear elements, the definition of equation (3-5) is used. In the case of linear material and geometry, equation (3-5) becomes

$$E = \frac{1}{2}u^T K_e u - \frac{1}{2}u^T P_{et} \tag{3-9}$$

Equation (3-9) assumes that the temperature varies linearly within a subcase. The user may request the definition of equation (3-9) to be applied to linear elements by adding PARAM,XFLAG,2 to the input file.

Example

The following MSC Nastran input file represents a simplified model of a mechanical clutch that consists of springs, beams, rigid elements, and gap elements. A geometric nonlinear analysis is performed. Both GPFORCE and ESE output requests are applied above all subcases. This is a good example to show the grid point force balance with both linear and nonlinear elements, that includes:

- applied loads
- element forces
- SPC forces
- MPC forces.

The model is shown in Figure 3-47. For clarity, only the elements are displayed.

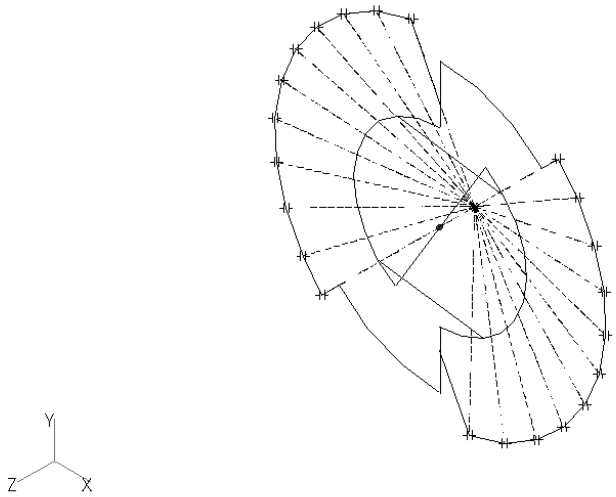


Figure 3-47

The dashed lines represent rigid elements, the rest are gap and beam elements. Spring elements are located at the intersection point of all dashed lines and are not visible in the figure because their connection points are coincident.

Grid Point Force Output

```
1    GPF005B - FORD ELETRO-MECHANICAL CLUTCH                                  SEPTEMBER  26, 2000  MSC Nastran  9/25/00   PAGE   248
      DISPLACEMENT IN MM., FORCE IN NEWTON, STRESS IN MPA
0                                                                                                                      SUBCASE 2
     LOAD STEP =  2.00000E+00
                                        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                  APP-LOAD       .0             .0             .0             .0             .0          -2.400000E+02
          1                  F-OF-SPC       .0             .0             .0             .0             .0           2.400000E+02
          1         10001    BEAM           .0             .0             .0             .0             .0             .0
          1         10002    BEAM           .0             .0             .0             .0             .0             .0
          1                  *TOTALS*       .0             .0             .0             .0             .0             .0
0       100                  F-OF-SPC     -5.366200E-25  -4.514879E-24    .0           4.535559E-23  -6.400176E-24    .0
        100                 F-OF-MPC      5.366200E-25   4.514879E-24    .0          -4.535559E-23   6.400176E-24   1.292178E-24
        100             3    ELAS2          .0             .0             .0             .0             .0          1.070983E-24
        100                  *TOTALS*       .0             .0             .0             .0             .0          2.363161E-24
0       101                  F-OF-SPC       .0             .0            -3.078859E-28  -3.549490E-26   2.535672E-24    .0
        101           101    BEAM         -3.701448E-08   2.907432E-07    .0             .0             .0          -5.474456E-09
        101          2004    BEAM         -3.888788E-07  -1.144153E-07   3.078859E-28   3.549490E-26  -2.535672E-24  -1.777338E-23
        101             1    GAP            1.270000E-10   1.438901E-27    .0             .0             .0             .0
        101                  *TOTALS*     -4.257663E-07   1.763280E-07    .0             .0             .0          -5.474456E-09
0       102           101    BEAM         -2.715682E-08  -2.918291E-07    .0             .0             .0          -2.549396E-09
        102           102    BEAM         -1.637746E-08   1.273060E-07    .0             .0             .0          -1.374512E-25
        102             2    GAP            1.270000E-10  -9.144001E-27    .0             .0             .0             .0
```

Element Strain Energy Output

```
1    GPF005B - FORD ELETRO-MECHANICAL CLUTCH                                  SEPTEMBER  26, 2000  MSC Nastran  9/25/00   PAGE   231
DISPLACEMENT IN MM., FORCE IN NEWTON, STRESS IN MPA
0                                                                                                                      SUBCASE 1
     LOAD STEP =  1.00000E+00
                                        E L E M E N T   S T R A I N   E N E R G I E S

               ELEMENT-TYPE = BEAM               * TOTAL ENERGY OF ALL ELEMENTS IN PROBLEM     =   4.422530E+02
               SUBCASE              1            * TOTAL ENERGY OF ALL ELEMENTS IN SET      -1 =   4.422530E+02
0
                                    ELEMENT-ID          STRAIN-ENERGY          PERCENT OF TOTAL    STRAIN-ENERGY-DENSITY
                                           101          1.211380E+01               2.6928              5.680606E-02
                                           102          1.049874E+00                .2334              4.923244E-03
                                           103          1.323984E-01                .0294              6.208644E-04
                                           104          1.323428E-01                .0294              6.206040E-04
                                           105          1.374425E-01                .0306              6.445181E-04
                                           106          1.407006E-01                .0313              6.597968E-04
                                           107          1.478413E-01                .0329              6.932822E-04
                                           108          1.483829E-01                .0330              6.958218E-04
                                           109          1.032130E-01                .0229              1.936014E-04
                                           110          3.163312E-01                .0703              7.020313E-04
                                           111          4.605607E-01                .1024              1.927717E-03
                                           112          7.067609E-01                .1571              2.958209E-03
                                           113          4.186914E-01                .0931              1.752469E-03
                                           121          3.068841E+00                .6822              2.905759E-02
```

Remarks

1. Linear and nonlinear elements can be mixed. The output for grid point force and element strain energy in nonlinear analysis follow the same format as in linear analysis.
2. The reference system for the grid point force output is the grid point global coordinate system. The user cannot specify another output coordinate system. In nonlinear, the grid point forces are not aligned with element edges. Therefore, PARAM,NOELOF and PARAM,NOELOP are ignored in nonlinear analysis.

3. In nonlinear analysis, the element strain energy must be calculated for each intermediate load step even if the output is requested only in the last load step. To save computations, the element strain energy is only calculated upon user request. The Case Control commands, GPFORCE or ESE must be present to activate grid point force output or element strain energy calculations and output. If the user wants to save the grid point forces and element strain energy on the database and wants an output later in an eventual restart run, the Case Control commands, GPFORCE(PLOT) or ESE(PLOT), must be specified in the cold start run.
4. By default, if GPFORCE or ESE is present, then the output of grid point forces or element strain energy is only available at the end of each subcase. For output of intermediate steps, the parameter INTOUT on the NLPARM Bulk Data entry must be set to YES or ALL.
5. For a model with superelements, the grid point force balance is not in equilibrium along the boundary of the residual and upstream superelements. This is because the applied load in the residual does not include the load contributed from upstream superelements when the grid point force balance is computed. The same situation also occurs in each upstream superelement whose applied load does not include the contribution from other superelements and the residual. When the total of the grid force balances of all superelements is added together, equilibrium is reached.

   For example, grid point 123 is located at the boundary between the residual (superelement ID = 0) and an upstream superelement (ID = 10) and is connected to two QUAD4 elements (ID 22 and 21). The output of the grid point force balance is shown below.

```
1    GPF006A - NONLINEAR GPFORCE TEST PROBLEM                    SEPTEMBER  26, 2000  MSC Nastran  9/26/00   PAGE    26
     Q4 ELEMENTS + MULTIPLE S.E.'S                                                      SUPERELEMENT 0
0    NONLINEARITY ANALYSIS - AUTO/NLPCI + INTOUT=YES                                        SUBCASE 1
     LOAD STEP =  1.00000E+00
                                        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
0       123                  APP-LOAD     -3.287216E-01   1.006007E-02  -5.592173E-03    .0             .0             .0
        123                  F-OF-SPC        .0              .0           5.127304E+00  -2.287795E-04  -1.193804E-02    .0
        123            21    QUAD4        -6.765498E-01   8.791827E+00  -2.111628E+00   8.897618E-04   1.142736E-02  -3.166363E-02
        123                  *TOTALS*     -1.005271E+00   8.801888E+00   3.010083E+00   6.609823E-04  -5.106755E-04  -3.166363E-02

1    GPF006A - NONLINEAR GPFORCE TEST PROBLEM                    SEPTEMBER  26, 2000  MSC Nastran  9/26/00   PAGE    33
     Q4 ELEMENTS + MULTIPLE S.E.'S                                                      SUPERELEMENT 10
0    NONLINEARITY ANALYSIS - AUTO/NLPCI + INTOUT=YES                                        SUBCASE 1
     LOAD STEP =  1.00000E+00
                                        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
        123            22    QUAD4         1.005275E+00  -8.801861E+00  -3.010083E+00  -6.609823E-04   5.106755E-04   3.166376E-02
        123                  *TOTALS*      1.005275E+00  -8.801861E+00  -3.010083E+00  -6.609823E-04   5.106755E-04   3.166376E-02
```

   Obviously, neither the residual nor the upstream superelement has the balanced grid point force. However, if both sides are added together, the balance will be reached.

6. Computational time and memory requirement increase when grid point force balance or element strain energy output is requested. For a large nonlinear model, allocation of sufficient memory is important. Spill logic is built in the code, but it may increase the overall computational time.

D Acoustic Cavity Modeling

- Introduction 725
- Assumptions and Limitations 726
- Acoustic Cavity Example Problem 727

Introduction

The Bulk Data entries used specifically for acoustic cavity analysis are described below, and the entry formats are exhibited in the MSC Nastran Quick Reference Guide. Their purposes are analogous to the use of structural data entries. A mesh is defined over the longitudinal cross section of an acoustic cavity, and finite elements are connected between these points to define the enclosed volume.

The points are defined by GRIDF entries for the axisymmetric central fluid cavity and by GRIDS for the radial slots. The GRIDF points are interconnected by finite elements via the CAXIF2, CAXIF3, and CAXIF4 entries to define a cross-sectional area of the body of rotation. The CAXIF2 element defines the area of the cross section between the axis and two points off the axis (the GRIDF points may not have a zero radius). The CAXIF3 and CAXIF4 entries define triangular or quadrilateral cross sections and connect three or four GRIDF points, respectively. The density and/or bulk modulus at each location of the enclosed fluid may also be defined on these entries.

The GRIDS points in the slot region are interconnected by finite elements via the CSLOT3 and CSLOT4 entries. These define finite elements with triangular and quadrilateral cross-sectional shapes, respectively. The width of the slot and the number of slots may be defined by default values on the AXSLOT entry. If the width of the slots is a variable, the value is specified on the GRIDS entries at each point. The number of slots, the density, and/or the bulk modulus of the fluid may also be defined individually for each element on the CSLOT3 and CSLOT4 entries.

The AXSLOT Bulk Data entry is used to define the overall parameters for the system. Some of these parameters are called the “default” values and may be selectively changed at particular cross sections of the structure. The values specified on the AXSLOT entry will be used if the corresponding fields on the GRIDS, CAXIFi, or CSLOTi entries are left blank. The fields RHOD and RHO (density) and BD and B (bulk modulus) are properties of the fluid. If the value given for bulk modulus is zero, the fluid is considered incompressible by the program. The fields MD and M (number of slots) and WD (slot width) are properties of the geometry. The fields MD and M define the number of equally spaced slots around the circumference, with the first slot located at $\phi = 0°$. The field N (harmonic number) is selected by the user to analyze a particular set of acoustic modes. The pressure is assumed to have the following distribution:

$$p(r,z,\phi) = (r,z)\cos N\phi$$

If $N = 0$, the breathing and longitudinal modes will result. If $N = 1$, the pressure at $\phi = 180°$ will be the negative of the pressure at $\phi = 0°$. If $N = 2$, the pressures at $\phi = 90°$ and $\phi = 270°$ will be the negative of that at $\phi = 0°$. Values of N larger than M/2 have no significance.

The interface between the central cavity and the slots is defined with the SLBDY Bulk Data entries. The data for each entry consist of the density of the fluid at the interface, the number of radial slots around the circumference, and a list of GRIDS points that are listed in the sequence in which they occur as the boundary is traversed. In order to ensure continuity between GRIDF and GRIDS points at the interface, the GRIDF points on the boundary between the cylindrical cavity and the slots are identified on the corresponding GRIDS entries rather than on GRIDF entries. Thus, the locations of the GRIDF points will be exactly the same as the locations of the corresponding GRIDS points.

Various standard bulk data entries may be used for special purposes in acoustic analysis. The SPCi Bulk Data entry may be used to constrain the pressures to zero at specified points, such as at a free boundary. The formats for these entries are included in the MSC Nastran Quick Reference Guide. Dynamic load entries, direct input matrices, and scalar elements may be introduced to account for special effects.

Assumptions and Limitations

The accuracy of the acoustic model will be dependent on the selection of the mesh of finite elements. The assumption for each element is that the pressure field has a linear variation over the cross section and a sinusoidal variation around the axis in the circumferential direction. In areas where the pressure gradient changes are large, such as near a sharp corner, the points in the mesh should be placed closer together so that large changes in flow may be defined accurately by the finite elements.

The shape of the finite elements play an important part in the accuracy of the results. It has been observed that long, narrow elements produce disproportionate errors. Cutting a large square into two rectangles will not improve the results, whereas dividing the square into four smaller squares may decrease the local error by as much as a factor of ten.

The slot portion of the cavity is limited to certain shapes because of basic assumptions in the algorithms. The cross section of the cavity normal to the axis must have a shape that is reasonably well defined by a central circular cavity having equally spaced, narrow slots. Various shapes are shown in Figure 4-1 in the order of increasing expected error.

It is recommended that shapes such as the cloverleaf and square cross section be analyzed with a full three-dimensional technique. The assumption of negligible pressure gradient in the circumferential direction within a slot is not valid in these cases.

The harmonic orders of the solutions are also limited by the width of the slots. The harmonic number, N, should be no greater than the number of slots divided by two. The response of the higher harmonics is approximated by the slot width correction terms discussed in Section 17.1 of *The MSC Nastran Theoretical Manual*.

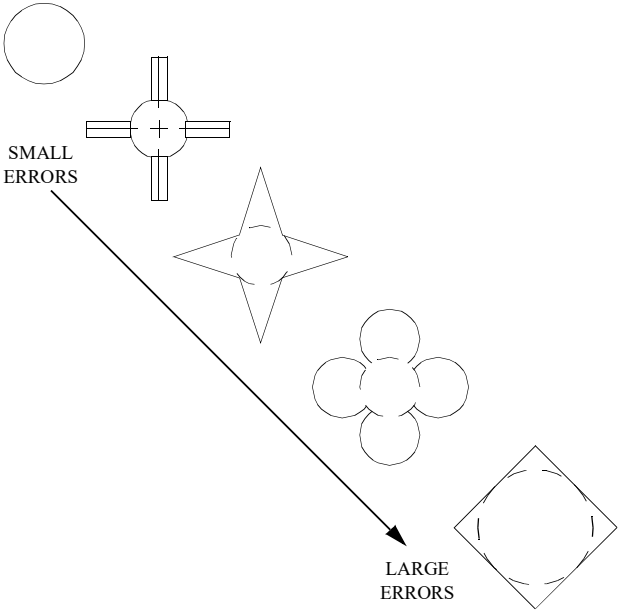


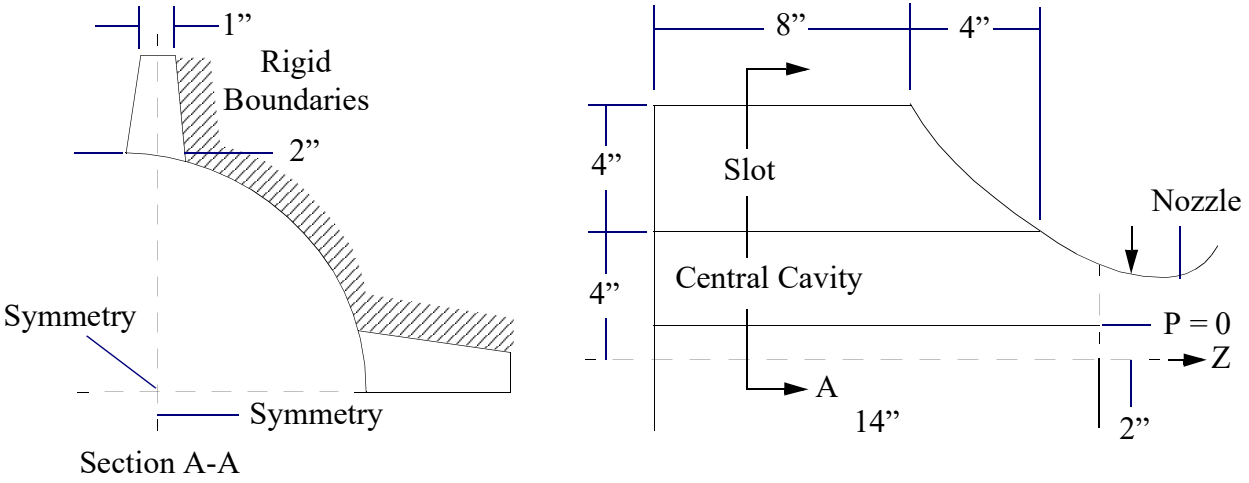
Figure 4-1 Modeling Errors for Various Shapes

The output data for the acoustic analysis consist of the values of pressure in the displacement vector selected via the Case Control command PRESSURE = n. The velocity vector components corresponding to each mode may be optionally requested by the Case Control command STRESS = n where n is the set number indicating the element numbers to be used for output, or by the words STRESS = ALL.

Plots of the finite element model and/or of the pressure field may be requested with the MSC Nastran plot commands. The central cavity cross section will be positioned in the XY plane of the basic coordinate system. The slot elements are offset from the XY plane by the width of the slot in the +Z direction. The radial direction corresponds to X and the axial direction corresponds to the Y direction. Pressures will be plotted in the Z direction for both the slot points and the central cavity points. The PLOTEL elements are used for plotting the acoustic cavity shape. The plot request entry SET n INCLUDE PLOTEL must be used where n is a set number.

Acoustic Cavity Example Problem

Table 4-1 contains a listing of the data entries used as a simple example of acoustic cavity analysis. The problem to be solved is illustrated in Figure 4-2. The model was subdivided into only ten finite elements in order to limit the number of data entries. For reasonable engineering accuracy, this model should be subdivided into at least four times that number of elements.



Parameters:Density: RHOD= 1.2×10^{-7} lb-sec^2/in^4

Bulk Modulus: BD= $ra^2 = gRT = 21.0$ lb/in^2

Harmonic: N=1

Number of Slots: MD=4

Finite Element Model:

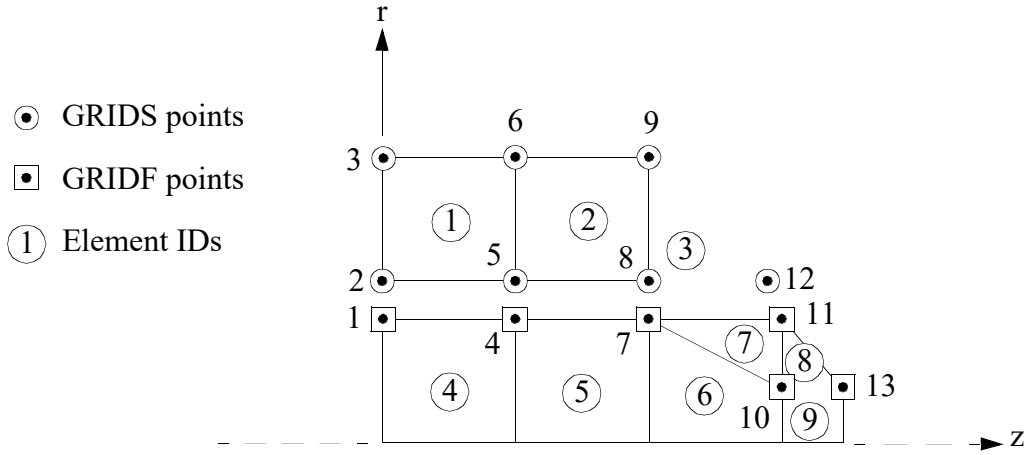


Figure 4-2 Description of Acoustic Cavity Example Problem

Each data entry in Table 4-1 is given a number on the left side. The format for each type of Bulk Data entry is given in parentheses above the group of that type. The following is a brief description of each part of the input data:

Lines	
1-4	Each statement in the Executive Control Section has the format of a request word and a selection separated by blanks or a comma. The CEND statement is last, but the statements above may appear in any order. Solution Sequence 103, normal modes analysis, was chosen. A limit of 2 minutes CPU time was set (TIME2).
5-7	The TITLE, SUBTITLE, and LABEL commands may contain any list of letters and numbers following the (=) sign. This list will appear on the first three lines of each output page.
8	The method of eigenvalue extraction is selected with the METHOD command. The number 11 refers to the identification number of an EIGRL Bulk Data entry which appears below as line 32.
9-11	A simple output request is illustrated with these commands. PRES = ALL will result in the printout of all pressures at the GRIDF and GRIDS points. STRESS = ALL will result in the printout of all velocities in the elements. This printout will occur for all extracted eigenvectors. Selected points or elements can be printed via the SET command.
12	The BEGIN BULK command denotes the beginning of the Bulk Data Section. The Bulk Data entries may occur in any order. Putting these entries in alphabetic sort will save sorting time in large problems, however.
13	In this problem, all the parameters except slot width WD are constant throughout the volume. The values on AXSLOT will be used whenever a corresponding field in the following entries is blank.
14-20	The location of points on the slot are defined with these entries. Entries 14, 16, 18, and 20 serve a dual purpose by defining a GRIDS point identification number in field 2 and a GRIDF point identification number in field 6. The two types of points thereby are forced to have the same locations at the interface.
21-22	The location of points within the axisymmetric fluid cavity are described by the GRIDF entry. No points are allowed to have a zero or negative radius.
23-31	These entries describe the elements shown in Figure 4-2. Each element is given a unique identification number and a list of the connected GRIDS or GRIDF points. Since the parameters p and B are constants, these fields are left blank, so the values on the AXSLOT entry will be used.
32	The EIGRL entry is used to define parameters for eigenvalue extraction (resonant frequencies). More than one of these entries may appear. The method to be used is selected with the METHOD command (command 8). With this particular command, the Lanczos method was selected with three (ND = 3) output mode shapes specified.
33	The SLBDY entry defines the boundary between the slot and the central cavity. Both the density (RHO and the number of radial slots (M) are blank, so the AXSLOT defaults are used; i.e., RHOD = 1.2×10^{-7} and M = 4. Only four GRIDS points are on the boundary, so a continuation entry is not necessary. Field 8 is left blank to signify the last entry.
34	The ENDDATA entry is required to denote the end of the Bulk Data Section. Any following entries will be ignored.

Table 4-1 Acoustic Cavity Example Problem Data

Line No.		
1	$DEMO PROBLEM D03D92 WITH SOL 103 AND LANCZOS	Executive Control statements
2	SOL 103	
3	TIME 2	
4	CEND	
5	TITLE = ACOUSTIC CAVITY EXAMPLE PROBLEM	Case Control commands
6	SUBTITLE = FIRST HARMONIC	
7	LABEL = NORMAL MODES (LANCZOS)	
8	METHOD = 11	
9	OUTPUT	
10	PRES = ALL	
11	STRESS = ALL	
12	BEGIN BULK	

Table 4-1 Acoustic Cavity Example Problem Data (continued)

Line No.		
	(AXSLOT, (RHOD, BD, N, WD, MD)	Bulk Data entries
13	AXSLOT,1.2-7,21.0,1,,4	
	(GRIDS,ID,R,Z,W,ID	
14	GRIDS,2,4.0,0.0,0.23 01,1	
15	GRIDS,3,8.0,.0,1.0	
16	GRIDS,5,4.0,4.0,2.0,4	
17	GRIDS,6,8.0,4.0,1.0	
18	GRIDS,8,4.0,8.0,2.0,7	
19	GRIDS,9,8.0,8.0,1.0	
20	GRIDS,12,4.0,1.2+1,2.0,11	
	(GRIDF,ID,R,Z)	
21	GRIDF,10,2.0,12.0	
22	GRIDF,13,2.0,1.4E1	
	(CSLOT4,1D,P_1,P_2,P_3,P_4,RHO,,M)	
23	CSLOT4,1 ,2,3,6,5	
24	CSLOT4,2,5,6,9,8	
	(CSLOT3,1D,P_1,P_2,P_3,,RHO,,)	
25	CSLOT3,3,8,9,12	
	(CAXIF2,1D,P_1,P_2,,,RHO)	
26	CAXIF2,4,1 ,4	
27	CAXIF2,5,4,7	
28	CAXIF2,6,7,10	
29	CAXIF2,9,10,13	
	(CAXIF3,ID,P_1,P_2,P_3,,RHO,B)	
30	CAXIF3,7,7,10,11	
31	CAXIF3,8,10,11,13	
	(EIGRL,SID,V1,V2,ND)	
32	EIGRL,11,,,3	
	(SLBDY,RHO,M,ID1,ID2,ID4,etc.)	
33	SLBDY,,,12,8,5,2	
34	ENDDATA	

E Cyclic Symmetry

- An Overview 733
- Symmetry 733
- Theory 738

An Overview

Symmetry has long been a subject on which philosophers and mathematicians have focused their attention. Their inquiries into the concepts of symmetry have led to powerful results that have, in turn, led to the formulation of analysis procedures of general applicability in applied mechanics. A familiar example is provided by the routine application of reflective or bilateral symmetry in many engineering disciplines. The principles of reflective symmetry allow the analyst to reduce the size of his analytical model by one-half for each plane of reflective symmetry in the system under investigation.

Reflective symmetry is only one of the types of symmetry that exist in nature. Among the other types is cyclic or rotational symmetry in which the properties of a body are repeated at equal intervals about an axis of symmetry. Mathematicians have developed a general theory of cyclic groups to deal with such problems, which electrical engineers have applied, in the method of symmetrical components originated by Fortescue in 1918, to the analysis of polyphase electrical networks.

An original method for the application of the concepts of cyclic symmetry to structural analysis is available in MSC Nastran. The method introduces no sacrifice in theoretical rigor or generality and provides substantial reductions in computer time for the analysis of linear systems that possess rotational and/or reflective planes of symmetry. These techniques are currently available in the following solution sequences:

SOLution 114	Superelement Static Analysis
SOLution 115	Superelement Normal Modes Analysis
SOLution 116	Cyclic Buckling
SOLution 118	Superelement Direct Frequency Response

In addition to significant savings in computer time, the analyst who grasps the concept of cyclic symmetry can realize a considerable reduction in problem preparation tasks. For example, if a body consists of N identical segments in the sense that the entire body can be generated through reflections and/or rotations of one (the fundamental region) of these N segments, the analyst is only required to define a finite element model for the fundamental region and to provide a list of the boundary grid points for this fundamental region. The finite element model may include any members of the MSC Nastran element library other than axisymmetric elements. All boundary conditions are automatically handled internally by MSC Nastran.

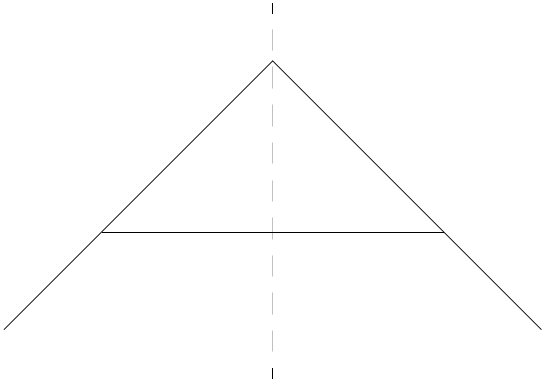
Complete generality in the definition of loading conditions is achieved by allowing the user to independently specify loads, enforced motions, and/or temperatures on each of the N segments, even though the fundamental region is the only segment explicitly represented in the analysis. Output requests are handled with the same degree of generality in that solutions are provided for each of the N segments.

Symmetry

Prior to specific consideration of the cyclic symmetry capability in MSC Nastran, it is useful to briefly review the aspects of symmetry that concerns the analyst. Specifically, the concern is with those symmetry operations that involve the permutations of a fundamental region of a body that leave the body unchanged. Several examples are presented to illustrate the points of interest.

A-Frame (Dihedral Symmetry)

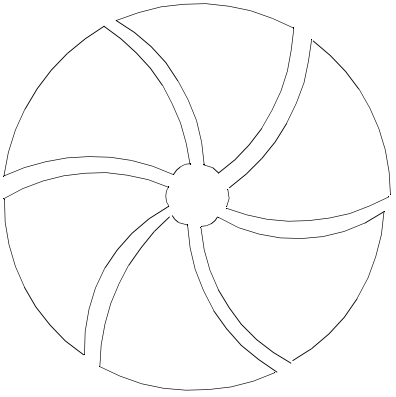
An A-frame has a plane of reflective symmetry as indicted by the dashed line in the figure below.



A reflection of the left half of the symmetric frame through the plane of symmetry results in the right half of the frame. This familiar reflective type of symmetry is known as dihedral (DIH) symmetry. Either symmetric half of the frame may be taken as the fundamental region.

Pinwheel (Rotational Symmetry)

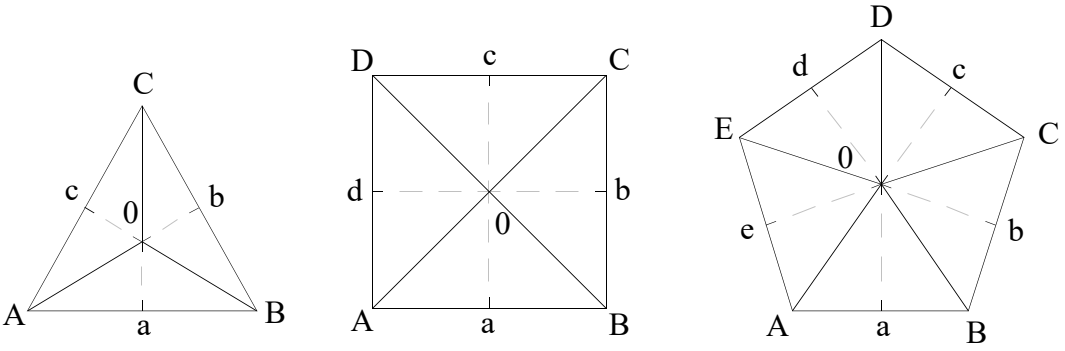
An example of a rotationally symmetric body is provided by the following example in which the "leaves" of a pinwheel have curved boundaries.



In this case, there are no planes of dihedral symmetry, but it is observed that each of the six "leaves" of the figure are identical and that rotations of the fundamental region (for example, segment 1) through the effectively distinct multiples of $360°/6$, will "generate" the complete pinwheel. This type of symmetry is known as rotational symmetry. In general, rotational symmetry includes N symmetry operations corresponding to the effectively distinct multiples $360°/N$ where N is the number of identical segments in the body.

N-Gons

In more general cases, a body may be susceptible to both DIH and ROT symmetry operations. For example, consider the three regular N-gons: the equilateral triangle, the square, and the pentagon.



N = 3 N = 4 N = 5

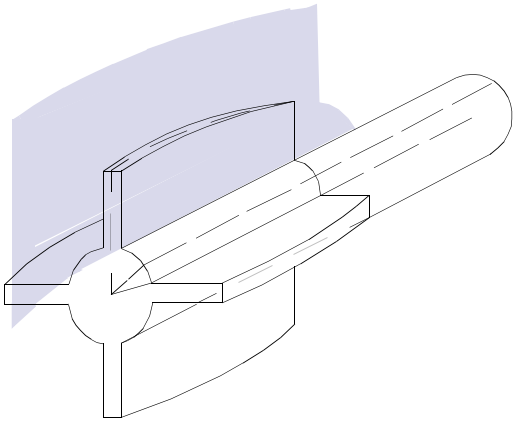
In these figures, the internal solid lines in conjunction with the edges define fundamental regions (for example, A0B) that may be permuted through the ROT symmetry operations. The axis of rotation is normal to the plane of the page and passes through the point 0.

In addition, the dotted internal lines such as a0 may be used to, for example, reflect A0a into a0B. Also, a0B may be reflected through B0 to obtain B0b, and so on around each N-gon until the identity of A0a is reached. Clearly, the fundamental region A0a for the DIH symmetry operations is smaller than the fundamental region A0B for ROT symmetry operations, and both of these fundamental regions are substantially smaller than the fundamental region required by traditional reflective symmetry techniques.

Three-Dimensional Bodies

The reader has undoubtedly noted that the previous examples involved planar bodies. However, the previous discussions would not in any way be altered if these examples were assumed to be three-dimensional bodies; that is, if the illustrations represent a cross-section of a uniform body perpendicular to the plane of the page. In fact, there are no restrictions with regard to changes in cross-sectional geometry as a function of distance along the axis of rotational symmetry.

In the example, the fundamental region is indicated by the shaded volume.



The symmetry operations are of the type DIH.

Cyclic Symmetry versus Traditional Reflective Symmetry Techniques

Engineers are generally familiar with the traditional techniques of reflective symmetry that could be employed in the analysis of any of the previous examples other than the pinwheel. The following list compares the major steps in the analysis procedures required with traditional reflective techniques and with cyclic symmetry techniques. The advantages of the cyclic symmetry techniques are most vivid.

- Analytical Model
  a. Traditional Techniques

     The size of the analytical model may be reduced by one-half for each plane of reflective symmetry.

  b. Cyclic Symmetry

     The size of the analytical model is at most equal to the size of the model required by traditional analysis techniques (A-frame). In general, the analytical model required for cyclic symmetry analysis techniques is substantially smaller.
- Boundary Conditions
  a. Traditional Techniques

     In the general case of p planes of reflective symmetry, the analyst must explicitly define 2^p sets of boundary conditions. These boundary conditions are defined through SPCi Bulk Data entries.

  b. Cyclic Symmetry

     The user supplies a list of the grid points on the boundary of the fundamental region. Boundary conditions are automatically imposed by the program.
- Applied Loads
  a. Traditional Techniques

The analyst must decompose the various loading conditions into symmetric and antisymmetric components.

b. Cyclic Symmetry

Applied loads are defined to act at arbitrary locations throughout the entire structure respective of whether any or all of these locations are in the explicitly defined fundamental region.

- Computer Solution

a. Traditional Techniques

For each set of defined boundary conditions, a lengthy matrix decomposition and equation solution are required. Static condensations are generally not economically feasible in static analysis.

b. Cyclic Symmetry

A static condensation is performed once to reduce the size of the analysis set. In static analysis, all but the boundary degrees of freedom should be omitted. In real eigenvalue analysis, buckling analysis, and dynamic analysis, the boundary degrees of freedom and selected internal points are kept in the analysis set. The static condensation involves a lengthy decomposition which, however, should be less lengthy than any one of the decompositions required by traditional techniques because the fundamental region for cyclic symmetry is generally substantially smaller than the region required for other analysis techniques. Several additional decompositions of relatively trivial magnitude are required after the static condensation.

- Data Recovery

a. Traditional Techniques

The results for each boundary condition must be combined to obtain the complete response of the structure. The user must define such combinations through `SUBCOM` or `SYMCOM` Case Control commands.

b. Cyclic Symmetry

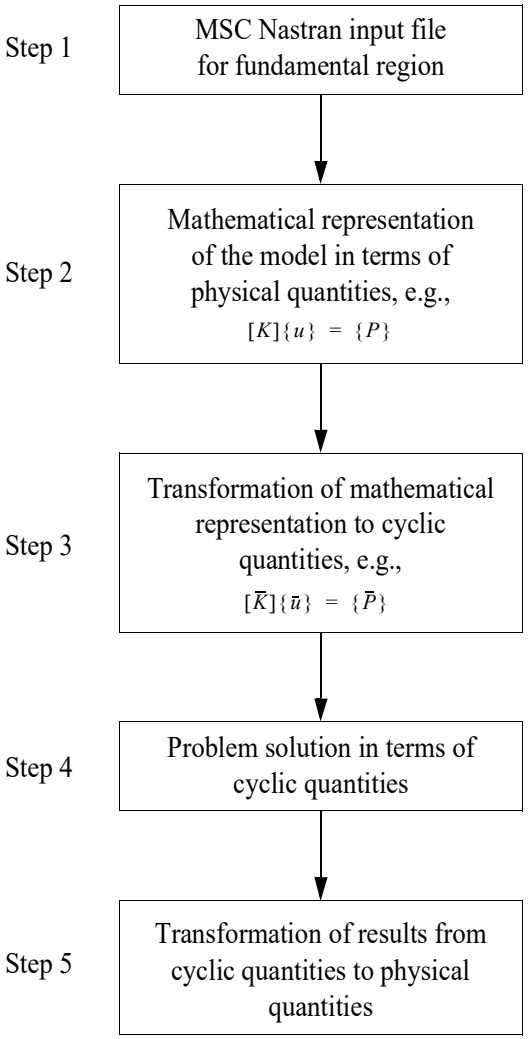
The response for the entire structure is automatically provided without any special user action. `SUBCOM`s and `SYMCOM`s are not necessary and must not be used.

In summary, cyclic symmetry provides a uniform treatment of symmetry with a potential for substantial savings in data preparation and solution cost.

Summary

The examples that have been discussed provide heuristic evidence that cyclic symmetry may be of significant value in structural analysis. One advantage is that the analyst will be required to provide an analytical model for only the fundamental regional of a structure, and this fundamental region is generally substantially smaller than the analytical model required by more traditional analysis techniques. In addition, the comparison between cyclic symmetry and more traditional analysis procedures provided in the previous section indicates that both user convenience and reductions in computer time requirements are uniquely available through cyclic symmetry procedures.

The validity of these statements relies upon the development of transformation relations that allow the analyst to manipulate the mathematical representation of the fundamental region in accordance with the appropriate symmetry operations. In graphical terms, the following operations are performed.



Steps 3 and 5 are of particular interest because they depend on the existence of the necessary transformations. Step 4 is also of interest in that it must be demonstrated that problem solution procedures in terms of cyclic quantities are valid and require less computer time than more traditional solution procedures. These matters are addressed in the next section.

Theory

The theory of cyclic symmetry is presented in some detail for the ROT case, and the corresponding results for the DIH case are summarized. This form of presentation will provide the interested user with sufficient information to independently

investigate many of the theoretical aspects of cyclic symmetry that cannot be adequately covered within the limits of this discussion.

Symmetrical Components

Consider an arrangement of N identical objects symmetrically disposed about a normal. The angle between these objects is

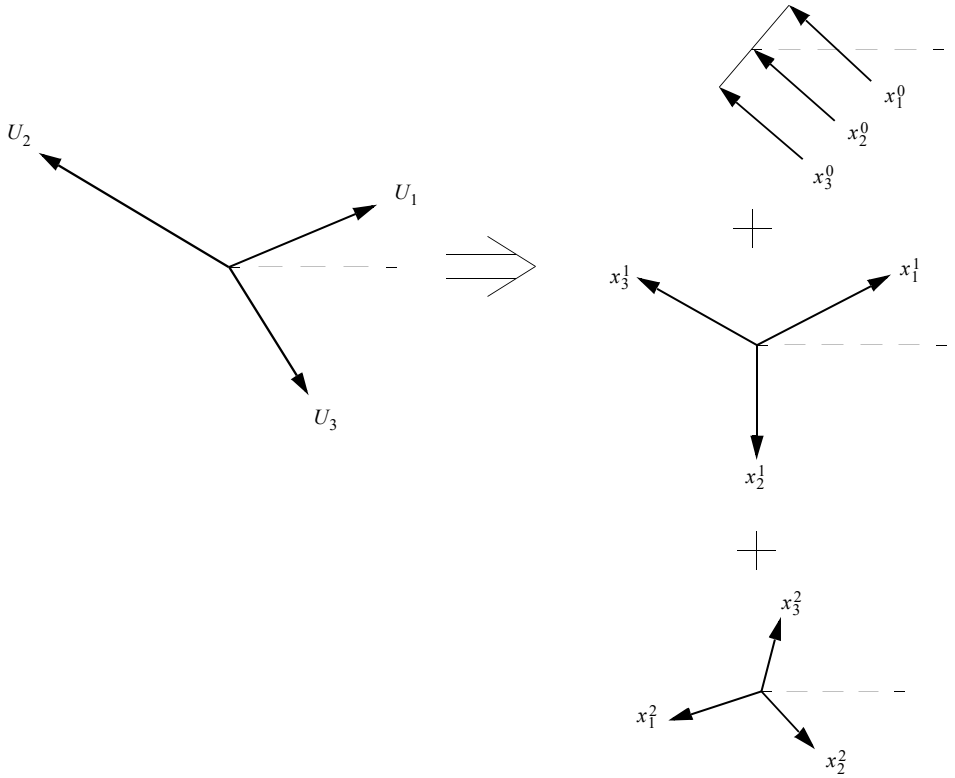
$$a = \frac{2\pi}{N} \tag{5-1}$$

and, in addition, let $U_1\ (r,t)$ be any variable displacement, force, stress, temperature, etc. in object 1, the fundamental region. The variables r and t denote distance and time respectively. The corresponding quantity in the n-th object is $U_n\ (r,t)$ and as a single valued solution is required

$$U_n = U_{n+N} \qquad n = 1,2,...,N$$

The various U_n's are, in general, unsymmetric quantities that are not related in a simple manner.

