Advanced CAE Applications for Professionals

Software that works — for you.[SM]

UAI/*NASTRAN User's Guide*

for Version 20.1

UNIVERSAL ANALYTICS, INC.

Publication ND-002

First Edition, February 1991
Second Edition, December 1992
Third Edition, April 1994
Fourth Edition, September 1995
Fifth Edition, December 1997
Sixth Edition, March 1999

UNIVERSAL ANALYTICS, INC.

3625 Del Amo Blvd., Suite 370
Torrance, CA 90503
Tel: (310) 214-2922
FAX: (310) 214-3420

FOREWORD

The **UAI/*NASTRAN*** User's Guide provides you with detailed information on the modeling and analytical disciplines of the system. It includes descriptions of finite elements, hints to improve your modeling practices, examples of Bulk Data use and many sample problems and their solutions. It is strongly recommended that you review this guide thoroughly before using a **UAI/*NASTRAN*** capability which is new to you.

This manual includes thirty-one Chapters which are divided into five parts:

INTRODUCTION

1. **FINITE ELEMENT MODELING**
2. **ANALYSIS DISCIPLINES**
3. **SPECIAL TOPICS**
4. **GRAPHICS**

The Introduction provides you with an overview of **UAI/*NASTRAN*** and an orientation to the system from both the software and engineering perspectives. Part I then describes the process of creating finite element models with **UAI/*NASTRAN***. It describes not only the finite element library, but provides general information regarding boundary conditions, loads, and material properties. Part II of the manual provides you with detailed descriptions of the analytical disciplines available in the system. Special analysis techniques are discussed in Part III, and, finally, Part IV illustrates the use of the **UAI/*NASTRAN*** graphics capabilities and interfaces with other software products.

The companion to this volume is the **UAI/*NASTRAN*** User's Reference Manual. The Reference Manual provides you with detailed information on the modeling describing all of the input data used by the system.

This page is intentionally blank.

VERSION 20.1 RELEASE NOTES

In Version 20.1 of **UAI/*NASTRAN***, there have been a number modifications to the ***User's Guide*** to describe new features. These are summarized, by Chapter, in this section. In addition, there have been many corrections and clarifications throughout the manual. Many of the data changes required for new and enhanced analytic capabilities are found in the ***User's Reference Manual***.

ONLINE DOCUMENTATION

As part of UAI's ongoing modernization program, all **UAI/*NASTRAN*** manuals, as well as those for other UAI software products, continue to be delivered in Adobe ***Portable Document File*** (PDF) format. This allows them to be used online with any computer having the Adobe Acrobat Reader (Version 3.0 or later). This reader is also delivered with our software.

To insure prompt updates to all documentation, any changes and enhancements may be downloaded from UAI's Web site at **`www.uai.com`**.

DOCUMENTATION TRACKING

Beginning with Version 20.1, the documentation has been slightly modified so that each page of specific commands and data entries has a revision date on it. The two forms are: New: V20.1, indicating that the page is new with the current release; and Rev: V20.1, which indicates that the page has been modified for the current release. In this manual, only the first page of new or revised Chapters is noted.

Chapter 2. Orientation: The Software System

- ❒ Description of the additional FEMAP interface and changes to the SDRC IDEA-S interface.

Chapter 5. The Finite Element Library

- Description of a new capability to compute equivalent Beam forces for sets of solid elements.
- Correction of solid element stress equations.
- Changes to the **NLSTRAIN** and **NLSTRESS** commands to add features missing from earlier documentation.

Chapter 6. Boundary Conditions and Equation Reduction

- Description of new capability to perform automatic reduction using the **AUTOREDUCE** command. Both Guyan reduction and Modal Reduction (including Craig-Bampton) may be performed using this feature. The reduced models may then be output, by using the new **EXPORT** command, on **DMIG** Bulk Data entries.

Chapter 9. Dynamics Modeling

- Modified to include information on the automatic static and Modal reduction features (**AUTOREDUCE**).

Chapter 10. Linear Static Analysis

- Expanded to include description of matrix conditioning checks for inertial relief problems.

Chapter 16. Material Nonlinear Analysis

- Reflects the name change of the **AUTOREDUCE** command to **NLREDUCE**, and provides updated results for sample problems.

Chapter 17. Geometric Nonlinear Analysis

- Typographic corrections

Chapter 19. Substructuring Analysis

- Additional description of **APP DMAP** with substructuring.

Chapter 25. Multidisciplinary Design Optimization

- Description of a new mode tracking feature used in the design of models with frequency constraints, and the addition of modal damping as a design variable.

TABLE OF CONTENTS

This page is intentionally blank.

LIST OF FIGURES

This page is intentionally blank.

LIST OF TABLES

This page is intentionally blank.

1

Chapter 1

OVERVIEW

The **UAI/*NASTRAN*** User's Guide has been designed to provide you with a comprehensive overview of the program, its software architecture, and its engineering analysis capabilities. The text has been organized in four broad parts:

- ❒ Finite Element Modeling
- ❒ Analysis Disciplines
- ❒ Special Topics
- ❒ Graphics

The first part, ***Finite Element Modeling***, includes chapters which: review the fundamentals of the ***Finite Element Idealization*** process; discuss the ***Finite Element Library*** including its limitations; define the manner in which ***Boundary Conditions*** are applied and ***Reductions*** are performed; summarize the types of environmental ***Loads*** which may be used; provide descriptions of the ***Material Properties***; and introduce the complex topic of ***Dynamics Modeling***.