The work by Fortescue on symmetrical component theory was referenced in the introduction of this section. This theory states that a general vector field on N-vectors with arbitrary phase relationships may be resolved into N-sets of N-symmetrically phased vectors of equal magnitude. For N = 3, the vectors may be illustrated by the following diagram.



Now, in accordance with the theory of symmetrical components, the following vector sums may be stated.

$$\begin{aligned} U_1 &= x_1^0 + x_1^1 + x_1^2 \\ U_2 &= x_2^0 + x_2^1 + x_2^2 \\ U_3 &= x_3^0 + x_3^1 + x_3^2 \end{aligned} \tag{5-2}$$

Here, the U_n's represent physical quantities and the x_n^k terms represent symmetrical components. The superscripts k denote the order of the symmetric components. The subscripts n are taken to be the number of a particular vector in the vector field. In the general context of cyclic symmetry, the subscripts n denote the number applied to each of the N identical objects under consideration. The object $n = 1$ is arbitrarily taken as the fundamental region.

Transformation from Physical Components to Cyclic Components (ROT)

These transformations for the symmetric components in terms of the general field vectors may be written in summation form for the general case of N vectors.

$$x_1^k = \frac{1}{N}\sum_{n=1}^{N} U_n e^{i(n-1)ka} \tag{5-3}$$

where a is given by equation (5-1).

Transformation from Cyclic Components to Physical Components (ROT)

After the solution phase of the problem, it will normally be necessary to transform the results from cyclic components back into physical components.

Equation (5-2) may be written in the following form.

$$\begin{aligned} U_n &= \sum_{k=0}^{N-1} x_1^k e^{-1(n-1)ka} \\ &= x_1^0 + [x_1^1 e^{-1(n-1)a} + x_1^{(N-1)} e^{i(n-1)a}] \\ &\quad + [x_1^2 e^{-1(n-1)a} + x_1^{(N-2)} e^{i(n-1)2a}] \\ &\quad + x_1^3 e^{-1(n-1)a} + x_1^{(N-3)} e^{i(n-1)3a} \\ &\quad + ,..., \\ &\quad + x_1^{N/2}(-1)^{(n-1)} \end{aligned} \tag{5-4}$$

Here,

$$e^{-1(n-1)(N-k)a} = e^{-1(n-1)ka} e^{-1(n-1)Na} = e^{-1(n-1)ka}$$

and

$$e^{-1(n-1)(N/2)a} = (-1)^{n-1}$$

If the exponentials in the bracketed terms in the above summation are replaced by the relations $e^{\pm ix} = cosx \pm sinx$, the typical bracketed term takes the following form:

$$[(x_1^k + x_1^{(N-k)})\,cos(n-1)ka - i(x_i^k - x_1^{(N-k)})\,sin(n-1)ka]$$

The purpose for the somewhat arbitrary introduction of equation (5-4) is now recognized, and the typical bracketed term assumes the following form:

$$[U^{kc}\,cos(n-1)ka + U^{ks}\,sin(n-1)ka]$$

The final form of the transformation from cyclic components to physical components is therefore,

$$U_n = x_1^0 + \sum_{k=1}^{K_L} [U^{kc}\,cos(n-1)ka + U^{ks}\,sin(n-1)ka] + (-1)^{n-1} X_1^{N/2} \tag{5-5}$$

$$n = 1, 2, 3, \ldots, N$$

In equation (5-5), U_n represents physical quantities such as displacement force, stress, temperature in the n-th segment in terms of the cyclic components in the fundamental region. The following points should be noted:

1. $K_L = \frac{(N-1)}{2}$ and $x_1^{N/2}$ is nonexistent when N is odd.
2. $K_L = \frac{(N-2)}{2}$ when N is even.
3. The notation for cyclic components of order $k = 0$ and $k = N/2$ is distinct from the notation for cyclic components of order $1 \le k \le k_L$ because of the somewhat peculiar nature of these two components.

Boundary Conditions and Problem Solution (ROT)

The transformation equations of Theory, 738 are assumed to allow one to transform a problem that is mathematically defined in terms of physical components into a set of mathematical statements written in terms of cyclic components. Specifically, the mathematical description of the n replications of the fundamental region may be written in the form

$$\{F_n\} = [Y_n]\{U_n\} \tag{5-6}$$

This expression represents the mathematical description of the system in terms of physical coordinates.

The application of previously defined transform equations may now be introduced to obtain the desired mathematical description of the system in terms of cyclic components. First, however, consider the matrix representation of the transform equations (5-3) and (5-5), respectively. Equation (5-3) assumes the form

$$\begin{Bmatrix} x_1^0 \\ U^{1c} \\ U^{1s} \\ U^{2c} \\ \\ U^{KLs} \\ x_1^{N/2} \end{Bmatrix} = \begin{bmatrix} \frac{1}{N} & \frac{1}{N} & \frac{1}{N} & \cdots & \frac{1}{N} \\ \frac{2}{N} & \frac{2}{N}\cos a & \frac{2}{N}\cos 2a & \cdots & \frac{2}{N}\cos(N-1)a \\ 0 & \frac{2}{N}\sin a & \frac{2}{N}\sin 2a & \cdots & \frac{2}{N}\sin(N-1)a \\ \frac{2}{N} & \frac{2}{N}\cos 2a & \frac{2}{N}\cos 4a & \cdots & \frac{2}{N}\cos(N-1)2a \\ \cdots & & & & \\ 0 & \frac{2}{N}\sin K_L a & \frac{2}{N}\sin 2K_L a & \cdots & \frac{2}{N}\sin(N-1)K_L a \\ \frac{1}{N} & -\frac{1}{N} & \frac{1}{N} & \cdots & \frac{1}{N}(-1)^{N-1} \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ \\ \\ \\ U_N \end{Bmatrix}$$

$$\{U^k\} = [T_{xU}]\{U_n\}$$

Equation (5-5) assumes the form

$$\begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ \hline \\ \hline \\ U_N \end{Bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & \dots & 0 & 1 \\ 1 & cos\,a & sin\,a & cos\,2a & \dots & sin\,K_L a & -1 \\ 1 & cos\,2a & sin\,2a & cos\,4a & \dots & sin\,2K_L a & 1 \\ - & - & - & - & - & - & - \\ - & - & - & - & - & - & - \\ 1 & cos(N-1)a & sin(N-1)a & cos(N-1)2a & \dots & sin(N-1)K_L a & (-1)^{(N-1)} \end{bmatrix} \begin{Bmatrix} x_0^1 \\ U^{1c} \\ \\ U^{1s} \\ \hline \\ U^K L^S \\ x_1^{N/2} \end{Bmatrix} \tag{5-7}$$

$$\{U_n\} = [T_{Ux}]\{U^k\}$$

It is easily demonstrated that

$$[T_{xU}][T_{Ux}] = [I]$$

Return now to equation (5-6), the mathematical representation of the system in terms of physical coordinates, and represent the energy of the system in the form

$$U = [U_n]^T\{F_n\} \tag{5-8}$$

From equations (5-6), (5-7), and (5-8)

$$U = \begin{cases} \{U_n\}^T\{Y_n\}\{U_n\} \\ \{U^k\}^T[T_{Ux}]^T[Y_n][T_{Ux}]\{U^k\} \end{cases}$$

The system energy in terms of cyclic components may be written as

$$U = \{U^k\}^T[Y_k]\{U^k\} \tag{5-9}$$

Since the energy must be invariant to the transformations, equations (5-9) and (5-10) may be equated

$$[Y_k] = [T_{Ux}]^T[Y_n][T_{Ux}] \tag{5-10}$$

That is, the equations of the system in terms of physical components are transformed into a set of equations in terms of cyclic components. As there is no coupling between the N identical objects of the system, the matrix Y_n may be written so that it consists of N identical blocks along the diagonal. These blocks represent, in general, the equations of motion of each of the N identical objects of the system.

As an example, assume $N = 4$ $(K_L = 1)$, and expand equation (5-10). Note that $Y_1 = Y_2 = Y_3 = Y_4$.

$$[Y_k] = Y_1 \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \cos a & \cos 2a & \cos 3a \\ 0 & \sin a & \sin 2a & \sin 3a \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} \\ I \\ \\ \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & \cos a & \sin a & -1 \\ 1 & \cos 2a & \sin 2a & 1 \\ 1 & \cos 3a & \sin 3a & -1 \end{bmatrix}$$

$$= Y_1 \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

In general, for N objects the above equation may be written

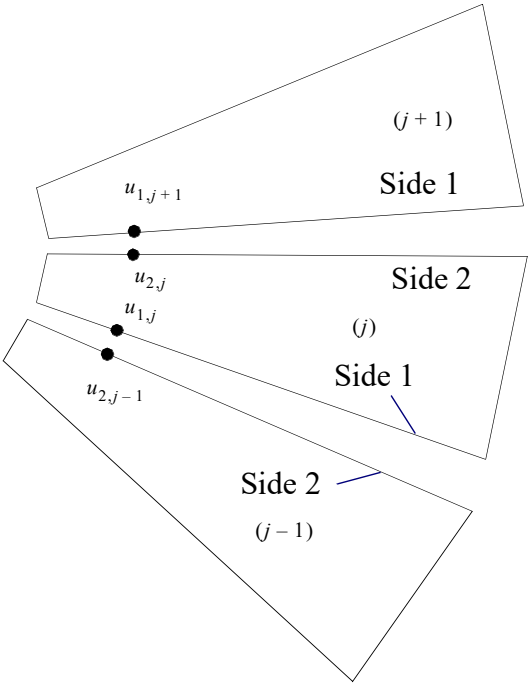
$$Y_k = Y_1 N \text{ for } k = 0, N/2$$

$$Y_k = \frac{N}{2} Y_1 \text{ for } 1 \le k \le K_L$$

These equations imply that the equations of motion in terms of cyclic components are either N or $N/2$ times the corresponding equations written in terms of physical components. This argument provides an explanation for the presence of the coefficients that appear in the terms of equation (5-6).

To this point, it has been found that except for boundary conditions, the equations for motion in terms of cyclic components ($k = 0, 1, ..., N/2$) may be developed from the equations of motion of the fundamental region ($n = 1$). In essence, an arbitrarily large physical system may be investigated in terms of a fundamental region which has been transformed into a number of models defined in terms of cyclic components. The boundary conditions between cyclic models must now be defined.

The only connection between the n identical objects in question exists along the boundaries between adjacent segments. Specifically, consider three of the N objects as indicated in the following figure.



These objects are displaced from one another in order to more clearly define the boundary conditions between the objects. Note the designation of the two sides of each of the elements as side 1 and side 2 as one proceeds in the counterclockwise direction. From the requirement of displacement compatibility, it is necessary that the typical boundary degrees of freedom, u, satisfy the equations

$$u_{2,j-1} = u_{1,j}$$
$$u_{2,j} = u_{1,j+1}$$

These physical degrees of freedom may be transformed into cyclic components of order K through equation (5-3).

$$(x^k)_{\text{side 1}} = \frac{1}{N} u_{1,j} e^{i(j-1)ka}$$
$$(x^k)_{\text{side 2}} = \frac{1}{N} u_{1,j+1} e^{ijka} \tag{5-11}$$

A comparison of equation (5-11) provides the result

$$(x^k)_{\text{side 2}} = (x^k)_{\text{side 1}} e^{ika} \tag{5-12}$$

for $1 \le k \le k_L$. For $k = 0$,

$$(x^k)_{\text{side 2}} = (x^k)_{\text{side 1}} \tag{5-13}$$

and for $k = \frac{N}{2}$,

$$(x^k)_{\text{side 2}} = -(x^k)_{\text{side 1}} \tag{5-14}$$

The cyclic component of equation (5-14) may be written in terms of the variables U^{kc} and U^{ks} in accordance with the definitions.

$$(U^{kc})_{\text{side 1}} = [(x^k)_{\text{side 1}} + (x^{N-k})_{\text{side 1}}]$$
$$(U^{ks})_{\text{side 1}} = -i[(x^k)_{\text{side 1}} - (x^{N-k})_{\text{side 1}}]$$
$$(U^{kc})_{\text{side 2}} = [(x^k)_{\text{side 2}} + (x^{N-k})_{\text{side 2}}]$$
$$(U^{ks})_{\text{side 2}} = -i[(x^k)_{\text{side 2}} - (x^{N-k})_{\text{side 2}}]$$

Equation (5-12) may now be introduced to yield the equations

$$(U^{kc})_{\text{side 2}} = [(x^k)_{\text{side 1}} e^{ika} + (x^{N-k})_{\text{side 1}} e^{-ika}]$$
$$(U^{kc})_{\text{side 2}} = -i[(x^k)_{\text{side 1}} e^{ika} - (x^{N-k})_{\text{side 1}} e^{-ika}]$$

Upon the replacement of the exponentials with Euler's equation, the appropriate boundary conditions may be represented as follow:

$$\begin{Bmatrix} (U^{kc})_{\text{side 2}} \\ (U^{ks})_{\text{side 2}} \end{Bmatrix} = \begin{bmatrix} \cos ka & -\sin ka \\ \sin ka & \cos ka \end{bmatrix} \begin{Bmatrix} (U^{kc})_{\text{side 1}} \\ (U^{ks})_{\text{side 1}} \end{Bmatrix} \tag{5-15}$$

These boundary conditions are of the form of constraint equations in which the variables on side 2 are dependent and the variables on side 1 are independent. Indeed, these equations may be viewed as MPC equations that couple kc and ks terms of side 1 to the corresponding terms of side 2.

The equations of motion in terms of cyclic components may be written in the following form.

$$[Y_1] \begin{bmatrix} \frac{N}{2} & & & & & & & \\ & N & & & & & & \\ & & N & & & & & \\ & & & N & & & & \\ & & & & \cdot & & & \\ & & & & & \cdot & & \\ & & & & & & \cdot & \\ & & & & & & & N \\ & & & & & & & & \frac{N}{2} \end{bmatrix} \begin{Bmatrix} x^0 \\ U^{1c} \\ U^{1s} \\ U^{2c} \\ \cdot \\ \cdot \\ \cdot \\ U^{KL^s} \\ x^{N/2} \end{Bmatrix} = \begin{Bmatrix} F^0 \\ F^{1c} \\ F^{1s} \\ F^{2c} \\ \cdot \\ \cdot \\ \cdot \\ F^{KL^s} \\ F^{N/2} \end{Bmatrix}$$

From the previous discussion on boundary conditions, the above equation may be rewritten in terms of the following disjoint problems.

$$k = 0 \qquad [Y_0]\{X^0\} = F^0$$

$$1 \le k \le K_L \qquad \begin{bmatrix} Y_k & 0 \\ 0 & Y_k \end{bmatrix} \begin{Bmatrix} U^{kc} \\ U^{ks} \end{Bmatrix} = \begin{Bmatrix} F^{kc} \\ F^{ks} \end{Bmatrix}$$

$$k = \frac{N}{2} \qquad [Y_{N/2}]\{x^{N/2}\} = \{F^{N/2}\}$$

The boundary equations (5-13), (5-14), and (5-15) must be introduced to remove the dependent variables that are defined to be on side 2 of the region. If, in addition, the user has taken advantage of the OMIT feature in MSC Nastran, the only independent degrees of freedom in the above disjoint problems are the degrees of freedom on side 1 that have not previously been declared dependent through MPC, SPCi, and/or OMITi Bulk Data entries. Thus, the matrices in the above disjoint problems, although dense, are of small size relative to, say, the matrices required to perform the same analysis by more traditional methods.

Summary of Transformations, Boundary Conditions, and Problem Solution (DIH)

Dihedral symmetry (DIH) is an advantageous option in the two general cases indicated below.

1. DIH takes advantage of reflective symmetry to reduce the fundamental region by one-half relative to the size of the fundamental region required in the ROT option.
2. The DIH option can be utilized for analyses that would normally be handled with traditional reflective symmetry techniques. The A-frame that was discussed above is an example of this situation.

The form of the transformations between cyclic and physical components and the boundary conditions between the segments of the object in question are, to some extent, dependent on the notation system. Figure 5-1 defines the notation system used in the DIH option in MSC Nastran.

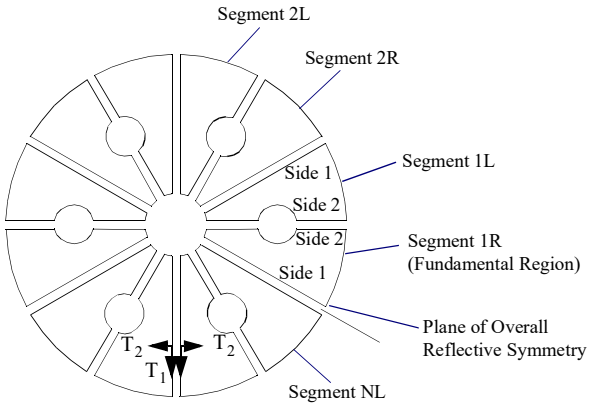


Figure 5-1 Example of Dihedral Option

Note that:

- The midplanes or relative planes are always denoted as side 2. The other boundary, which must also be planar, is always denoted as side 1. In the specific case of a structure with a single overall plane of reflective symmetry (that is, the complete structure is represented by segments 1R and 1L) the reflective plane may be defined as side 1 or side 2.
- The two halves of a segment that join at a reflective plane are denoted as the right, “R”, and left, “L”, halves.
- The fundamental region is taken as 1R.
- R halves use a right-handed coordinate system and L halves use a left-handed coordinate system. Therefore, at each boundary a degree of freedom is associated with its mirror image on the adjacent segment so that compatibility always involves a left- and a right-hand coordinate system. The degrees of freedom on the boundaries are further categorized into odd and even components in accordance with the following table.

Component	Degrees of Freedom
Even	Displacement vectors parallel to the boundary and rotation vectors normal to the boundary.
Odd	Displacement vectors normal to the boundary and rotation vectors parallel to the boundary.

The several equations of interest for the DIH option are indicated below along with the equation number for the corresponding expression under the ROT option. In the following equations, the starred (*) terms donate anti-symmetric motions and unstarred terms denote symmetric motions. The subscripts R and L refer to the right and left halves respectively.

Transformation from Physical Components to Cyclic Components (DIH)

(See equation (5-5) for the corresponding equations for the ROT option.)

$$\left.\begin{matrix} U^{kc} \\ U^{kc*} \end{matrix}\right\} = \frac{\delta}{N}\sum_{n}[U_{n,R}\cos(n-1)ka \pm U_{n,L}\cos(N-n)ka]$$

$$\left.\begin{matrix} U^{ks} \\ U^{ks*} \end{matrix}\right\} = \frac{\delta}{N}\sum_{n}[U_{n,R}\sin(n-1)ka \pm U_{n,L}\sin(N-n)ka]$$

Here,

$$\delta = \begin{cases} 1/2 \text{ for } k = 0, N/2 \\ 1 \text{ for } 1 \le k \le K_L \end{cases}$$

and the positive sign is associated with the unstarred terms.

Transformation from Cyclic Components to Physical Components (DIH)

(See equation (5-15) for the corresponding relation for the ROT option.)

$$U_{n,R} = \sum_k (U^{kc} + U^{kc*}) cos(n-1)ka + (U^{ks} + U^{ks*}) sin(n-1)ka$$

$$U_{n,L} = \sum_k (U^{kc} + U^{kc*}) cos(N-n)ka + (U^{ks} + U^{ks*}) sin(N-n)ka$$

$$0 \le k \le \frac{N}{2}$$

Boundary Conditions and Problem Solution (DIH)

(See equation (5-5) for the corresponding boundary conditions for the ROT option.)

Side 1, EVEN

$$U^{kc*} = 0$$

$$U^{ks} = 0$$

Side 1, ODD

$$U^{kc} = 0$$

$$U^{ks*} = 0$$

Side 2, EVEN

$$U^{kc} sin\frac{ka}{2} + U^{ks} cos\frac{ka}{2} = 0$$

$$U^{kc*} cos\frac{ka}{2} - U^{ks*} sin\frac{ka}{2} = 0$$

Side 2, ODD

$$U^{kc} cos\frac{ka}{2} - U^{ks} sin\frac{ka}{2} = 0$$

$$U^{kc*} sin\frac{ka}{2} + cos\frac{ka}{2} = 0$$

In static analysis with cyclic symmetry, the equation

$$[\bar{K}_{aa}]^K \{\bar{u}_a\}^K = \{\bar{P}_a\}^K$$

is solved in SSG3 (Static Solution Generator – Part 3) for $\{\bar{u}_a\}^K$

$$[\bar{K}_{aa}]^K = [G_{ck}^T K_{aa} G_{ck}] + [G_{sk}^T K_{aa} G_{sk}]$$

and

$$\{\bar{P}_a\}^K = [G_{ck}^T]\{\bar{P}\}^{KC} + [G_{sk}^T]\{\bar{P}\}^{ks}$$

are formed in CYCLIC3 (Cyclic Symmetry Transformation – Module 3).

The symmetrical components of the loads $\{\bar{P}\}^{kc}$ and $\{\bar{P}\}^{ks}$ are subsets of $\{\bar{P}\}^{x}$ which is obtained in CYCLIC3 by multiplying the load vector formed in SSG1 by the GFORE matrix formed in CYCLIC1 module

$$\{\bar{P}\}^{x} = \{P_l\}\{GFORE\}$$

where $\{P_l\}$ is formed in SSG2 (Static Solution Generator – Part 2) and the transformation matrix [GFORE] is formed in CYCLIC1 module.

The cyclic component constraint data (CYCD) used in CYCLIC3 for the formulation of the transformation matrices $[G_{ck}]$ and $[G_{sk}]$ are generated in CYCLIC2 (Cyclic Symmetry – Module 2).

The solution $\{u_a\}^{K}$ is transformed to symmetric components in CYCLIC4 as follows:

$$\{u_x\}^{kc} = [G_{ck}]\{\bar{u}_a\}^{K}$$
$$\{u_x\}^{kc} = [G_{sk}]\{\bar{u}_a\}^{K}$$

The symmetric components for each harmonic index are appended to form $\{\bar{u}\}^{x}$ in the CYCLIC4 module. The physical segment displacements are obtained as follows:

$$\{u_a\} = \{\bar{u}\}^{x}[GBACK]$$

where the transformation matrix is formed in the CYCLIC1 module.

In the case of vibration analysis with cyclic symmetry, the following equation is used in READ to determine the eigenvalues and eigenvectors,

$$[\bar{K}_{aa} - \lambda\bar{M}_{aa}]\{\bar{u}\}^{K} = 0$$

where $[\bar{M}_{aa}]^{K}$ is formed in a manner similar to $[\bar{K}_{aa}]$ in CYCLIC3. The symmetrical components of the eigenvectors are recovered in CYCLIC4 and the physical segment data is recovered using the GBACK1 matrix created in CYCLIC4 module.

Since all of the transformation matrices are extremely sparse, none of the matrix multiplications indicated above will require large amounts of computer time. The most significant operation is the triple product associated with the determination of $[\bar{K}_{aa}]^{K}$. In the usual application this matrix is dense and approximately a-size. The computer time for the triple product can be estimated as ten matrix packing operations on a full matrix of a-size.

Using Cyclic Symmetry

Many structures, including pressure vessels, rotating machines and antennas for space communications, are made up of virtually identical segments that are symmetrically arranged with respect to an axis. As shown in Figure 5-2 and Figure 5-3, there are two types of cyclic symmetry: simple rotational symmetry, in which the segments do not have planes of reflective symmetry and the boundaries between segments may be general doubly-curved surfaces; and dihedral symmetry, in which each segment has a plane of reflective symmetry and the boundaries between segments are planar. The use of cyclic

symmetry allows the user to model only one of the identical substructures. There will also be a large saving of computer time for most problems.

The total model consists of N identical segments which are numbered consecutively from 1 to N. The user generates the model for one segment, using regular elements and standard modeling techniques. All other segments and their coordinate systems are automatically rotated to equally spaced positions about the polar axis by MSC Nastran. The boundaries must be conformable; that is, when the segments are put together, the grid points and the displacement coordinate systems of adjacent segments must coincide. This is easiest to ensure if a cylindrical or spherical coordinate system is used, but such is not required.

The two boundaries are called Sides 1 and 2. In the case of rotational symmetry, Side 2 of segment n is connected to Side 1 of segment n+1, as shown in Figure 5-1. In the case of dihedral symmetry, Side 1 is on the boundary of the plane of overall reflective symmetry, and Side 2 is on the midplane or reflective plane of the segment, as shown in Figure 5-3. In either case, the grid point numbers on Sides 1 and 2 must be specified on the Bulk Data entry, CYJOlN.

The type of symmetry (rotational, dihedral, or axisymmetric) and the number of segments in the model are specified on the Bulk Data entry CYSYM. Grid points that lie on the axis of symmetry must be listed on the CYAX Bulk Data entry. The harmonics to be used in the cyclic symmetry analysis are specified on the HARMONICS Case Control command. The HARMONICS command must be present in Case Control, and it must appear above the subcase level. In buckling analysis the program limits the static preload to the zero harmonic and the information on the HARMONICS command is used only for the eigenvalue analysis.

No restrictions are placed on the use of the single-point constraint, the multipoint constraint, or the OMIT feature. Constraints between segments are automatically applied to the degrees of freedom at grid points specified on CYJOIN Bulk Data entries which are not otherwise constrained. The SPCD Bulk Data entry may be used to vary the magnitude of enforced displacements for each of the segments.

In the case of static analysis, the OMIT feature may be used to remove all degrees of freedom at internal grid points without any loss of accuracy. Since this reduction is applied to a single segment prior to the symmetry transformations, it can greatly reduce the amount of subsequent calculation. In the case of vibration analysis, the OMIT feature is used in the usual way to reduce the size of the analysis set and involves the usual approximations. The SUPORTi entries for the free bodies cannot be used with cyclic symmetry.

Provision is made for the use of rigid body supports with the use of the CYSUP Bulk Data entry. All supported degrees of freedom for a rigid body must be specified at a single grid point. Additional restrictions on the location of the support point and the uses of coordinate systems are given on the CYSUP Bulk Data entry.

Special provision is made in DIH-type problems for reflective symmetry about one or two planes. Symmetrical or antisymmetrical boundary conditions are specified on the DSYM Case Control command. A single plane of reflective symmetry must be with respect to Side 1, and if a second plane of reflective symmetry exists, it must be perpendicular to Side 1. In the case of two planes of symmetry, the model must contain an even number of whole segments.

Static loads are applied to the model with a LOADCYH (harmonic load), LOADCYN (physical segment load), or LOADCYT (AXI option) Bulk Data entry. Only a single subcase is needed for each loading condition, and it must contain the keyword LOAD, TEMP(LOAD), or DEFORM to select a LOADCYH, LOADCYN, or LOADCYT Bulk Data entry. The LOADCYH and LOADCYN entries in turn reference FORCE, MOMENT, PLOAD, SLOAD, SPCD, DEFORM, and TEMP entries. The LOADCYH entry may also reference an RFORCE or GRAV entry in order to specify centrifugal or

gravity loading, respectively. The LOADCYT entry is used only with the AXI option to specify loads as a function of azimuth angle.

The NOUTPUT Case Control command is used to request physical output within a subcase for all output quantities, such as forces and displacements, requested in the subcase. The HOUTPUT Case Control command is used to request harmonic output within a subcase for all output quantities requested in the subcase. In the case of thermal or deformation loading, element force and stress output can only be requested consistent with the form of the loading, i.e., physical output if the loads are defined with a LOADCYN Bulk Data entry and harmonic output if the loads are defined with a LOADCYH Bulk Data entry.

The use of a single subcase for each loading condition introduces the need to refer to "coded" subcases for SORT2 output and plotting purposes. Since a single subcase now refers to NSEG segments (in the case of ROT and AXI options) and two times NSEG segments (including Right and Left halves, in the case of DIH option), a unique coded subcase ID is necessary to refer to a particular segment of the structure for plotting purposes in PLOT commands wherever a subcase ID is to be supplied. This coded ID is obtained as follows:

Coded Subcase ID for a Particular Segment

$$= \text{SUBCASE ID} \cdot 10000 + \text{SEGMENT ID} \cdot 10 + \text{zero} \quad \text{(for ROT or AXI option)}$$

$$= \text{SUBCASE ID} \cdot 10000 + \text{SEGMENT ID} \cdot 10 + \begin{Bmatrix} 1 & \text{(if R segment)} \\ 2 & \text{(if L segment)} \end{Bmatrix}$$

Coded Subcase ID for a Particular Harmonic

$$= \left[\text{SUBCASE ID} \cdot 10000 + \text{HARMONIC ID} \cdot 10 + \begin{Bmatrix} 1 & \text{(if component = C)} \\ 2 & \text{(if component = S)} \\ 3 & \text{(if component = C*)} \\ 4 & \text{(if component = S*)} \end{Bmatrix} \right]$$

Solution of Axisymmetric Problems

The Cyclic Symmetric Solution Sequences provide a convenient and effective procedure for solving axisymmetric problems without the need for special axisymmetric elements. The following procedure is recommended for the solution of static load problems:

1. Select the AXI symmetry option on the CYSYM Bulk Data entry.
2. Select a large number of segments (for example, 180) on the CYSYM Bulk Data entry.
3. Construct a model with two rows of grid points and a single row of elements in the meridional direction.
4. Define a set of harmonics on a SET Case Control command. The set of harmonics should be reasonably small for efficiency, but sufficient in size to yield reasonably accurate results.

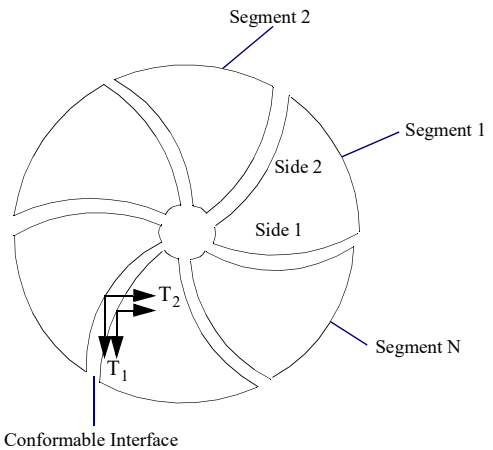
5. Select the defined set of harmonics on the HARMONICS Case Control command. Convergence of the harmonic output towards zero for the higher harmonics will verify that the number of harmonics specified on the HARMONICS Case Control command and used in the analysis is sufficient.
6. Select a static loading condition with the LOAD, TEMP(LOAD), and DEFORM Case Control commands which reference LOADCYN, LOADCYH, or LOADCYT Bulk Data entries.
7. Place static loads only on Side 1 or on elements. The boundary conditions will take care of loading on Side2.
8. Prepare the referenced TABLED1 entries if LOADCYT entries are used to specify loads as a function of azimuth angle. The number of x-y pairs in a given table should be sufficient to produce reasonably accurate results when linearly interpolated.
9. Select the physical output for a reasonably small number of segments with the NOUTPUT Case Control command or ALL of the harmonic output with the HOUTPUT Case Control command.

The following procedure is recommended to obtain the vibration modes of an axisymmetric structure:

1. Select the AXI symmetry option on the CYSYM Bulk Data entry.
2. Select a large number of segments (for example, 90) on the CYSYM Bulk Data entry.
3. Construct a model with two rows of grid points and a single row of elements in the meridional direction.
4. Define a set of harmonics on a SET Case Control command where the number of harmonics is reasonably small for efficiency, but sufficient to cover the frequency range of interest.
5. Select the defined set of harmonics on the HARMONICS Case Control command.
6. Select ALL of the harmonic output with the HOUTPUT Case Control command. This will give the eigenvectors, in a concise form, which may be interpreted by the formula

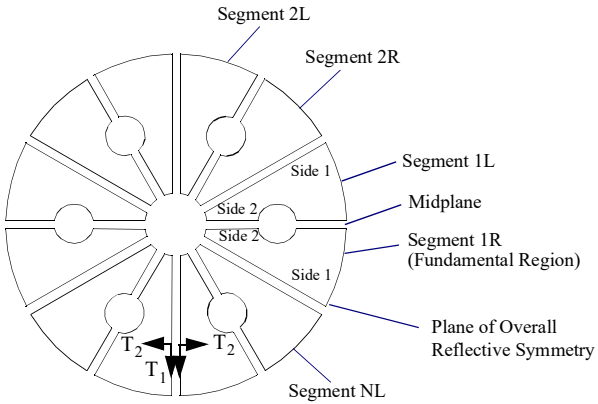
   $$U(\phi) = U^{kc} \cos k\phi + U^{ks} \sin k\phi$$

   where ϕ is the azimuth angle measured from Side 1 of the fundamental region and K is the harmonic index.



1. The user models one segment.
2. Each segment has its own coordinate system which rotates with the segment.
3. Segment boundaries may be curved surfaces. The local displacement coordinate system must conform at the joining points. The user gives paired list of points on Side 1 and Side 2 which are to be joined.

Figure 5-2 Rotational Symmetry



1. The user models one-half segment (an R segment). The L half segments are mirror images of the R half segments.
2. Each half segment has its own coordinate system which rotates with the segment. The L half segments use left-hand coordinate systems.
3. Segment boundaries must be planar. Local displacement systems axes, associated with the inter-segment boundaries, must be in the plane or normal to the plane. The user lists the points on Side 1 and Side 2 which are to be joined.

Figure 5-3 Dihedral Symmetry

F p-Elements

- Introduction 756
- p-Elements in MSC Nastran 756

Introduction

One of the main applications of p-version elements is detailed stress analysis. The p-elements have higher-order polynomials, which provide better representation of complex stress fields. For these complex stress fields, the geometry, loads, and boundary conditions must be represented accurately. This includes modeling fillets instead of sharp corners, distributed loads and constraints instead of point loads and constraints, etc. A more detailed model leads to more detailed results.

The p-Elements have several modeling advantages.

- With the higher-order polynomials, the geometry and loads can be represented more accurately. Generally, fewer p-version elements with curvature than h-version elements with straight edges are required. The accuracy of the analysis is controlled primarily by the polynomial level, not by the element size. Generally the user needs to only use the minimum number of elements necessary to adequately model the geometry, independent of expected result characteristics.
- Polynomials levels can then be assigned based on areas of the model in which the user is interested, and areas of the model in which the user is not. Of course, the same polynomial level can be assigned everywhere for a minimum of user involvement but this is not as efficient.
- If a more accurate answer is necessary, the polynomial levels can be increased, either manually by the user or automatically by the program, until the answers reach the specified accuracy, without changing the mesh.
- Adding degrees of freedom until the error decreases to a specified level is known as adaptivity.
- By adding higher-order polynomials instead of refining the mesh, a faster rate of convergence, and therefore fewer iterations, can be achieved.

p-Elements in MSC Nastran

In MSC Nastran, p-elements allow different polynomial orders in the three coordinate directions, and use bubble functions at $p = 1$ and optimum integration algorithms at $p \geq 2$ to achieve better accuracy. The elements will also pass the patch tests and rigid body motion at any allowable p-level. These additional features improve the accuracy over conventional p-elements.

Compatibility with elements of different p-orders, and with the existing solid elements, is automatically enforced. The user does not have to write constraint equations or use special elements to provide displacement continuity. Loads and boundary conditions can be applied to the elements themselves, or to the geometry on which they are defined.

The definition of geometry has been added to MSC Nastran. This geometry can be defined with midside points, as in the existing elements; with rational parametric cubics or equations; or even with any external geometry. A facility has been set up to invoke remote procedure calls to any independent program and database that provides a toolkit for geometry queries, which would allow proprietary geometries of other preprocessors or CAD packages to be used. Associativity has been added so that the elements are defined on the geometry, and the element geometry calculated accordingly.

The error estimator requires only a single analysis, not the difference between two analyses. For linear elements, it is based on the grid-point stress discontinuity; for higher polynomial orders, it considers the contribution of the additional terms for the individual edge, face, and body functions. Therefore, it provides efficient information because of the directional sensitivity within the elements.

The adaptivity method is very flexible, working on an element-by-element basis. Using the error estimator, the polynomial level within each element can be varied independently in the three directions. At the most basic level, the edges are varied, since they determine the face and body functions. Different error criteria may be used in different regions, allowing areas of primary interest to have lower error tolerances. Multiple subcases may be used.

Combining p- and h-Elements

The main advantage of implementing p-elements in MSC Nastran is the ability to combine them with the existing h-elements for global/local analysis. The mesh could consist mostly of h-version elements, with p-version elements in the areas of interest. This method of global/local interface is the most accurate, because it provides both stiffness and loads information, rather than transferring just boundary displacements or tractions into the local model.

h-version elements still have several advantages. They are better for global behavior, such as loads analysis, where the exact geometry is not used. They are better for strongly singular problems, such as nonlinear problems. Approximations, such as sharp corners, point loads, point constraints, MPC's, etc. may be used. For a given iteration, they tend to have shorter solution times and use less disk space. Finally, h-elements are a very mature technology.

Putting both approaches in the same program combines the advantages for general problems. The p-elements may be connected directly to the existing h-elements, and continuity is automatically enforced. This provides global/local analysis in a single run.

Most general problems have both some global and some local aspects; that is, regions where h-elements and p-elements, respectively, are appropriate. Therefore the best approach is combined h- and p-adaptivity, with different criteria in different parts of the model.

p-Version Capabilities

Following is a limited description of the Bulk Data entries and Case Control commands for the p-version elements. Table 6-1 contains a comparison with the conventional h-version elements. A full description of the entries may be found in the *MSC Nastran Quick Reference Guide*.

Table 6-1 Bulk Data Entries for h-Elements and p-Elements

	h-Version Elements	p-Version Elements
Geometry		GMCURV GMSURF
Geometrical Associativity		FEEDGE FEFACE
Coordinate Systems	CORDij	CORDij GMCORD
Grids Points	GRID	GRID POINT

Table 6-1 Bulk Data Entries for h-Elements and p-Elements (continued)

	h-Version Elements	p-Version Elements
Elements	CHEXA CPENTA CTETRA CQUAD CTRIA CBEAM	CHEXA CPENTA CTETRA CQUAD CTRIA CBEAM
Element Properties	PSOLID PSHELL PBEAM	PSOLID PSHELL PBEAM
Material Properties	MATi	MATi
Loads Thermal Loads Gravity Loads	 PLOAD4 FORCE TEMP, TEMPD TEMPP1 TEMPRB GRAV	GMLOAD PLOAD4 FORCE* TEMPF TEMP, TEMPD TEMPP1 TEMPRB GRAV
Constraints Boundary Conditions	 SPC, SPC1 SPCD	GMSPC SPC, SPC1* GMBC SPCD*
Polynomial Values		PVAL PSET
Adaptivity Control		ADAPT
Data Recovery		OUTPUT OUTRCV
Miscellaneous		DEQATN TABLE3D
Sets Definition		SETS DEFINITION

*Should only be used under specific conditions.

Solution Sequences Supported

Solution Sequences 101, 103, 107, 108, 109, 110, 111, 112, and 101 (Heat Transfer).

Geometry

With the h-version elements, geometry was determined by the addition of midside grid points to the element definition. With the p-version elements, geometrical entities have been introduced into MSC Nastran. The edges and faces of elements can be defined to lie on curves and surfaces, which can span an arbitrary number of elements, and midside grid points do not have to be defined.

The Bulk Data entries to define the associativity between the finite element model and the geometrical model are the FEEDGE and FEFACE entries for the edges and faces, respectively. These entries reference the grids that make up the edges and faces, not the elements themselves, and therefore do not have to be repeated for each element that shares the edge or face. The old method of midside grid points is still available for the FEEDGE entry, which would reference one or two GRIDs or POINTs. The POINT entry is a new entry, very similar to the GRID entry, except that it defines a location in space with no associated degrees of freedom.

The more useful way of defining the FEEDGEs and FEFACEs is to have them reference geometrical curves and surfaces, defined by the new GMCURV and GMSURF entries. GMCURVs and GMSURFs are functions of one or two parametric variables, respectively, and can be defined as rational parametric cubics or general equations. For instance, a hole could be defined with the parametric equations for a cylinder, which all the element faces on the hole would then reference. The GMCURVs and GMSURFs could also reference external geometry from a CAD or other database, using remote procedure calls with an external programming interface. When geometry has been specified, it is approximated as cubics for the element-level computations. Using cubics ensures that the elements pass the rigid-body and patch tests.

The two different methods of defining the geometry are shown in Figure 6-1.

Convective coordinate systems may be defined on the FEEDGEs, FEFACEs, GMCURVs, and GMSURFs. The directions of the systems are normal and tangent to the entities at every point. These coordinate systems can then be used for material properties, loads, boundary conditions, etc.