The second part, ***Analysis Disciplines***, describes the most frequently used **UAI/*NASTRAN*** analysis disciplines and capabilities. These include:

- ❒ Linear Static Analysis
- ❒ Normal Modes Analysis
- ❒ Complex Eigenvalue Analysis
- ❒ Frequency and Random Response Analysis
- ❒ Transient Response Analysis
- ❒ Response Spectra Analysis
- ❒ Geometric Nonlinear Analysis
- ❒ Buckling Analysis
- ❒ Material Nonlinear Analysis

REV: V20.1

The third part, ***Special Topics***, describes other more advanced capabilities. These include:

- ❒ Substructuring Analysis
- ❒ Axisymmetric Harmonic Modeling
- ❒ Axisymmetric Ring Modeling
- ❒ Cyclic Symmetry Modeling
- ❒ Acoustic and Hydroelastic Analysis
- ❒ Heat Transfer Analysis
- ❒ Multidisciplinary Design Optimization
- ❒ Design Sensitivity Analysis
- ❒ Mesh Error Estimates
- ❒ Aeroelastic Analysis

The fourth and final part, ***Graphics***, describes the manner in which you may interface **UAI/*NASTRAN*** to other pre- and post-processing software systems, and describes in detail the **UAI/*NASTRAN*** plotting capability. This latter feature allows you to generate both structural plots and X-Y response plots.

It is hoped that the information found in this manual will assist you in performing more timely, accurate, and cost-effective engineering analyses with **UAI/*NASTRAN***.

The **UAI/*NASTRAN*** system is developed and maintained by Universal Analytics, Inc. (UAI). It is constantly being improved and expanded to new disciplines. New releases of the program are made regularly to meet the needs of the user community.

1.1 THE DOCUMENTATION SUITE

There are four technical manuals in the **UAI/*NASTRAN*** documentation suite:

- ❒ User's Reference Manual
- ❒ User's Guide
- ❒ Installation Guide and System Support Manual
- ❒ ***eBase*** ARCHIVE Database Schemata

The ***User's Reference Manual*** is the primary document. It describes the input data requirements of **UAI/*NASTRAN*** in full detail. The ***User's Guide*** provides an overview of the capabilities of the system and gives examples of and guidelines for its use. Italso provides examples of analyses and verification of the system by solving problems with known solutions. The last document is the Installation Guide and System Support Manual. This is used by your computer systems staff to install **UAI/*NASTRAN*** at your computing facility and to tune it for optimal performance.

In addition to these documents, there are four other manuals which describe UAI's advanced scientific database product, ***eBase***™. These manuals include:

- ❒ ***eShell***™ User's Manual
- ❒ ***eBase*:*applib***™ and ***eBase*:*matlib***™ Programming Manual (One Volume)

The ***eShell*** User's Manual describes how the ***eBase*** interactive shell is used to access **UAI/*NASTRAN*** data. The ***eBase*:*applib*** and ***eBase*:*matlib*** Programming Manual provides a detailed description of the Fortran Application Programming Interface (API) to the ***eBase*** database. The API allows you to create programs which can communicate effectively with **UAI/*NASTRAN*** ARCHIVE databases. The final manual, the ARCHIVE Database Schemata, provides a detailed description of the format, or schema, of all of the **UAI/*NASTRAN*** data structures which contain input data and solution results. These schemata allow the effective use of ***eShell***, ***applib***, and ***matlib***.

1

1.2 FORMAT OF THIS MANUAL

The previous sections of this Chapter have described the basic organization of this manual. The table below illustrates typographic styles used in the text to help clarify the meaning of the various input data items.

TYPE STYLE	DESCRIPTION
Differential Stiffness ***vonMises Stress***	Bold italics indicate the definition of an important term.
GRID eBase	San serif letters indicate a **UAI/*NASTRAN*** term that has come into common usage among NASTRAN users. Such terms may or may not be the same as those used by other FEA programs.
CQUAD4 **PSHELL**	Bold computer typing signifies the name of a **UAI/*NASTRAN*** command or data entry. For Bulk Data entries, it may also reference the name of a field of an entry.

Another convention is used to assist you in learning **UAI/*NASTRAN*** methodology. This is a callout which provides important information.

This is important information!!

1.3 REPORTING PROBLEMS

There are two special forms available for you to report problems that you may find while using **UAI/*NASTRAN*** or reading its documentation. The first is called the ***Software Problem Report***, SPR, form, and the second is called the ***Documentation Error Report***, DER. A blank form of each type is included in this Chapter. You may use these as masters, copying them freely as needed.

The SPR is used to report any erroneous results that you may obtain during your analysis work. Please note that it is crucial for you to enclose sufficient materials to allow UAI to recreate the problem during the maintenance cycle. Ideally, it is best if you send to UAI the input data stream and a copy of the output file produced by the job. You may place this data on a magnetic tape generated on your host computer, or you may place them on an IBM-PC compatible 3 1/2 inch floppy disk, a JAZ disk, a ZIP disk, or a CD. Workstation users may use an 8 mm cartridge tape, a 1/4 in cartridge tape, a 4 mm DDS, or a CD.

If your site has a direct telecommunications link with UAI, then you may contact Customer Support for instructions on how to FTP files to us. Finally, you may also use eMail by sending all required information to:

```
support@uai.com
```

You may use the DER form to report any errors, omissions or criticisms of this, or any, **UAI/*NASTRAN*** documentation. Generally, you include copies of the document with errors annotated. Other suggestions may be written on separate pages indicating where the new information or changes belong in a specific manual. Through the years, UAI has received many valuable comments from users which have been integrated into the documentation suite.

1

This page is intentionally blank.

UAI/*NASTRAN* SOFTWARE PROBLEM REPORT

NAME:		DATE:	
COMPANY:		PHONE:	
ADDRESS:		PROGRAM VERSION:	
		HOST COMPUTER:	
		OPERATING SYSTEM:	
eMail:			

ENCLOSED MATERIALS:

DESCRIPTION OF ERROR

Please attach additional sheets as necessary

1

This page is intentionally blank.

UAI/*NASTRAN* DOCUMENTATION ERROR REPORT

NAME: _____ DATE: _____

COMPANY: _____ PHONE: _____

ADDRESS: _____ eMail: _____

_____

_____

ERROR IN:

USER'S GUIDE		***eShell***		
USER'S REFERENCE		***eBase:applib***		
INSTALLATION GUIDE		***eBase:matlib***		
		ARCHIVE SCHEMATA		

DESCRIPTION OF ERROR OR SUGGESTION

Please attach additional sheets as necessary

1

This page is intentionally blank.