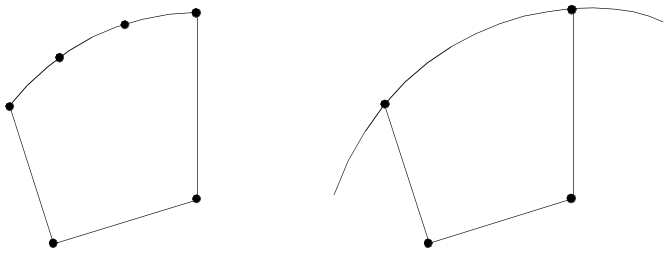


Figure 6-1 Methods of Defining Element Geometry

Elements

Only the CHEXA, CPENTA, CTETRA, CQUAD, CTRIA, and CBEAM elements may be defined as p-elements. Midside grid points may not be specified on p-version elements; curvature is defined by the FEEDGE or FEFACE entries, as previously described.

The standard Bulk Data entries, PSOLID, PSHELL, and PBEAM are used to defined the element properties. The p-version properties are defined in the PVAL and ADAPT entries, which are described later.

The p-version elements may be joined to existing h-version elements with a few considerations. FEEDGEs and FEFACEs may not be defined on the interface, since the h-elements can not accept them. Midside grid points may not be defined on the interface, since the p-elements can not accept them. The polynomial levels of the p-elements will be lowered automatically so that displacement continuity is enforced.

Materials

The standard isotropic, orthotropic, and anisotropic materials are available for p-elements. Orthotropic and anisotropic material properties may be defined in a curvilinear material system.

Loads

For p-version elements in detailed stress analysis, it is important to use distributed loads. Concentrated forces cause singularities in the stress field, and therefore should be avoided. The GMLOAD Bulk Data entry has been defined for the p-version elements. It can apply a distributed load to FEEDGEs, FEFACEs, GMCURVs, or GMSURFs. The load can be distributed as constant, linear, quadratic, cubic, equation, or table, and can be given in any coordinate system. The PLOAD4 and GRAV entries can also be applied to p-elements. The FORCE entry can be used, but should only be used where the adjacent elements have a fixed p-level and the results are not of interest.

For thermal loads, the TEMPF entry allows specification of an equation or table to describe the variation of the temperature field over the elements. The TEMP and TEMPD entries may also be used, but will result in linear interpolation of the temperature from the grids. The TEMPP1 entry defines a constant temperature and temperature gradient distribution in a shell element. The TEMPRB entry defines a linear temperature and temperature gradient distribution in a beam element.

Boundary Conditions

As with the loads, it is important to use distributed boundary conditions for p-version elements in detailed stress analysis. Concentrated boundary conditions also cause singularities in the stress field and should be avoided. The GMSPC and GMBC Bulk Data entries have been defined for the p-version elements. These can apply distributed boundary conditions to GRIDs, FEEDGEs, FEFACEs, GMCURVs, or GMSURFs. Note that they should be applied to GRIDs only to prevent rigid-body motion. The imposed boundary conditions can be distributed as constant, linear, quadratic, cubic, equation, or table, and can be given in any coordinate system. The current SPC and SPC1 entries can be used, but should only be used to prevent rigid-body motion or used where the adjacent elements have a fixed p-level and the results are not of interest.

p-Version Adaptivity

In order for an element to be recognized as a p-version element, it must be assigned the polynomial values in the new PVAL Bulk Data entry, which is referenced in the ADAPT Bulk Data entry. In order for the p-element to be adaptive, the adaptivity parameters must be assigned in the ADAPT entry.

The new PVAL Bulk Data entry assigns the polynomial levels for the three directions to a single element or a set of elements, and is referenced in the ADAPT Bulk Data entry. By default, the three directions are along the element edges. If a coordinate system is specified, the polynomial level of each edge is set to a weighted combination of the coordinate directions. There

may be multiple PVALs with the same ID; all of the elements do not have to use the same entry. Each p-version element must have a starting, minimum, and maximum PVAL ID; however, these do not need to be unique. Since the polynomial levels can not decrease, the starting and minimum PVAL IDs can be the same. If the starting and maximum PVAL IDs are the same, the element will have a fixed p-level.

After the p-value distribution has been defined, all of the values will be resolved to ensure displacement continuity. This includes selecting the highest p-level specified for elements with common edges, decreasing the p-level for p-elements adjacent to h-elements, and increasing the p-level on curved edges to adequately map them.

The new ADAPT Bulk Data entry controls the p-adaptivity. The ADAPT Bulk Data entry is referenced with the ADAPT Case Control command, and there may only be one unique ADAPT command in the Bulk Data file. The ADAPT entry contains the IDs for the starting, minimum, and maximum p-value distributions and the maximum number of iterations. It also contains the adaptivity parameters, which may be assigned differently to different sets of elements. These sets of elements do not have to correspond with any sets used in the PVAL entries. The adaptivity type may be element-by-element p-adaptive, uniform p-adaptive, no change, or list of p-distributions. For each set in the ADAPT entry, the error estimator method, error tolerance, and stress and strain tolerances are specified. If a p-version element is not included in one of the sets, it is not adaptive.

For a nonadaptive p-version solution, the ADAPT Bulk Data entry may specify a maximum of one iteration, or may specify an adaptivity type of no change for all the elements. A third way is to use the PSET Bulk Data entries, which have the same format as the PVAL entry, and reference them with the ADAPT Case Control command.

Data Recovery

Since the p-version analysis may have multiple iterations, the output control must have the capability of differing among the iterations. For this reason, the OUTPUT Bulk Data entry was added. For a given set of elements, the conventional displacements, stresses, and strains, as well as the new element errors and polynomial values, may be printed, plotted, or punched for the given iteration. The OUTPUT entry is referenced by the DATAREC Case Control command. The conventional Case Control commands still apply to all iterations, but the OUTPUT entry has precedence.

The p-version elements also tend to be larger than h-version elements, and have higher-order distributions of displacements, stresses, and strains. Therefore results only at the grid points are not adequate, especially for most plotting packages, which use linear interpolation. For data recovery, the p-version elements are divided into view-grids and view-elements in order to better visualize the results. The new OUTRCV Bulk Data entry defines the number of view-elements in each element direction and the output coordinate system for the view-grids and view-elements. The OUTRCV entry is referenced by the OUTRCV Case Control command. The coordinates of the view-grids may be output using the VUGRID Case Control command.

p-Element Example Problems

This section demonstrates the p-element and p-adaptivity capabilities. Several examples are included in this section to show how p-elements are used. Most of the input files of these examples have been annotated and are included in the Test Problem Library (TLP) directory. The detailed descriptions of the input commands and the interpretation of the results can be found in the MSC Nastran Quick Reference Guide.

Examples

Each example contains the following parts:

Objectives	Outlines the problem objectives.
Keywords	Serves as index and show what p-element related Bulk Data entries being used.
Problem Description	Describes the geometry and data of the problem.
Theoretical Results	Shows the closed form solution, if available.
Setup	Describes what and how to use the Bulk Data entries.
Results	Compares the calculated results with theory.
References	

Square Plate with a Circular Hole

Objectives

This example computes the stress concentration factor of a square plate with a circular hole under axial loads. The objective of this problem is to validate the stress concentration factor for a square plate of 2.73 around the hole.

Problem Description

Plate size	= 10 in x 10 in
Thickness	= 0.1 in
Hole radius	= 2.0 in
Modulus of Elasticity	= 1.0E7 psi
Poisson's ratio	= 0.3

Figure 6-2 shows the quarter model that requires only two CHEXA solid p-elements.

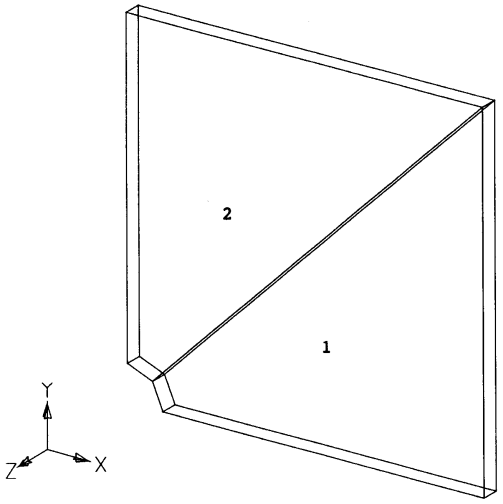


Figure 6-2 Geometry of a Square Plate with a Circular Hole

A portion of the file which has been annotated is shown in Listing 6-1. The complete input file, PEV0101.dat, can be found in the TPL directory.

Theoretical Results

Stress concentration factor of 2.73 around the hole as described in Reference 1.

Setup

The Bulk Data entry FEEDGE is used to describe the circular arc of the hole. The quadratic order is used for demonstration purposes; a cubic order curve can be easily accommodated with an additional point for each curve. The load and boundary conditions are also specified using FEFACE.

Since the file has been heavily annotated, the description of the Bulk Data entries is not repeated here. Please refer either to the following partial list or the complete input file in the TPL directory for detailed descriptions.

Listing 6-1 Partial Bulk Data Entries

```
TITLE = PLATE WITH HOLE -- QUARTER MODEL (p-element model; Point option)
SUBTITLE = p-adaptivity; maximum allowable p-value 5X5X3
$
$   p-elements Case Control commands
$
$     Select Bulk Data ADAPT ID for adaptivity control
$
ADAPT = 150
$
```

```
$     Request output destination(s) of locations (i.e. coordinates)
$       for p-element data recovery points
$
VUGRID(PLOT,PUNCH)=all
$
$     Request output (STRESS, STRAIN, DISP, PVAL) for p-elements only
$     DATAREC = n, where n refers to a Bulk Data OUTPUT ID
$
DATAREC = 301
$
$     Select output options for p-elements
$     OUTRCV = n, where n defines an OUTRCV ID in Bulk Data
$
OUTRCV = 401
$
$     Select set(s) of elements for adaptivity and stress output
$     SETS DEFINITION must be defined at the end of the Case Control
$     SET 501 defines a set of all elements
$       referenced by ELSETID in ADAPT entry for ERRTOL of 10%
$
SETS DEFINITION
set 501 = all
$
BEGIN BULK
$
$ p-elements Bulk Data
$
$    Adaptivity control
$      PSTRTID specifies the PVAL ID 120 for the starting pvalues p1, p2 and p3
$        for the element SET 501 (all elements)
$      PMINID  specifies the PVAL ID 120 for the minimum  pvalues p1, p2 and p3
$      PMAXID  specifies the PVAL ID 150 for the maximum  pvalues p1, p2 and p3
$
$ADAPT  SID                     PSTRTID PMINID  PMAXID
$       PART    ELSETID
$
ADAPT   150                     120     120     150
        PART=ALL,ELSET=501
$
$    Specify p-levels for element SET 501 (all elements)
$     POLY1, POLY2 and POLY3 correspond to p1, p2 and p3 respectively and
$       can have different p-values
$     p-values specified by PMINID should be = PSTRTID < PMAXID
$     starting p-level p1=p2=p3=2
$     minimum  p-level p1=p2=p3=2
$     maximum  p-level p1=p2=5 p3=3
$
$PVAL   ID      POLY1   POLY2   POLY3           SETTYP  ID
PVAL    120     2       2       2               SET     501
PVAL    150     5       5       3               SET     501
$
$     Select various output options;
$      Displacements, Stresses, Strains and p-values
$
$       SID
$       ELSET=n,  DISP=(PRINT,PLOT,PUNCH),
$       STRESS=(PRINT,PLOT,PUNCH), STRAIN=(PRINT,PLOT,PUNCH),
```

```
$       PVAL=(PRINT,PLOT,PUNCH), BY=n
$
OUTPUT  301
        ELSET=501, DISP=PRINT, STRESS=PRINT
        STRAIN=PLOT, PVAL=PRINT,  BY=1
$
$     Define various output options;
$      for Displacements, Stresses and Strains
$
$       SID     SETID
$       CID= 0 or n         (default 0; basic coordinate system)
$       VIEW=i*j*k          (default 3*3*3)
$         For less output, reduce i,j and k to 2 or 1
$
OUTRCV  401     501
        VIEW=2*2*2
$
CHEXA   1       1       1       25      28      4       101     125
+       128     104
CHEXA   2       1       32      4       28      56      132     104
+       128     156
$
$   The following geometry specifications demonstrate the use of
$     different options on specifying actual geometry
$     to achieve more accurate results
$
$     Define FEEDGEs and FEFACEs to specify
$       geometry, loading and constraints
$
$    Use POINT, FEEDGE to describe curves 104, 108, 204 and 208
$
POINT   3               1.7305341.0026220.0
POINT   30              1.0026221.7305340.0
POINT   103             1.7305341.0026221.0
POINT   130             1.0026221.7305341.0
$
$       EDGEID  G1      G2              GEOMIN   ID1
FEEDGE  104     4       1               POINT    3
FEEDGE  204     4       32              POINT    30
FEEDGE  108     104     101             POINT    103
FEEDGE  208     104     132             POINT    130
$
$    Specify FEFACE for BC and loads
$
$       FACEID  G1      G2      G3      G4
FEFACE  1001    1       25      125     101
FEFACE  1002    25      28      128     125
FEFACE  2002    56      156     132     32
$
$    Use GMSPC,GMLOAD and FEFACE to specify BC and loads
$      on geometry instead of element data
$
$    Boundary Conditions
$
$       SID     C       ENTITY  ID
GMSPC   999     2       FEFACE  1001
GMSPC   999     1       FEFACE  2002
```

```
$
$     LOADS
$
$        LID              N1       N2       N3       ENTITY   ID       METHOD
GMLOAD   998              1.0      0.0      0.0      FEFACE   1002     CONSTANT
         90.
$
ENDDATA
```

Results

Table 6-2 compares MSC Nastran results with the theory, and they agree very well with only two p-elements. Closer results could have been obtained with more elements or higher p-values.

Table 6-2 Comparison of MSC Nastran and Theoretical Results

Stress Concentration Factor		Percent Difference
Theory	MSC Nastran	
2.73	2.77	1.5%

Spherical Structure Under Internal Pressure

Objectives

This example computes the effects of internal pressure on a spherical structure. The deflections at the inner and the outer surfaces are compared with the theoretical results.

Problem Description

Internal pressure (P)	= 3000.0 psi
Inner radius (b)	= 5.0 in
Outer radius (a)	= 10.0 in
Modulus of Elasticity (E)	= 1.0E7 psi
Poisson ratio (ν)	= 0.3

Theoretical Results

The theoretical results on radial deflections can be found in Reference 1 and are listed as follows:

$$\text{Inner radial deflection} = \frac{Pb}{E}\left[\frac{(1-\nu)(a^3+2b^3)}{2(a^3-b^3)}+\nu\right]$$

Outer radial deflection = $\frac{Pa}{E}\frac{3(1-\nu)b^3}{2(a^3-b^3)}$

Setup

This example demonstrates how the higher cubic order of the geometry for a spherical structure can be specified using the EQUATION option of the GMCURV and GMSURF Bulk Data entries. An octant model is analyzed with four CPENTA elements. The boundaries of the CPENTA elements are specified along the great circles. Figure 6-3 shows the geometry and the element boundaries.

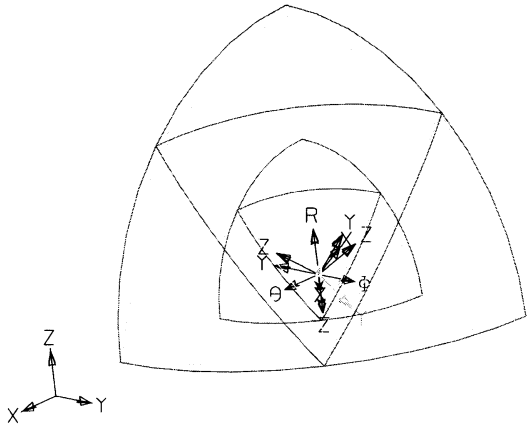


Figure 6-3 Spherical Structure

Partial Bulk Data entries are given in Listing 6-2. Equations of the inner circle along the sphere are described using Bulk Data entries GMCURV and DEQATN. The equations (IDs 1, 2, and 3) are the spatial locations of the curve along the great circle (circumference of a sphere) expressed in terms of the radius and an angle (U). The first and second derivatives of the equations are described by DEQATN entries 4 to 9. The first and second derivatives are optional inputs. If they are missing, they will be calculated by the program.

Listing 6-2 Partial Bulk Data Entries

```
$
$ ************** CURVE AND SURFACE GEOMETRY *************
$
$ Inner circle equations
$
$
DEQATN  1       X(U) = 5.0*COS(U)
DEQATN  2       Y(U) = 5.0*SIN(U)
DEQATN  3       Z(U) = 0.0
DEQATN  4       XP(U) = -5.0*SIN(U)
DEQATN  5       YP(U) =  5.0*COS(U)
DEQATN  6       ZP(U) =  0.0
DEQATN  7       X2P(U) = -5.0*COS(U)
```

```
DEQATN  8       Y2P(U) = -5.0*SIN(U)
DEQATN  9       Z2P(U) =  0.0
$
$ Inner circle on x-y plane
$
GMCURV  1       MSCGRP1  0       0                                       +
+       EQUATION  0.0   1.570796327                                     +
+       1 2 3   4 5 6   7 8 9
$
$ Inner circle on y-z plane
$
GMCURV  2       MSCGRP1  0       0                                       +
+       EQUATION  0.0   1.570796327                                     +
+       3 2 1   6 5 4   9 8 7
$
$ Inner circle on x-z plane
$
GMCURV  3       MSCGRP1  0       0                                       +
+       EQUATION  0.0   1.570796327                                     +
+       2 3 1   5 6 4   8 9 7
$
```

The GMCURV entry specifies the great circle geometry on the x-y plane with equations 1 through 9 for the curve ID 1, and ranges from 0° to 90° (0 to 1.57 radians). Similarly, the geometry on the y-z and x-z planes, curves 2 and 3, can be specified using the same equations with different orders. Note that the labels and arguments of the equations may be chosen for user convenience.

Geometry of the inner surface is described using GMSURF and DEQATN entries. The equations (IDs 41, 42, and 43) are the spatial locations of the surface using a spherical coordinate system expressed in terms of radius and two angles (U and V). The first and second derivatives of the equations are described by DEQATNs 44 to 59. Again, the first and second derivatives are optional inputs.

The GMSURF entry specifies the surface geometry with equations 41 through 59 for the surface ID 1, and ranges from 0 to 90 degrees in two directions. Besides describing geometry, the specification of this surface facilitates the application of internal pressure with the GMLOAD command. The GMLOAD specifies a constant pressure load of 3000 psi on the surface GMSURF 1. The commands are shown below.

```
$
$ Inner sphere
$
DEQATN  41      SX(U,V) =   5.0*SIN(U)*COS(V)
DEQATN  42      SY(U,V) =   5.0*SIN(U)*SIN(V)
DEQATN  43      SZ(U,V) =   5.0*COS(U)
DEQATN  44      SXU(U,V) =  5.0*COS(U)*COS(V)
DEQATN  45      SYU(U,V) =  5.0*COS(U)*SIN(V)
DEQATN  46      SZU(U,V) = -5.0*SIN(U)
DEQATN  47      SXV(U,V) = -5.0*SIN(U)*SIN(V)
DEQATN  48      SYV(U,V) =  5.0*SIN(U)*COS(V)
DEQATN  49      SZV(U,V) =  0.
DEQATN  51      SXUU(U,V) = -5.0*SIN(U)*COS(V)
DEQATN  52      SYUU(U,V) = -5.0*SIN(U)*SIN(V)
DEQATN  53      SZUU(U,V) = -5.0*COS(U)
DEQATN  54      SXVV(U,V) = -5.0*SIN(U)*COS(V)
DEQATN  55      SYVV(U,V) = -5.0*SIN(U)*SIN(V)
DEQATN  56      SZVV(U,V) =  0.
```

```
DEQATN  57      SXUV(U,V) = -5.0*COS(U)*SIN(V)
DEQATN  58      SYUV(U,V) =  5.0*COS(U)*COS(V)
DEQATN  59      SZUV(U,V) =  0.
$
GMSURF  1       MSCGRP1 0       0                                       +
+       EQUATION    0.0   1.570796327   0.0  1.570796327                +
+       41 42 43    44 45 46    47 48 49                                +
+       51 52 53    54 55 56    57 58 59
$
$ THIS SECTION CONTAINS LOADS
$
GMLOAD  1       1       1.      0.      0.      GMSURF 1        CONST   +
+       3000.
$
```

Since only an octant model is analyzed, the symmetric boundary conditions are required. The following partial commands demonstrate how GMSPC, GMSURF, and DEQATN work together. The GMSURF 201 is expressed in terms of radius (R) and an angle (U), ranging from 5 to 10 inches and 0 to 1.57 radians. GMSPC specifies the constraint in the basic z-direction.

```
$
$ Surface on x-y plane
$
DEQATN  61      SX(R,U) =   R*COS(U)
DEQATN  62      SY(R,U) =   R*SIN(U)
DEQATN  63      SZ(R,U) =   0.
DEQATN  64      SXU(R,U) =    COS(U)
DEQATN  65      SYU(R,U) =    SIN(U)
DEQATN  66      SZU(R,U) =  0.
DEQATN  67      SXV(R,U) = -R*SIN(U)
DEQATN  68      SYV(R,U) =  R*COS(U)
DEQATN  69      SZV(R,U) =  0.
DEQATN  71      SXUU(R,U) = 0.
DEQATN  72      SYUU(R,U) = 0.
DEQATN  73      SZUU(R,U) = 0.
DEQATN  74      SXVV(R,U) = -R*COS(U)
DEQATN  75      SYVV(R,U) = -R*SIN(U)
DEQATN  76      SZVV(R,U) =  0.
DEQATN  77      SXUV(R,U) = -  SIN(U)
DEQATN  78      SYUV(R,U) =    COS(U)
DEQATN  79      SZUV(R,U) =  0.
$
GMSURF  201     MSCGRP1 0       0                                       +
+       EQUATION    5.0  10.0  0.0  1.570796327                        +
+       61 62 63    64 65 66    67 68 69                                +
+       71 72 73    74 75 76    77 78 79
$
$ x-y plane in xyz basic
$
GMSPC   1       3       GMSURF  201
$
```

Other inputs, such as FEEDGE and FEFACE, are used to specify the geometry, the element boundaries, loads, and boundary conditions. For example, the following FEEDGE command is used to specify that the element edge between grids 3 and 4 will follow the geometry of GMCURV ID 1. The complete input file has been annotated and stored in the TPL directory for review.

```
$
$ Inner radius x-y plane
$
FEEDGE   1        3        4        0        GMCURV   1
```

Results

The p-element model requires two adaptivity cycles to converge to the desired error tolerance of 10% at p-value of4. Table 6-3 tabulates the results with the theory.

Table 6-3 Comparison of Displacements

Results			Percent Difference
Deflections	Theory	MSC Nastran	
Inner	1.20E-03	1.19E-03	1.0%
Outer	4.50E-04	4.45E-04	1.1%

Motor Blade – Normal Modes

Objectives

A normal mode analysis of a simplified motor blade is analyzed and compared using the h- and p-elements. The natural frequencies between 0 and 2000 Hz are computed.

Problem Description

Figure 6-4 shows the geometry of the motor blade consisting of a base and a blade. The whole assembly part is made of aluminum.

Modulus of Elasticity	= 1.0E7 psi
Poisson's ratio	= 0.3
Mass density	= 2.59E-4 lb/in^3

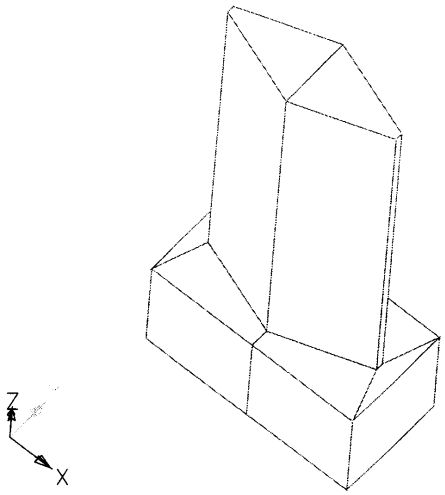


Figure 6-4 Motor Blade – p-Element Model

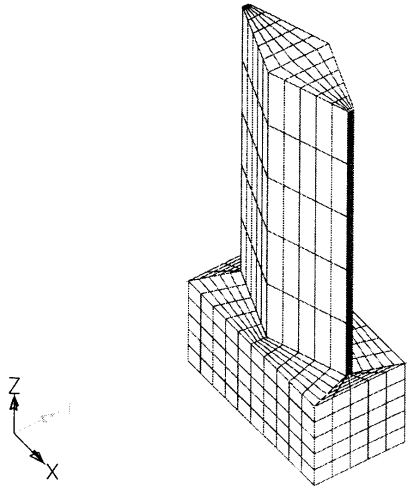


Figure 6-5 Motor Blade -- Fine h-Model

Theoretical Results

There is no theoretical result. Results of three different meshes of h-elements — coarse, medium, and fine — are compared with a p-element model.

Setup

Figure 6-4 and Figure 6-5 show the p-model and the h-model with fine mesh respectively. Since only the natural frequencies are requested, the geometry of both the h- and p-meshes are simplified and the boundaries are modeled with straight edges. If a detailed stress analysis is required, the geometry should be specified more precisely with higher-order geometry.

Results

All models produce six rigid body modes, so they are not discussed here. Table 6-4 tabulates the computed elastic natural frequencies between 0.0 to 2000.0 Hz, the number of elements, and the number of degrees of freedom with respect to different models. The p-element results agree very well with the fine meshed h-model showing the power and usefulness of the p-elements.

Table 6-4 Comparison of Natural Frequencies of a Motor Blade

Descriptions		No. of Elements	No. of DOFs	Elastic Modes			
	Mesh			1st	2nd	3rd	4th
h-model	Coarse	4	90	622.5	651.2	1204.9	–
	Medium	80	441	604.5	954.7	1289.6	1956.2
	Fine	1250	4878	659.7	1051.5	1404.8	1744.7
p-adaptive	cycle=1	10	276	737.7	1064.8	1526.9	–
model	cycle=2		1161	675.8	1070.8	1426.9	1796.6
	cycle=3		2112	674.6	1071.2	1427.8	1765.2

Torsion of Elliptical Bar

Objectives and Problem Description

This example illustrates torsion of an elliptical bar.

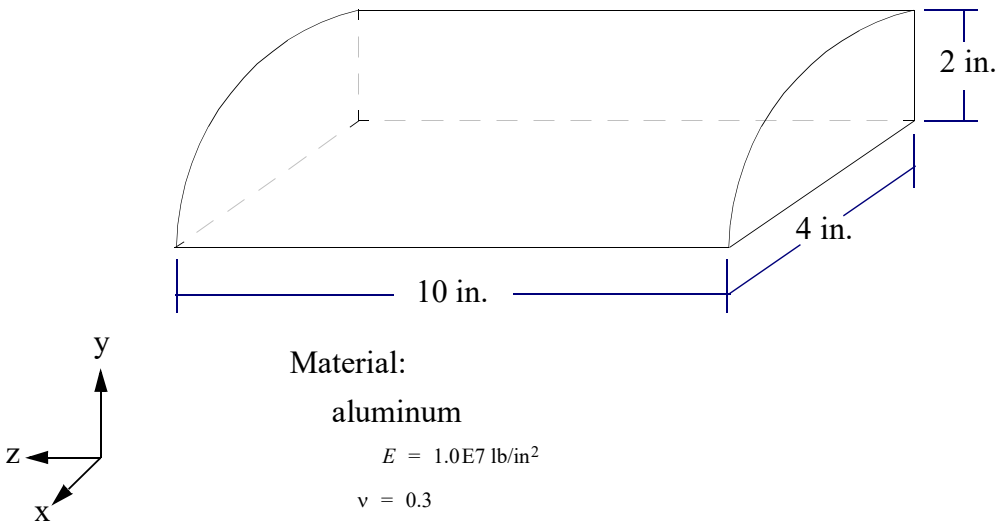


Figure 6-6 Octant of the Elliptical Bar Model

Setup

The model is constructed from four CPENTA elements. There are two coordinate systems used: the cylindrical coordinate system (CORD2C entry) to define grids of the model, and the convective coordinate system (GMCORD entry) to define loads applied to the bar.

The elliptical cross section of the model is described using GMCURV and GMSURF entries, both with the equation method. For the GMCURV entry:

The general ellipse equation:

$$\frac{x^2}{a^2}+\frac{y^2}{b^2} = 1$$

as shown in Figure 6-7.

Define x and y in polar coordinates:

$$x = r\cos\theta$$
$$y = r\sin\theta$$
$$r = \frac{ab}{\sqrt{a^2\sin^2\theta + b^2\cos^2\theta}} = \frac{a}{\sqrt{\cos^2\theta + \frac{a^2}{b^2}\sin^2\theta}}$$

$$x = \frac{a}{\sqrt{\cos^2\theta + \frac{a^2}{b^2}\sin^2\theta}}\cos\theta$$
$$y = \frac{a}{\sqrt{\cos^2\theta + \frac{a^2}{b^2}\sin^2\theta}}\sin\theta$$

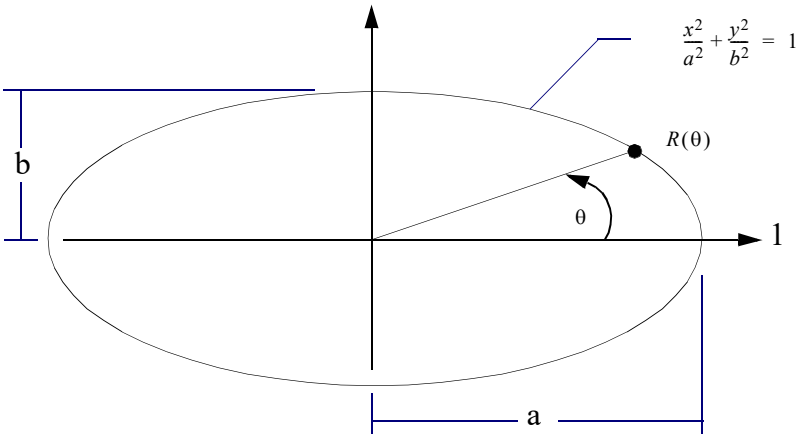


Figure 6-7 Elliptical Cross Section

For the GMSURF entry, three different types of parameterization were used. In the first method,

$$x_s = \frac{v}{\sqrt{\cos^2 u + \frac{a^2}{b^2}\sin^2 u}} \cos u \qquad 0.0 \le u \le \frac{\pi}{2}$$

$$y_s = \frac{v}{\sqrt{\cos^2 u + \frac{a^2}{b^2}\sin^2 u}} \sin u \qquad 0.0 \le v \le a$$

as shown in Figure 6-8.

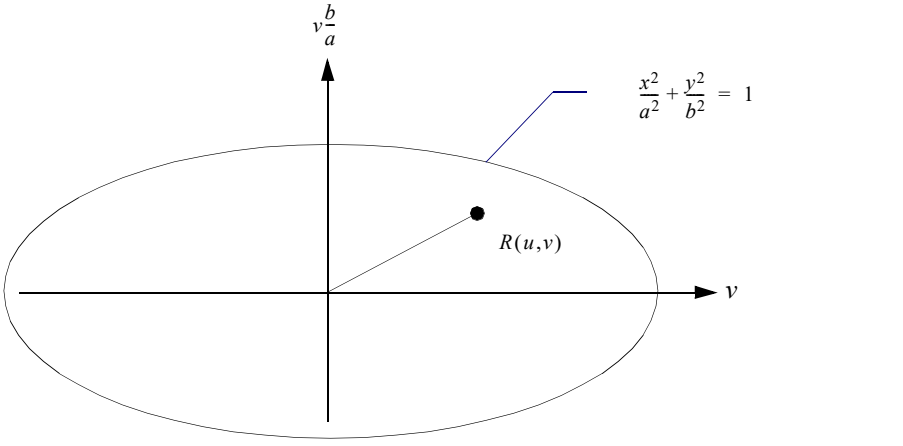


Figure 6-8 Equation of Ellipse using First Method

In the second method,

$$x_s = a v \cos u \qquad 0.0 \le u \le \frac{\pi}{2}$$

$$y_s = b v \sin u \qquad 0.0 \le v \le 1.0$$

The third method, using hyperbolic functions,

$$\begin{aligned} x + iy &= e(\cosh r \cos t + i \sinh r \sin t) \\ &= e \cosh(r + it) \end{aligned}$$

$$x = e \cosh r \cos t \qquad 0.0 < r < R$$

$$y = e \sinh r \sin t \qquad 0.0 < t < \frac{\pi}{2}$$

$$e = \sqrt{a^2 - b^2} \qquad R = \tanh^{-1}\left(\frac{b}{a}\right)$$

as shown in Figure 6-9, with contours of t displayed.

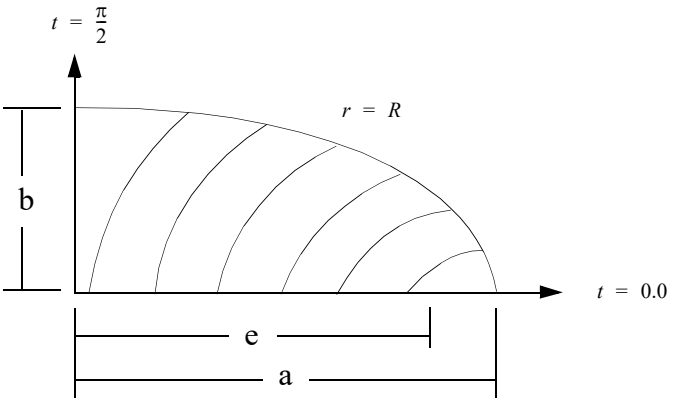


Figure 6-9 Equation of Ellipse using Third Method

For the GMSURF entry, the order of the input IDs of DEQATN entries for the surface is important. First, we input the IDs of DEQATN entries providing equations for the x, y, and z coordinates of the surface. Then, the IDs of DEQATN entries of the first derivatives of x, y, and z functions with respect to the surface parameter u. Then, the IDs of DEQATN entries of the first derivatives of x, y, and z functions with respect to the surface parameter v. Then, the IDs of DEQATN entries of the second derivatives of x, y, and z functions with respect to the surface parameter u. Then, the IDs of DEQATN entries of the second derivatives of x, y, and z functions with respect to the surface parameter v. Finally, the IDs of DEQATN of the mixed second derivatives of the x, y, and z functions with respect to both surface parameters u and v.

The torsion in this example is defined by applying an edge load (GMLOAD entry) using the convective coordinate system (GMCORD entry). The load is applied tangent to the GMCURV 3. This is illustrated in Figure 6-10. Note that an edge load on solid elements can cause singularities, but it is used here for demonstration purposes.

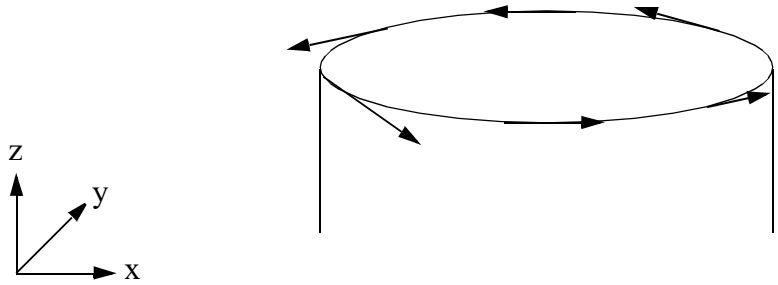


Figure 6-10 Torsional Load on Shaft

The input file of this model is listed in Listing 6-3.

Listing 6-3 Input File for the Model

```
ID TORSION, STATIC
TIME 500
SOL 101 $ STATIC ANALYSIS
CEND
TITLE = TORSION, ELLIPSE CSA
SUBTITLE = METHOD 2
$
$ DEFINE ANALYSIS ENTRY
$
ADAPT = 100
$
LOAD = 1
SPC = 1
$
$ DEFINE OUTPUT REQUEST OPTION
$
DATAREC = 100
OUTRCV = 200
VUGRID (PUNCH) = ALL
$
$ DEFINE THE SETS OF ELEMENTS REQUIRED FOR THE ADAPTIVITY ANALYSIS
$ THIS IS REQUIRED WHEN SETTYP=SET IN PVAL ENTRY
$ (REFER TO PVAL ENTRY ALSO)
$
SETS DEFINITION
SET 120 = ALL
$
BEGIN BULK
PARAM   VUPENTA CPENTA
$
$ BOTH THE ADAPT AND PVAL ENTRY ARE USED TO
$ DEFINE ADAPTIVITY OPTION ENTRY FOR ALL ELEMENTS
$
$ FOR THE ADAPT ENTRY:
$ ID      = ADAPT ENTRY ID SELECTED IN CASE CONTROL
$ ADGEN   = ID OF THE FIRST PVAL ENTRY GENERATED IN THE ADAPTIVE PROCESS;
$           MUST BE LARGER THAN PSTRID,PMINID, AND PMAXID
$ MAXITER = NUMBER OF ITERATIONS REQUESTED BEFORE THE ADAPTIVE PROCESS STOPPED
```

```
$ PSTRTID = ID OF THE STARTING PVAL ENTRY
$ PMINID  = ID OF THE MINIMUM PVAL ENTRY
$ PMAXID  = ID OF THE MAXIMUM PVAL ENTRY
$ PART    = ANY NAME CHOSEN FOR THE ELEMENTS;
$           USED FOR ERROR ESTIMATOR REFERENCE
$ ELSETID = ID OF SETS OF ELEMENTS REQUESTED FOR ADAPTIVE PROCESS
$ TYPE    = EBEP : THE P-ORDER IS INCREASED ELEMENTS-BY-ELEMENT
$ ERREST  = ERROR ESTIMATOR METHOD USED FOR ADAPTIVE PROCESS
$ ERRTOL  = ERROR TOLERANCE (IN FRACTION), REQUIRED IF MAXITER IS NOT DEFINED
$ SIGTOL  = STRESS TOLERANCE
$
$       SID     ADGEN   MAXITER PSTRID  PMINID  PMAXID
ADAPT   100     1001    10      10      20      30                      +000001
++000001PART=ALL,ELSET=120,TYPE=EBEP,ERREST=1,ERRTOL=0.02,SIGTOL=5.0    +000002
$
$ DEFINE THE POLYNOMIAL ORDER DISTRIBUTION:
$ PVAL ID = ID OF PVAL DEFINED IN THE ADAPT ENTRY ABOVE
$ PX      = THE POLYNOMIAL ORDER DISTRIBUTION IN X-DIRECTION
$ PY      = THE POLYNOMIAL ORDER DISTRIBUTION IN Y-DIRECTION
$ PZ      = THE POLYNOMIAL ORDER DISTRIBUTION IN Z-DIRECTION
$ SETTYP  = TYPE OF SET PROVIDED. IT CAN BE EITHER SET OR ELID
$ ID      = SET ID OR ELEMENT ID, WHICHEVER IS APPROPRIATE
$
$       ID      Px      Py      Pz              SETTYP  ID
PVAL    10      2       2       2               SET     120
PVAL    20      2       2       2               SET     120
PVAL    30      4       4       4               SET     120
$
$ DEFINE THE OUTPUT REQUEST OPTION
$ REQUEST DISPLACEMENT AND STRESS OUTPUT FOR SET 120 FOR THE LAST
$ ADAPTIVITY ANALYSIS
$
OUTPUT  100                                                             +000003
++000003ELSET=120,DISP=PRINT,STRESS=PRINT,STRAIN=NONE                   +000004
++000004FIRST=NO,LAST=YES,BY=2                                          +000005
$
$ THE OUTPUT COORDINATE SYSTEM IS THE BASIC COORDINATE SYSTEM (CID=0)
$
OUTRCV  200     120                                                     +000003
++000003CID=0,VIEW=2*2*2                                                +000004
$
$ DEFINE THE CYLINDRICAL COORDINATE SYSTEM FOR
$ GRID POINTS
$
CORD2C  1       0       0.0     0.0     0.0     0.0     0.0     1.0     +000005
++0000051.0     0.0     0.0                                             +000006
$
$ DEFINE THE CONVECTIVE COORDINATE SYSTEM FOR THE APPLIED LOAD
$ IF GMCURV IS USED AS THE ENTITY TO DEFINE THE COORDINATE SYSTEM,
$ TWO ID'S ARE REQUIRED. THE FIRST ID IS THE GMCURV ID AND THE SECOND
$ ONE IS THE CORRESPONDING GMSURF ID
$
$       CID     ENTITY  ID 1    ID 2
GMCORD  10      GMCURV  3       2
$
$ DEFINE PENTA ELEMENTS
$
```

```
CPENTA  1       1       12      1       2       11      4       5
CPENTA  2       1       12      2       3       11      5       6
CPENTA  3       1       11      4       5       10      7       8
CPENTA  4       1       11      5       6       10      8       9
$
$ DEFINE GRID POINTS
$
GRID    1       1       4.      0.0     0.0
GRID    2       1       2.52982245.     0.
GRID    3       1       2.0     90.0    0.0
GRID    4       1       4.0     0.0     5.0
GRID    5       1       2.52982245.0    5.0
GRID    6       1       2.0     90.0    5.0
GRID    7       1       4.0     0.0     10.0
GRID    8       1       2.52982245.0    10.0
GRID    9       1       2.0     90.0    10.0
GRID    10              0.0     0.0     10.0
GRID    11              0.0     0.0     5.0
GRID    12              0.0     0.0     0.0
$
$ DEFINE FEEDGES
$ FEEDGES BY DEFAULT ARE STRAIGHT LINES
$ GMCURVS ARE REQUIRED FOR FEEDGES THAT DEFINE CURVED EDGES
$
$ FEEDGE 18 AND FEEDGE 19 MAKE GMCURV 1
$ FEEDGE 20 AND FEEDGE 21 MAKE GMCURV 2
$ FEEDGE 22 AND FEEDGE 23 MAKE GMCURV 3
$
$       ID      GRID 1  GRID 2  CID     GEOMIN  ID
FEEDGE  1       12      1       0
FEEDGE  2       12      2       0
FEEDGE  3       12      3       0
FEEDGE  4       11      4       0
FEEDGE  5       11      5       0
FEEDGE  6       11      6       0
FEEDGE  7       10      7       0
FEEDGE  8       10      8       0
FEEDGE  9       10      9       0
FEEDGE  10      12      11      0
FEEDGE  11      1       4       0
FEEDGE  12      2       5       0
FEEDGE  13      3       6       0
FEEDGE  14      11      10      0
FEEDGE  15      4       7       0
FEEDGE  16      5       8       0
FEEDGE  17      6       9       0
FEEDGE  18      1       2       0       GMCURV  1
FEEDGE  19      2       3       0       GMCURV  1
FEEDGE  20      4       5       0       GMCURV  2
FEEDGE  21      5       6       0       GMCURV  2
FEEDGE  22      7       8       0       GMCURV  3
FEEDGE  23      8       9       0       GMCURV  3
$
$ DEFINE CURVE GEOMETRY USING EQUATION METHOD
$
DEQATN  1       X(U) = 4.0*COS(U)/(SQRT(1.0+3.0*SIN(U)*SIN(U)))
DEQATN  2       Y(U) = 4.0*SIN(U)/(SQRT(1.0+3.0*SIN(U)*SIN(U)))
```

```
DEQATN   3       Z(U) = 0.0
DEQATN   4       Z1(U) = 5.0
DEQATN   5       Z2(U) = 10.0
DEQATN   6       XP(U) = -16.0*SIN(U)/((1.0+3.0*SIN(U)*SIN(U))**1.5)
DEQATN   7       YP(U) = 4.0*COS(U)/((1.0+3.0*SIN(U)*SIN(U))**1.5)
DEQATN   8       ZP(U) = 0.0
DEQATN   9       XPP(U) =(-16.0*COS(U)+96.0*SIN(U)*SIN(U)*COS(U))/        +000009
++000009         (1.0+3.0*SIN(U)*SIN(U))**2.5                             +000010
DEQATN   10      YPP(U) =(-4.0*SIN(U)*(10.0-6.0*SIN(U)*SIN(U)))/          +000011
++000011         ((1.0+3.0*SIN(U)*SIN(U))**2.5)                           +000012
DEQATN   11      ZPP(U) =0.0
$
$
GMCURV   1       MSCGRP1 0       0                                         +000013
++000013EQUATION,0.0,1.5708,1,2,3,6,7,                                    +000014
++0000148        9       10      11                                       +000015
GMCURV   2       MSCGRP1 0       0                                         +000016
++000016EQUATION,0.0,1.5708,1,2,4,6,7,                                    +000017
++0000178        9       10      11                                       +000018
GMCURV   3       MSCGRP1 0       0                                         +000019
++000019EQUATION,0.0,1.5708,1,2,5,6,7,                                    +000020
++0000208        9       10      11                                       +000021
$
$ DEFINE FEFACES
$
$       ID      GRID 1  GRID 2  GRID 3  GRID 4  CID     SURFID
FEFACE   1       12      1       2               0       1
FEFACE   2       12      2       3               0       1
FEFACE   3       11      4       5               0
FEFACE   4       11      5       6       0
FEFACE   5       10      7       8               0       2
FEFACE   6       10      8       9               0       2
FEFACE   7       12      1       4       11      0
FEFACE   8       12      3       6       11      0
FEFACE   9       11      4       7       10      0
FEFACE   10      11      6       9       10      0
$
$ DEFINE SURFACE GEOMETRY USING EQUATION METHOD
$
DEQATN   13      X(U,V)   = 4.0*V*COS(U)
DEQATN   14      Y(U,V)   = 2.0*V*SIN(U)
DEQATN   15      Z1(U,V) = 0.0
DEQATN   16      Z2(U,V) = 10.0
DEQATN   17      XU(U,V) = -4.0*V*SIN(U)
DEQATN   18      YU(U,V) = 2.0*V*COS(U)
DEQATN   19      ZU(U,V) = 0.0
DEQATN   20      XV(U,V) = 4.0*COS(U)
DEQATN   21      YV(U,V) = 2.0*SIN(U)
DEQATN   22      ZV(U,V) = 0.0
DEQATN   23      XUU(U,V)= -4.0*V*COS(U)
DEQATN   24      YUU(U,V)= -2.0*V*SIN(U)
DEQATN   25      ZUU(U,V)= 0.0
DEQATN   26      XVV(U,V)= 0.0
DEQATN   27      YVV(U,V)= 0.0
DEQATN   28      ZVV(U,V)= 0.0
DEQATN   29      XUV(U,V)= -4.0*SIN(U)
DEQATN   30      YUV(U,V)= 2.0*COS(U)
```

```
DEQATN  31      ZUV(U,V)= 0.0
$
$
GMSURF  1       MSCGRP1 0       0                                       +000022
++000022EQUATION,0.0,1.5708,0.0,1.0,13,14,15,                           +000023
++00002317,18,19,20,21,22,23,24,                                        +000024
++00002425,26,27,28,29,30,31                                            +000025
GMSURF  2       MSCGRP1 0       0                                       +000026
++000026EQUATION,0.0,1.5708,0.0,1.0,13,14,16,                           +000027
++00002717,18,19,20,21,22,23,24,                                        +000028
++00002825,26,27,28,29,30,31                                            +000029
$
$ DEFINE CONSTRAINT IN THE X-DIRECTION
$
$       SID     C       ENTITY  ID
GMSPC   1       1       FEFACE  1
GMSPC   1       1       FEFACE  2
GMSPC   1       1       FEFACE  7
GMSPC   1       1       FEFACE  9
$
$ DEFINE CONSTRAINT IN THE Y-DIRECTION
$
$       SID     C       ENTITY  ID
GMSPC   1       2       FEFACE  1
GMSPC   1       2       FEFACE  2
GMSPC   1       2       FEFACE  8
GMSPC   1       2       FEFACE  10
$
$ DEFINE CONSTRAINT IN THE Z-DIRECTION
$
$       SID     C       ENTITY  ID
GMSPC   1       3       FEFACE  7
GMSPC   1       3       FEFACE  8
GMSPC   1       3       FEFACE  9
GMSPC   1       3       FEFACE  10
$
$ DEFINE THE MATERIAL PROPERTY (ALUMINUM)
$
MAT1    1       1.+7            0.3
PSOLID  1       1                       GRID            SMECH
$
$ CONSTANT LOAD OF 100 LB/INCH IS APPLIED TANGENTIALLY TO
$ GMCURV 3
$
$ LID      = LOAD IDENTIFICATION NUMBER DEFINED IN CASE CONTROL SECTION
$ CID      = COORDINATE SYSTEM USED TO DEFINE THE APPLIED LOAD
$ N1,N2,N3 = DIRECTION OF THE LOAD VECTOR
$ ENTITY   = ENTITY AT WHICH LOAD IS APPLIED TO
$ ID       = ID OF THE ENTITY
$ METHOD   = METHOD USED TO DEFINE THE LOAD
$
$       LID     CID     N1      N2      N3      ENTITY  ID      METHOD
GMLOAD  1       10      1.0     0.0     0.0     GMCURV  3       CONSTANT +000007
$
$       VALUE OF LOAD DENSITY
++000007100.0                                                           +000008
$
```

```
ENDDATA
```

Results

The OUTPUT entry requests the displacements and stress output solely for the last adaptivity iteration. An abridged output for this model is shown in Figure 6-11.

```
1    TORSION, ELLIPSE CSA                                                DECEMBER  22, 1993  MSC/NASTRAN  12/20/93   PAGE    73
      METHOD 2
0*** USER INFORMATION MESSAGE 5293 FOR DATA BLOCK KLL
    LOAD SEQ. NO.            EPSILON              EXTERNAL WORK      EPSILONS LARGER THAN  .001 ARE FLAGGED WITH ASTERISKS
               1          5.2309875E-15         1.6239751E-02
 *** USER INFORMATION MESSAGE 6502 (ADAPT)
     THE CURRENT SUPERELEMENT REQUIRES NO FURTHUR P-ADAPTIVE ANALYSIS BECAUSE THERE ARE NO CHANGES IN P-VALUES.
0*** USER WARNING MESSAGE 4594. THE POSTCDB CONTAINS NO LEGITIMATE SURFACE OR VOLUME DEFINITIONS.
1    TORSION, ELLIPSE CSA                                                DECEMBER  22, 1993  MSC/NASTRAN  12/20/93   PAGE    74
      METHOD 2                                                          ADAPTIVITY INDEX=      3
0

                                        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      G       .0              .0              .0              .0              .0              .0
             2      G       .0              .0             -8.692895E-07    .0              .0              .0
             3      G       .0              .0              .0              .0              .0              .0
             4      G       .0             1.357096E-05     .0              .0              .0              .0
             5      G     -3.062882E-06   -2.173771E-06    -1.839771E-06    .0              .0              .0
             6      G      1.593498E-05     .0              .0              .0              .0              .0
             7      G       .0             1.513064E-04     .0              .0              .0              .0
             8      G      1.887783E-06    -4.328616E-05     1.990558E-05    .0              .0              .0
             9      G      1.121671E-04     .0              .0              .0              .0              .0
            10      G       .0              .0              .0              .0              .0              .0
            11      G       .0              .0              .0              .0              .0              .0
            12      G       .0              .0              .0              .0              .0              .0
     201001001      G       .0              .0             -6.008400E-23    .0              .0              .0
     201001002      G       .0              .0             -3.919389E-23    .0              .0              .0
     201001003      G       .0              .0             -1.764301E-22    .0              .0              .0
     201001004      G       .0              .0             -4.865105E-07    .0              .0              .0
     201001005      G       .0              .0             -8.088551E-07    .0              .0              .0
     201001006      G       .0              .0             -8.692895E-07    .0              .0              .0
     201001007      G     -3.961429E-23    1.151926E-22     6.248783E-24    .0              .0              .0
     201001008      G     -1.317319E-22    2.320618E-06    -1.101103E-22    .0              .0              .0
     201001009      G     -1.197587E-22    3.720583E-06    -2.934319E-24    .0              .0              .0
     201001010      G     -7.882488E-07    1.256443E-06    -2.743236E-07    .0              .0              .0
     201001011      G     -1.074418E-06    2.716110E-06    -1.095060E-06    .0              .0              .0
     201001012      G     -1.967381E-06    1.468333E-06    -7.467912E-07    .0              .0              .0
     201001013      G     -9.475001E-23    4.980112E-22    -3.150599E-23    .0              .0              .0
     201001014      G     -3.235471E-22    3.547284E-06    -1.786648E-22    .0              .0              .0
     201001015      G     -4.842875E-22    1.357096E-05     6.322074E-23    .0              .0              .0
     201001016      G     -8.565316E-07    4.229675E-06    -7.164187E-07    .0              .0              .0
     201001017      G     -3.000400E-06    4.619390E-06    -5.842737E-07    .0              .0              .0
     201001018      G     -3.062882E-06   -2.173771E-06    -1.839771E-06    .0              .0              .0
     201002001      G       .0              .0              3.004200E-23    .0              .0              .0
     201002002      G       .0              .0             -4.865105E-07    .0              .0              .0
     201002003      G       .0              .0             -8.692895E-07    .0              .0              .0
     201002004      G       .0              .0              2.921973E-23    .0              .0              .0
     201002005      G       .0              .0              5.445964E-07    .0              .0              .0
     201002006      G       .0              .0              4.219457E-22    .0              .0              .0
     201002007      G     -1.446019E-22   -5.887475E-23    -3.124391E-24    .0              .0              .0
     201002008      G     -7.882488E-07    1.256443E-06    -2.743236E-07    .0              .0              .0
     201002009      G     -1.967381E-06    1.468333E-06    -7.467912E-07    .0              .0              .0
     201002010      G     -8.927043E-07    3.961755E-23    -6.052314E-24    .0              .0              .0
     201002011      G     -2.660616E-06    7.911209E-07     5.446515E-08    .0              .0              .0
     201002012      G     -7.240385E-08    1.330381E-22    -1.194728E-22    .0              .0              .0
     201002013      G     -8.723581E-22   -3.184499E-22     1.575299E-23    .0              .0              .0
     201002014      G     -8.565316E-07    4.229675E-06    -7.164187E-07    .0              .0              .0
     201002015      G     -3.062882E-06   -2.173771E-06    -1.839771E-06    .0              .0              .0
     201002016      G     -1.335821E-06    1.364713E-22     2.932544E-23    .0              .0              .0
     201002017      G     -7.250181E-06    1.251031E-06     1.185184E-06    .0              .0              .0
     201002018      G      1.593498E-05    2.834062E-22     4.533070E-22    .0              .0              .0
     201003001      G     -9.475001E-23    4.980112E-22    -3.150599E-23    .0              .0              .0
     201003002      G     -3.235471E-22    3.547284E-06    -1.786648E-22    .0              .0              .0
```

Figure 6-11 Abridged Output for the Model

```
1    TORSION, ELLIPSE CSA                                                    DECEMBER  22, 1993  MSC/NASTRAN  12/20/93   PAGE    77
      METHOD 2                                                              ADAPTIVITY INDEX=      3
0                                                                           PVAL ID=    1002 (P1,P2,P3=  3  2  3), SUBCASE        1
          S T R E S S E S   I N   P - V E R S I O N   P E N T A H E D R O N   S O L I D   E L E M E N T S   ( P E N T A )
  VIEW-ELEMENT ID=100001001, P-ELEMENT ID =      1, OUTPUT COORD. ID=      0, P OF EDGES =  3  3  4  4  3  4  4  4  4
    VUGRID    SIGMA-X/1      SIGMA-Y/2      SIGMA-Z/3       TAU-XY         TAU-YZ         TAU-XZ     MEAN-PRESSURE    VON-MISES
 201001001  2.507561E-15   2.507561E-15   5.850975E-15  -2.450845E-31  -3.490772E+00   3.045360E-16
            3.490772E+00  -3.490772E+00   2.881587E-15                                                -4.069153E-15   6.046195E+00
 201001002 -9.512720E-17  -6.704689E-16   3.533783E-16   7.719806E-32   7.080933E-01  -7.649862E-16
            7.080933E-01  -7.080933E-01  -2.876032E-16                                                 2.068900E-16   1.226454E+00
 201001004  5.034403E+00   5.034403E+00   1.174694E+01   4.981801E-15  -1.109684E+01  -4.317494E+00
            2.076182E+01  -3.980472E+00   5.034403E+00                                                -7.271916E+00   2.168871E+01
 201001007  1.006225E+01   2.347857E+01   1.006225E+01  -2.168649E+00   3.630430E-01  -1.325106E-16
            2.382975E+01   9.711069E+00   1.006225E+01                                                -1.453436E+01   1.394641E+01
 201001008 -3.267787E+00  -7.624836E+00  -3.267787E+00  -1.073696E+00  -6.464741E-01  -1.551473E-15
           -2.933002E+00  -7.959620E+00  -3.267787E+00                                                 4.720137E+00   4.867867E+00
 201001010 -1.527811E+00  -2.566048E+00  -9.243130E-01  -5.258331E-02  -3.754325E-01  -2.206367E+00
            1.018136E+00  -3.538538E+00  -2.497769E+00                                                 1.672724E+00   4.135702E+00
          S T R E S S E S   I N   P - V E R S I O N   P E N T A H E D R O N   S O L I D   E L E M E N T S   ( P E N T A )
  VIEW-ELEMENT ID=100001002, P-ELEMENT ID =      1, OUTPUT COORD. ID=      0, P OF EDGES =  3  3  4  4  3  4  4  4  4
    VUGRID    SIGMA-X/1      SIGMA-Y/2      SIGMA-Z/3       TAU-XY         TAU-YZ         TAU-XZ     MEAN-PRESSURE    VON-MISES
 201001002 -9.512720E-17  -6.704689E-16   3.533783E-16   7.719806E-32   7.080933E-01  -7.649862E-16
            7.080933E-01  -7.080933E-01  -2.876032E-16                                                 2.068900E-16   1.226454E+00
 201001003  2.474018E-14   5.246847E-14   2.999879E-14  -1.088513E-15   2.914146E+00  -2.396744E-15
            2.914146E+00  -2.914146E+00   2.316292E-14                                                -3.525450E-14   5.047449E+00
 201001005 -7.025239E-01  -7.025239E-01  -1.639222E+00   3.874331E-15   1.205412E+01  -6.993091E+00
            1.277274E+01  -1.511449E+01  -7.025239E-01                                                 1.014757E+00   2.415559E+01
 201001008 -3.267787E+00  -7.624836E+00  -3.267787E+00  -1.073696E+00  -6.464741E-01  -1.551473E-15
           -2.933002E+00  -7.959620E+00  -3.267787E+00                                                 4.720137E+00   4.867867E+00
 201001009  6.025518E+00   1.405954E+01   6.025518E+00   7.571502E-01   6.719486E+00   7.331966E-16
            1.790772E+01   2.177344E+00   6.025518E+00                                                -8.703526E+00   1.420281E+01
 201001011  2.021004E+00  -5.901449E+00  -1.849734E+00   2.802238E+00   2.882026E+00  -1.129690E+00
            2.922884E+00  -8.277588E+00  -3.754748E-01                                                 1.910060E+00   9.969281E+00
1    TORSION, ELLIPSE CSA                                                    DECEMBER  22, 1993  MSC/NASTRAN  12/20/93   PAGE    78
      METHOD 2                                                              ADAPTIVITY INDEX=      3
0                                                                           PVAL ID=    1002 (P1,P2,P3=  3  2  3), SUBCASE        1
          S T R E S S E S   I N   P - V E R S I O N   P E N T A H E D R O N   S O L I D   E L E M E N T S   ( P E N T A )
  VIEW-ELEMENT ID=100001003, P-ELEMENT ID =      1, OUTPUT COORD. ID=      0, P OF EDGES =  3  3  4  4  3  4  4  4  4
    VUGRID    SIGMA-X/1      SIGMA-Y/2      SIGMA-Z/3       TAU-XY         TAU-YZ         TAU-XZ     MEAN-PRESSURE    VON-MISES
 201001005 -7.025239E-01  -7.025239E-01  -1.639222E+00   3.874331E-15   1.205412E+01  -6.993091E+00
            1.277274E+01  -1.511449E+01  -7.025239E-01                                                 1.014757E+00   2.415559E+01
 201001004  5.034403E+00   5.034403E+00   1.174694E+01   4.981801E-15  -1.109684E+01  -4.317494E+00
            2.076182E+01  -3.980472E+00   5.034403E+00                                                -7.271916E+00   2.168871E+01
 201001002 -9.512720E-17  -6.704689E-16   3.533783E-16   7.719806E-32   7.080933E-01  -7.649862E-16
            7.080933E-01  -7.080933E-01  -2.876032E-16                                                 2.068900E-16   1.226454E+00
 201001011  2.021004E+00  -5.901449E+00  -1.849734E+00   2.802238E+00   2.882026E+00  -1.129690E+00
            2.922884E+00  -8.277588E+00  -3.754748E-01                                                 1.910060E+00   9.969281E+00
 201001010 -1.527811E+00  -2.566048E+00  -9.243130E-01  -5.258331E-02  -3.754325E-01  -2.206367E+00
            1.018136E+00  -3.538538E+00  -2.497769E+00                                                 1.672724E+00   4.135702E+00
 201001008 -3.267787E+00  -7.624836E+00  -3.267787E+00  -1.073696E+00  -6.464741E-01  -1.551473E-15
           -2.933002E+00  -7.959620E+00  -3.267787E+00                                                 4.720137E+00   4.867867E+00
          S T R E S S E S   I N   P - V E R S I O N   P E N T A H E D R O N   S O L I D   E L E M E N T S   ( P E N T A )
  VIEW-ELEMENT ID=100001004, P-ELEMENT ID =      1, OUTPUT COORD. ID=      0, P OF EDGES =  3  3  4  4  3  4  4  4  4
    VUGRID    SIGMA-X/1      SIGMA-Y/2      SIGMA-Z/3       TAU-XY         TAU-YZ         TAU-XZ     MEAN-PRESSURE    VON-MISES
 201001004  5.034403E+00   5.034403E+00   1.174694E+01   4.981801E-15  -1.109684E+01  -4.317494E+00
            2.076182E+01  -3.980472E+00   5.034403E+00                                                -7.271916E+00   2.168871E+01
 201001005 -7.025239E-01  -7.025239E-01  -1.639222E+00   3.874331E-15   1.205412E+01  -6.993091E+00
            1.277274E+01  -1.511449E+01  -7.025239E-01                                                 1.014757E+00   2.415559E+01
 201001006 -3.200166E-02  -3.200166E-02  -7.467054E-02  -8.733983E-14   1.493832E+01  -4.544772E+00
            1.556104E+01  -1.566771E+01  -3.200166E-02                                                 4.622462E-02   2.704490E+01
 201001010 -1.527811E+00  -2.566048E+00  -9.243130E-01  -5.258331E-02  -3.754325E-01  -2.206367E+00
            1.018136E+00  -3.538538E+00  -2.497769E+00                                                 1.672724E+00   4.135702E+00
 201001011  2.021004E+00  -5.901449E+00  -1.849734E+00   2.802238E+00   2.882026E+00  -1.129690E+00
            2.922884E+00  -8.277588E+00  -3.754748E-01                                                 1.910060E+00   9.969281E+00
 201001012  2.319424E+00  -3.111391E+00  -1.438776E+00  -7.330344E+00   8.622318E+00  -9.914967E+00
            1.681246E+01  -1.124202E+01  -7.801181E+00                                                 7.435812E-01   2.650212E+01
```

Figure 6-12 Abridged Output for the Model (continued)

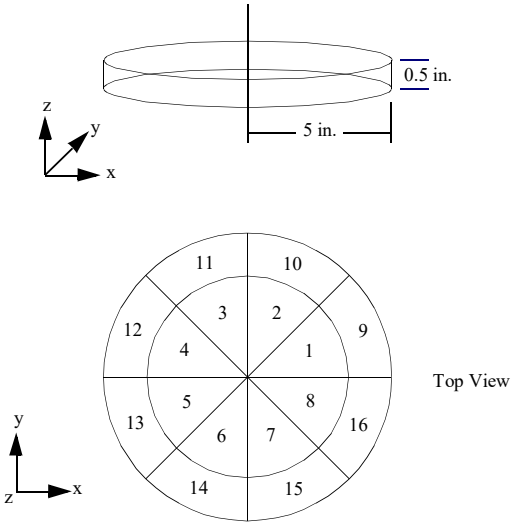
Circular Solid Model

Objectives and Problem Description

This example is a circular solid with a simply supported edge. It is made of aluminum and has the following properties:

E = 1.0E7 lb/in^2
ρ_w = 0.098 lbf/in^3
ν = 0.3

It is illustrated in Figure 6-13.



element 1 thru 8: CPENTA elements
element 9 thru 16: CHEXA elements (boundary elements)

Figure 6-13 The Circular Solid Model

Theoretical Results

The behavior of this circular solid model can be compared to the theoretical circular plate model described in Reference 1.

$$f = \frac{K_n}{2\pi}\sqrt{\frac{Dg}{wr^4}}$$

where:

$$D = \frac{Et^3}{12(1-\nu^2)}$$

w = uniform load per unit area including own weight

$$= \frac{(\rho_w)(Volume)}{Area}$$

$$= \frac{(\rho_w)(Area)(t)}{Area}$$

$$= (\rho_m)(g)(t)$$

$$f = \frac{K_n}{2\pi}\sqrt{\frac{Et^2}{12(1-\nu^2)(\rho_m)(r^4)}} \text{ cycles/sec}$$

where:

E = modulus of elasticity
K_n = constant, n refers to the mode of vibration
t = thickness of the plate
r = radius of the plate
ν = Poisson's ratio

For aluminum,

E = 1.0E7 lb/in^2
ν = 0.3
ρ_w = 0.098 lbf/in^3
ρ_m = 2.538E-4 lbm/in^3

For the circular plate, radius $(r) = 5$ inches and thickness $(t) = 0.5$ inches. The theoretical natural frequencies for all modes:

$$f = \frac{K_n}{2\pi}\sqrt{\frac{(10^7)(0.5^2)}{12(1-0.3^2)(2.538\times10^{-4})(5^4)}} = 191.2\ K_n \text{ cycles/sec}$$

The theoretical natural frequencies for the first several modes are calculated and tabulated in Table 6-5.

Table 6-5 Theoretical Natural Frequency Values for Circular Plate

Mode	*Kn*	Natural Frequency (cycles/sec)
Fundamental	4.99	954.1
One Nodal Diameter	13.9	2657.7
Two Nodal Diameters	25.7	4913.9
One Nodal Circle	29.8	5697.8

Setup

The model is constructed from 8 CPENTA elements and 8 CHEXA elements. The CHEXA elements are the boundary elements. The circular edge of the model is described using the circle equation (GMCURV entry, method=equation). Since the GMCURV can not be a closed curve, the full circle must be described with multiple GMCURV entries. In this model, one full circle is defined by two GMCURV entries. The first GMCURV entry has an input range from 0.0 to 3.1416 radians, and the second GMCURV entry has an input range from 3.1415 to 6.2832 radians. Note that there is some overlap, so that both segments include π. In addition, since the circle equation is a straightforward equation, the first and the second derivatives of the equation are not necessarily required. In this case, the derivatives are then computed numerically.

For the boundary conditions, GMSPC entries are used. Due to the hierarchy order which places grids as the highest priority, GMSPC entries that are imposed on grid 4 and 8 in the x-direction will override GMSPC entries on the adjacent FEFACES in the z-direction. Thus, it is necessary to add two more GMSPC entries for grids 4 and 8 in the z-direction to restore the constraints in the z-direction.

For the material property, the MAT1 entry is used. In the MAT1 entry, the weight density term ρ_w must be converted to the mass density ρ_m for consistency of units. Therefore,

$$\rho_m = (0.098\ \text{lbf/in}^3)(1/32.174\ \text{ft/s}^2)(1/12\ \text{in/ft})$$
$$= 2.538\text{E-4}\ \text{lbm/in}^3$$

The input file for this model is shown in Listing 6-4.

Listing 6-4 Input File for the Circular Solid Model

```
ID NORMAL, MODES
TIME 200
SOL 103 $ NORMAL MODE ANALYSIS
CEND
TITLE = CIRCULAR PLATE
SUBTITLE = EDGE SIMPLY SUPPORTED
METHOD = 100
$
$ DEFINE ADAPTIVITY ANALYSIS ENTRY
$
ADAPT = 1
$
$ OUTPUT REQUEST
$
DISPLACEMENT = ALL
ECHO = BOTH
$
SPC =1
$
$ REQUEST FOR VIEW COORDINATES OF THE P-ELEMENTS TO BE
$ PRINTED IN THE PUNCH FILE. THE VIEW COORDINATES
$ ARE THE POINTS AT WHICH THE DISPLACEMENTS ARE
$ CALCULATED.
$
VUGRID(PUNCH) = ALL
$
$ DEFINE THE SETS OF ELEMENTS REQUIRED FOR THE ADAPTIVITY
```

```
$ ANALYSIS. THIS IS REQUIRED WHEN SETTYP=SET IN PVAL ENTRY
$ ( REFER TO PVAL ENTRY ALSO )
$
SETS DEFINITION
SET 10 = ALL
$
BEGIN BULK
PARAM   POST    0
$
PARAM   VUHEXA  CHEXA
PARAM   VUPENTA CPENTA
$
$ BOTH THE ADAPT AND PVAL ENTRY ARE USED TO
$ DEFINE ADAPTIVITY OPTION ENTRY FOR ALL ELEMENTS,
$
$ FOR THE ADAPT ENTRY:
$ ID      = ADAPT ENTRY ID SELECTED IN CASE CONTROL
$ ADGEN   = ID OF THE FIRST PVAL ENTRY GENERATED IN THE ADAPTIVE PROCESS;
$           MUST BE LARGER THAN PSTRID, PMINID, AND PMAXID
$ MAXITER = NUMBER OF ITERATIONS REQUESTED BEFORE THE ADAPTIVE PROCESS STOPPED
$ PSTRTID = ID OF THE STARTING PVAL ENTRY
$ PMINID  = ID OF THE MINIMUM PVAL ENTRY
$ PMAXID  = ID OF THE MAXIMUM PVAL ENTRY
$ PART    = ANY NAME CHOSEN FOR THE ELEMENTS;
$           USED FOR ERROR ESTIMATOR REFERENCE
$ ELSETID = ID OF SETS OF ELEMENTS REQUESTED FOR ADAPTIVE PROCESS
$ TYPE    = EBEP : THE P-ORDER WILL INCREASE ONLY IN THE ELEMENTS WHOSE ERROR
$                  IS GREATER THAN ERROR TOLERANCE.
$ ERREST  = ERROR ESTIMATOR METHOD USED FOR ADAPTIVE PROCESS
$ ERRTOL  = ERROR TOLERANCE (IN FRACTION), REQUIRED IF MAXITER IS NOT DEFINED
$ SIGTOL  = STRESS TOLERANCE
$
$       ID      ADGEN   MAXITER PSTRTID PMINID  PMAXID
ADAPT   1       1001    5       10      20      30                      +000001
$
++000001PART=ALL,ELSET=10,TYPE=EBEP,ERREST=1,ERRTOL=0.02,SIGTOL=0.0     +000002
$

$ DEFINE THE POLYNOMIAL ORDER DISTRIBUTION:
$ PVAL ID = ID OF PVAL DEFINED IN THE ADAPT ENTRY ABOVE
$ Px      = THE POLYNOMIAL ORDER DISTRIBUTION IN X-DIRECTION
$ Py      = THE POLYNOMIAL ORDER DISTRIBUTION IN Y-DIRECTION
$ Pz      = THE POLYNOMIAL ORDER DISTRIBUTION IN Z-DIRECTION
$ SETTYP  = TYPE OF SET PROVIDED. IT CAN BE EITHER SET or ELID
$ ID      = SET ID or ELEMENT ID WHICHEVER IS APPROPIATE
$
$       ID      Px      Py      Pz              SETTYP  ID
PVAL    10      2       2       2               SET     10
PVAL    20      4       4       2               SET     10
PVAL    30      6       6       2               SET     10
$
$ DEFINE A CYLINDRICAL COORDINATE SYSTEM
$ USED IN DEFINING THE GRIDS
$
CORD2C  1       0       0.0     0.0     0.0     0.0     0.0     1.0     +000003
++0000031.0     0.0     0.0
$
```

```
$ DEFINE SOLID ELEMENTS
$ HEXA ELEMENTS ARE THE BOUNDARY ELEMENTS
$
CPENTA  1       1       1       21      22      10      29      30
CPENTA  2       1       1       22      23      10      30      31
CPENTA  3       1       1       23      24      10      31      32
CPENTA  4       1       1       24      25      10      32      33
CPENTA  5       1       1       25      26      10      33      34
CPENTA  6       1       1       26      27      10      34      35
CPENTA  7       1       1       27      28      10      35      36
CPENTA  8       1       1       28      21      10      36      29
CHEXA   9       1       2       3       22      21      11      12       +000005
++00000530      29                                                      +000006
CHEXA   10      1       3       4       23      22      12      13       +000007
++00000731      30                                                      +000008
CHEXA   11      1       4       5       24      23      13      14       +000009
++00000932      31                                                      +000010
CHEXA   12      1       5       6       25      24      14      15       +000011
++00001133      32                                                      +000012
CHEXA   13      1       6       7       26      25      15      16       +000013
++00001334      33                                                      +000014
CHEXA   14      1       7       8       27      26      16      17       +000015
++00001535      34                                                      +000016
CHEXA   15      1       8       9       28      27      17      18       +000017
++00001736      35                                                      +000018
CHEXA   16      1       9       2       21      28      18      11       +000019
++00001929      36
$
$DEFINE EIGRL CARD (LANCZOS METHOD)
$
EIGRL   100                     10
$
$DEFINE THE CURVE GEOMETRY, METHOD = EQUATION, U IS THE CURVE PARAMETER
$
$DEFINE THE OUTER BOTTOM CIRCLE OF RADIUS=5.0 AT Z=0.0 USING DEQATN 1,2,AND 3.
$THE OUTER BOTTOM CIRCLE IS DEFINED BY TWO GMCURV ENTRIES:
$GMCURV 1 WITH THE RANGE OF CURVE PARAMETER,U = 0.0    TO 3.1416 RADIANS
$GMCURV 2 WITH THE RANGE OF CURVE PARAMETER,U = 3.1415 TO 6.2832 RADIANS
$
DEQATN  1       X(U) = 5.0*COS(U)
DEQATN  2       Y(U) = 5.0*SIN(U)
DEQATN  3       Z(U) = 0.0
$
GMCURV  1       MSCGRP1 0       0                                       +000021
++000021EQUATION,0.0,3.1416,1,2,3                                       +000022


GMCURV  2       MSCGRP1 0       0                                       +000025
++000025EQUATION,3.1415,6.2832,1,2,3                                    +000026
$
$ DEFINE THE TOP OUTER CIRCLE USING THE SAME EQUATION, DEQATN 1 AND DEQATN 2,
$ BUT WITH Z=0.5 (DEQATN 4). THIS TOP OUTER CIRCLE IS DEFINED BY TWO GMCURV
$ ENTRIES:
$ GMCURV 3 WITH THE RANGE OF CURVE PARAMETER,U = 0.0    TO 3.1416 RADIANS
$ GMCURV 4 WITH THE RANGE OF CURVE PARAMETER,U = 3.1415 TO 6.2832 RADIANS
$
DEQATN  4       Z1(U) = 0.5
```

```
$
GMCURV  3       MSCGRP1 0       0                                       +000023
++000023EQUATION,0.0,3.1416,1,2,4                                       +000024
GMCURV  4       MSCGRP1 0       0                                       +000027
++000027EQUATION,3.1415,6.2832,1,2,4                                    +000028
$
$ DEFINE THE BOTTOM INNER CIRCLE OF RADIUS = 4.0 AT Z=0.0 USING DEQATN 5,6,
$ AND 7. THE BOTTOM INNER CIRCLE IS DEFINED BY TWO GMCURV ENTRIES:
$ GMCURV 5 WITH THE RANGE OF CURVE PARAMETER,U = 0.0    TO 3.1416 RADIANS
$ GMCURV 6 WITH THE RANGE OF CURVE PARAMETER U = 3.1415 TO 6.2832 RADIANS
$
DEQATN  5       X(U) = 4.0*COS(U)
DEQATN  6       Y(U) = 4.0*SIN(U)
DEQATN  7       Z(U) = 0.0
$
GMCURV  5       MSCGRP1 0       0                                       +000029
++000029EQUATION,0.0,3.1416,5,6,7                                       +000030
GMCURV  6       MSCGRP1 0       0                                       +000033
++000033EQUATION,3.1415,6.2832,5,6,7                                    +000034
$
$ DEFINE THE TOP INNER CIRCLE USING THE SAME EQUATION, DEQATN 5 AND DEQATN 6,
$ BUT WITH Z=0.5 (DEQATN 8). THIS TOP INNER CIRCLE IS DEFINED BY TWO GMCURV
$ ENTRIES:
$ GMCURV 7 WITH THE RANGE OF CURVE PARAMETER,U = 0.0    TO 3.1416 RADIANS
$ GMCURV 8 WITH THE RANGE OF CURVE PARAMETER U = 3.1415 TO 6.2832 RADIANS
$
DEQATN  8       Z1(U) = 0.5
$
GMCURV  7       MSCGRP1 0       0                                       +000031
++000031EQUATION,0.0,3.1416,5,6,8                                       +000032
GMCURV  8       MSCGRP1 0       0                                       +000035
++000035EQUATION,3.1415,6.2832,5,6,8                                    +000036
$
$ DEFINE FEEDGES
$ FEEDGE BY DEFAULT IS A STRAIGHT LINE.
$ GMCURVS ARE REQUIRED FOR FEEDGES THAT DEFINE CURVED EDGES
$
$ FEEDGES 1, 2, 3 AND 4 MAKE GMCURV 1
$ FEEDGES 5, 6, 7 AND 8 MAKE GMCURV 2
$ FEEDGES 9, 10, 11 AND 12 MAKE GMCURV 3
$ FEEDGES 13, 14, 15 AND 16 MAKE GMCURV 4
$ FEEDGES 17, 18, 19 AND 20 MAKE GMCURV 5
$ FEEDGES 21, 22, 23 AND 24 MAKE GMCURV 6
$ FEEDGES 25, 26, 27 AND 28 MAKE GMCURV 7
$ FEEDGES 29, 30, 31 AND 32 MAKE GMCURV 8
$
$       ID      GRID 1  GRID 2  CID     GEOMIN ID
FEEDGE  1       2       3       0       GMCURV  1
FEEDGE  2       3       4       0       GMCURV  1
FEEDGE  3       4       5       0       GMCURV  1
FEEDGE  4       5       6       0       GMCURV  1
FEEDGE  5       6       7       0       GMCURV  2
FEEDGE  6       7       8       0       GMCURV  2
FEEDGE  7       8       9       0       GMCURV  2
FEEDGE  8       9       2       0       GMCURV  2
FEEDGE  9       11      12      0       GMCURV  3
FEEDGE  10      12      13      0       GMCURV  3
```

```
FEEDGE  11      13      14      0       GMCURV  3
FEEDGE  12      14      15      0       GMCURV  3
FEEDGE  13      15      16      0       GMCURV  4
FEEDGE  14      16      17      0       GMCURV  4
FEEDGE  15      17      18      0       GMCURV  4
FEEDGE  16      18      11      0       GMCURV  4
FEEDGE  17      21      22      0       GMCURV  5
FEEDGE  18      22      23      0       GMCURV  5
FEEDGE  19      23      24      0       GMCURV  5
FEEDGE  20      24      25      0       GMCURV  5
FEEDGE  21      25      26      0       GMCURV  6
FEEDGE  22      26      27      0       GMCURV  6
FEEDGE  23      27      28      0       GMCURV  6
FEEDGE  24      28      21      0       GMCURV  6
FEEDGE  25      29      30      0       GMCURV  7
FEEDGE  26      30      31      0       GMCURV  7
FEEDGE  27      31      32      0       GMCURV  7
FEEDGE  28      32      33      0       GMCURV  7
FEEDGE  29      33      34      0       GMCURV  8
FEEDGE  30      34      35      0       GMCURV  8
FEEDGE  31      35      36      0       GMCURV  8
FEEDGE  32      36      29      0       GMCURV  8
$
$ DEFINE FEFACES FOR BOUNDARY CONDITION
$
$       ID      GRID 1  GRID 2  GRID 3  GRID 4  CID
FEFACE   1      2       3       12      11      0
FEFACE  .2      3       4       13      12      0
FEFACE   3      4       5       14      13      0
FEFACE   4      5       6       15      14      0
FEFACE   5      6       7       16      15      0
FEFACE   6      7       8       17      16      0
FEFACE   7      8       9       18      17      0
FEFACE   8      9       2       11      18      0
$
$ DEFINE CONSTRAINTS IN THE Z-DIRECTION
$
$ SID    = SPC ID THAT IS DEFINED IN CASE CONTROL
$ C      = CONSTRAINT COMPONENT NUMBER IN GLOBAL COORDINATE SYSTEM.
$          IT HAS TO BE A SINGLE INTEGER (1 OR 2 OR 3).
$ ENTITY = ENTITY AT WHICH THE CONSTRAINT IS APPLIED
$ ID     = ID OF THE ENTITY SELECTED ABOVE
$
$       SID     C       ENTITY  ID
GMSPC   1       3       FEFACE   1
GMSPC   1       3       FEFACE   2
GMSPC   1       3       FEFACE   3
GMSPC   1       3       FEFACE   4
GMSPC   1       3       FEFACE   5
GMSPC   1       3       FEFACE   6
GMSPC   1       3       FEFACE   7
GMSPC   1       3       FEFACE   8
$
$ DEFINE CONSTRAINTS IN THE Y-DIRECTION
$
GMSPC   1       2       GRID    1
$
```

```
$ DEFINE CONSTRAINTS IN THE X-DIRECTION AND Z-DIRECTION
$ FOR BOTH GRID 4 AND GRID 8. MULTIPLE GMSPC ENTRIES ARE
$ REQUIRED FOR CONSTRAINING MULTIPLE COMPONENTS.
$
GMSPC   1       1       GRID    4
GMSPC   1       3       GRID    4
GMSPC   1       1       GRID    8
GMSPC   1       3       GRID    8
$
$ DEFINE CONSTRAINTS FOR ROTATION FOR ALL GRIDS
$
GRDSET          0                                       0       456
$
$ DEFINE GRIDS
$
GRID    1               0.0     0.0     0.0
GRID    2       1       5.0     0.0     0.0
GRID    3       1       5.0     45.0    0.0
GRID    4       1       5.0     90.0    0.0
GRID    5       1       5.0     135.0   0.0
GRID    6       1       5.0     180.0   0.0
GRID    7       1       5.0     225.0   0.0
GRID    8       1       5.0     270.0   0.0
GRID    9       1       5.0     315.0   0.0
GRID    10              0.0     0.0     0.5
GRID    11      1       5.0     0.0     0.5
GRID    12      1       5.0     45.0    0.5
GRID    13      1       5.0     90.0    0.5
GRID    14      1       5.0     135.0   0.5
GRID    15      1       5.0     180.0   0.5
GRID    16      1       5.0     225.0   0.5
GRID    17      1       5.0     270.0   0.5
GRID    18      1       5.0     315.0   0.5
GRID    21      1       4.0     0.0     0.0
GRID    22      1       4.0     45.0    0.0
GRID    23      1       4.0     90.0    0.0
GRID    24      1       4.0     135.0   0.0
GRID    25      1       4.0     180.0   0.0
GRID    26      1       4.0     225.0   0.0
GRID    27      1       4.0     270.0   0.0
GRID    28      1       4.0     315.0   0.0
GRID    29      1       4.0     0.0     0.5
GRID    30      1       4.0     45.0    0.5
GRID    31      1       4.0     90.0    0.5
GRID    32      1       4.0     135.0   0.5
GRID    33      1       4.0     180.0   0.5
GRID    34      1       4.0     225.0   0.5
GRID    35      1       4.0     270.0   0.5
GRID    36      1       4.0     315.0   0.5
$
$ DEFINE THE MATERIAL PROPERTIES (ALUMINUM)
$
MAT1    1       1.+7            0.3     2.538-4
PSOLID  1       1                       GRID            SMECH
$
ENDDATA
```

This example illustrates a normal modes analysis, SOL 103, using the Lanczos method (EIGRL entry). The output requests are eigenvectors for all modes (DISPLACEMENT = ALL). Adaptivity analysis is required for all the elements. For this purpose, both the ADAPT and PVAL entries are used in conjunction with the ADAPT command in Case Control Section.