Chapter 2

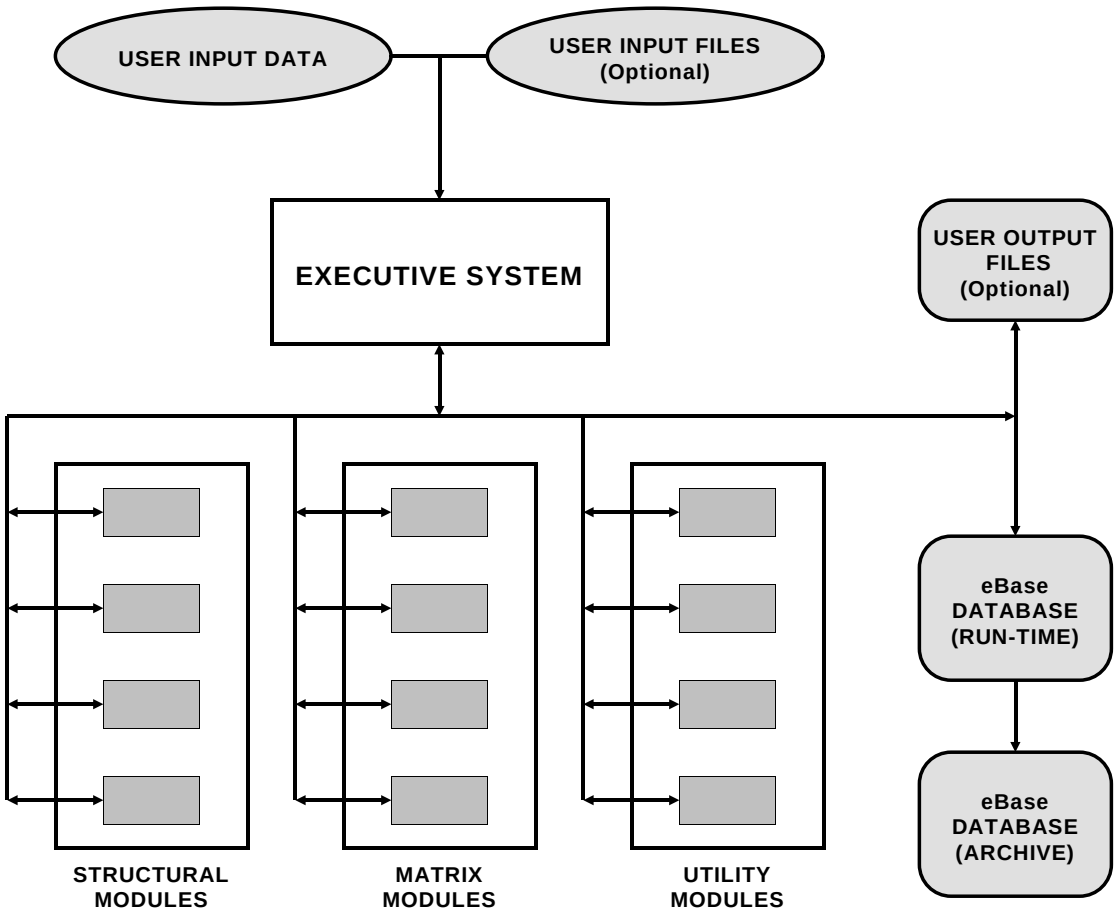
ORIENTATION: THE SOFTWARE SYSTEM

This chapter provides you with an orientation to the software architecture and data flow of **UAI/*NASTRAN***. It describes the organization of the system, the input data stream, how the program is controlled, the dynamic memory model, the database management system, and how the program communicates with other external applications software. This information will allow you to understand the basic principles of the system which are useful in applying it to your analysis task.

2.1 THE SYSTEM ARCHITECTURE

UAI/***NASTRAN*** is a fully modular software system with a centralized ***Executive System*** and a powerful ***Database Management System***, called ***eBase***. Figure 2.1 illustrates this architecture and the flow of information within it. Your input data, and optional input files, are read and processed by the executive system. These data determine the type of analysis that you are performing and select a predefined ***Direct Matrix Abstraction Program***, or DMAP, that controls the flow of **UAI/*NASTRAN*** during execution. The actual work is performed by ***Modules*** within the program. As shown in the figure, modules are categorized for convenience according to the type of function that they perform. A module is a software component which performs a single well defined task such as generating finite element matrices or solving a system of simultaneous linear equations. During execution, the modules communicate through the ***eBase*** database. Additional output files, used to communicate with other software products, may also be created during the job. Each of these topics is discussed in more detail in following sections of this Chapter.

Figure 2-1. UAI/NASTRAN ARCHITECTURE



2.2 THE UAI/NASTRAN INPUT DATA STREAM

The input data stream for **UAI/*NASTRAN*** consists of four data packets, one of which is optional. These packets, described below, must be assembled in the following order:

- ❒ Executive Control Command Packet
- ❒ Substructure Control Command Packet (Optional)
- ❒ Case Control Command Packet
- ❒ Bulk Data packet

2.2.1 The Executive Control Command Packet

The Executive Control Command Packet contains commands which identify the job and the type of solution to be performed. It also defines the general conditions under which the job is to be executed such as: maximum time allowed; files to be used; and checkpoint or restart conditions. If you use Direct Matrix Abstraction, the complete DMAP sequence must appear in the Executive Command Packet. These DMAP instructions allow you to specify your own sequence of matrix operations independent of the standard sequences provided by the Rigid Formats. All of the Executive Control commands are described in Chapter 2 of the ***User's Reference Manual***.

2.2.2 The Substructure Control Command Packet

The Substructure Control Command Packet is an optional part of the **UAI/*NASTRAN*** input data stream. It is used only when you are using the substructuring feature of the program. Substructuring is described in detail in Chapter 19 of this manual and in Chapter 3 of the ***User's Reference Manual***.