Results

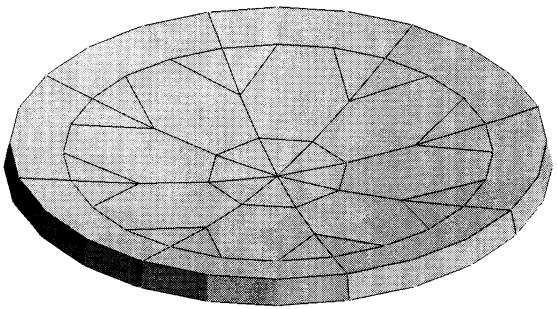
```
1    CIRCULAR PLATE                                                    DECEMBER  13, 1993  MSC/NASTRAN  12/10/93   PAGE   929
      EDGE SIMPLY SUPPORTED
0
0
                          E I G E N V A L U E   A N A L Y S I S   S U M M A R Y   (LANCZOS ITERATION)
                                    BLOCK SIZE USED .....................    7
                                    NUMBER OF DECOMPOSITIONS .............    2
                                    NUMBER OF ROOTS FOUND ................   10
                                    NUMBER OF SOLVES REQUIRED ............   10
                                    TERMINATION MESSAGE :  REQUIRED NUMBER OF EIGENVALUES FOUND.
1    CIRCULAR PLATE                                                    DECEMBER  13, 1993  MSC/NASTRAN  12/10/93   PAGE   930
      EDGE SIMPLY SUPPORTED
0
                                                R E A L   E I G E N V A L U E S
    MODE    EXTRACTION      EIGENVALUE            RADIANS             CYCLES            GENERALIZED         GENERALIZED
     NO.       ORDER                                                                      MASS                STIFFNESS
        1         1       3.465334E+07        5.886709E+03        9.368988E+02        1.000000E+00        3.465334E+07
        2         2       1.182291E+08        1.087332E+04        1.730543E+03        1.000000E+00        1.182291E+08
        3         3       1.483814E+08        1.218119E+04        1.938696E+03        1.000000E+00        1.483814E+08
        4         4       2.502755E+08        1.582010E+04        2.517847E+03        1.000000E+00        2.502755E+08
        5         5       2.654028E+08        1.629119E+04        2.592823E+03        1.000000E+00        2.654028E+08
        6         6       3.027782E+08        1.740052E+04        2.769379E+03        1.000000E+00        3.027782E+08
        7         7       8.642510E+08        2.939815E+04        4.678860E+03        1.000000E+00        8.642510E+08
        8         8       8.645740E+08        2.940364E+04        4.679734E+03        1.000000E+00        8.645740E+08
        9         9       1.166431E+09        3.415306E+04        5.435628E+03        1.000000E+00        1.166431E+09
       10        10       2.009605E+09        4.482862E+04        7.134696E+03        1.000000E+00        2.009605E+09

1    CIRCULAR PLATE                                                    DECEMBER  13, 1993  MSC/NASTRAN  12/10/93   PAGE   931
      EDGE SIMPLY SUPPORTED
0
 *** USER INFORMATION MESSAGE 6502 (ADAPT)
     THE CURRENT SUPERELEMENT REQUIRES NO FURTHUR P-ADAPTIVE ANALYSIS BECAUSE ALL P-ELEMENTS REACHED MAXIMUM P-VALUES.
0*** USER WARNING MESSAGE 4594. THE POSTCDB CONTAINS NO LEGITIMATE SURFACE OR VOLUME DEFINITIONS.

1    CIRCULAR PLATE                                                    DECEMBER  13, 1993  MSC/NASTRAN  12/10/93   PAGE   932
      EDGE SIMPLY SUPPORTED                                          ADAPTIVITY INDEX=     5
0
      EIGENVALUE =  3.465334E+07
          CYCLES =  9.368987E+02        R E A L   E I G E N V E C T O R   N O .         1

      POINT ID.   TYPE          T1              T2              T3              R1              R2              R3
             1      G      6.661354E-06    .0             -1.868621E+01    .0              .0              .0
             2      G      1.333745E+00   -2.678128E-05    .0              .0              .0              .0
             3      G      9.431428E-01    9.430584E-01    .0              .0              .0              .0
             4      G       .0             1.333743E+00    .0              .0              .0              .0
             5      G     -9.430543E-01    9.431386E-01    .0              .0              .0              .0
             6      G     -1.333741E+00    2.663938E-05    .0              .0              .0              .0
             7      G     -9.431665E-01   -9.430221E-01    .0              .0              .0              .0
             8      G       .0            -1.333743E+00    .0              .0              .0              .0
             9      G      9.430305E-01   -9.431752E-01    .0              .0              .0              .0
            10      G      6.681371E-06   -6.129073E-08   -1.868621E+01    .0              .0              .0
            11      G     -1.333732E+00   -3.440074E-04    .0              .0              .0              .0
            12      G     -9.428723E-01   -9.433200E-01    .0              .0              .0              .0
            13      G      3.471338E-04   -1.333743E+00    .0              .0              .0              .0
            14      G      9.433249E-01   -9.428771E-01    .0              .0              .0              .0
            15      G      1.333754E+00    3.440111E-04    .0              .0              .0              .0
            16      G      9.429217E-01    9.432840E-01    .0              .0              .0              .0
            17      G     -3.258019E-04    1.333743E+00    .0              .0              .0              .0
            18      G     -9.432753E-01    9.429132E-01    .0              .0              .0              .0
            21      G      1.343029E+00   -8.329611E-05   -5.443803E+00    .0              .0              .0
            22      G      9.497321E-01    9.495923E-01   -5.443810E+00    .0              .0              .0
            23      G      8.442988E-05    1.343021E+00   -5.443810E+00    .0              .0              .0
            24      G     -9.495862E-01    9.497259E-01   -5.443810E+00    .0              .0              .0
            25      G     -1.343013E+00    8.315749E-05   -5.443817E+00    .0              .0              .0
            26      G     -9.497337E-01   -9.495776E-01   -5.443811E+00    .0              .0              .0
            27      G     -7.570248E-05   -1.343021E+00   -5.443810E+00    .0              .0              .0
            28      G      9.495844E-01   -9.497408E-01   -5.443809E+00    .0              .0              .0
            29      G     -1.343016E+00   -2.135180E-04   -5.443803E+00    .0              .0              .0
            30      G     -9.495108E-01   -9.498014E-01   -5.443810E+00    .0              .0              .0
            31      G      2.196022E-04   -1.343021E+00   -5.443810E+00    .0              .0              .0
            32      G      9.498075E-01   -9.495170E-01   -5.443810E+00    .0              .0              .0
            33      G      1.343026E+00    2.135190E-04   -5.443817E+00    .0              .0              .0
            34      G      9.495386E-01    9.497868E-01   -5.443811E+00    .0              .0              .0
            35      G     -2.027499E-04    1.343021E+00   -5.443810E+00    .0              .0              .0
            36      G     -9.497795E-01    9.495314E-01   -5.443809E+00    .0              .0              .0
```

Figure 6-14 Abridged Output from the Circular Solid Model

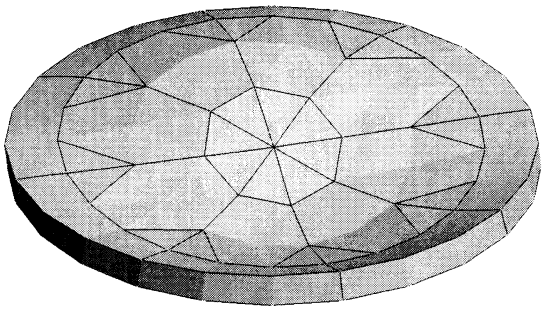
201001001	G	1.333745E+00	-2.678128E-05	.0	.0	.0	.0
201001002	G	1.288204E+00	3.459260E-01	.0	.0	.0	.0
201001003	G	1.155554E+00	6.657130E-01	.0	.0	.0	.0
201001004	G	9.431428E-01	9.430584E-01	.0	.0	.0	.0
201001005	G	1.354012E+00	-1.118739E-07	-1.801449E+00	.0	.0	.0
201001006	G	1.307760E+00	3.511871E-01	-1.801487E+00	.0	.0	.0
201001007	G	1.173140E+00	6.758680E-01	-1.801314E+00	.0	.0	.0
201001008	G	9.574580E-01	9.574031E-01	-1.801455E+00	.0	.0	.0
201001009	G	1.357469E+00	-5.671428E-05	-3.626504E+00	.0	.0	.0
201001010	G	1.311071E+00	3.521154E-01	-3.626570E+00	.0	.0	.0
201001011	G	1.176151E+00	6.776471E-01	-3.626329E+00	.0	.0	.0
201001012	G	9.599283E-01	9.598196E-01	-3.626513E+00	.0	.0	.0
201001013	G	1.343029E+00	-8.329611E-05	-5.443803E+00	.0	.0	.0
201001014	G	1.297096E+00	3.483891E-01	-5.443859E+00	.0	.0	.0
201001015	G	1.163644E+00	6.704805E-01	-5.443617E+00	.0	.0	.0
201001016	G	9.497321E-01	9.495923E-01	-5.443810E+00	.0	.0	.0

Figure 6-15 Abridged Output from the Circular Solid Model (continued)

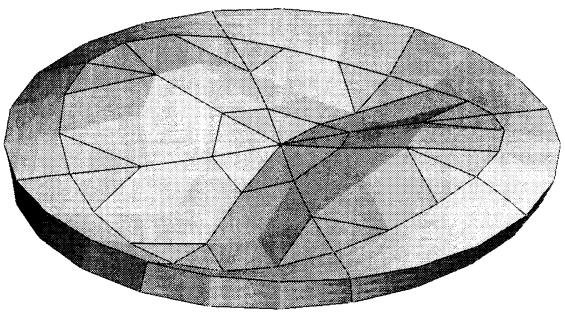
The mode shapes for modes 1, 5, 8, and 9 are shown in Figure 6-16. Mode 1 is the fundamental mode, mode 5 is the one nodal diameter mode, mode 8 is the two nodal diameters mode, and mode 9 is one nodal circle mode.



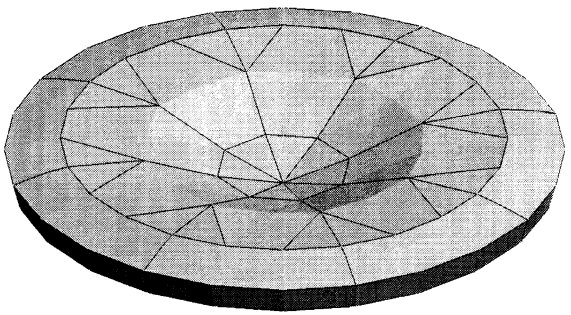
Mode 1: Fundamental Mode



Mode 5: One Nodal Diameter Mode



Mode 8: Two Nodal Diameter Mode



Mode 9: One Nodal Circle Mode

Figure 6-16 The Mode Shapes for Modes 1, 5, 8, and 9

Comparison of the natural frequencies obtained from the adaptivity analysis for the solid circular model to the theoretical natural frequencies for the circular plate is shown in Table 6-6. The discrepancies between the two values decrease as the p-order increases.

Table 6-6 Comparison of the Natural Frequencies

Mode	Theoretical (Hertz)	Numerical (Hertz)
Fundamental	954.1	936.9
One Nodal Diameter	2657.7	2592.8
Two Nodal Diameters	4913.9	4679.7
One Nodal Circle	5697.8	5435.6

Further studies were performed in which the p-order was kept constant and the thickness of the solid model was varied. Results showed that as the thickness of the circular solid model decreases, the solid model behaves like a plate model, which is to be expected.

Reference:

1. Roark, R. J. and Young, W. C. *Formulas for Stress and Strain*. 5th edition. McGraw-Hill, Inc., 1975, p. 578.

Clamped Rhombic Plate

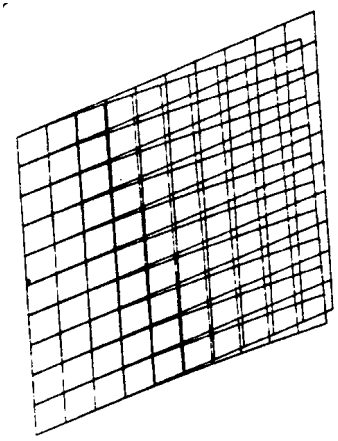
The first eigenfrequency of a rhombic plate is calculated using SOL 103. Figure 6-17 shows a plate clamped at the left end and the first eigenmode. Compare the convergence behavior of three different models:

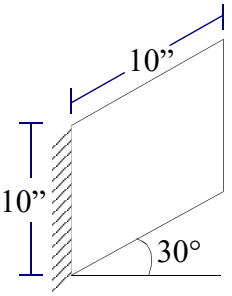
- One `CHEXA` p-element with p-orders varying from p = 1 to p = 8
- One CQUAD p-element with p-orders varying from p = 1 to p = 8
- Multiple CQUAD h-elements with varying meshes (1×1, 2×2, 4×4, 8×8, 16×16, 32×32)

The convergence behavior of the three models is shown in Figure 6-18. The first eigenfrequency is normalized with 2.932 rad/sec, which is the value from a series solution using Kirchhoff plate theory. All three models exhibit excellent convergence behavior.

The solid p-element and the shell p-element models converge when the p-level is increased. The `CQUAD4` h-element model converges when the mesh is refined. If the error rate in the first eigenfrequency must be under 0.2%, then either: (a) a single shell or solid p-element with p = 8, or (b) 16×16 shell h-elements should be used. The shell p-element needs the smallest number of degrees of freedom to achieve the required accuracy. For a comparison of the degrees of freedom, see Table 6-7.

First Eigenmode





$t = 0.5''$
$E = 10.6 \times 10^6$ psi
$\nu = 0.3$

Figure 6-17 Rhombic Plate

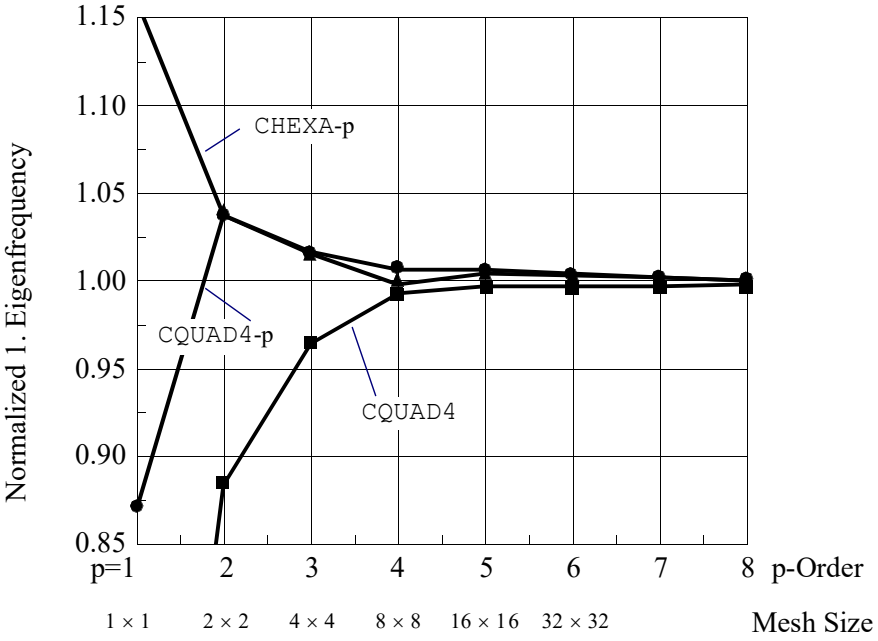


Figure 6-18 Normalized First Eigenfrequency for Different p-Levels and Mesh Sizes

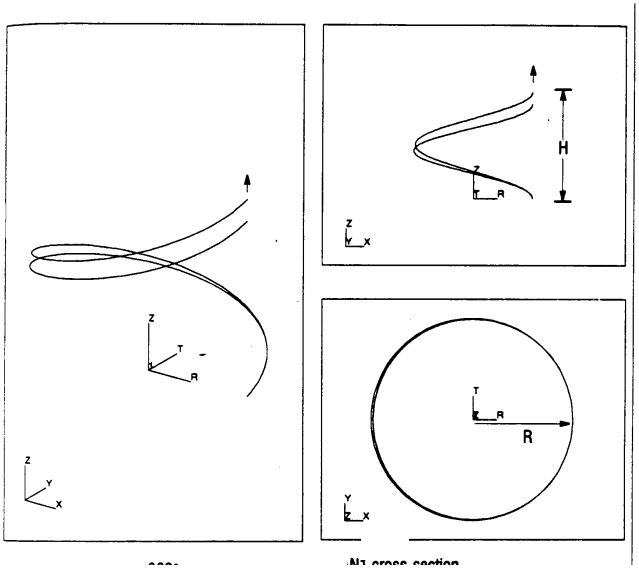
Table 6-7 Comparison of the Number of Degrees-of-Freedom for 0.2% Accuracy in the First Eigenfrequency

Element	Mesh Size and p-Level	Normalized Eigenfrequency	Degrees of Freedom
CQUAD4	16 × 16	0.9977	1445
CQUAD4-p	p = 8	1.0012	264
CHEXA-p	p = 8	1.0017	510

Helix Modeled with CBEAM p-Elements

A 360° helix is modeled with nonadaptive beam h-elements and with adaptive beam p-elements. The helix is clamped at the bottom and free at the top, with a single force applied at the top. The undeformed and deformed helix is shown in Figure 6-19. The normalized deflection at the tip is shown in Figure 6-20 for different mesh sizes.

Both beam h- and p-elements converge well. If the error in the deflection must be under 0.2%, then 8 beam p-elements with p = 3 (45° angle for each element) or 64 beam h-elements should be used. A comparison of the degrees of freedom (see Table 6-8) shows that the beam p-elements require a smaller number of degrees of freedom than the number required by the beam h-elements in order to achieve the required accuracy.



θ = 360° CHAN1 cross-section
H = 9. height E = 1.5e+7
R = 5. radius ν = 0.33

Figure 6-19 Helix

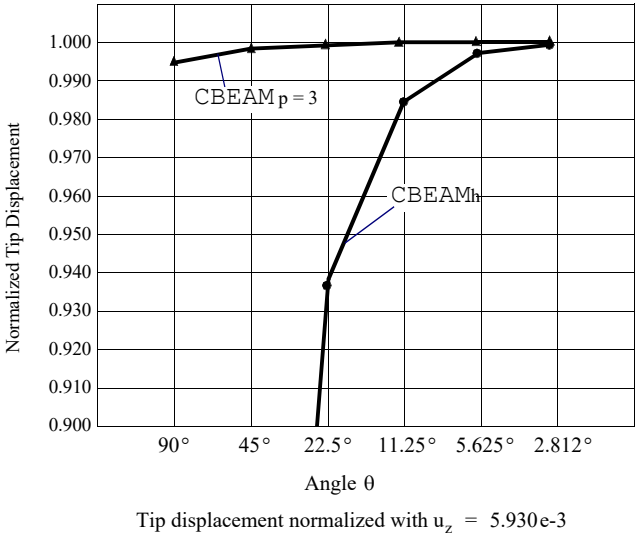


Figure 6-20 Normalized Tip Displacement for Different Mesh Sizes

Table 6-8 Comparison of the Number of Degrees of Freedom for 0.2% Accuracy in the Tip Displacement

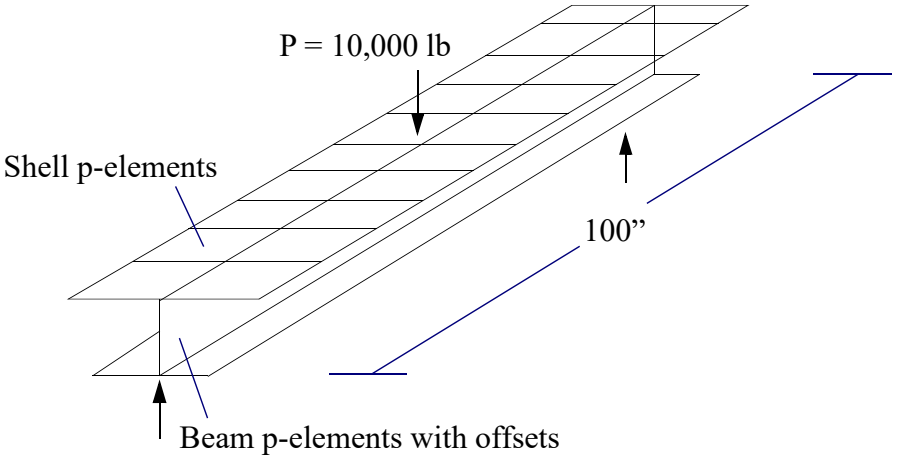
Element and Mesh	Normalized Tip Displacement	Degrees of Freedom
CBEAM h, $\theta = 5.6°$; 64 Elements	0.9970	390
CBEAM p, $\theta = 45°$; p = 3, 8 Elements	0.9978	150

Plate with Stiffener

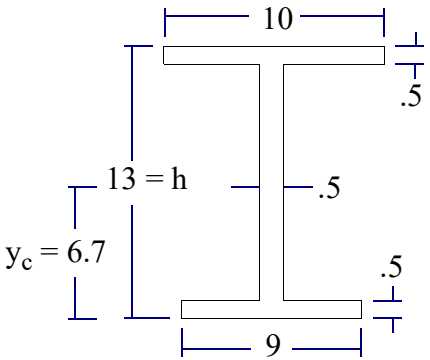
In stiffened structures, an efficient practice is to model the stiffeners using beam elements with offsets. Typically, nonadaptive h-elements are used. The example below demonstrates the benefits of p-elements for this class of problems.

A plate with stiffener is loaded in the middle and simply supported at the ends (see Figure 6-21). The plate is modeled with 10×2 shell elements and the stiffener is modeled with 10 beam elements with offsets. Compare the stress results of two models, an h-element model and a p-element model, with a uniform p-order of p = 2 for shells and beams. The solution is compared with the solution derived from beam theory.

The nonadaptive h-elements produce very inaccurate stresses with errors around 30%, as shown in Figure 6-22. The adaptive p-elements with a uniform p-order of p = 2 yield very accurate results — the error rate is only around 4%. In the h-element model, the normal forces in the shell and beam elements are constant within each element, which causes the high stress error. In the p-element model, the normal forces in each element are linear (p = 2), which improves the stress distribution in the cross-section of the beam.



Cross Section



$$I_z = 442.662$$

$$M_{\text{at center}} = \frac{P}{2} \cdot \frac{l}{2} = 250{,}000 \text{ lb-in}$$

$$\sigma_{\text{top}} = -\frac{M(h - y_c)}{I_z} = -3560 \text{ psi}$$

$$\sigma_{\text{bottom}} = \frac{M(y_c)}{I_z} = 3786 \text{ psi}$$

Figure 6-21 Plate with Stiffener

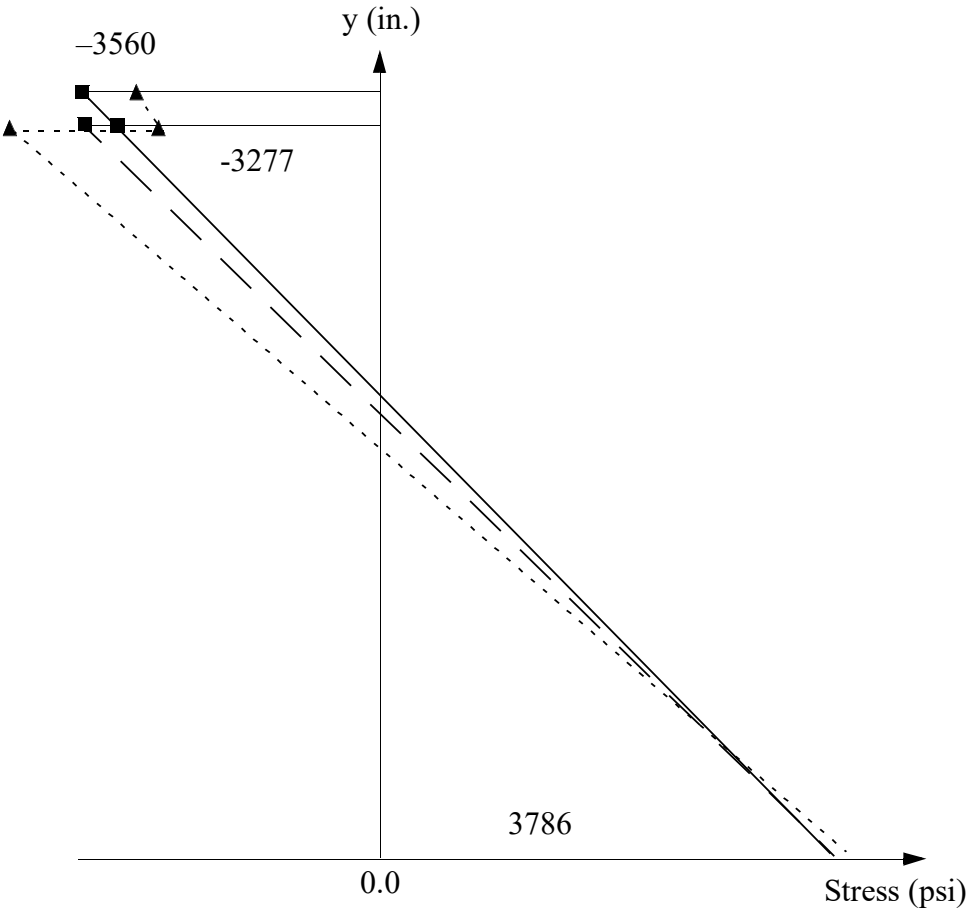


Table 1:

Stress [psi]	y [in]
3786.	0.
–3277.	12.5
–3560.	13.0

beam theory

Table 2:

Stress [psi]	y [in]
3978.	0.
–4311.	12.5
–2804.	12.5
–3082.	13.0

h-elements

Table 3:

Stress [psi]	y [in]
3834.	0.
–3545.	12.5
–3250.	12.5
–3590.	13.0

p=elements
p = 2

Figure 6-22 Stress Distribution in the Middle of the Beam

Solid p-Elements in Linear Dynamic Solution Sequences

Solid p-elements are available in the dynamic solution sequences SOL 107, to SOL 112. The p-elements do not adapt; the analysis is done for a fixed p-level. Displacements, velocities, accelerations, stresses and strains are output in SORT1 or SORT2 format. The complex output quantities are in rectangular format (real/imaginary) or in polar format (magnitude/phase).

User Interface in MSC Nastran

Output for p-elements is requested with the Case Control command DATAREC, where SORT1 or SORT2 formats can be chosen. For complex eigenvalues (SOL 107, SOL 110) and frequency response (SOL 108, SOL 111), the default is SORT1. For transient response (SOL 109, SOL 112), the default is SORT2.

The output quantities are requested with DISP, VELO, ACCE, STRESS, or STRAIN in the Case Control, and/or in the OUTPUT Bulk Data entry. In the complex solution sequences, the formats are rectangular (REAL, IMAG) or polar (MAGNITUDE, PHASE); the rectangular format is the default. REAL, IMAG or PHASE is requested in the Case Control command or in the OUTPUT Bulk Data entry.

Current Limitations and Recommendations

The following limitations and recommendations apply to solid p-element usage in linear dynamic solution sequences:

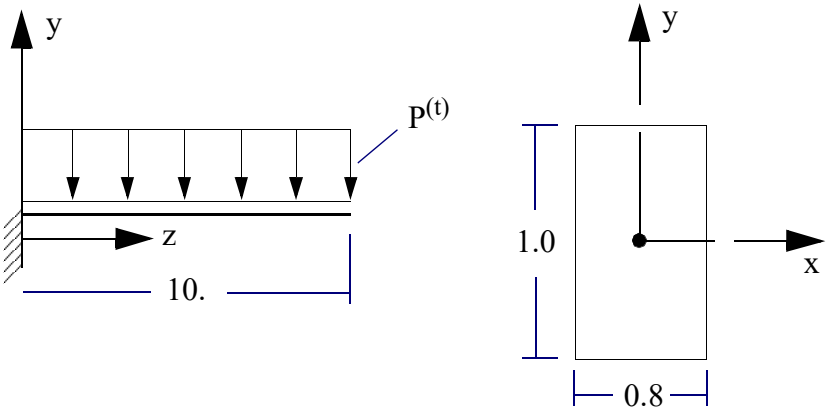
- The output request must be all in SORT1 or all in SORT2 format for a single run.
- PARAM,CURVPLOT,1 must be specified to get SORT1 output in transient response (SOL 109 and SOL 112).
- If an ADAPT entry is specified with multiple p-levels, the analysis is done with the first p-level.

Example

The transient response of a cantilever subject to a constant step load is calculated in SOL 109 (see Figure 6-23). The following is a comparison of the results between the two different models:

- HEXA8 h-elements, $4 \times 4 \times 40$ element mesh.
- HEXA p-elements, $1 \times 1 \times 4$ element mesh with p = 3.

The plot displacement versus time graph shown in Figure 6-24 illustrates that the results are in good agreement.



E = 3.e+7
ν = 0.25
PARAM,W3,9.7
PARAM,G,0.0647

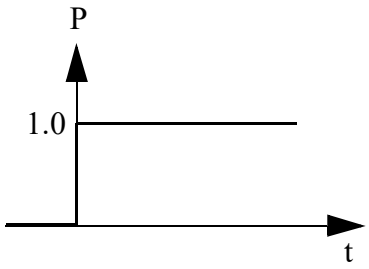


Figure 6-23 Cantilever with Step Load

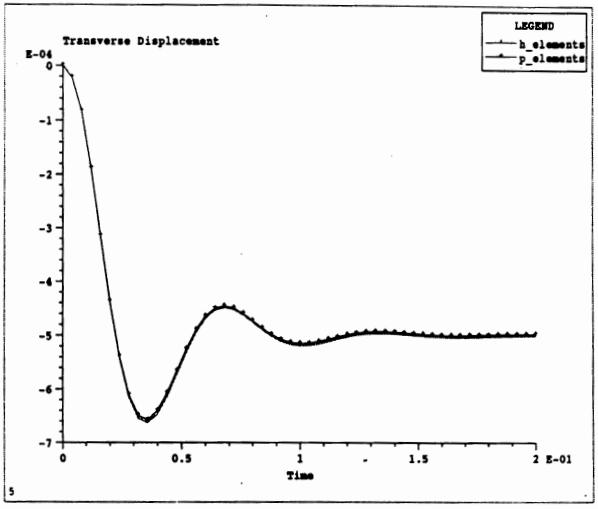


Figure 6-24 Tip Displacement of Cantilever

p-Element Thermal Analysis

MSC Nastran supports p-version elements for linear steady state thermal analysis (SOL 101). The definitions of model geometry, elements, p-values, and error estimation methods follow the general terminologies of the p-element structural analysis. A major difference exists in the specification of thermal loads and boundary conditions, which requires new or modified Bulk Data entries for thermal analysis.

The following sections provide usage guidelines and examples for the p-version thermal analysis.

Geometry

The geometry definitions are the same as those defined for the p-element structural analysis.

Elements

Six kinds of elements — `CHEXA`, `CPENTA`, `CTETRA`, `CQUAD4`, `CTRIA3`, and `CBEAM` — may be defined as p-elements for thermal analysis.

Materials

The standard isotropic, orthotropic, and anisotropic materials are available for p-version thermal elements. The only material property required is thermal conductivity which must be constant for linear steady state analysis.

Loads and Boundary Conditions

The following thermal loads and boundary conditions are supported in p-version thermal analysis.

- Prescribed temperatures applied on point, curve, 2-D edge, surface, or 3-D surface.
- Normal or directional heat fluxes applied on point, curve, 2-D edge, surface, 3-D surface, or solid.
- Volumetric heat generation within conduction elements.
- Free convection applied on point, curve, 2-D edge, surface, or 3-D surface.

Table 6-9 summarizes the Bulk Data entries for p-version thermal loads and boundary conditions. A full description of these entries may be found in the MSC Nastran Quick Reference Guide.

Table 6-9 Bulk Data Entries for p-Version Thermal Loads and Boundary Conditions

Load or Boundary Condition Type	Bulk Data
Prescribed Temperature	GMBC
Normal or Directional Heat Flux	GMLOAD
Volumetric Heat Generation	GMQVOL
Free Convection	GMCONV

p-Version Adaptivity

p-values and error estimation methods are the same as those for p-element structural analysis.

Data Recovery

Similar to structural output, the DATAREC and OUTRCV Case Control commands as well as the OUTPUT and OUTRCV Bulk Data entries define the output of p-version thermal analysis. Temperatures, temperature gradients, heat fluxes, polynomial values, and element errors may be printed, plotted, or punched. The TEMP output command of the OUTPUT Bulk Data entries requests for calculating temperatures while the FLUX output command of the OUTPUT Bulk Data entries requests for recovering temperature gradients and heat fluxes for conduction elements.

Thermal Stress Analysis

The interface to input temperature results as temperature loads for thermal stress analysis is described in the Case Control command TSTRU (Case) (p. 631) in the *MSC Nastran Quick Reference Guide*.

Limitations and Recommendations

- Only Solution Sequence 101 (linear steady state) performs p-element thermal analysis.
- Temperature boundary conditions can be spatially varying, but must be independent or time.
- Heat fluxes and volumetric heat generation can be spatially varying, but must be independent of temperature or time.
- The convection heat transfer coefficients and ambient temperatures of free convection boundary conditions can be spatially varying, but must be independent of temperature or time.
- It is important to select the appropriate entity type while applying loads and boundary conditions. Concentrated loads or boundary conditions will cause singularities or inaccurate results and should be avoided. For example, the entity type `FEFACE` or GMSURF should be used to prescribe temperatures on a surface, rather than the entity type `FEEDGE`, `GMCURV`E, or `GRID`.

G Local Adaptive Mesh Refinement

- Introduction 806
- Adaptive Mesh Refinement Loop 806
- Refinement by Regular Subdivision 808
- Transference of Analysis Data Between Unrefined and Refined Meshes 832

Introduction

The basics of local adaptive meshing is documented in Chapter 18: Adaptive Meshing of the *MSC Nastran Linear Static Analysis User's Guide*. A more detailed explanation of the local meshing refinements are covered in this chapter.

Adaptive Mesh Refinement Loop

During the adaptive mesh refinement process, a sequence of analysis supported over a sequence of different finite element meshes is sequentially performed within an automatic loop (Figure 7-1).

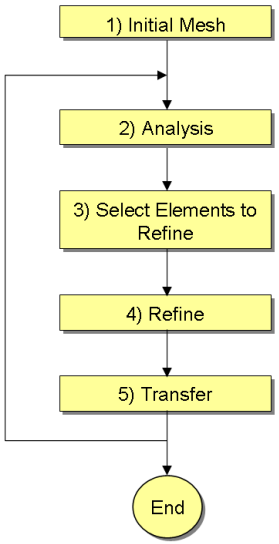


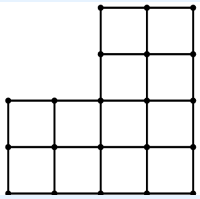
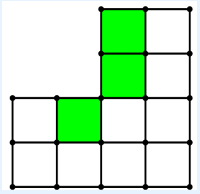
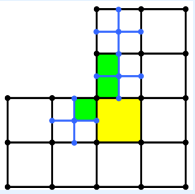
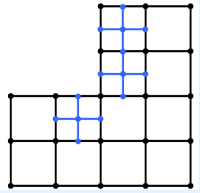
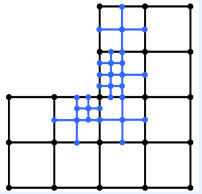
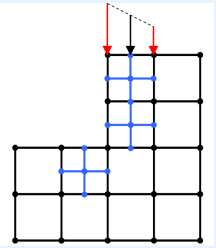
Figure 7-1 Adaptive Mesh Refinement Loop

This *adaptive mesh refinement loop* can be summarized as follows:

1. The user inputs an *initial*, preferably coarse, finite element mesh.
2. An *analysis* is run and a finite element solution (supported on the current mesh) is computed.
3. Some elements of the previous mesh are scheduled or *marked* for refinement. These elements are chosen according to a user specified *adaptivity criterion* and implicit *refinement propagation* rules.
4. The elements scheduled for refinement are *refined* and a new finite element mesh with new elements and grid points are thus created.
5. Element properties, boundary conditions, constraints and loads are *transferred* or mapped from the previous mesh to the new mesh.
6. Steps 2 to 5 are repeated until a *termination criterion* is met.

Table 7-1 schematically illustrates the first two iterations of the adaptive mesh refinement loop. Elements scheduled for refinement due to the user specified adaptivity criterion are depicted in green. Notice that there are neighboring elements that are also refined during the process (yellow elements). Implicit rules to propagate the refinement from elements meeting the user specified criterion to their neighbors are explained in Propagation of Refinement, 827.

Table 7-1 First Two Iterations of the Adaptive Mesh Refinement Loop

	First Iteration	Second Iteration
1. Initial Mesh		
2. Analysis	...	...
3. Mark for refinement		
4. Refine		
5. Transfer		...

Refinement by Regular Subdivision

Mesh refinement in MSC Nastran is accomplished by the so-called *regular or isotropic subdivision* of a subset of parent elements into offspring or children sub-elements. Figure 7-2 illustrates the subdivision rules for individual elements of the different types:

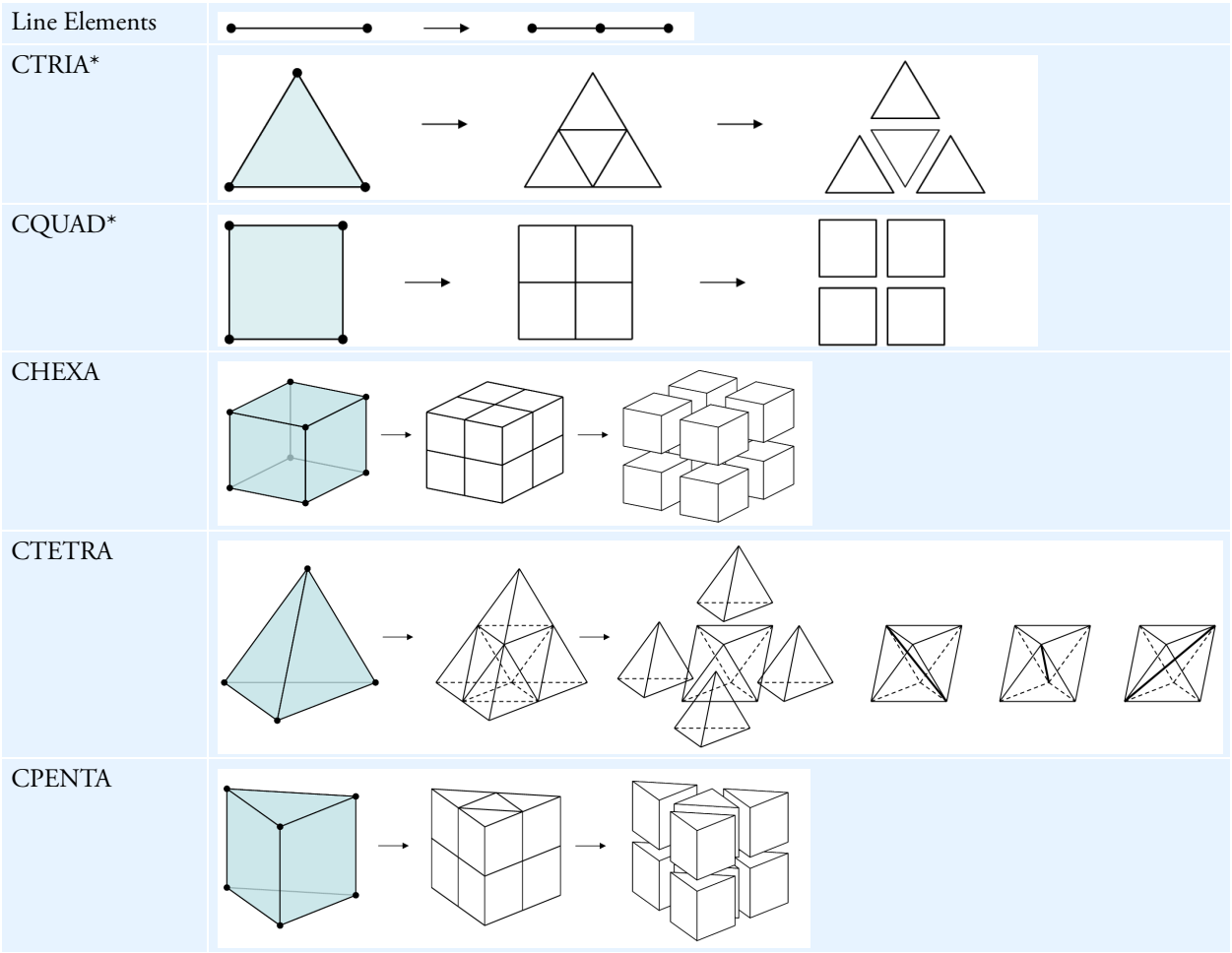


Figure 7-2 Regular (isotropic) subdivision rules for individual elements of different types

These subdivision rules are called *regular* or *isotropic* because all edges in the boundary of a refined element are subdivided into the same number of segments (two) as opposed to, for example, the subdivision of a quadrilateral or a triangular element by bisection (Figure 7-3).

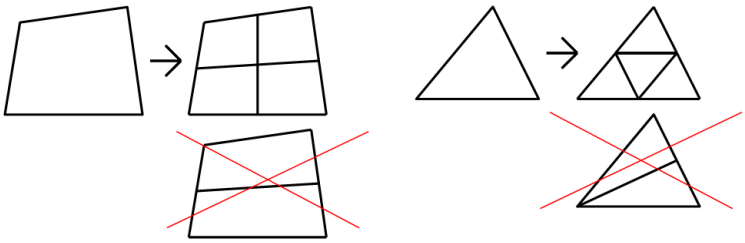


Figure 7-3 Isotropic vs. Anisotropic Subdivision of a Quadrilateral and a Triangular Element

For a tetrahedron, there are three possible regular subdivision schemes into eight children tetrahedra. These three schemes are obtained as follows: first, each edge is bisected. This defines four corner tetrahedra and an internal octahedron. Then, the latter might be subdivided into four additional tetrahedra in three different ways, according to each of its three diagonals.

In MSC Nastran, the diagonal selected to subdivide the internal octahedron is the one connecting node 7 (mid-node of the edge 1-3) to node 9 (mid-node of edge 2-4) as depicted in Figure 7-4. Furthermore, corner nodes of the children tetrahedra are numbered according to the special rule illustrated in Figure 7-4. This special labeling convention of nodes along with the selection of the diagonal connecting nodes 7 to 9 to subdivide the inner octahedron ensures minimization of element distortion with successive refinements.

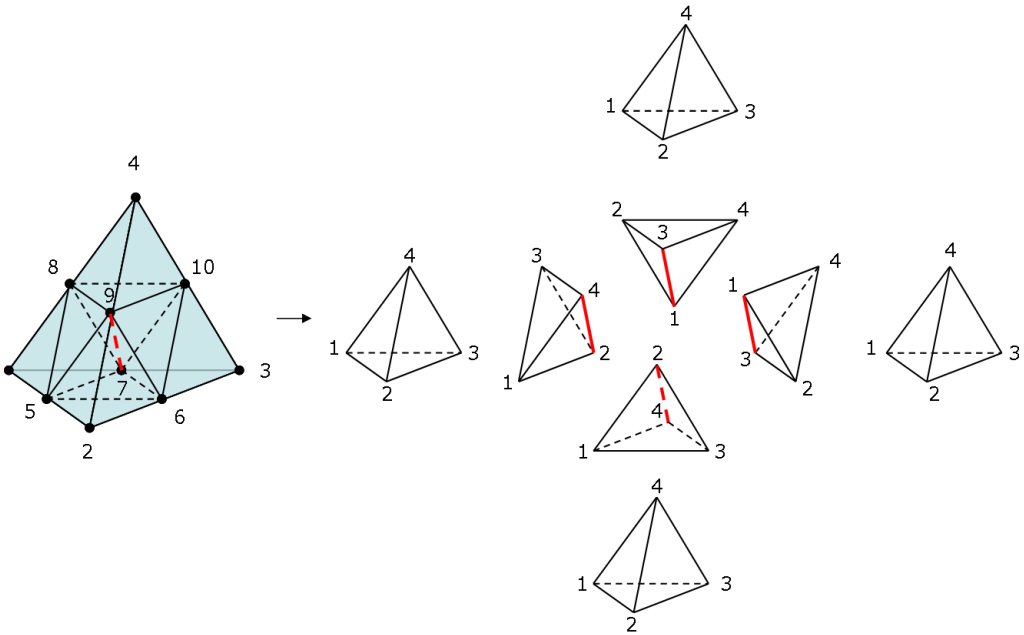


Figure 7-4 Labeling Convention for Corner Nodes of Children Tetrahedra

During subdivision, the following properties are preserved:

1. Preservation of element type: Quadrilateral elements are subdivided into quadrilateral elements, triangular elements into triangular elements, etc., as opposed to, for example, subdivision of a quadrilateral into triangles or subdivision of a triangle into quadrilaterals (Figure 7-5).

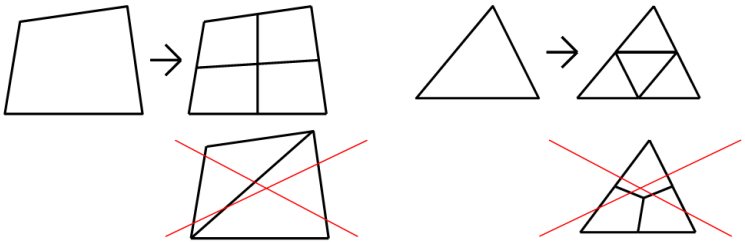


Figure 7-5 Preservation of Element Type during Subdivision

2. Preservation of element orientation (Figure 7-6): The outlining nodes of children elements will be listed in clockwise (respectively counterclockwise) order when the father element have been defined in clockwise (respectively counterclockwise) order.

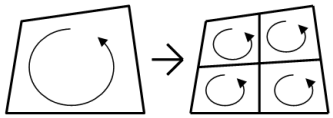


Figure 7-6 Preservation of Orientation during Subdivision

Similarly, linear (3-noded) triangular elements (CTRIA3, CTRIAR) will be subdivided into four linear (3-noded) triangular elements whereas quadratic (6-noded) triangular elements will be subdivided into four quadratic triangular elements (Figure 7-7). The linear case requires the creation of three new grid points whereas the quadratic case involves the creation of nine new grid points, two on each edge and one on each internal edge.

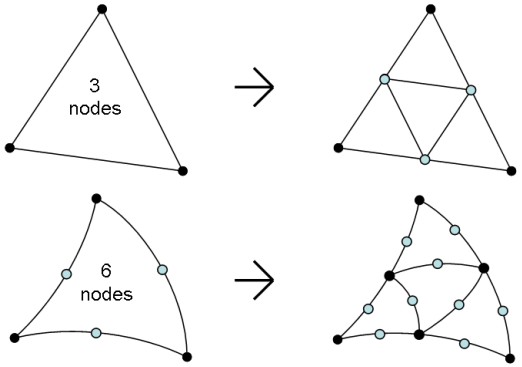


Figure 7-7 Preservation of Interpolation Order in Triangular Elements during Subdivision

The same rule applies to 3-D elements (CTETRA, CPENTA, CTRIA); that is, a linear tetrahedron (4-noded), pentahedron (6-noded) or hexahedron (8-noded) will be respectively subdivided into eight linear tetrahedra, pentahedra or hexahedra requiring respectively the creation of 6, 12, and 19 new grid points (Figure 7-8).

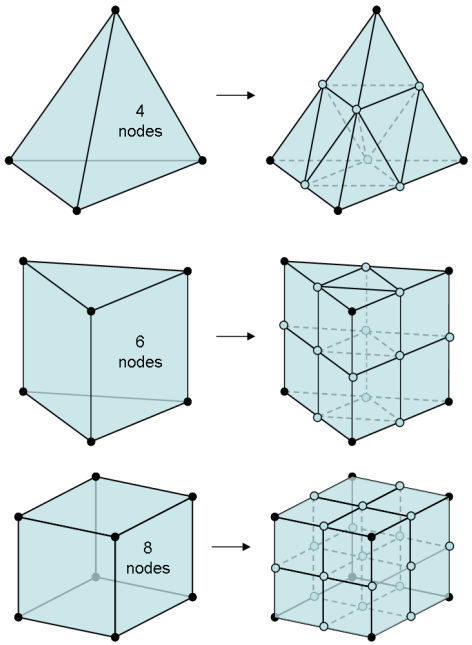


Figure 7-8 Preservation of Interpolation Order in 3-D Linear Elements during Subdivision

Location of New Grid Points

In linear elements, new *edge nodes* are placed at the mid-side of the (straight) edge. Similarly, new *face nodes* of linear quadrilateral surface elements (CQUAD4, CQUADR) or quadrilateral faces of linear pentahedral and hexahedral elements (6-noded CPENTA or 8-noded CHEXA) or new *internal nodes* of linear tetrahedral or hexahedral elements (4-noded CTETRA or 8-noded CHEXA) are placed at the *baricenter* of the surface element, face or 3-D element, that is, at the position obtained by averaging the position of the corner nodes (Figure 7-9).

Mid-edge nodes are placed at the mid-side of the straight edge, mid-face nodes at the baricenter of the face, and internal nodes at the baricenter of the element.

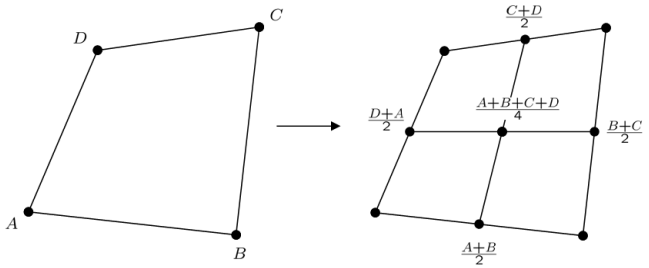


Figure 7-9 Location of New Mid-edge and Mid-face Nodes in a Linear Element

In quadratic elements new nodes are positioned by making use of the isoparametric mapping. The parametric space of the element is uniformly bisected, and mid-edge and mid-face nodes are mapped back to the physical space using the element (isoparametric) shape functions (Figure 7-10).

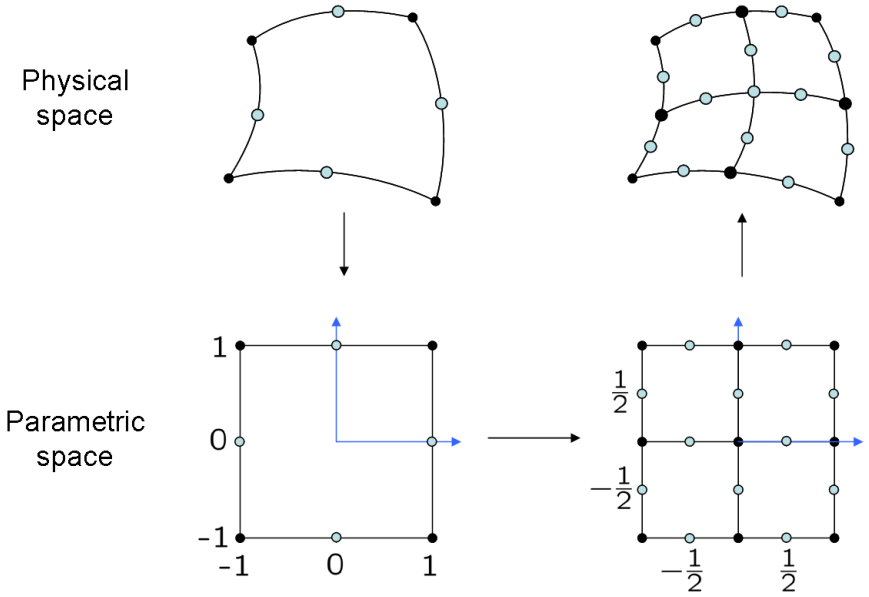


Figure 7-10 Uniform Subdivision of the Parametric Domain and Resulting Subdivision in Physical Space

No special provisions are taken during refinement of very distorted quadratic elements. In this case, the user should expect distorted children elements (Figure 7-11).

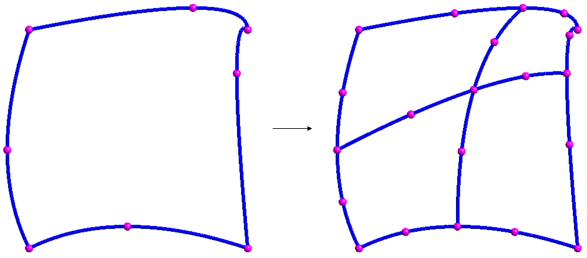


Figure 7-11 Subdivision of a Distorted Quadratic (Quadrilateral) Element

The default method of placement of mid-edge nodes on mid-side edges might render inaccurate solutions when the initial mesh provided by the user is very coarse and the boundary of the domain of analysis is therefore poorly approximated.

For example, consider the analysis of a circular planar shell subjected to compression and initially discretized with very few elements as illustrated in Figure 7-12. If mid-edge nodes are placed on the mid-side of edges, inaccurate results are obtained because the circular domain remains poorly approximated during all mesh refinement cycles. Default location for mid-edge nodes is on the mid-side of edges.

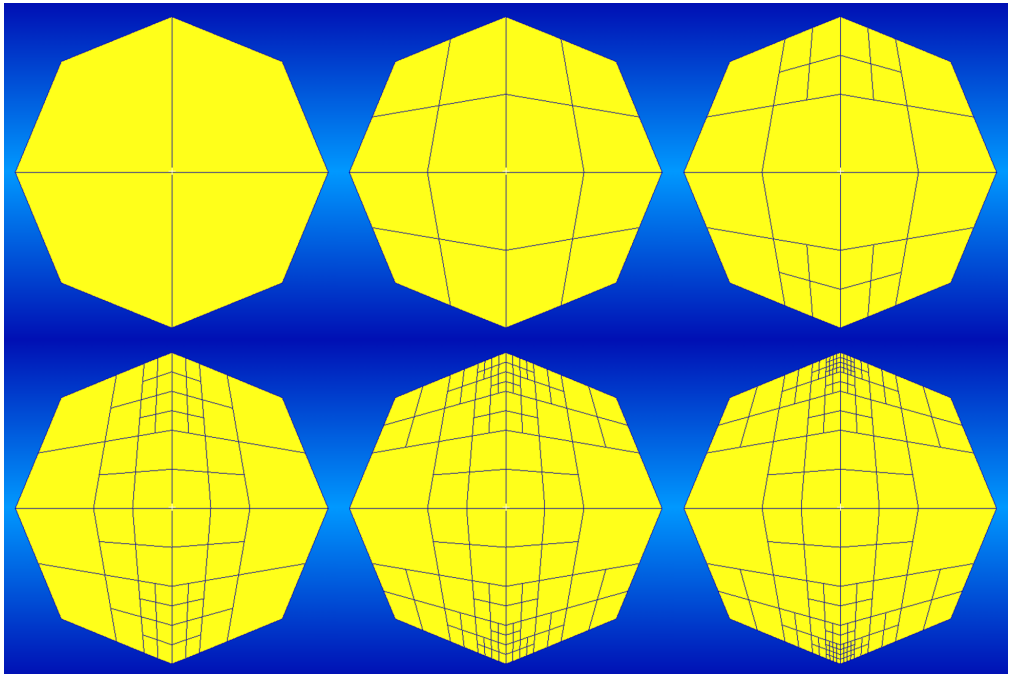


Figure 7-12 Compression of a Circular Planar Shell

To address this inaccuracy, the user can request to place mid-edge nodes on a *smooth approximation of the analysis domain boundary interpolated from the initial mesh*.

Given the initial mesh, a smooth curve is interpolated using the nodes located on the mesh boundary to approximate the analysis domain boundary. Then, mid-edge nodes are projected onto this smooth approximation. Figure 7-13 depicts this alternative for the case of the compressed circular shell example shown in Figure 7-12 with the projection of mid-edge nodes onto a smooth approximation of the geometric boundary interpolated from the initial mesh.

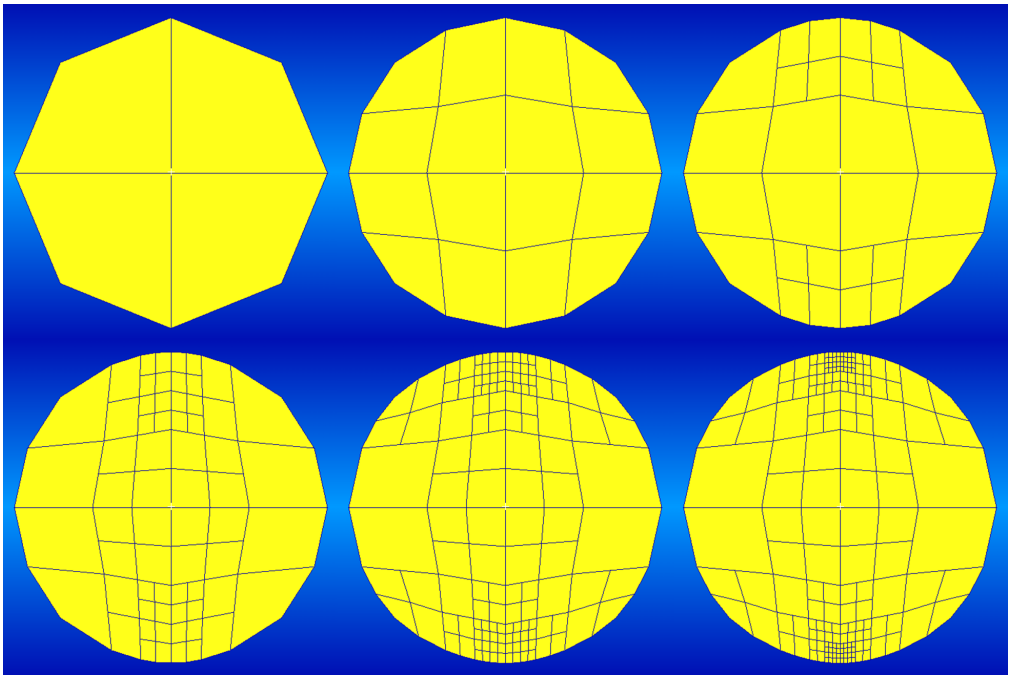


Figure 7-13 Projection of Mid-edge Nodes onto a Smooth Approximation of the Geometric Boundary Interpolated from the Initial Mesh

It's important to note that the smooth approximation of the boundary is computed using the boundary nodes of the initial mesh provided by the user and that the accuracy of this approximation is determined by the coarseness of this initial mesh.

The success and accuracy of this smooth boundary approximation depends also on appropriate detection of corners and edges. In order to identify corners and edges, the initial mesh is preprocessed using an automatic *Geometric Feature Detection Algorithm*, see Detection of Material, and Superelement Interfaces, 845.

The alternative method of projecting edge-nodes onto a smooth approximation of the mesh boundary is available only for edge nodes belonging to edge boundaries of 2-D and 3-D geometries. No repositioning for edge and face nodes *belonging to 3-D surfaces and face-boundaries of D geometries* is supported.

Hanging Nodes and Multipoint Constraints on Hanging Nodes

When an element is refined (subdivided) but its adjacent elements are not refined a non conforming mesh is generated. Nodes created on the boundary between a refined and a non refined element are referred to as hanging nodes (Figure 7-14).

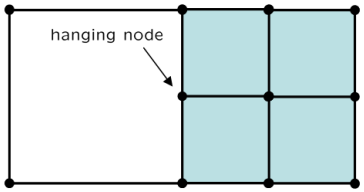


Figure 7-14 Hanging Node

Displacement on hanging nodes need to be constrained or tied to the displacement of corner nodes to avoid a discontinuity in the displacement field, as illustrated in Figure 7-15:

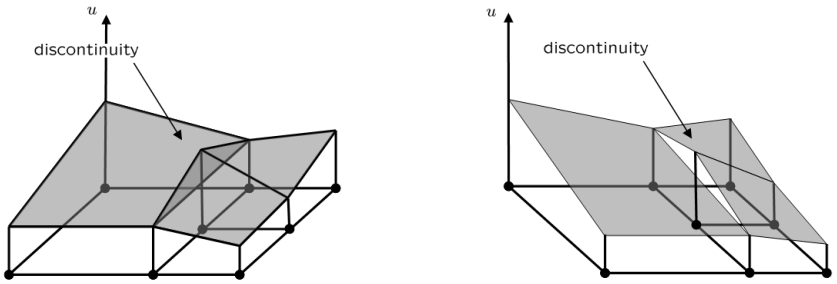


Figure 7-15 Displacement Field over an Incompatible Mesh created due to the Presence of an Unconstrained Hanging Node

All degrees of freedom (1 to 6) associated to a hanging node are automatically constrained using internal Multipoint Constraint (MPC) equations derived from the isoparametric mapping.

Figures 7-16 and 7-17 depicts the MPC equation for hanging nodes laying respectively on a linear and a quadratic edge. Notice that in the linear case, the MPC equation ties each component of the hanging node displacement U_M with those corresponding to the corner nodes 1 and 2 whereas in the quadratic case each components of the hanging node displacements U_M needs to be tied to the corresponding displacements of both the corner nodes 1, 2 and the mid-edge node 3.

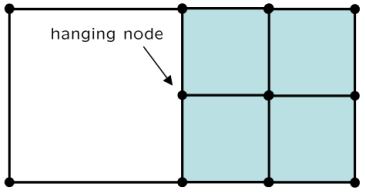
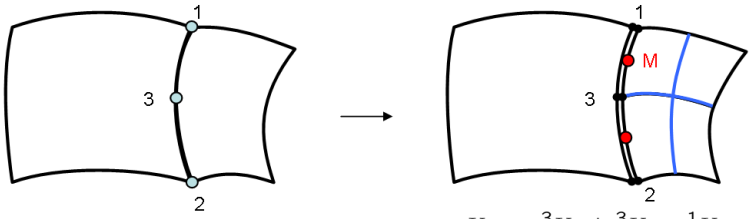


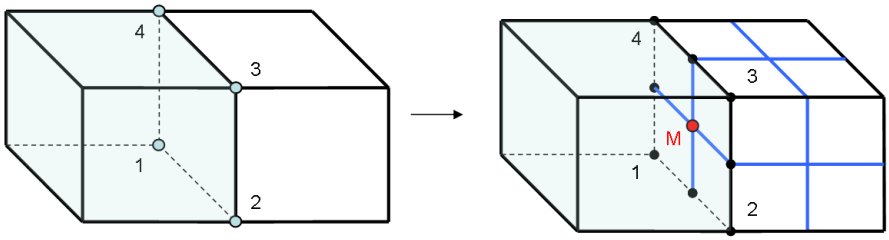
Figure 7-16 Constraint Equations for Hanging Nodes laying on a Linear Edge



$$U_M = \frac{3}{8}U_1 + \frac{3}{4}U_3 - \frac{1}{8}U_2$$

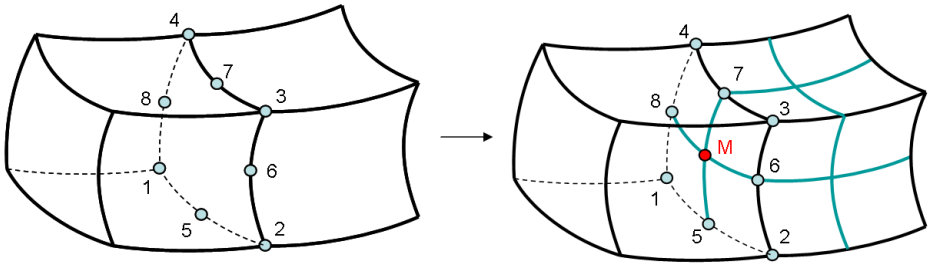
Figure 7-17 Constraint Equations for Hanging Nodes laying on a Quadratic Edge

Figures 7-18 and 7-19 show the MPC equations for hanging nodes laying respectively on a linear and a quadratic face. In the linear case, the MPC equation ties each component of the hanging node displacement with those corresponding to the corner nodes 1, 2, 3 and 4 whereas in the quadratic case, each component of the hanging node displacements needs to be tied to the corresponding displacements of both the corner nodes 1, 2, 3, 4 and the mid-edge nodes 5, 6, 7 and 8.



$$U_M = \frac{1}{4}(U_1 + U_2 + U_3 + U_4)$$

Figure 7-18 Constraint Equation for Hanging Nodes laying on a Linear Face



$$U_M = -\frac{1}{4}(U_1 + U_2 + U_3 + U_4) + \frac{1}{2}(U_5 + U_6 + U_7 + U_8)$$

Figure 7-19 Constraint Equation for Hanging Nodes laying on a Quadratic Face

An example illustrating the need of enforcing constraints on hanging nodes is depicted in Figures 7-20, 7-21, and 7-22 concerning the deformation of a cylindrical shell subjected to a central concentrated load. Figure 7-20 shows the finite element mesh after the third iteration in the adaptive mesh refinement loop along with the hanging nodes involved in MPC equations. Figures 7-21 and 7-22 compare the deformed configuration and von Mises stresses obtained when the hanging

node constraints are enforced (left) or not enforced (right). Notice when the hanging node constraints are not enforced, an incompatible configuration is obtained.

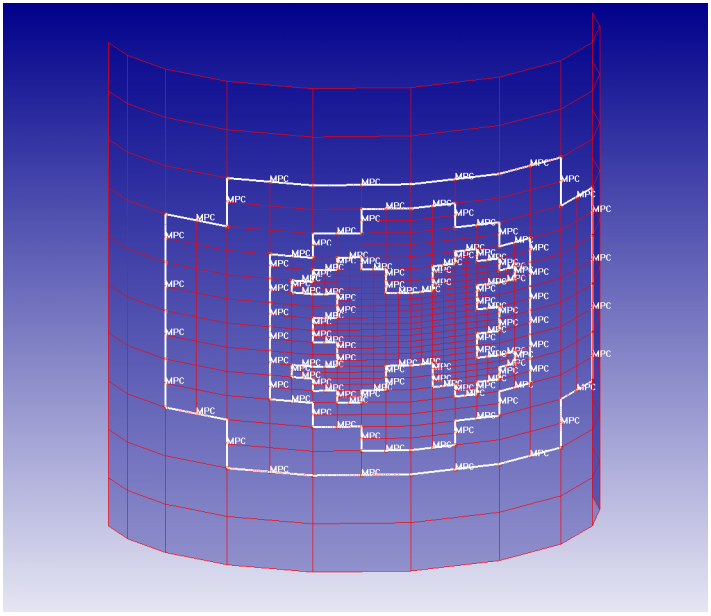


Figure 7-20 Multipoint Constraints on Hanging Nodes

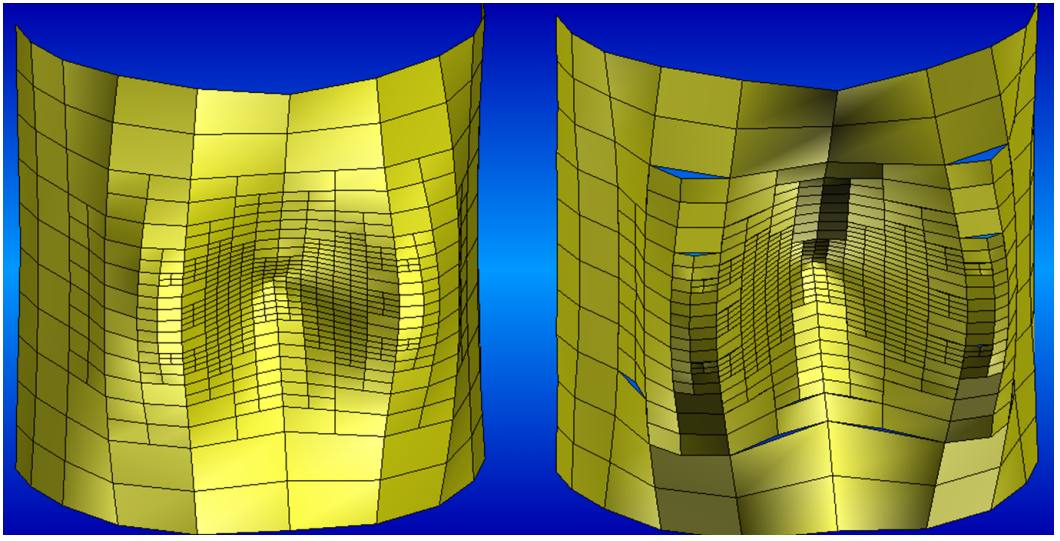


Figure 7-21 Compatible (left) and Incompatible (right) Deformations obtained when Multipoint Constraints are Enforced (left) or not (right)

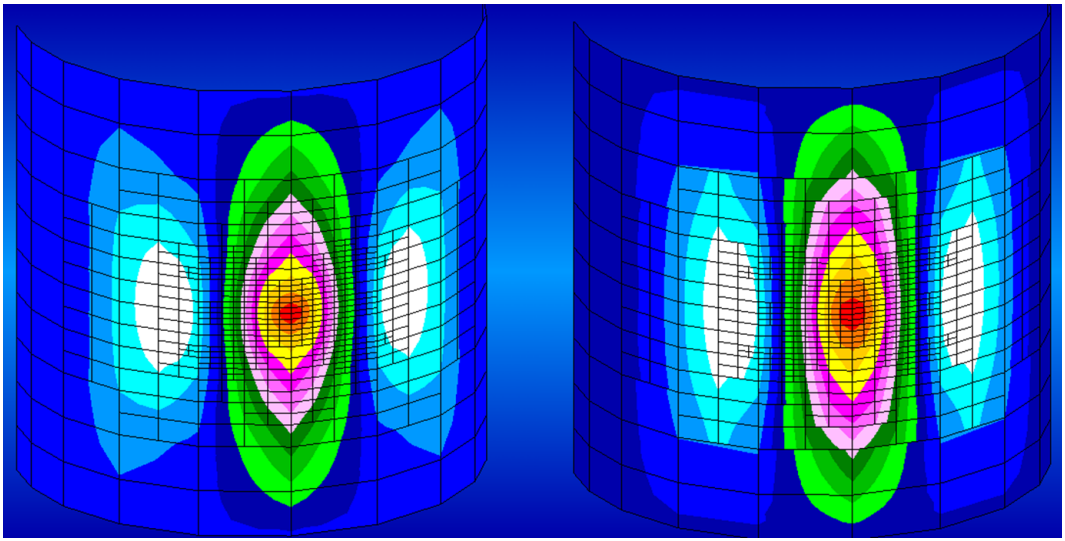


Figure 7-22 Compatible (left) and Incompatible (right) von Mises Stresses obtained when Multipoint Constraints are Enforced (left) or not (right)

Selection of Elements to Refine

Elements that will be refined during a given iteration in the adaptive loop are selected in two steps; first, all elements meeting the user-specified adaptivity criterion are searched for and scheduled for refinement. Second, some of the elements adjacent to the latter are also scheduled for refinement according to a set of implicit propagation rules. If no elements meeting the user-specified criterion are found, the adaptivity loop is terminated.

MSC Nastran currently supports four refinement criteria (see Refinement Criteria, 819):

1. Error indicator based criterion
2. Nodes within a spherical spatial region criteria
3. Nodes within an orthogonal spatial region criteria
4. Nodes in contact criteria

The set of implicit refinement propagation rules are the following (see Propagation of Refinement, 827):

1. Horizontal propagation (2-to-1 rule)
2. Horizontal propagation from inner to outer children in triangular and tetrahedral elements
3. Vertical propagation
4. Propagation across superelement boundaries.

Refinement Criteria

The mesh refinement criteria available in MSC Nastran are the following:

Error Indicator Based Criterion

In this case an *error indicator* E_e is computed over each element 'e' in the finite element mesh. Then, an element is refined if

$$E_e^2 \geq f\bar{E}^2$$

where f is a scalar factor such that $0 \leq f \leq 1$ and chosen by the user (as part of the Bulk Data entry HADACRI) and $\bar{E}$ is the *quadratic mean of the error indicator* defined as:

$$\bar{E}^2 = \frac{1}{N}\sum_{e=1}^{N} E_e^2$$

with N the total number of elements in the element set where element 'e' belongs, Thus, an element is refined if its estimated error is larger than a fixed percentage of the quadratic mean.

Figure 7-23 schematically depicts a mesh with its corresponding elemental error indicator over a 1-D mesh (shown in blue) with N elements and corresponding quadratic mean (shown in red). Error indicator distribution over a 1-D mesh is in blue with N elements and corresponding quadratic mean is shown in red.

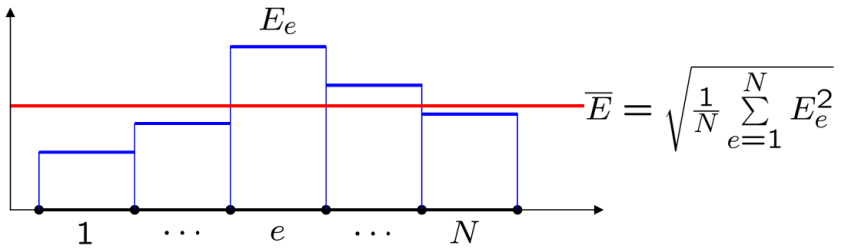


Figure 7-23 Mesh with Element Error Indicator and Qadratic Mean

The factor f is specified by the user in the F1 field of the HADACRI Bulk Data entry (HADACRI (p. 2125) in the).

The error indicator E_e is a scalar, elemental magnitude that provides a relative measure of the discretization error; that is, the error between the finite element solution and the analytical solution of the underlying differential equations of the problem under analysis. It is computed using the grid point stresses and element stress discontinuity following the procedure utilized by the ELSDCON Case Control command and described in the *MSC Nastran Reference Guide,* Section 8.3.

This procedure can be summarized as follows:

- Let $\sigma_{aij} = \sum_{e=1}^{N_a} W_a^e \sigma_{aij}^e$ be the weighted average over all elements 'e' concurrent to a given node 'a' of each component 'ij' of the grid point stresses σ_{aij}^e where W^e is a weighting factor assigned to element 'e' and N_a is the number of elements connected to the given node 'a'.
- An estimate of the error in a particular component of stress 'ij' at a grid point 'a' is then be computed as

$$E_{aij}^2 = \sum_{e=1}^{N} W_a (\sigma_{aij}^e - \sigma_{aij})^2$$

Averaging the latter over the different stress components, 'ij', over the different shell fibers (for shell elements) and over the different grid points 'a' connected by a given element 'e' the elemental, scalar error indicator E_e is obtained.

Figure 7-24 shows an example using the error indicator based adaptivity criterion involving the analysis of a 2-D mode-I fracture specimen. Notice that this criterion tends to cluster the refinement near areas of stress concentration. This is due to the fact that stress gradients (and therefore element stress discontinuities and error indicators) are considerably higher in those zones than in the rest of the mesh.

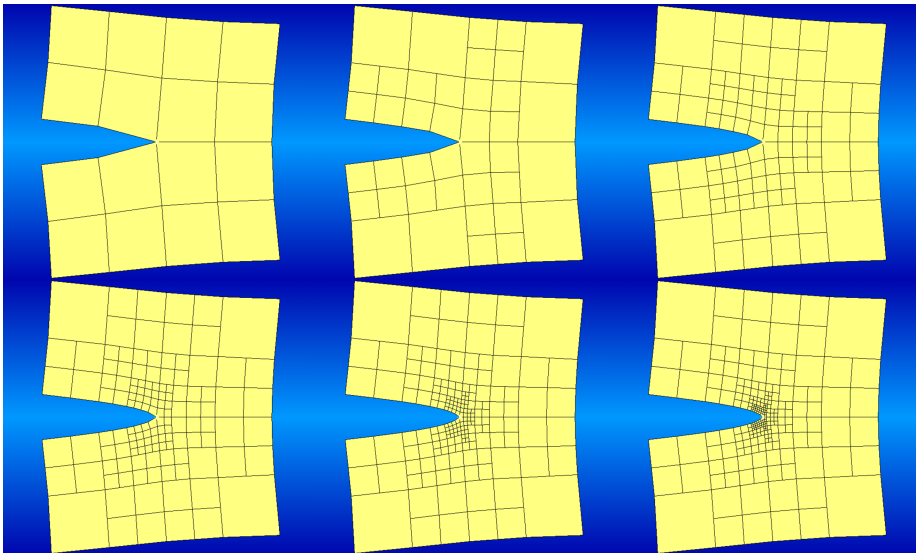


Figure 7-24 Analysis of a 2-D Mode-I Fracture Specimen

This refinement criterion is available for any of the surface or volume elements, namely CTRIA3, CTRIAR, CQUAD4, CQUAD8, CTETRA, CPENTA, CHEXA. It is not available however for the family of line elements. The latter might be subdivided either by using any other criteria of the remaining refinement criterion, or because they are connected to the boundary of a surface or volume element (see Vertical Propagation, 830).

Elements Within a Spatial Spherical Region Criterion

In this criteria, the user defines a spherical region in space by specifying its center in basic coordinate system (X_0, Y_0, Z_0) and its radius R. Then, all elements with at least one node with basic coordinates (X, Y, Z) within the spherical region (i.e., such that $(\|(X, Y, Z) - (X_0, Y_0, Z_0)\| < R)$ || will be refined.

Figure 7-25 shows the mesh obtained in an example involving a 3-D cylindrical body using this criterion. Figure 7-26 shows a detail of the mesh obtained after the fourth refinement cycle along with the spatial spherical region selected for refinement with only the bottom half of the cylinder is shown in the right graphic.