2.2.3 The Case Control Command Packet

You use the Case Control Command packet to provide titling information, to select sets of data from the Bulk Data packet, to generate the subcase structure, to make output requests, and to create structural or X-Y plots. Specific Case Control Commands are often determined by the type of analysis being performed. Examples of Case Control are given throughout the remaining Chapters of this manual. All of the available commands are described in Chapter 4 of the ***User's Reference Manual***.

2.2.4 The Bulk Data Packet

The great majority of data required for a structural analysis are contained in the Bulk Data Packet. Bulk Data are used to define the structural model and the various other engineering data needed to specify the analysis environment. Also, Bulk Data may be used to specify the parameters and limits required to control the analysis.

In general, all data submitted describing the physical properties of the model are used directly by **UAI/*NASTRAN***. Those data used to specify conditions of the environment such as constraints, temperatures, and loads must be selected with Case Control commands if they are to be applied to the model. Methods of analysis, including the associated data and limits, must also be selected from Bulk Data by Case Control commands. Because these last two categories of information are fully under your control, the Bulk Data Deck may include several sets of

these data at once. The subcase structure of Case Control provides you with the facility to define unique combinations of these data as separate subcases to be analyzed, since more than one subcase may be solved in a single run with **UAI/*NASTRAN***.

For large problems, the Bulk Data Packet may consist of many thousands of entries. The Bulk Data Packet may contain entries in any order because they are sorted prior to the execution of an analysis. You may obtain a printed copy of either or both the unsorted and sorted bulk data by using a Case Control Command. A sorted listing is necessary in order for you to modify data for a restart of a previous execution. This listing is automatically provided as an echo of the Bulk Data unless you specifically disable it. All of the Bulk Data entries available are described in Chapter 7 of the ***User's Reference Manual***. Additionally, many examples of their use is provided throughout this manual.

2.3 CONTROL MECHANISMS FOR UAI/*NASTRAN*

The Direct Matrix Abstraction Program (DMAP) is the original *programming language* of **UAI/*NASTRAN***. It controls the execution logic and sequence of most program operations. Each DMAP instruction causes a **UAI/*NASTRAN*** module to be executed, and each Rigid Format is simply a sequence of DMAP instructions permanently stored in the program. You may code and execute a DMAP sequence instead of using a Rigid Format. You may also make changes to the Rigid Formats through the use of ALTERs. In either case, these instructions are always included with the Executive Control Commands.

Many operations may be performed with DMAP. Some perform executive functions, such as defining labels, controlling loops, and checkpointing files. Others perform matrix arithmetic operations. A third group performs utility functions such as printing matrices and tables, or modifying parameter values. The final group is the structural modules.

The first three categories are intended for general user applications. However, the structural modules require specialized input tables and are intended for use in the Rigid Formats. The format and detailed description of the user-oriented DMAP instructions are found in Chapter 9 of the ***User's Reference Manual***.

UAI introduced the Advanced Programmable Executive (APEX) with Version 11.7 of the software. This new concept in program control allows standard Fortran code to be used for writing high-level analysis procedures. APEX has been used to implement the Multidisciplinary Analysis and Design Optimization solution sequence.

Future releases of **UAI/*NASTRAN*** will allow you to modify and enhance the APEX programs delivered with the system — you will be able to incorporate your own Fortran programs into your analysis procedures. Also, this facility will allow you to read from and write to the ***eBase*** run-time database. As a result, you will have complete access to all input data, output results, and intermediate computational data from within a **UAI/*NASTRAN*** execution.

2.4 DYNAMIC MEMORY MANAGEMENT

The architecture of **UAI/*NASTRAN*** allows the modeling and analysis of finite element models of virtually unlimited size. Most numerical calculations perform at maximum efficiency when all data for the operation fit in the ***working memory*** space of the program. Many operations may be performed even when all data that they require do not fit in memory by using what is called ***spill logic***. Spill logic simply involves the paging of data to and from disk storage devices as necessary. For very large jobs, spill commonly occurs. In such cases, providing **UAI/*NASTRAN*** with additional working memory can often improve performance. On the other hand, you do not want to give **UAI/*NASTRAN*** excess memory, because it will reduce resources that could be used for other processes on your system. Under certain circumstances, excess memory may actually degrade the performance of **UAI/*NASTRAN*** and, in extreme cases, even your computer system.

The working memory for **UAI/*NASTRAN*** is dynamically acquired during execution. The amount of space that is actually used by the program is typically controlled by the **UAI/*NASTRAN*** execution procedure or the **MEMORY** Executive Control command. Some host computers have alternate means of controlling this memory. You should refer to Chapter 2 of the **UAI/*NASTRAN User's Reference Manual*** and contact your Support Specialist for details relative to your specific host computer.

UAI/***NASTRAN*** has a second independent dynamic memory which is used to operate on databases that are attached to the execution. This memory is typically much smaller than the working memory. The main factor influencing the amount of database memory required is the block size used by the active databases. This is an advanced topic that also requires that you contact your Support Specialist for details relative to your specific host computer.

2.5 THE *eBase* DATABASE

The Engineering Database Management System, ***eBase***, is an integral part of **UAI/*NASTRAN***. This advanced scientific database technology greatly enhances the data handling capabilities of **UAI/*NASTRAN*** while removing many of the inconveniences of the older I/O system which used sequential files.

2.5.1 Multischematic Database

eBase is a powerful database management system especially designed for the *number crunching* required by scientific applications. It is called a ***multischematic*** database because it supports a number of data structures in a uniform manner. The following sections describe these data structures, each of which forms an ***entity class***. These classes include:

- ❒ Relational Entities
- ❒ Matrix Entities
- ❒ Freeform Entities
- ❒ Stream Entities

2.5.1.1 Relational Entities

Relational entities are basically tables of data. The columns of these tables are called ***attributes***, and the rows are called ***entries***. Each of the attributes has certain characteristics such as its numeric type, whether its data must exist, and so on. The set of attributes and characteristics of a relation are called its ***Schema***. A data value at a given entry and attribute position is called a ***Field*** in the relation.

Relations have been introduced into **UAI/*NASTRAN*** for several reasons. The first is that they result in well-defined data structures that are easier to access. The second is that they provide better tools for developing advanced capabilities such as Multidisciplinary Design Optimization.

In Version 11.7, relations have been defined to store input data and analysis results on an ARCHIVE database, described in Section 2.5.4. The schema of these relations has followed the international PDES/STEP standard wherever possible.

2.5.1.2 Matrix Entities

One of the most important data structures encountered in scientific software applications is the matrix. Matrix algebra forms the basis for algorithms used in **UAI/*NASTRAN***. ***eBase*** stores matrix data in a special compressed format which is called ***packing***. Packing is performed in ***column-major form***. Using this method, matrices are created and retrieved by their columns.

Only columns which have at least one non-null row are stored in the entity. The use of packing provides three important advantages. The first is that significant amounts of disk space may be saved. For very large finite element analyses, this saving can often reach 98% and more. The second is that the performance of computational algorithms can often be enhanced because unnecessary computations can be avoided. Finally, algorithms may be written to allow virtually unlimited problem size by using ***spill logic***. The use of spill logic allows computations to proceed on only portions of a matrix and then uses the ***eBase*** database to accumulate results.

Finally, ***eBase*** matrices are defined by their numeric types, which may be ***Real Single Precision***, ***Real Double Precision***, ***Complex Single Precision***, or ***Complex Double Precision***. Their general topology is also specified. Forms supported are ***General Rectangular***, ***Symmetric***, ***Square***, ***Diagonal*** and ***Identity***. Two additional types, which are used for computational efficiency, are ***Upper*** and ***Lower Triangular Factors***.

2.5.1.3 Freeform Entities

A Freeform Entity is a heterogeneous collection of data for which no schema is defined. It may be thought of as a collection of binary records containing any data types. These entities are used to store data that are used on an all-or-nothing basis.

2.5.1.4 Stream Entities

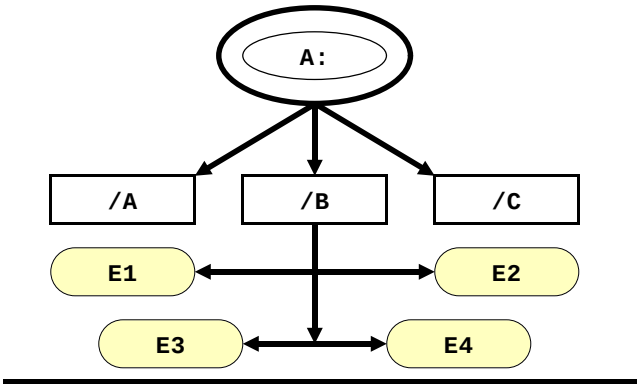
A stream entity is a continuous stream of data values, each of which has a position in the entity. Each of the data values may be directly and randomly addressed. This entity class has been specially developed to support the high-performance sparse matrix algorithms within **UAI/*NASTRAN***.

2.5.1.5 Directory Structure and Subscripted Entities

All of the ***eBase*** entities may be organized into a hierarchy of ***Directories***, as shown in Figure 2-2. Directories allow you to group related data into the same storage area. Several commands allow you to direct data to specific directories, most notable of these is the **ARCHIVE** Case Control command. Additionally, there may be a number of versions of a specific entity. This is accomplished by using ***Subscripted Entities***.

For detailed information describing the directory structure and entity classes, please see the ***eBase*** manual suite.

Figure 2-2. eBase DIRECTORY HIERARCHY



2.5.2 The Two Types of Databases

There are two types of ***eBase*** databases. The first type is the ***run-time database***, or RUNDB. This database is used to store the freeform entities and matrices which are used in performing your analysis task. At the end of your job, the RUNDB is usually deleted. The second type is the ***archival database***. This type of database is saved from one execution to the next. The three archival databases available are the SOF database, used in performing Substructuring Analyses, the NLDB database, used when you perform Nonlinear Material Analyses, and the ARCHIVE database used to save input data and solution results for use with ***eShell***™, ***eBase***:***applib***™, and ***eBase***:***matlib***™.

2.5.3 The Logical and Physical Views of the Database

To fully understand the database technology, you must understand the two views of the database. Each database, whether a RUNDB, an ARCHIVE, an SOF, or an NLDB, is called a ***logical database.*** This term is used because from an engineering viewpoint, the database is a single entity which is used in its entirety. The manner in which the logical database is stored on your host computer depends on the amount of data it contains and the availability of disk storage devices. The physical view is a mapping of a logical database to some number of physical files on your host computer. These two views are discussed in the following sections.