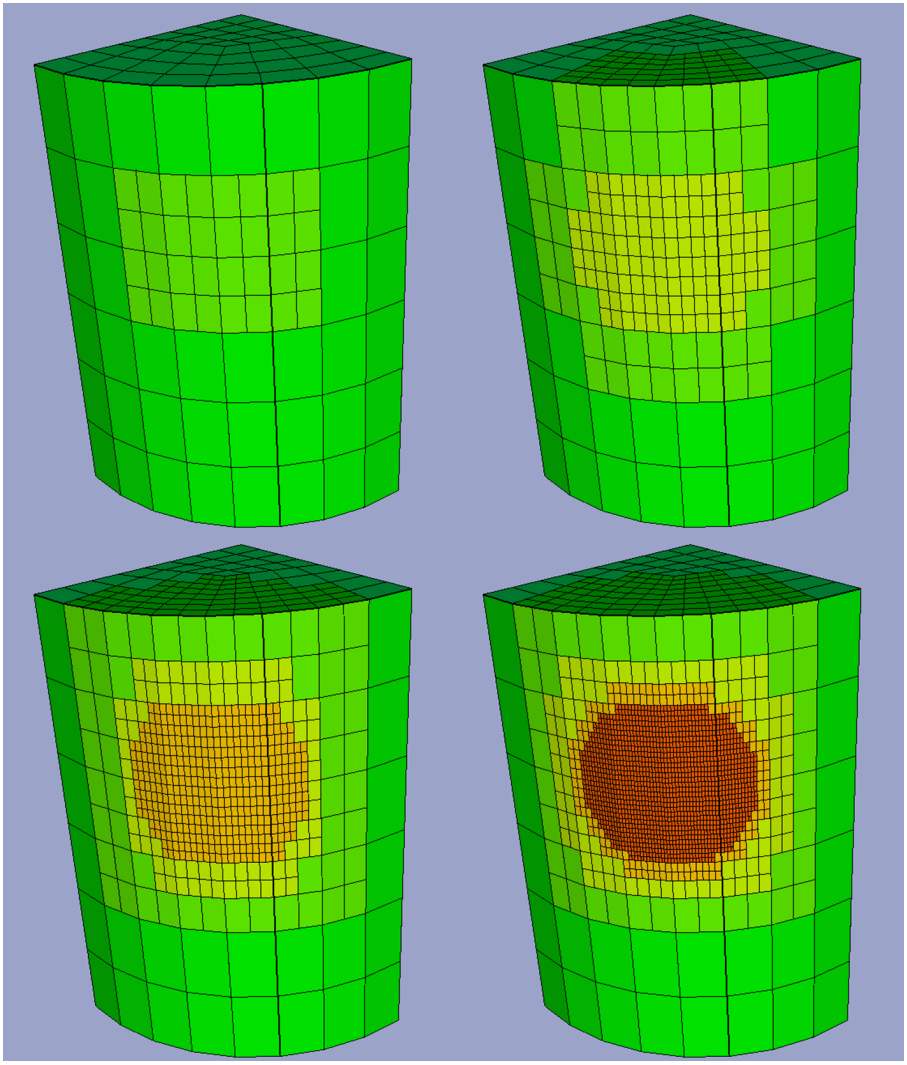


Figure 7-25 Sequence of Meshes obtained on a 3-D Cylindrical Body using the “Elements within a Spherical Region” Adaptivity Criterion

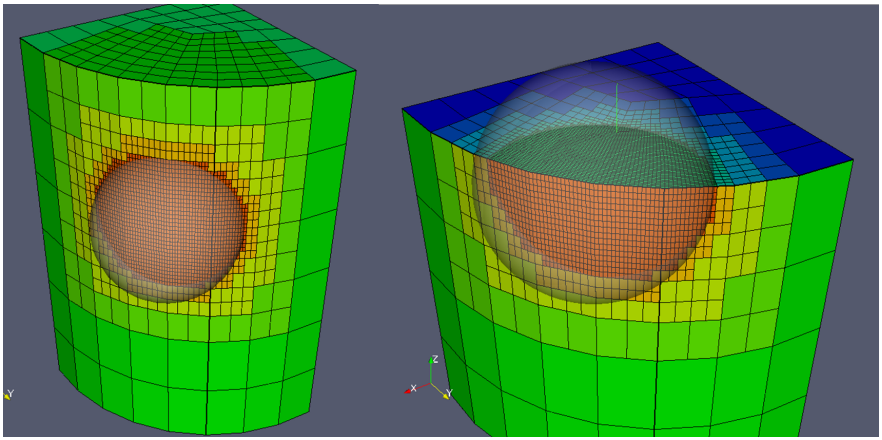


Figure 7-26 Mesh obtained after the Third Refinement Cycle of 3-D Cylindrical Body using the “Elements within a Spherical Region” Adaptivity Criterion

Elements Within a Spatial Orthogonal Region Criterion

In this refinement criterion, the user defines an hexahedral region in space or box aligned with the basic coordinates system by specifying the basic coordinates of opposite corners (X_0, Y_0, Z_0) and (X_1, Y_1, Z_1) of the box. Then, all elements with at least one node with basic coordinates (X, Y, Z) within the specified hexahedral region (i.e., such that $X_0 \le X \le X_1$, $Y_0 \le Y \le Y_1$ and $Z_0 \le Z \le Z_1$) will be refined.

Figure 7-27 shows the mesh obtained in the same 3-D cylindrical body used in Figure 7-25 but with the nodes within a box refinement criterion. A detail of the mesh obtained after the third refinement along with the orthogonal refinement region is shown in Figure 7-28 with only the bottom half of the cylinder is shown in the right graphic.

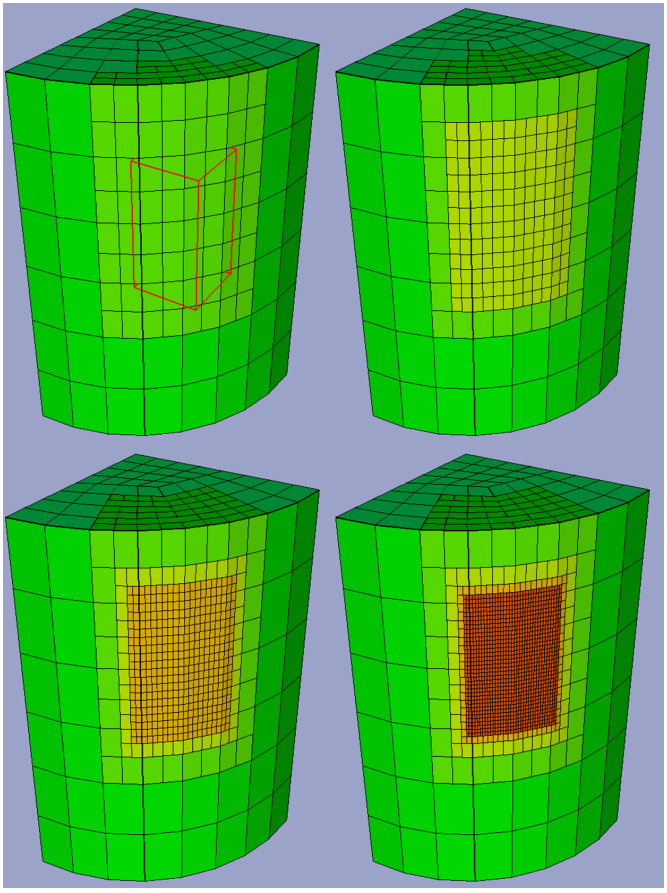


Figure 7-27 Sequence of Meshes obtained on a 3-D Cylindrical Body using the “Elements within an Orthogonal Region” Adaptivity Criterion

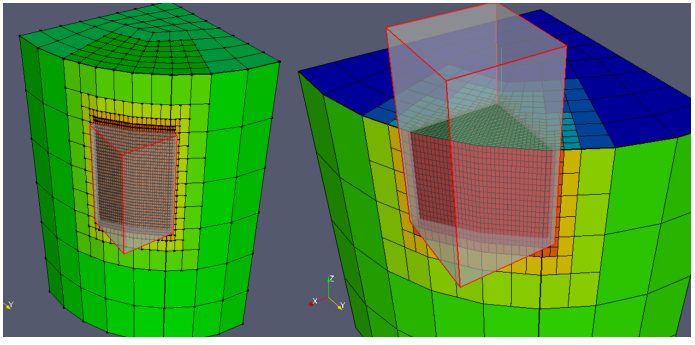


Figure 7-28 Mesh obtained after the Third Refinement Cycle of 3-D Cylindrical Body using the “Elements within an Orthogonal (Box) Region” Adaptivity Criterion

Elements in Contact Criterion

In this criterion, all touching elements with at least one node involved in contact and touched elements with at least one face in contact will be refined. MSC Nastran supports glued contact, rigid-to-deformable body contact, and deformable-to-deformable body contact situations. Figure 7-29 shows the initial mesh of two 3D deformable bodies composed exclusively of linear hexahedral elements and brought into contact after a vertical displacement is applied on the top body.

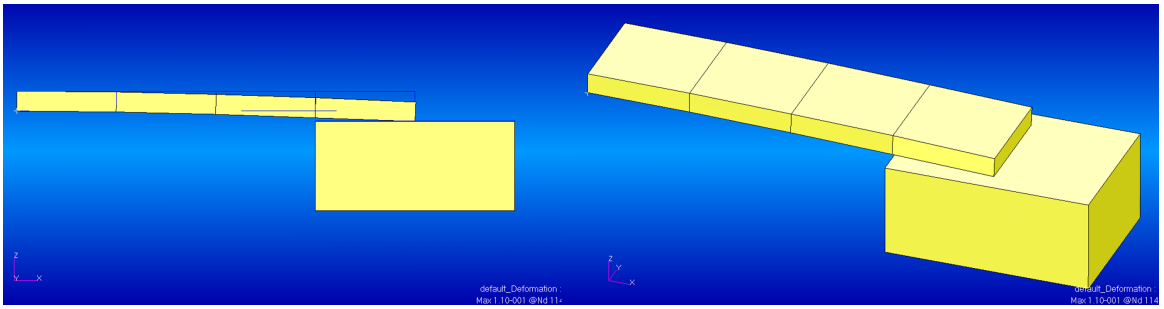


Figure 7-29 Two 3-D Deformable Bodies in Contact

Figure 7-30 shows the sequence of meshes obtained during the mesh refinement process.

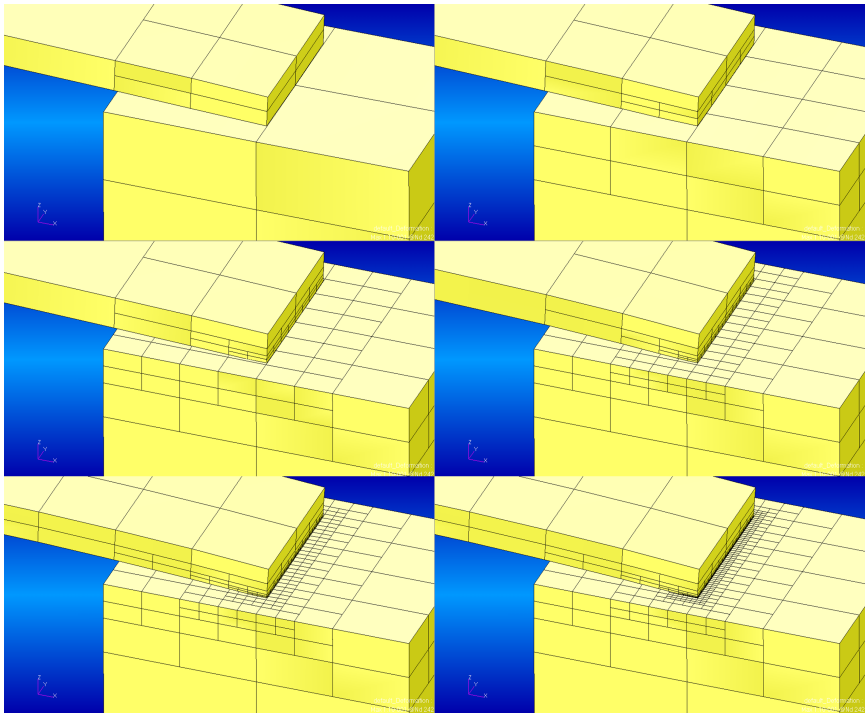


Figure 7-30 Sequence of Meshes during Adaptive Mesh Refinement Process using the “Nodes in Contact” Criterion on Two 3-D Deformable Bodies In Contact

Figures 7-31 and 7-32 depict the sequence of meshes obtained during mesh refinement using the “Nodes in Contact” refinement criterion in a situation involving 3-D rigid-to-deformable contact with 8-noded `CHEXA` elements and 3-D glue contact between two deformable bodies with 10-noded `CTETRA` elements, respectively.

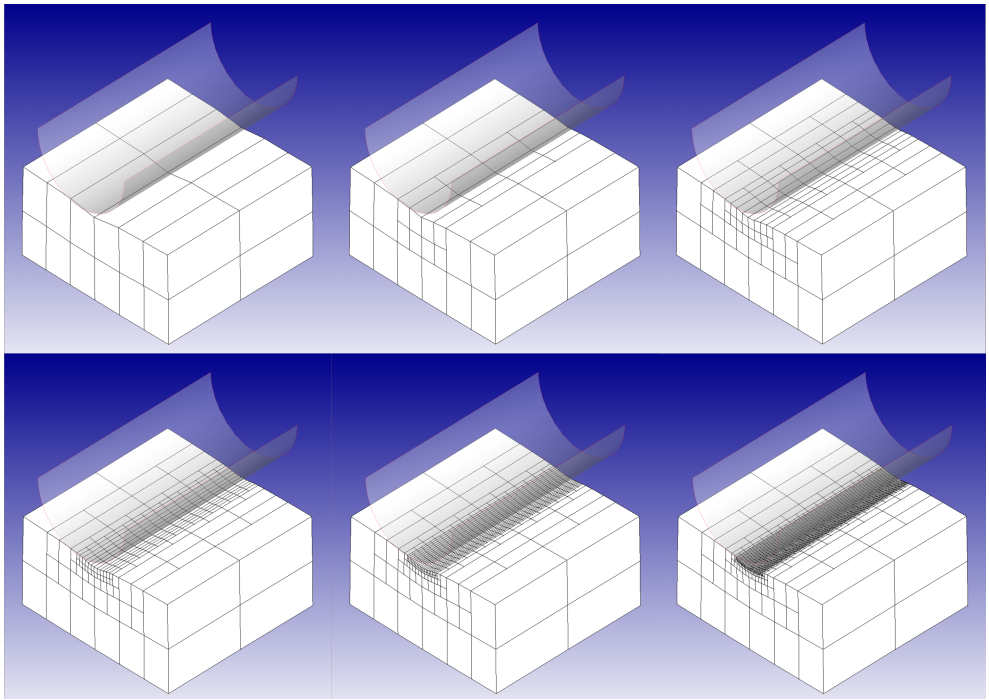


Figure 7-31 Sequence of Meshes during Adaptive Mesh Refinement Process using the “Nodes in Contact” Criterion on a Rigid-to-Deformable Body Contact Setting

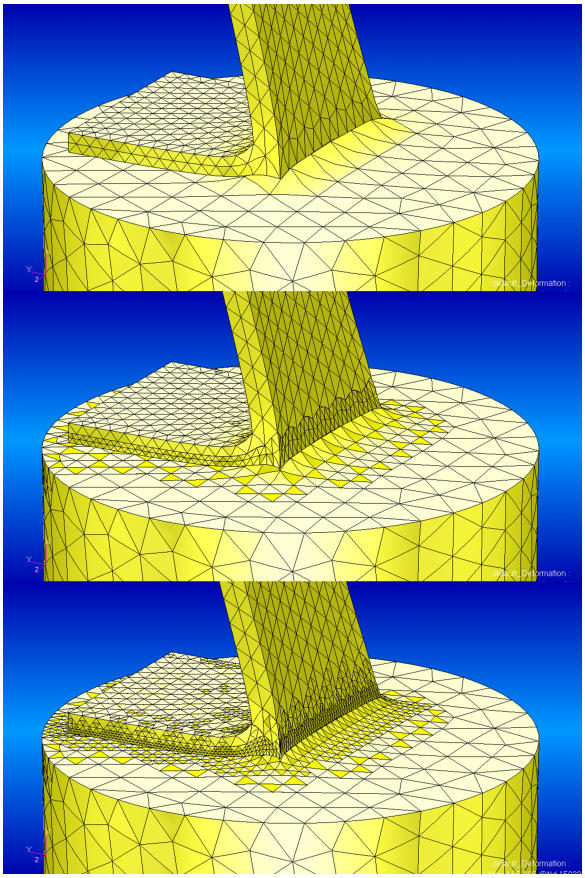


Figure 7-32 Sequence of Meshes during Adaptive Mesh Refinement Process using the “Nodes in Contact” Criterion on Glue Contact

Propagation of Refinement

Once elements meeting the refinement criteria are scheduled for refinement, the refinement is propagated to some of their adjacent elements according to a set of implicit propagation rules. These rules are the following:

Horizontal Propagation (2 to 1 rule)

The first refinement propagation rule is the 2-to-1 rule. The 2-to-1 rule restricts the number of hanging nodes on each edge to one, see Hanging Nodes and Multipoint Constraints on Hanging Nodes, 814. To this end, all the edge neighbors of elements scheduled for refinement are selected for refinement as well. This is illustrated in Figure 7-33. When one element scheduled for refinement (green element) is refined, two hanging nodes are created on its adjacent edges. To restrict the number of hanging nodes to one, the refinement is propagated to its edge-neighbors (yellow elements).

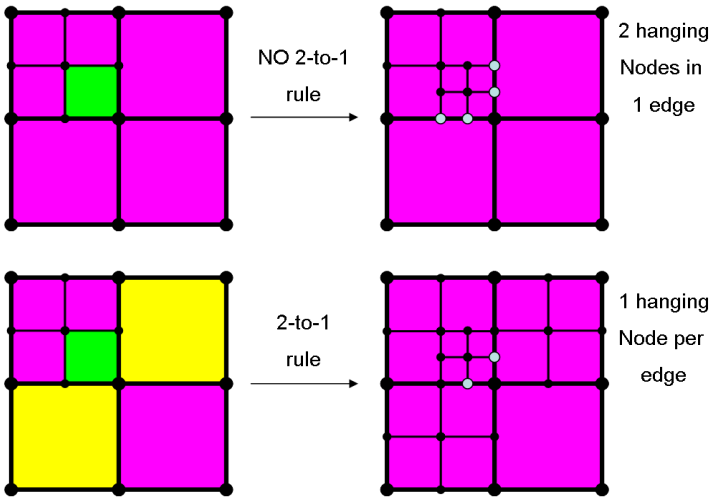


Figure 7-33 Edge Neighbors of Elements Scheduled for Refinement

Notice that in 3-D meshes, there might be more than one edge-neighbor per edge as illustrated in Figure 7-34.

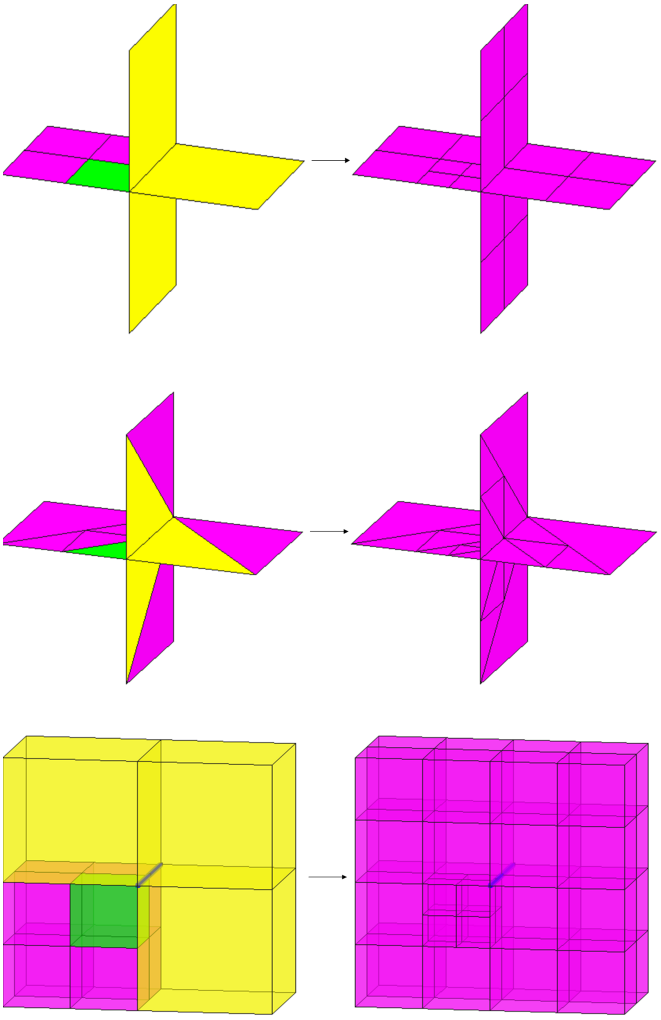


Figure 7-34 Propagation of Refinement from an Element Meeting the User-specified Refinement Criterion (green element) to its Edge-neighbors (yellow elements) to Enforce the 2-to-1 Propagation Rule

Horizontal Propagation in Triangles and Tetrahedra

Consider a refined triangular or tetrahedral element. If the internal triangle or internal tetrahedra are scheduled for a second refinement, then the external triangle or tetrahedral are automatically selected for refinement as well. This is to avoid the creation of redundant degrees of freedom on internal edges or faces that would otherwise be constrained with no net gain of mesh resolution. As shown in Figure 7-35, if the internal (green) triangle of a refined triangular element is further refined,

no net addition of degrees of freedom is obtained (top row). To avoid this redundancy, the refinement is automatically propagated to all external (yellow) triangles of the refined triangular element (bottom row).

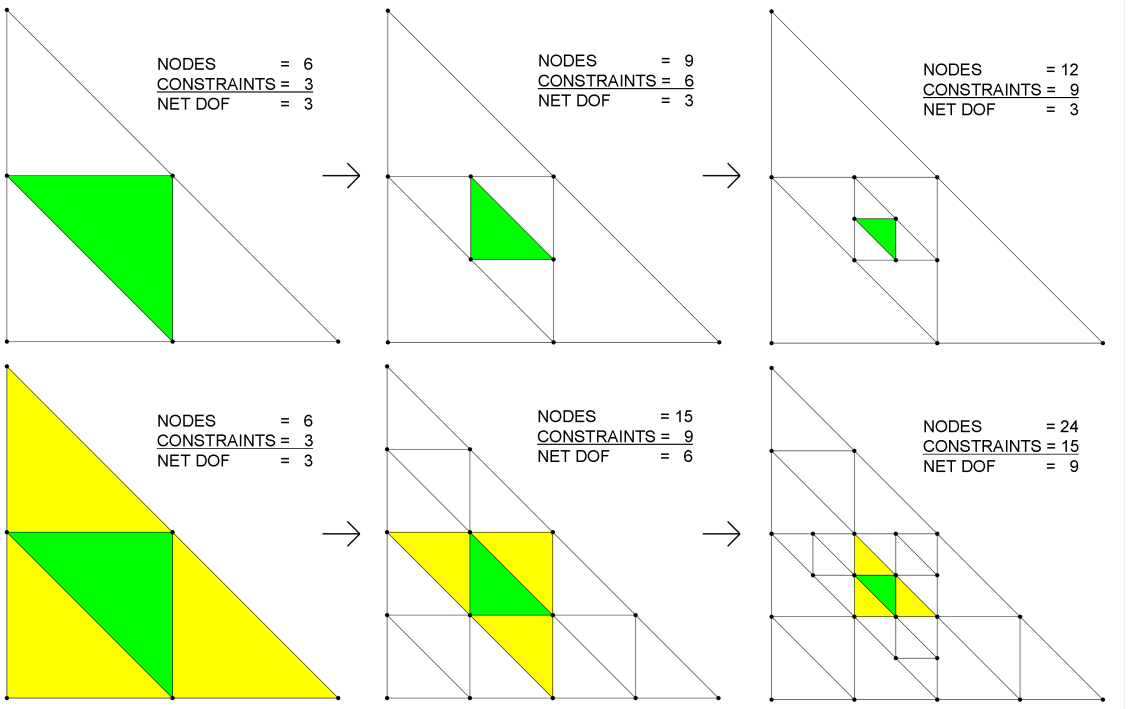


Figure 7-35 Refined Triangular Element

Vertical Propagation

Consider a line element (CBEAM, CBEND, CBAR, CONROD, CROD, CTUBE, CVISC) attached to an edge of a surface element (CQUAD4, CQUADR, CTRIA3, CTRIAR) of a surface element attached to the face of a 3-D element (CTETRA, CPENTA, CHEXA). Then, if the element of higher dimensionality (green) is scheduled for refinement, then the element of lower dimensionality (yellow) attached to its face is automatically selected for refinement as well as shown in Figure 7-36.

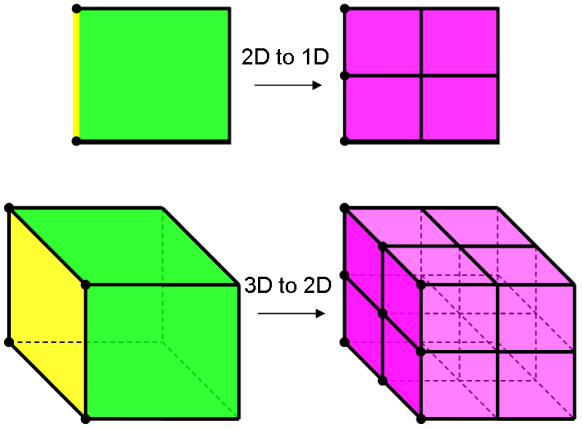


Figure 7-36 Automatic Propagated Refinement - Higher to Lower

The same rule applies in the opposite direction: if an element of lower dimensionality attached to the boundary of another element of higher dimensionality is scheduled for refinement, then the latter is automatically selected for refinement as shown in Figure 7-37.

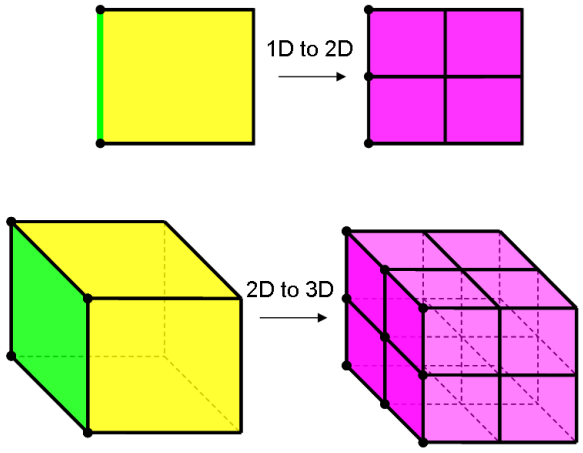


Figure 7-37 Automatically Propagated Refinement - Lower to Higher

Propagation of Refinement Across Partitioned Superelement Boundaries

Hanging nodes cannot occur at partitioned superelement boundaries because their corresponding degrees of freedom cannot belong simultaneously to two different Degree-of-Freedom Sets (see Degree-of-Freedom Sets in the *MSC Nastran Quick Reference Guide*). In order to prevent this condition, the refinement is automatically propagated across superelement boundaries. In this way, hanging nodes are moved from the boundary to the interior of the affected superelements. When

elements on a given superelement are scheduled for refinement (green elements), the refinement is propagated into the neighboring superelement (yellow elements) to avoid the creation of hanging nodes on the superelement boundaries as shown in Figure 7-38.

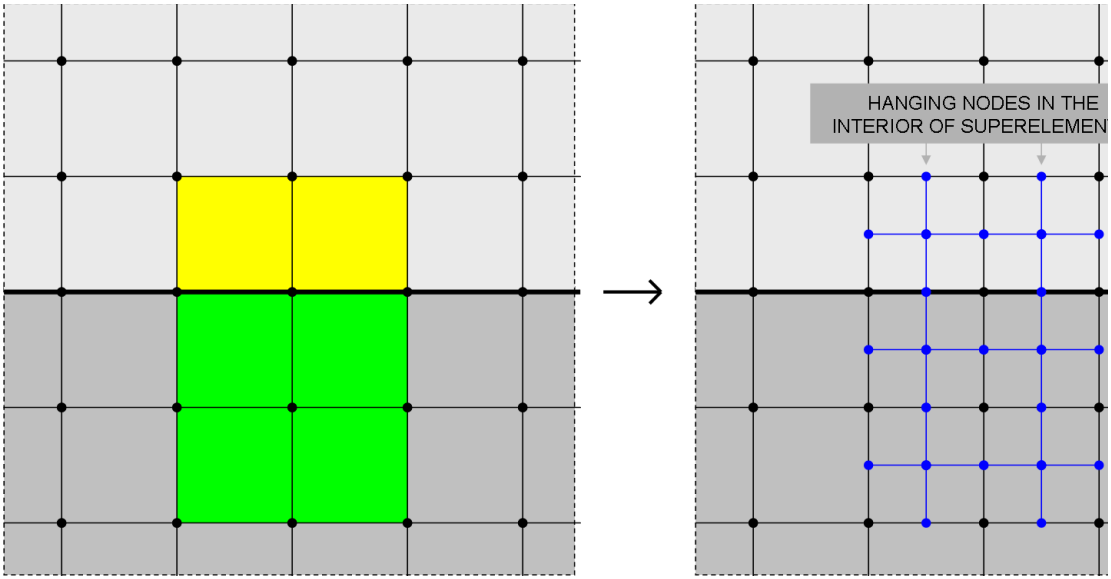


Figure 7-38 Automatically Propagated Refinement - Superelement

Transference of Analysis Data Between Unrefined and Refined Meshes

Once a refined mesh is obtained by subdividing selected elements from the previous mesh, analysis data must be communicated or transferred from the old mesh to the new mesh in order set up the next analysis on the new mesh. Analysis data to transfer includes element properties, shell thicknesses, material orientations, pressure loads and permanent and single point constraints (displacement boundary conditions). The rules to transfer this data are described in the following sections.

Transference of Element Properties

Children elements created after refinement inherit their parent's property ID. In surface elements (CQUAD4, CQUADR, CTRIA3, CTRIAR) with nonuniform shell thicknesses, the thickness for corner nodes in the children elements are linearly interpolated from the corner nodes of the parent element as shown in Figure 7-39.

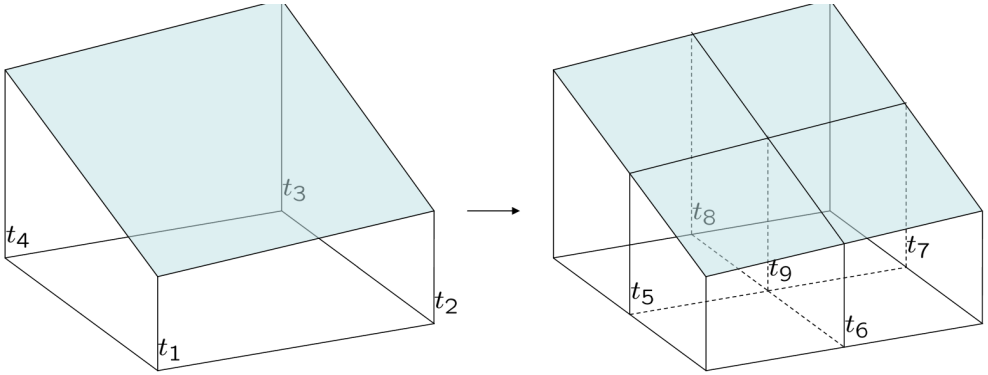


Figure 7-39 Transference of Non-uniform Shell Thickness Data from the Parent to the Children Elements

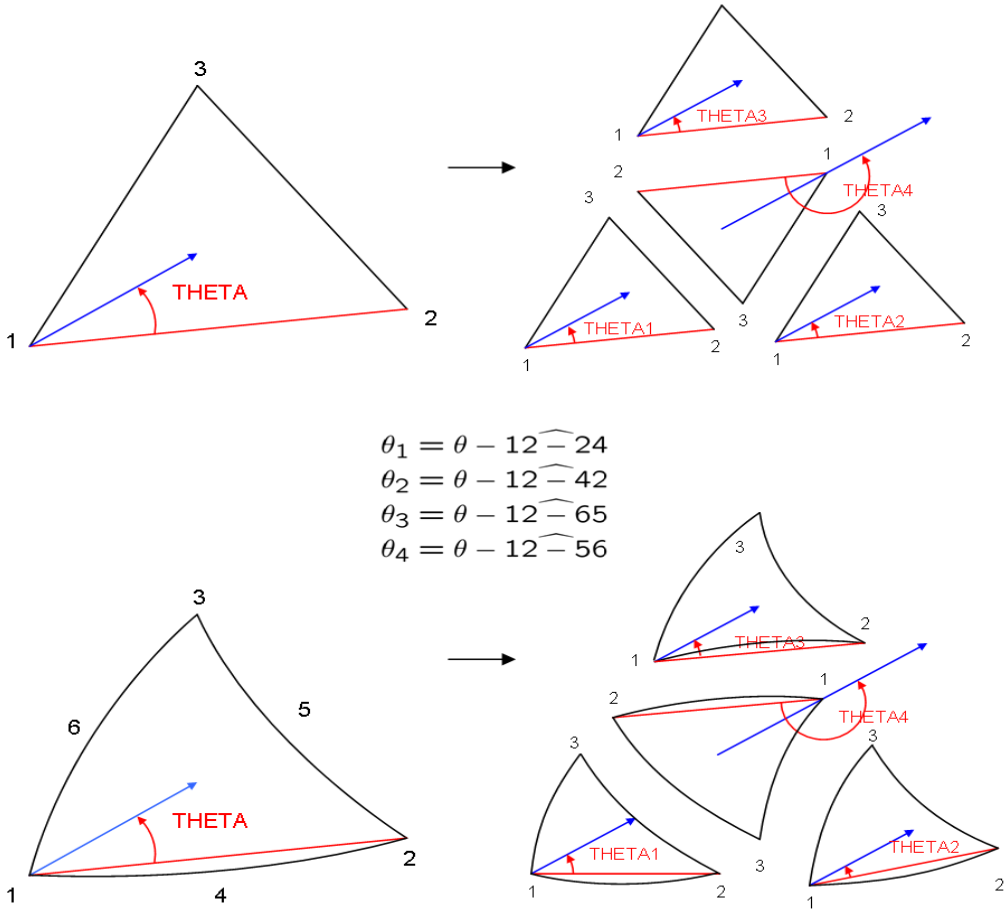


Figure 7-40 Transference of Material Orientation Angle from the Parent of the Children Elements

Furthermore, children elements inherit the material orientation angle (see THETA field on the CQUAD4, CQUADR, CTRIA3, CTRIAR entries). The orientation angle THETA for children element is computed using the equations given in Figure 7-40 with the blue arrow pointing in the direction of the material orientation which is conserved during refinement. Notice that this does not imply the conservation of the THETA angle. This takes into account that THETA is defined as the angle between the edge joining nodes 1 and 2 of the element and the material direction and, therefore, might not be uniform within children elements. Notice that uniform material orientation does not imply uniform angle THETA.

Transference of Distributed Loads and Concentrated Forces

Pressure loads (PLOAD, PLOAD2, PLOAD4) distributed over parent elements are automatically redistributed over children elements.

Thus, a uniform pressure load distributed over a quadrilateral surface element (PLOAD, PLOAD2) is copied over children elements and redistributed with same magnitude and direction over their smaller area.

Similarly, the magnitude of non-uniform pressure loads distributed over surface elements faces of 3-D elements or faces of surface elements (PLOAD4) are not just copied over but interpolated linearly from corner pressures (Figure 7-41) and applied to children elements with the same direction. Mid-edge and mid-face pressures are linearly interpolated from corner values.

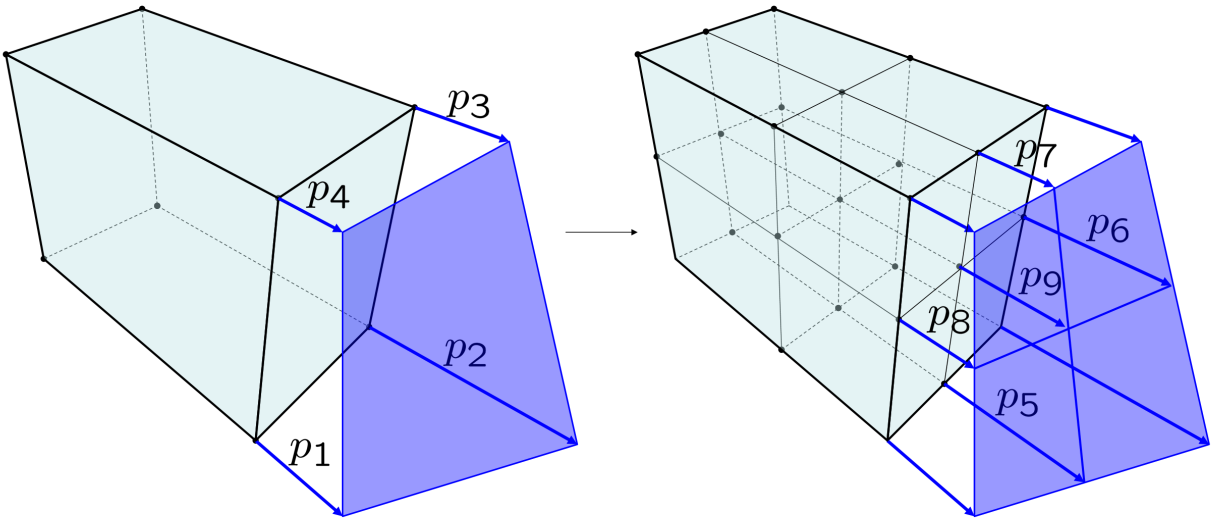


Figure 7-41 Transference of PLOAD4 applied to the Face of a 3-D (CHEXA) Element

No special provisions are taken regarding transference of concentrated forces or moments. Thus, no concentrated loads will be created or new grids created for refinement but just carried over existing grid points in the old mesh to the same grid points in the new mesh.

Transference of Displacement Coordinate System, Displacement Boundary Conditions, and Constraints

Displacement coordinate systems, permanent single point constraints, single point constraints and multipoint constraints on the new grid point *c* created during refinement are internally enforced according to the following rules:

- If the coordinates systems defined on corner grid points (specified on the CD field in the GRID Bulk Data entry) are identical, then the same displacement coordinate system is assigned for a new mid-edge or mid-face node created during refinement. Furthermore, for edges of surface or volume elements, each degree of freedom associated to the new mid-node created on the edge is either permanently constrained (assigned to degree of

freedom set *SG*), or explicitly constrained via an internal Single point constrain SPC (assigned to the degree of freedom set *SB*) or constrained to corner nodes via an internal Multipoint Constraints *M* (see Degree-of-Freedom Sets in the *MSC Nastran Quick Reference Guide*) depending upon the degree of freedom set where the corner nodes.

For faces of volume elements, each degree of freedom of a new mid-face node created during refinement is also either permanently constrained, or explicitly constrained (via internal SPC) or tied to corner nodes (via internal MPC) according to the permanent constraints, enforced displacement or multipoint constraints defined on the corner nodes.

The specific constraints to be enforced internally for mid-edge or mid-face nodes are:

- If permanent constraints have been defined on a degree of freedom *i* at both corner nodes, then a permanent constraint is also assigned to the degree of freedom *i* at the mid-edge of mid-face node. For example (as shown in Figure 7-42), if both corner nodes *a* and *b* on an edge have been permanently constrained in all degrees of freedom (by defining $PS_a = 123456$ and $PS_b = 123456$ on the PS field in the GRID Bulk Data entry), then node *c* will be also constrained permanently in all degrees of freedom (by defining $PS_c = 123456$ internally).

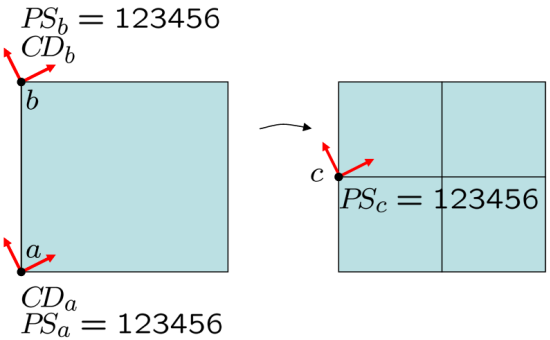


Figure 7-42 Coordinate systems (CD_a and CD_b)

Coordinate systems (CD_a and CD_b) defined on corner node *a* and *b* (using the CD field in the Grid Bulk Data entry) must match in order for constraints to be enforced on mid-edge node c. Permanent constraints (PS field on the GRID Bulk Data entry) on each degree of freedom are then carried over the mid-edge node.

- If single point constraints have been defined on a degree of freedom *i* for all corner nodes, then the degree of freedom *i* on the mid-edge or mid-face node will be tied internally to the corner nodes according to the same multipoint constraint equations used for hanging nodes on straight edges (see Hanging Nodes and Multipoint Constraints on Hanging Nodes, 814, Figures 7-16 and 7-18).

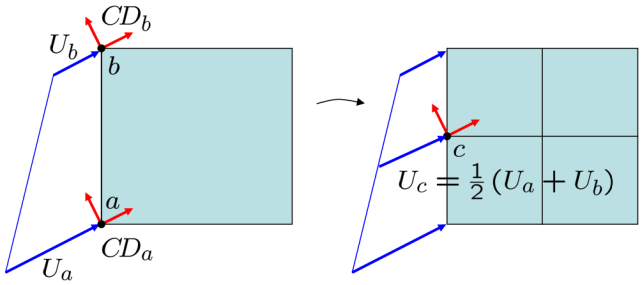


Figure 7-43 If displacement is enforced on a given degree of freedom on corner nodes a and b (using SPC or SPC1 entries), then a multipoint constraint that ties the mid-edge node c with both corner nodes is internally enforced, provided that displacement coordinate systems for all corner nodes coincide (CD field in the GRID entry).

Notice that this is equivalent to imposing an SPC on the mid-edge node, degree of freedom *i*, with a value for enforced displacement averaged from the value at corner nodes. For example, if a displacement with value U_a along direction 1 is enforced on node *a* (by defining an SPC on node *a*, direction 1) and a displacement in the same direction with value U_b is enforced on node *b* (by defining another SPC on node *b*, direction 1), then a multipoint constrain for node *c* (direction 1) is internally defined such that $U_c = 1/2(U_a + U_b)$ as shown in Figure 7-43.

- The same multipoint constraint equation is applied on the mid-edge node *c* (in a given direction *i)* if one of the corner nodes has been constrained permanently and the other corner node has been constrained via a Single Point Constraint (SPC) on the same direction *i*. For example, if a displacement with value U_a along direction 1 is enforced on node *a* (by defining an SPC on node *a*, direction 1) and a permanent constraint in the same direction has been enforced for corner node *b* (by defining $PS_b = 1$ on the *PS* field in its GRID Bulk Data entry), then a multipoint constraint for node *c* (direction 1) is internally defined such that $U_c = 1/2(U_a + U_b) = 1/2(U_a + 0) = 1/2U_a$ as shown in Figure 7-44.