2.5.3.1 The Logical Model

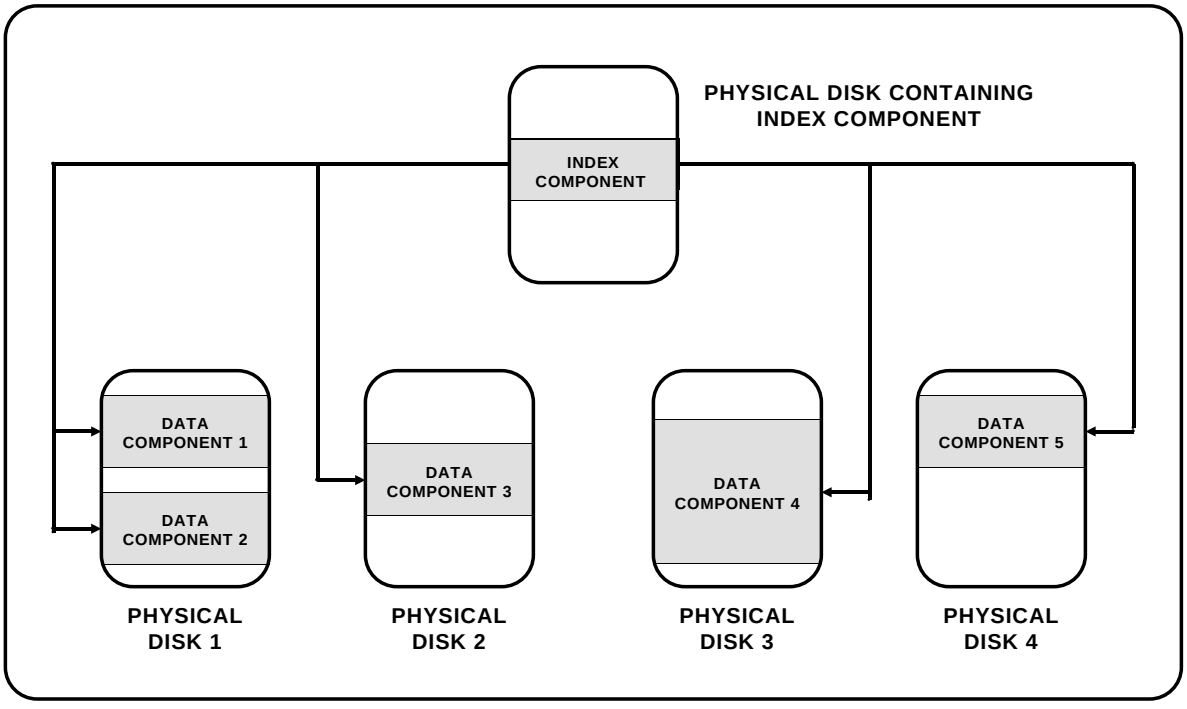
In nearly every case, you may think of your ***eBase*** database as a single object which contains a collection of relational, matrix, freeform and stream entities. When performing an analysis or using DMAP ALTERs to write files that will be exported to other applications programs, you simply refer to the various entities by the name which has been assigned to them in the DMAP program. When you reference an entity from **UAI/*NASTRAN***, the program finds the physical data and performs the required operation. There is absolutely no need for you to know where the physical data may be located.

2.5.3.2 The Physical Model

There are times, however, when it may be necessary for you to understand the physical model because, for very large analyses, it may be more efficient to organize the actual files in a manner that allows higher performance on your host. Each ***eBase*** database, regardless of its use, has two components which are stored in a minimum of two physical files. The first of these components is called the INDEX component. This component is always a single physical file. It contains information which identifies and locates actual database entities. These entities themselves are stored in the DATA component. You define databases with the Executive Control command **ASSIGN**. Part of this command requires that you specify a name prefix for your database. **UAI/*NASTRAN*** uses the prefix to create the actual file names where the data is stored. You must contact your Support Specialist for information describing the naming convention on your host computer.

To provide the maximum flexibility for a wide variety of data storage requirements, the data components may be stored in a number of different physical files, as illustrated in Figure 2-3. Most database systems are organized in this manner, because the index component is generally small in size and referenced often, while the data component may be extremely large and not fit in a single file or even on a single disk drive. It is again necessary that you contact your Support Specialist to obtain the information that you need to perform the distribution of your database across multiple disk drives on your host computer.

Figure 2-3. THE PHYSICAL DATABASE



2.5.4 The Archive Database

The ARCHIVE database is part of the ***eBase*** software suite. This database is intended to be used with the ***eShell***™ interactive query program and as an interface to other software products. The ARCHIVE database contains all of the **UAI/*NASTRAN*** input data and solution results in ***rationalized*** form. This means that the data are stored as relational entities which are easy to access. The Schemata, or format, of these relations is given in the ***eBase ARCHIVE Database Schemata Description Manual***.

2.5.5 The Application Programming Interface

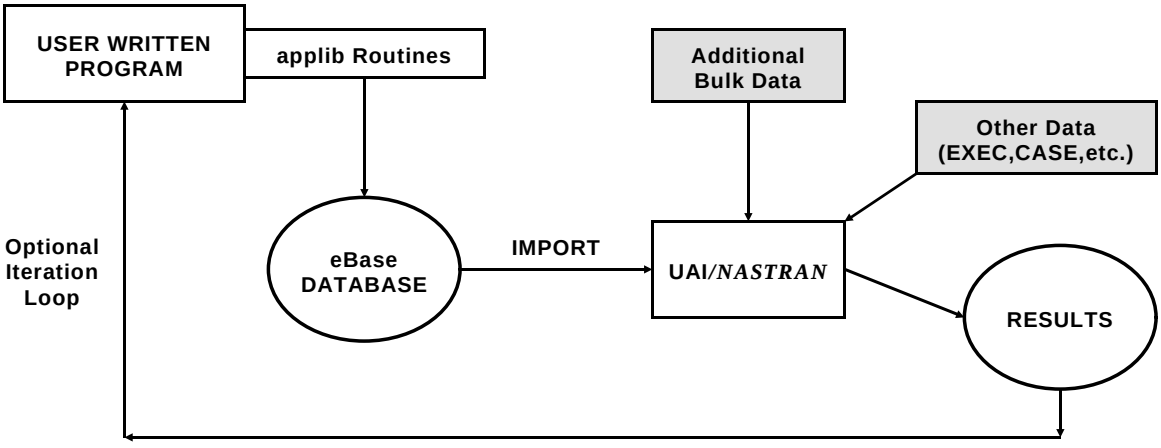
You may also access the ARCHIVE database from Fortran programs which use the ***eBase***:***applib***™ and ***eBase***:***matlib***™ subroutine libraries. The libraries are thoroughly described in the ***applib Programmer's Manual*** and the ***matlib Programmer's Manual*** which are found in the same volume. The ARCHIVE database approach greatly simplifies interfacing with **UAI/*NASTRAN***.

2.5.6 Alternate Bulk Data Source

Unless doing a RESTART, described in the next section, Bulk data for a **UAI/*NASTRAN*** execution is typically entered into the input data stream. However, those using the ***eBase*** tools can create their own input data and use it directly. This is illustrated in Figure 2-4.

User's may create their own programs which generate data. For example, graphic pre-processors that create GRID points and elements, or special load generators. These programs may then create an ***eBase*** database to store these data. As long as each type of data is created using the required schema (as defined in the ***eBase ARCHIVE Database Schemata Description Manual***) the database may be imported directly into **UAI/*NASTRAN***. You may then input other data into **UAI/*NASTRAN***, typically those data which are not generated by programs. The data in the ARCHIVE database are merged with any data found in the Bulk Data section of the input data stream. If any duplication of data is detected, the program terminates.

Figure 2-4. IMPORTING BULK DATA FROM A DATABASE



2.6 The CHECKPOINT/RESTART Concept

UAI/***NASTRAN*** has a sophisticated checkpoint and restart capability. This capability allows you to efficiently recover after either scheduled or unscheduled interruptions to program execution. Examples of such occurrences include:

- A job which aborts due to insufficient time, a data error, or insufficient memory.
- A job for which you requested termination at selected steps during the analysis for the purpose of examining intermediate results prior to continuing the run.
- A job which you wish to rerun for only a portion of the analysis with limited data changes.
- Requesting additional output recovery.
- Changing Solution Algorithms.

Restart provides two convenient features. Firstly, a restart selectively executes only the modules necessary to accomplish data change. You are able to change any part of your problem including model data changes, additional subcases, or more output requests. Secondly, restarts are automatic and eliminate the need for intervention. You need only checkpoint your original run and submit changes to the original run on subsequent executions. The following sections describe how you use the checkpoint and restart feature.

2.6.1 The Checkpoint Execution

You may Checkpoint your **UAI/*NASTRAN*** execution by including the **CHKPNT** Executive Control command in your input data stream. You must also **ASSIGN** the appropriate files to hold the Checkpoint Dictionary and the Checkpoint File. The typical commands to do this are:

```
ASSIGN logical_file=phys_file,NEW,USE=DICT
ASSIGN logical_file=phys_file,NEW,USE=CHKPNT
```

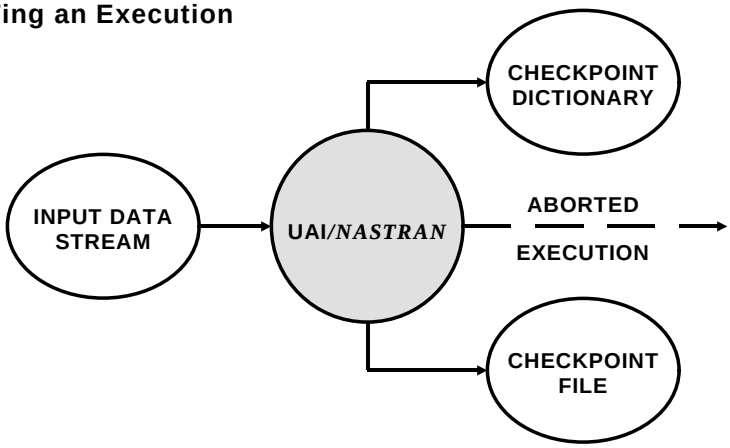
The sequence of events that occur during a checkpoint execution are illustrated in Figure 2-5a. The Checkpoint Dictionary represents a ***Table of Contents*** of the Checkpoint File. It contains the names of data blocks and parameter status. The Checkpoint File itself contains the actual **UAI/*NASTRAN*** data blocks that have been created during the execution.

2.6.2 The Restart Execution

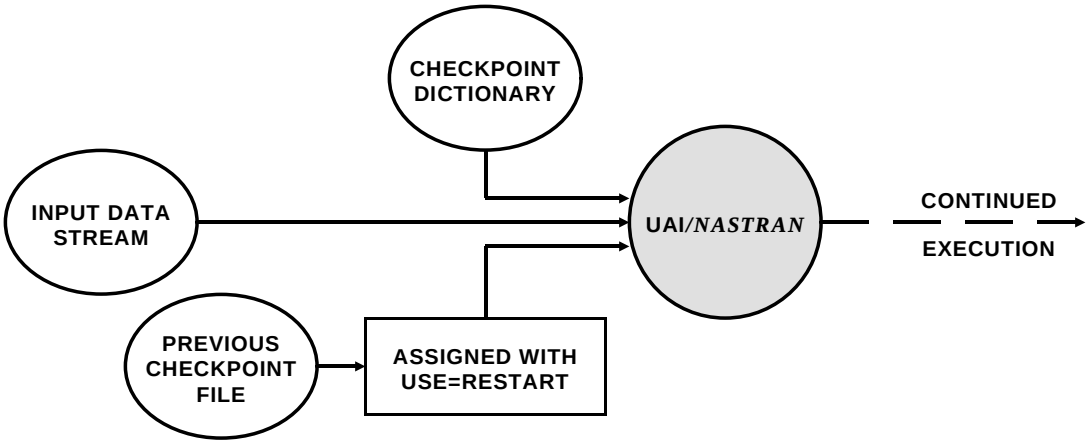
After you have executed in the Checkpoint mode, it is possible to perform a Restart. Figure 2-5b illustrates the data flow for a Restart operation. Your previously created Checkpoint File is **ASSIGN**ed with **USE=RESTART**. The input data stream may contain any changes you desire to your Executive Control, Case Control or Bulk Data packets. The Bulk Data packet *must only contain changes* to the original data. Such changes are implemented using the "**/**" Bulk Data entry, described in the Chapter 7 of the ***User's Reference Manual***, to delete any previously input Bulk Data. In order to use the "**/**" entry, you will need to reference the Bulk Data entries from the original Checkpoint execution. The Bulk Data entry identification numbers are given in the sorted Bulk Data echo. Any new Bulk Data are simply included in the Bulk Data packet. If you modify any of the data in your input stream then **UAI/*NASTRAN*** determines which modules within the program must be executed. This is also true if you change from one Rigid Format to another.