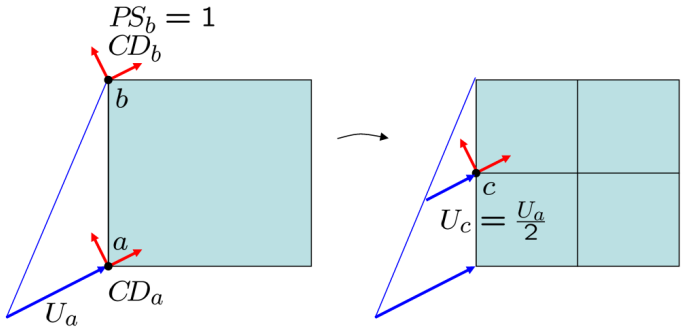


Figure 7-44 If a displacement is enforced on a given degree of freedom on corner node a (using SPC or SPC1 entries) and a permanent constraint is enforced on the same degree of freedom on corner node b (PS=1 on its GRID Bulk Data entry), then a multipoint constraint that ties the mid-edge node c with both corner nodes is internally enforced, provided that displacement coordinate systems for all corner nodes coincide (CD field in GRID entry).

- Similarly, the same multipoint constraint is applied on the mid-edge or mid-face nodes (in a given degree of freedom *i)* if some of the corner nodes are involved in a permanent constraint (PS field in the GRID Bulk Data entry) or single point constraint (SPC or SPC1 Bulk Data entry) and the others corner nodes are involved in a multipoint constraint (MPC Bulk Data entry) on the same degree of freedom *i*.
- If the corner nodes are involved in contact, either as touching nodes or touched element faces or edges, then the mid-edge nodes are regarded as nodes potentially in contact. Therefore, constraints on any of the degrees of freedom associated to the latter are determined by the contact detection algorithm.

The set of relations just outlined is summarized in the following table:

Corner node a	Corner node b	Mid-node c
SG	SG	SG
SB	SB	M
SG SB	SB SG	M
SG or SB M	M SG or SB	M
Node in contact	Node in contact	Node potentially in contact

- If the displacement coordinate system defined on corner nodes on an edge or face are different, then the displacement coordinate system for the mid-node on the edge or face is set to the basic coordinate system. Furthermore, no constraints are enforced on any of its associated degrees of freedom independently of the constraints that might have been imposed on corner nodes as shown in Figure 7-45.

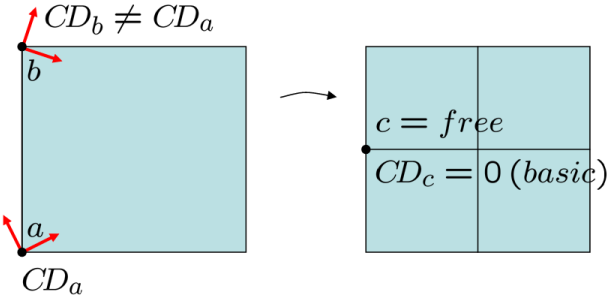


Figure 7-45 If coordinate systems (CD_a and CD_b) defined on corner nodes a and b (using the CD field in the GRID Bulk Data entry) are different then the mid-edge node c is left free and its displacement coordinate system is set to the basic.

Detection of Geometric Features

Prior to the initiation of the adaptive mesh refinement loop, the initial mesh provided by the user is preprocessed using an automatic *Geometric Feature and Material interface Detection Algorithm* aimed to identify:

- Geometric features such as:
  - Sharp corners and edges
  - Non-manifold edges and vertices, i.e., edges joining more than two surfaces (in 3D) or vertices joining more than two curves in 2-D or 3-D.

The Geometric Feature Detection Algorithm identifies edges and corners by comparing the angle between each pair of adjacent elements with the "*Feature Angle*" VARPHI. The feature angle is a scalar parameter specified by the user (PARAM,VARPHI) that defines how sharp a mesh edge or vertex should be in order to be considered as a geometric feature.

More precisely, face outward normal vectors N_1, N_2 of each pair of adjacent mesh faces and the edge oriented tangents T_1, T_2 of each pair of adjacent mesh edges are computed (Figure 7-46) by the geometric feature detection algorithm. If the angle between N_1 and N_2 for mesh faces, or between T_1 and T_2 for mesh edges is larger than the feature angle VARPHI then the common edge or vertex will be considered a splitting edge or vertex where surfaces or lines are broken and a geometric feature is thus defined.

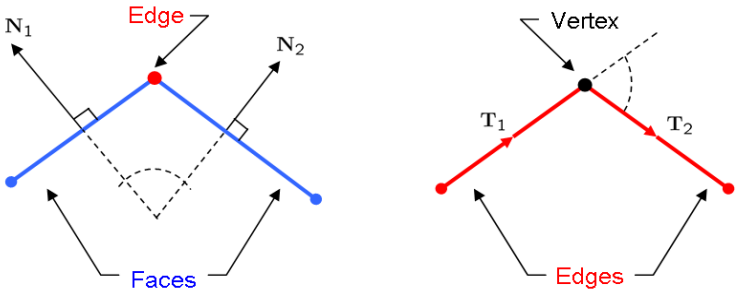


Figure 7-46

Mesh faces and elements are preprocessed to ensure consistent orientation and that the appropriate sign of face normal vectors and edge tangents will be accounted for during the computation of their mutual angle.

Figures 7-47, 7-48, 7-49, and 7-50 show the edges identified by the Geometric Feature detection algorithm on a surface mesh (Figure 7-47) and three 3-D volume meshes (Figures 7-47, 7-48, and 7-49).

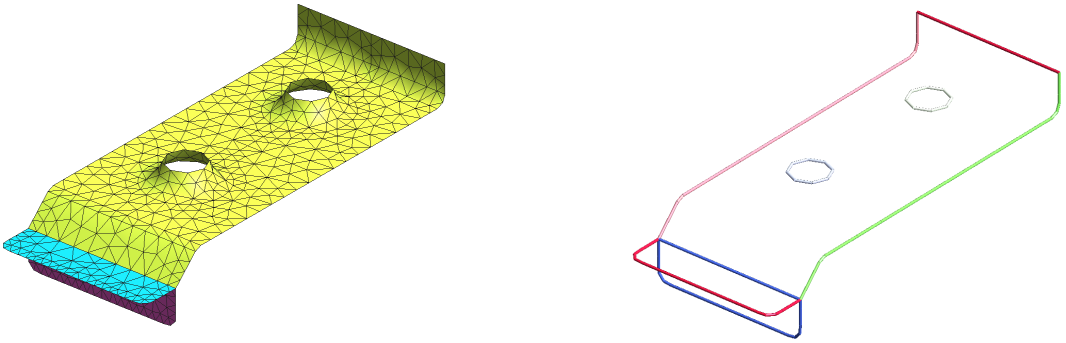


Figure 7-47 Edges detected by the geometric feature detection algorithm in a surface mesh of triangular elements of a mechanical part with a non-manifold edge

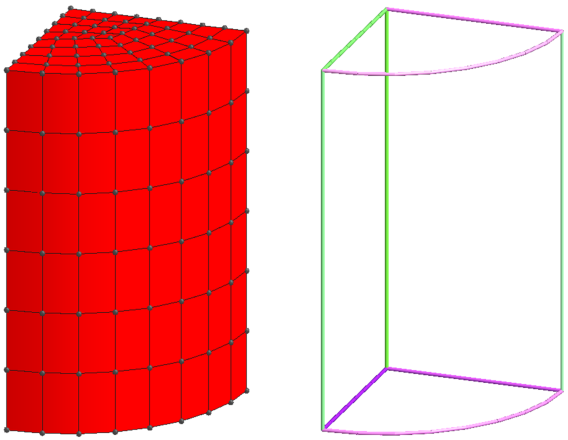


Figure 7-48 Edges detected by the geometric feature detection algorithm in a 3-D hexahedral mesh of a cylindrical body

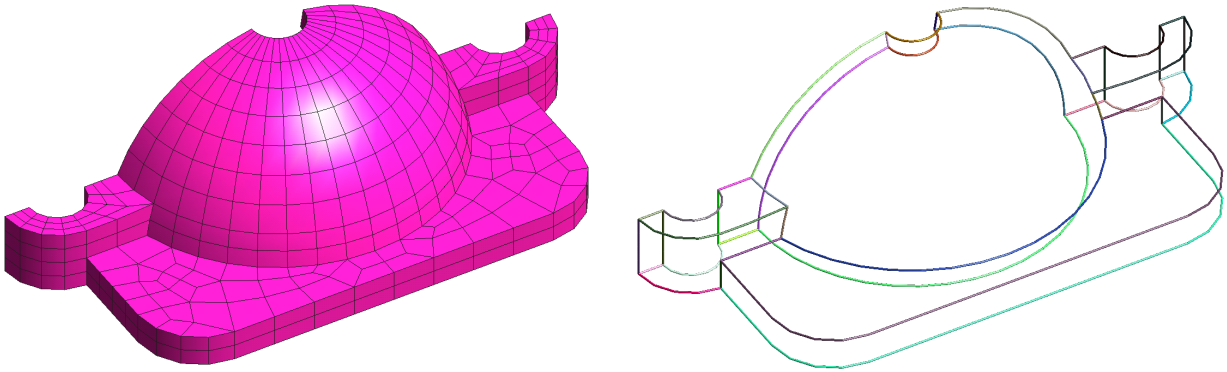


Figure 7-49 Edges detected by the geometric feature detection algorithm in a 3-D hexahedral mesh of an engine cup

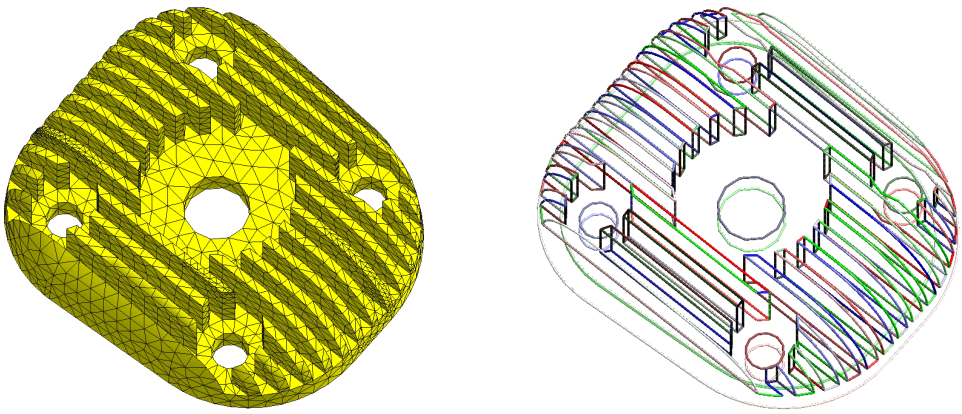


Figure 7-50 Edges detected by the geometric feature detection algorithm in a 3-D hexahedral mesh of an engine cylinder head

Adequate identification of geometric features (by appropriately adjusting the feature angle VARPHI) is essential to ensure the convergence of the mesh refinement process to expected results.

Thus, if the error indicator based refinement criterion is selected and sharp edges are not properly identified by the geometric feature detection algorithm, the refinement might cluster indefinitely in the neighborhood of the undetected sharp edges as shown in Figure 7-47. This anomaly occurs mainly on sharp intersections between shells due to the fact that the error indicator indirectly measures membrane stress jumps between adjacent elements and the latter might be very high due to the abrupt change in shell normal directions at sharp intersections.

Figures 7-51 and 7-52 depict two orthogonal planar shells joined on a common edge and subjected to a vertical displacement on the top. The feature angle must be chosen smaller than $\pi/2$ in order for the joining edge to be detected and the error estimator to ignore (or filter) the large membrane stress discontinuity across this edge as shown in Figure 7-51.

If the feature angle is not appropriately chosen, then the geometric feature algorithm will fail to identify the joining edge and the mesh refinement will cluster in its neighborhood as shown in Figure 7-52.

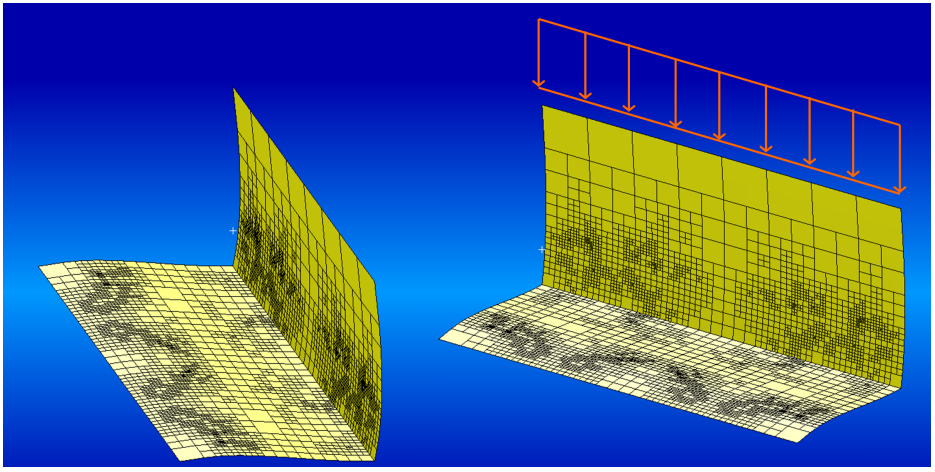


Figure 7-51 Clustering during Mesh Refinement

An adequate value of the VARPHI parameter ($\pi/4$ in this case) will ensure that the sharp edge shared by both planar shells is detected by the geometric feature detection algorithm and the big membrane stress jumps occurring at the edge are filtered out.

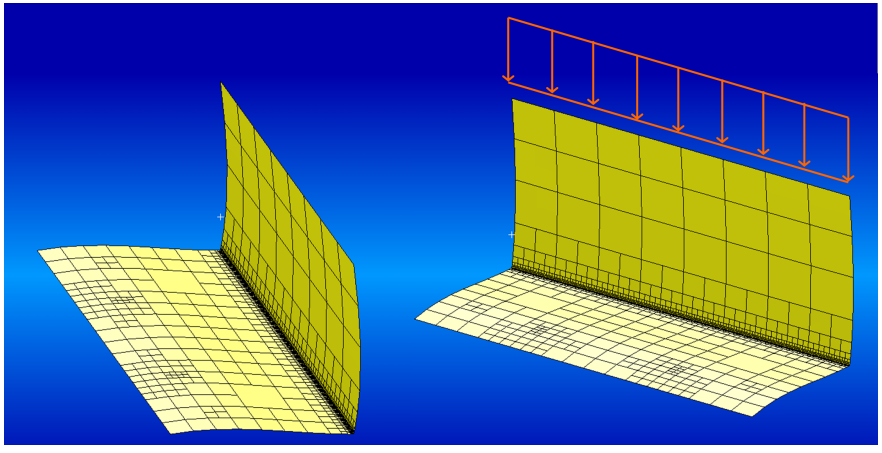


Figure 7-52 Filtering of Stress Jumps at Edges

The geometric feature detection algorithm fails to detect the sharp edge shared by both planar shells because the VARPHI parameter is too large. As a consequence, the big membrane stress jumps occurring at the edge are not filtered out and the refinement clusters in the neighborhood of the sharp edge.

The adequate identification of corners is also required to improve the smooth approximation of the analysis domain boundary constructed by interpolating the mesh boundary nodes and used as a method to place new mid-edge nodes during refinement (see Location of New Grid Points, 811) alternative to the default location at the mid-side of the edge.

Figures 7-53 and 7-54 compare the two edge-node placement methods (mid-side placement and projection of mid-edge nodes onto a smooth approximation of the boundary) in an example involving a 2-D treble shaped planar shell subjected to compression. The boundary of this mesh exhibits three sharp corners located at the intersection of each pair of circular leaves. The mesh is refined everywhere (uniform refinement). This is accomplished by selecting the "nodes within a spatial sphere" refinement criteria (see Refinement Criteria, 819) with a spherical refinement region big enough to contain the whole mesh.

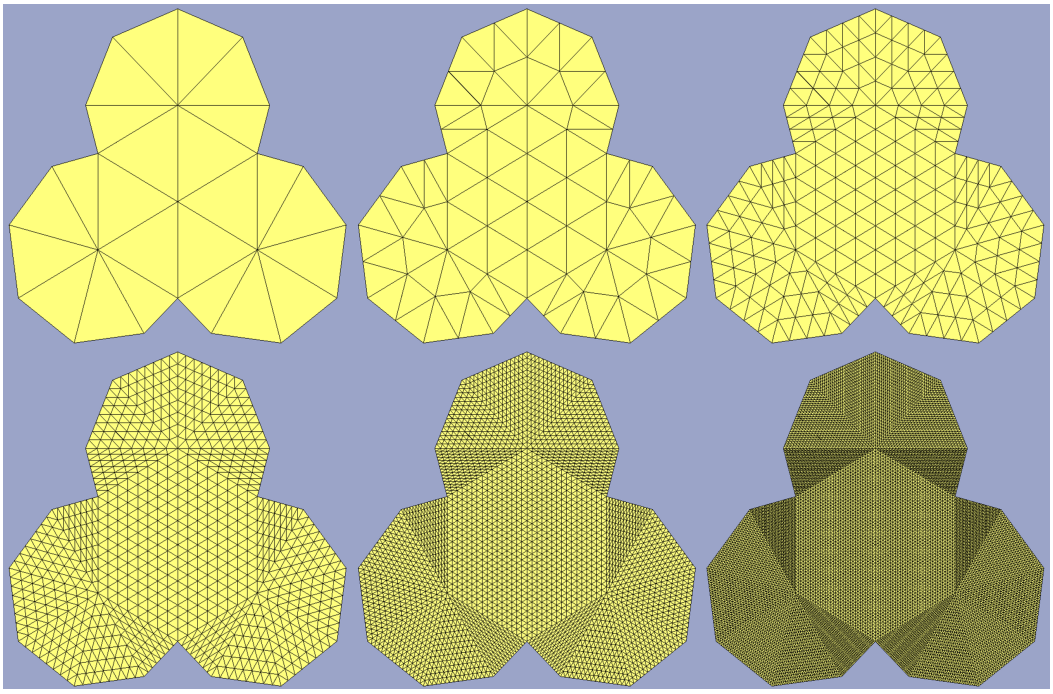


Figure 7-53 Mid-edge Nodes are placed in the Mid-side of Edges

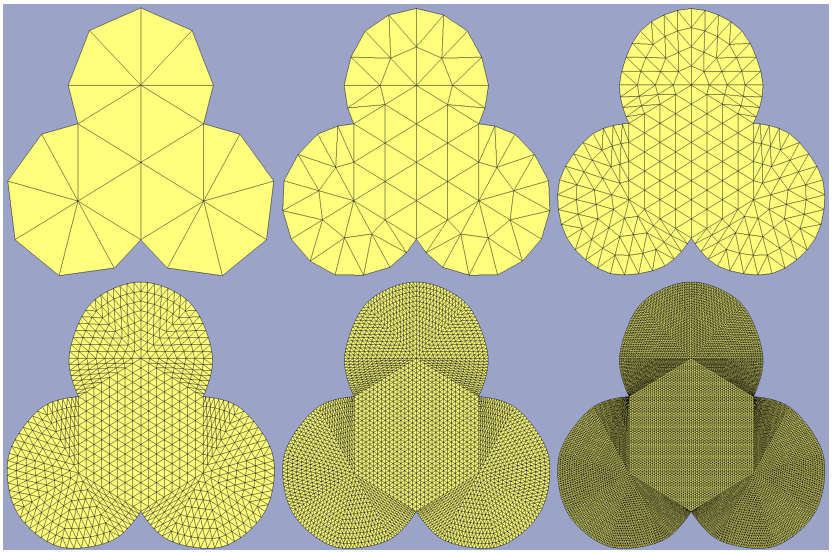


Figure 7-54 Projection of Mid-edge Nodes onto a Smooth Approximation of the Geometric Boundary Interpolated from the Initial Mesh

Notice that the three sharp corners are appropriately detected by the Geometric Feature Detection Algorithm and kept as hard points during the mesh refinement process. By contrast, sharp corners might become smeared out if the geometric feature detection algorithm is not successful due to an inadequate selection of the feature angle (parameter VARPHI) (Figure 7-55).

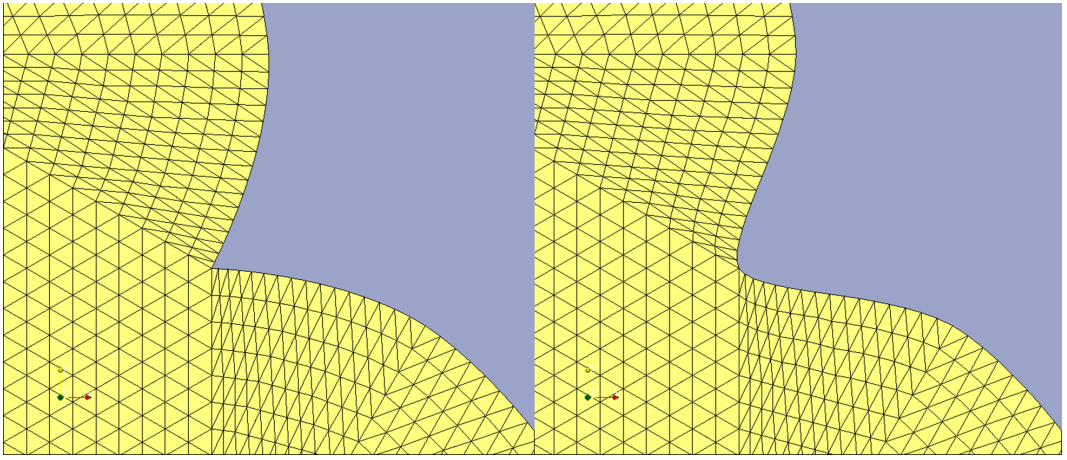


Figure 7-55 Corners might be smeared out if they are not appropriately detected by the automatic geometry feature detection algorithm. Corner detection can be controlled by the user adjusting the VARPHI parameter.

Detection of Material, and Superelement Interfaces

Interfaces between mesh regions with different properties IDs or superelement IDs are also automatically detected during the preprocessing phase prior to the beginning of the adaptive mesh refinement loop.

Different properties or superelement may reference different materials or different shell thicknesses. Therefore, stress of different order of magnitude are expected in areas with different properties or superelements. This type of discontinuities must be filtered out by the error indicator (which averages stress jumps across interelement boundaries) in order to be able to capture the discontinuities introduced by the finite element discretization exclusively.

Index

MSC Nastran Reference Manual

A

ACCELERATION, 28, 29
ACSRCE, 306
ADAPT, 197, 760, 763, 764, 791
adaptive bisection, 715
ADPCON, 189
ADUM8, 177
ADUM9, 177
ADUMi, 47
aeroelastic response, 21, 28
AEROF, 26, 28
ANALYSIS, 16, 22, 23
angular acceleration, 299
angular velocity, 299
APRESSURE, 26
ASETi, 333
ASSIGN, 456
Augmented Lagrange Multiplier Method, 164
AUTOASGN, 458
automatic time stepping, 713
AUTOSPC, 133, 331
axial damping element, 205
axisymmetric elements, 46, 175
AXSLOT, 725

B

B2PP, 20, 31
BAR, 246
bar element, 65
BAROR, 67
BCONP, 189
BEAM, 246
beam element, 55
BEGIN BULK delimiter, 12
BFRIC, 189
BLSEG, 189
BOUTPUT, 189
BSBRCD subroutine, 85
BSBRGD subroutine, 85
BSBRID subroutine, 85
BSBRPD subroutine, 85
BSBRT subroutine, 85
BSCON subroutine, 84
BSGRQ subroutine, 85
BSMSG subroutine, 85
buckling analysis, 20, 27, 691
Bulk Data Section, 12
BUSH1D, 205, 206, 659
BWIDTH, 189

C

Case Control commands for SOLs
- ADAPT, 16, 20
- GRID, 16
- METHOD(FLUID), 20
- OUTRCV, 16, 20
- SDMAPING(FLUID), 20
- SMETHOD, 16
- SPC, 16
- SUPORT1, 16
- VUGRID, 20

Case Control Section, 12, 15
CAXIF2, 725
CAXIF3, 725
CAXIF4, 725
CAXIFi, 48, 50, 53
CBAR, 48, 50, 53, 55, 65
CBARAO, 298
CBEAM, 48, 50, 53, 55, 59, 61
CBEAM3, 48
CBEND, 48, 50, 53, 55, 67
CBUSH, 48, 50, 53, 201
- frequency dependency, 201
- load-displacement dependency, 201

CBUSH1D, 48, 50, 53, 205
CCONEAX, 46, 48, 51, 53
CDAMP1, 175
CDAMP2, 175

CDAMP3, 175
CDAMP4, 175
CDAMP5, 175
CDAMPi, 48, 51, 53
CDUMi, 47
CELAS1, 174
CELAS2, 174
CELAS3, 174
CELAS4, 174
CELASi, 48, 51, 53
CEND delimiter, 12
CFAST, 48
CFLUIDi, 49, 51, 53
CGAP, 49, 51, 53, 184, 599
CHBDYG, 51
CHBDYi, 49, 53
CHBDYP, 51
CHEXA, 49, 51, 53, 151, 247
CLOAD, 25
CMASS1, 175
CMASS2, 175
CMASS3, 175
CMASS4, 175
CMASSi, 49, 51, 53
CMETHOD, 21, 695
commands, 16
comments, 12
complex eigenvalue analysis, 20, 27
component mode synthesis, 333
Composite Rotations, 162
Composites
  MAT8, 269
composites
  data recovery, 279
  failure criteria, 269
  failure indices, 280
  Hill's theory, 281
  Hoffman's theory, 282
  lamina, 267
  MAT2, 269
  material properties, 268
  maximum strain theory, 283
  PCOMP, 269
  PSHELL, 269
  Tsai-Wu theory, 282
concentrated mass elements, 190
CONM1, 40, 47, 190
CONM2, 40, 47, 190
CONMi, 48, 51, 53
CONROD, 48, 51, 53, 55, 71
constraint elements, 47
contact, 677
convergence, 677
CORD1C, 38
CORD1R, 38
CORD1S, 38
CORD2C, 38
CORD2R, 38
CORD2S, 38
COUPMASS, 56, 67, 71, 124
CPENTA, 49, 51, 53, 151, 247
CQUAD4, 49, 51, 53, 123, 247
CQUAD8, 49, 51, 53, 123
CQUADR, 49, 51, 54, 123, 247
CQUADX, 46, 175
CRAC2D, 48, 51, 53, 177
CRAC3D, 48, 51, 53, 177
crack tip elements, 177
CROD, 49, 51, 54, 55, 71
CSEAM, 49
CSHEAR, 49, 51, 54, 121
CSLOT3, 725
CSLOT4, 725
CSLOTi, 49, 51, 54
CTETRA, 49, 51, 54, 151, 247
CTRIA3, 49, 51, 54, 123, 247
CTRIA6, 49, 51, 54, 123
CTRIAR, 49, 52, 54, 123, 247
CTRIAX, 46, 175
CTRIAX6, 46, 49, 52, 54, 175
CTUBE, 50, 52, 54, 55, 71
CVISC, 47, 50, 52, 54, 72
CWELD, 50, 52, 54, 211, 216
cyclic symmetry, 23
  dihedral symmetry, 734
  N-gons, 735
  reflective symmetry, 733, 736
  rotational symmetry, 734
  theory, 738
  three-dimensional bodies, 735

CYJOIN, 751
cylindrical coordinate system, 38, 39
CYSYM, 751

D

DAREA, 23, 304, 305, 306, 307, 317
database, 455
DATAREC, 764
DBCLEAN, 458
DBLOAD, 459
DBLOCATE, 457
DBSETDEL, 457
DBsets, 455
    DBALL, 455
    MASTER, 455
    SCRATCH, 455
    USROBJ, 455
    USRSOU, 455
DBUNLOAD, 459
DDRMM, 72, 175
DEFORM, 16, 20, 302
DEFUSET, 335
Degree of Freedom Sets
    Bulk Data entries, 328
    definitions, 326
DELAY, 304, 305, 318
delivery database, 457
DEQATN, 197, 767, 775
design optimization, 23
differential stiffness, 691
direct input matrices, 40
Direct Matrix Abstraction Program, 453
DISPLACEMENT, 26, 27, 28, 29
displacement coordinate system, 38
dissimilar meshes, 208
DLOAD, 21, 22, 23, 31, 304, 305, 317, 318
DMAP, 453
DMIG, 40, 193, 333
DPHASE, 304
dummy elements, 47
DYNSPCF, 391

E

EDE, 28, 29
EIGC, 21, 695
Eigenvalue problem, 575
EIGR, 18, 23
EIGRL, 18, 23
EKE, 27, 28, 29
element deformation, 302
element geometry checks, 245
element offsets, 38
element strain energy, 722
ELEMID, 212, 216
ELPAT, 213
ELSDCON, 436, 449
ENDDATA delimiter, 12
enforced motion, 303
EPOINT, 40
ESE, 26, 27, 28, 29, 723
Euler angles, 161
Executive Control Section, 12
Executive Control statements, 15

F

failure criteria, 285, 286, 287
    HILL, 285
    HOFFMAN, 286
    TSAI-Wu, 287
Failure Index (FI), 284
FEEDGE, 40, 194, 197, 309, 759, 763, 765, 769
FEFACE, 40, 195, 197, 309, 759, 763, 765, 769
FEFACES, 785
File Management Section (FMS), 12, 15
flexibility factors, 68
FLUTTER, 21
FLUX, 29, 30
FMETHOD, 21
follower force, 715
follower stiffness, 695
FOLLOWK, 690
FORCE, 26, 27, 28, 29, 298, 313
FORCE1, 298, 313
FORCE2, 298, 313
FORCEAX, 298
FREQ, 21, 304
FREQ1, 21, 304
FREQ2, 21, 304
FREQUENCY, 21, 31
frequency dependency, 201
frequency response, 21, 28

G

GAP, 185
gap element, 184
gaps, 677
GENEL, 47, 190, 193
general element, 190
GEOMCHECK, 246, 248
GEOMIN, 194
Gimbal angles, 161
global coordinate system, 38
global/local analysis, 757
GMBC, 197, 310
GMCORD, 40, 197, 775
GMCURV, 40, 195, 197, 310, 759, 767, 773, 785
GMLOAD, 197, 309, 766, 768, 775
GMSPC, 197, 310, 765, 785
GMSURF, 40, 195, 197, 310, 759, 767, 773
GPFORCE, 26, 27, 29, 723
GPSDCON, 436, 449
GPSTRAIN, 27
GPSTRESS, 27, 29, 436, 445, 449, 451
GRAV, 299
gravity load, 299
GRDPNT, 33
GRDSET, 40
GRID, 36, 38
grid point force, 722
grid point stress, 436, 437, 450
  exception points, 443
  geometric method, 441
  output, 445
  topological method, 438
grid points, 36, 40
GRIDF, 725
GRIDID, 212, 216
GRIDS, 725
Guyan reduction, 333

H

HARMONICS, 751
HDOT, 30
heat transfer, 677
HEXA, 246
HOUTPUT, 752
hyperelastic elements, 197
hyperelastic materials, 124
hyperelasticity
  curve-fitting algorithm, 618
  nonlinear stress output, 646

I

IC, 22, 23
INCLUDE statement, 12
inertia relief, 332, 376, 398, 403
INIT, 456
INIT MASTER, 457
initial conditions, 317
INREL, 388
interface elements, 206
  dissimilar meshes, 208

K

K2GG, 16, 19, 23, 25
K2PP, 20, 31
K6ROT, 133

L

Lagrange Elimination Method, 160
Lagrange elimination method, 165
Lagrange rigid element, 159
Lanczos method, 581
Large mass method, 582
Left Rotation, 162
LGDISP, 649, 690
line elements, 55
Linear Elimination Method, 163
linear steady state heat transfer, 22, 29
LOAD, 16, 20, 22, 24
LOADCYH, 308, 751
LOADCYN, 308, 751
LOADCYT, 751
load-displacement dependency, 201
LOADSET, 21, 22, 25, 304, 305, 306, 314, 317
local coordinate systems, 36
LOOPID, 22, 23, 25
LSEQ, 21, 22, 25, 304, 306, 308, 314, 317

M

M2GG, 16, 19, 23, 25, 193

M2PP, 20, 31
MacNeal/Harder test, 137
mass properties, 40
MAT1, 46, 59, 66
MAT2, 46
MAT3, 46
MAT8, 46, 268
MAT9, 46
material properties, 46
MATGPR module, 335
MATHP, 47, 197
MATS1, 47, 55
MATT1, 46
MATT2, 46
MATT3, 46
MATT9, 46
maximum stress (strain), 288
Mechanism, 579
mechanism, 372
METHOD, 18, 23, 25, 695
MODES, 19
MOMAX, 298
MOMENT, 298, 313
MOMENT1, 298, 313
MOMENT2, 298, 313
MPC, 18, 31, 157, 330
MPCADD, 330
MPCAX, 330
MPCF, 27, 28, 29
MSC.Marc, 677
MSGLIMIT, 246
MSGTYPE, 246
multipoint constraint, 329

N

nastran command, 11
NASTRAN statement, 12, 14
NDDL
    Nastran Data Definition Language, 478
Newton's method, 694
Newton-Raphson method, 161
NLLOAD, 30, 317
NLPARM, 24, 675, 677, 678
NLPCI, 680
NLRGAP, 318, 715
NLSTRESS, 27
NOCOMPS, 269
NOLIN, 713, 715
NOLIN1, 306, 716
NOLIN2, 306, 716
NOLIN3, 306, 716
NOLIN4, 306, 716
NOLINEAR, 317
NOLINi, 205, 318
NONLINEAR, 22, 23, 29
nonlinear analysis, 312
    3-D slideline contact, 664
    elastic-plastic material, 622
    element library, 599
    elements, 627
    gap element, 660
    geometric nonlinearity, 599, 627
    hyperelastic elements, 644
    hyperelastic material, 599, 615
    material nonlinearity, 599, 613
    modeling, 611
    nonlinear analysis with superelements, 719
    nonlinear elastic material, 614
    nonlinear transient response analysis, 701
    parameters, 608
    plot codes, 611
    recommendations, 613
    restarts, 708
    small strain elements, 640
    SOL 106, 602
    SOL 129, 602
    SOL 153, 602
    SOL 159, 602
    static analysis, 673
    subcase structure, 604
    temperature dependent material, 627
    viscoelastic material, 626
nonlinear static analysis, 24
nonlinear steady state heat transfer, 22
nonlinear transient heat transfer, 22, 30
nonlinear transient response analysis, 25
nonstructural mass, 250
normal modes analysis, 18, 27
NOUTPUT, 752

O

Offsets, 117

OFREQUENCY, 21, 28, 392
OLOAD, 26, 27, 28, 29, 30
OMITAX, 333
OMITi, 333
OTIME, 392
OUTPUT, 197, 761, 765, 781
OUTPUT(PLOT), 27, 28, 29
OUTPUT(POST), 436, 445, 449, 451
OUTPUT(XYPLOT), 28, 29
OUTRCV, 197, 761, 764

P

P2G, 16, 23, 25
PARAM, 285
  BUCKLE, 694
  FKSYMFAC, 690, 695
  LANGLE, 167
  LMFACT, 167
  NDAMP, 715
  NLTOL, 677
  NOELOF, 722
  NOELOP, 722
  ORIGID, 167
  PENFN, 167
  QRMETH, 139
  SHLDAMP, 151
  SRCOMPS,YES/NO, 285
  XFLAG, 721
PARAML module, 335
PARAMs
  BAILOUT, 580
  MAXRATIO, 578
PARTPAT, 213
PBAR, 66, 72
PBARL, 72
PBCOMP, 55, 58, 59
PBEAM, 55, 58, 59, 61, 72
PBEAML, 72
PBEND, 67
PBUSH1D, 205
PBUSHT, 201
PCOMP, 268
PCOMPG, 268
PDAMP, 175
PDUMi, 47
PELAS, 174
p-element analysis, 40
p-elements, 193, 210
  adaptivity, 760
  boundary conditions, 760
  combining p- and h-elements, 757
  data recovery, 761
  elements, 759
  error estimator, 756
  example problems, 761
  geometry, 756, 759
  loads, 760
  materials, 760
  solution sequences, 758
PENTA, 246
PGAP, 185
PLOAD, 313
PLOAD1, 55, 298
PLOAD2, 299, 313
PLOAD4, 299, 313
PLOADX1, 175, 299
PLOTSUP, 461
PLPLANE, 197
PLSOLID, 197
POINT, 759, 765
POINTAX, 298
POST, 472
PostScript plot, 470
PRAC2D, 178
PRESAX, 299
prestressed normal mode, 689, 690
PROD, 71
PROJECT, 458
PSET, 197
PSHEAR, 121
PSHELL, 124, 128
PSOLID, 47, 152
PTUBE, 71
punch, 480
PVAL, 197, 760, 764, 791
PVISC, 72

Q

QBDYi, 22, 23
QHBDY, 22, 23

QUAD4, 133, 136, 246
QUAD8, 136
QUADR, 136, 246
quaternion, 161
QVECT, 22, 23
QVOL, 22, 23

R

RANDOM, 21, 480
random response, 21, 28, 305
RANDPS, 21, 305
RANDT1, 305
RANDTi, 21
RBAR, 47, 128, 156, 330
RBAR1, 167, 168
RBE1, 47, 156, 330
RBE2, 47, 156, 330
RBE3, 47, 157, 330
rectangular coordinate system, 38, 39
REPCASE, 16, 18, 19, 22, 31
RESTART, 458
restarts, 685
RFORCE, 299
Right Rotation, 162
RIGID, 166, 167
rigid elements, 159
RJOINT, 167, 168
RLOAD1, 303, 304, 306
RLOAD2, 303, 304, 306
rod element, 71
Rodrigues parameters, 161
RROD, 47, 156, 330
RSPLINE, 47, 157, 330
RSSCON, 157, 330
RTRPL1, 167
RTRPLT, 47, 156, 330
RTRPLT1, 168

S

SACCELERATION, 28, 29
scalar elements, 174
SDAMPING, 20, 31
SDISP, 30
SDISPLACEMENT, 28, 29, 30
SEALL, 26
SEPLOT, 461
SETS DEFINITION, 764
shear panel, 121
shear relief, 60
shell elements, 123
single point constraint, 331
Slideline contact
    guidelines, 187
slideline contact
    input, 189
    master line, 186
    penetration, 187
    slave line, 186
    slideline contact region, 186
SLOAD, 298, 313
SLOOPID, 23, 25
SOL 106, 674
SOL 129, 701
SOL 400, 169
SOL DBTRANS, 459
solution sequence, 337
    rigid format, 337
SORT1, 26
SORT2, 26
SPC, 18, 31, 303, 331
SPC1, 304, 331
SPCADD, 331
SPCAX, 303, 331
SPCD, 303
SPCF, 26, 27, 28, 29
SPCFORCE, 29, 30
spherical coordinate system, 38, 39
split database operations, 458
SPOINT, 40
static analysis, 16, 26
STATSUB, 20
steady state heat transfer, 30
STIME, 23, 25
STRAIN, 27, 28
strain energy, 720
strength ratio, 296
STRESS, 26, 27, 28, 29, 436
stress discontinuities, 436, 451
stress intensity factor, 178
STRFIELD, 451
SUBCASE, 17, 18, 19

subcase combinations, 308
subcases, 16, 18, 20
SUBCASID, 22
SUBCOM, 17, 31, 300, 301, 308, 737
SUBID, 22, 25
subroutines
BSBRGD, 85
BSBRID, 85
BSBRPD, 85
BSBRT, 85
BSCON, 84
BSGRQ, 85
BSMSG, 85
SUBSEQ, 31
SUPAX, 332
SUPER, 26, 32
superelement, 25, 33
superelement analysis, 419
SUPORT, 332
SUPORT1, 18, 31, 332
SURFACE, 436, 445, 449, 451
SVECTOR, 27, 28
SVELO, 30
SVELOCITY, 28, 29, 30
SYM, 18, 301
SYMCOM, 18, 300, 301, 308, 737
system cell
210, 478

T

TABDMP1, 20
TABLE3D, 197
TABLEDi, 304, 305, 318
TABLEM1, 46
TABLEM2, 46
TABLEM3, 46
TABLEM4, 46
TABLES1, 47
TABLEST, 47
TABRNDi, 305
tapered beam, 62
TEMP, 22, 23, 29, 30, 40, 300, 314
TEMP(LOAD), 301
TEMPAX, 40, 300
TEMPD, 22, 23, 40, 300, 314
TEMPERATURE, 16, 20
TEMPERATURE(INIT), 25
TEMPERATURE(INITIAL), 22
TEMPERATURE(LOAD), 24
temperatures, 299
TEMPF, 197
TEMPP1, 299, 314
TEMPP2, 299
TEMPP3, 299, 314
TEMPRB, 299, 314
TETRA, 246
TF, 333
TFL, 31
TFLAG, 148
THERMAL, 29, 30
thermal field, 40, 300
thermal load, 300
TIC, 22
TLOAD1, 303, 304, 305, 306, 317, 715
TLOAD2, 303, 304, 305, 306, 317
TLOADi, 21, 23
transient response, 21, 29
TRIA3, 246
TRIAR, 246
TRIM, 16
TSTEP, 22, 31
TSTEPNL, 23, 25, 206, 704

U

UMERGE module, 335
UMERGE1 module, 335
UPARTN module, 335
USET, 335
USET1, 335

V

VEC module, 335
VECTOR, 27, 28
VELO, 30
VELOCITY, 28, 29, 30
VOLUME, 436, 445, 449, 451
VUGRID, 764

W

warping, 60, 61