Figure 2-5. CHECKPOINT AND RESTART

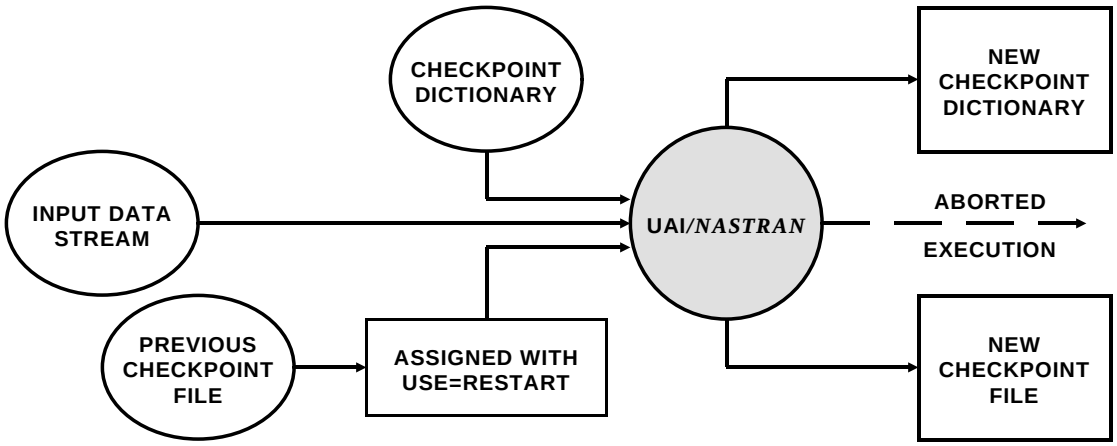
a. CHECKPOINTing an Execution



b. The RESTART Procedure



c. CHECKPOINTING When RESTARTing



The Restart execution may itself be Checkpointed also, as shown in Figure 2-5c. This process may be continued until a solution is finally obtained. At each step you use the new Checkpoint Dictionary and the new Checkpoint File as input to the next **UAI*/NASTRAN*** execution.

2.7 WHAT IS A CONFIGURATION?

In general, UAI's suite of engineering software products uses computing resources intensively. As a result, there are many parameters that may be set to customize the software and to achieve optimal resource management on a given host computer. These parameters, taken as a group, are called the ***Configuration*** of the products. The configuration is provided through several ***Preference Files***. These files include parameters which are used for controlling such things as database locations, physical file characteristics, memory utilization, and algorithm control.

For maximum flexibility, configurations may be controlled by the site, i.e. the UAI System Support Specialist, or the end user. Many different configurations may be defined for a site. For example, when configuring **UAI/*NASTRAN***, the System Support Specialist may create different configurations for very small and for very large analyses.

The configuration of UAI products begins with the Default Preference Files included in your delivery. There are one or more of these files for each UAI product. Those used by **UAI/*NASTRAN*** are:

rel_dir/nastran#/nastran.pref –	**UAI/*NASTRAN*** parameters
rel_dir/applib#/applib.pref –	***eBase*** kernel, ***applib*** and ***matlib*** parameters

The parameters contained in each of these files may be overridden by the contents of other Preference Files that the user provides. The actual configuration used for a given execution is determined by:

- Processing the Default Preference Files and setting all parameters included in these files to their specified values and then applying the specified Preference Files in the following sequence:
- First, any selected System Preference File is processed, and any parameters included in it replace those previously defined. Typically System Preference Files are created by your System Support Specialist to define special parameters for different types of jobs run at your site. By default, System Preference Files are stored in the directory where UAI software products have been installed.
- Second, any selected User Preference File is processed, and any parameters included in it replace those previously defined. Typically users create User Preference Files to define special parameters that they wish to use for all of their jobs. By default, User Preference Files are stored in a user's home directory.
- Third, any selected Local Preference File is processed, and any parameters included in it replace those previously defined. Typically users create Local Preference Files to define special parameters that apply to only a single, or small number of, jobs. By default, Local Preference Files are stored in the current working directory from which the job is submitted.

In summary, the final configuration is the union of the Preference files. The Default Preference files contain a value for every parameter used by the product suite. The selection of System, User, and Local Preference Files is usually done with the command that invokes the program. See subsequent sections of this manual for how this is done for each product. The other Preference Files need only contain those parameters that differ from, and override, the default values. They can also override any of the parameters contained in the Default Preference Files.

Each Preference File is composed of several ***Sections***. Each UAI software product may use one or more sections from each file. The sections used by **UAI/*NASTRAN*** are:

- The Host Section (**`applib.pref`**)
- The ***eBase*** Section (**`applib.pref`**)
- The ***eBase*:*applib*** Section (**`applib.pref`**)
- The ***eBase*:*matlib*** Section (**`applib.pref`**)
- The **UAI/*NASTRAN*** Section (**`nastran.pref`**)

The format of the Preference File and a brief description of its various sections are described in the following sections.

2.7.1 The Format of Preference Files

A Preference File is a text file which is composed of several Sections as indicated above. Each Section includes a header followed by the parameters associated with the Section. For ease of use, each Section is divided into subsections which contain related parameters, The format of these files is shown in the following sections.

2.7.1.1 applib.pref Default Preference File

The **`applib.pref`** file consists of the sections shown below:

```
[Host]
   < Site Description >
   <Preference Override Information >

[eBase]
   < Computing Resources >
   < I/O System Parameters >
   < Program Authorization >

[eBase:applib]
   < Computing Resources >

[eBase:matlib]
   < Solver Options >
   < General MATLIB Options >
   < Timing Constants >
   < Program Authorization >
```

2.7.1.2 nastran.pref Default Preference File

The **nastran.pref** file consists is shown below:

```
[UAI/NASTRAN]
   < Print File Controls >
   < Computing Resources >
   < Matrix Conditioning >
   < Data Checking >
   < Analysis Output Control >
   < Solution Techniques >
   < Element Options >
   < I/O System Parameters >
   < Assign Processing >
   < Index Archive Control >
   < Entity Redirections >
   < Program Authorization >
```


2.7.1.3 System, User, and Local Preference Files

These files can contain any of the parameters defined in these Default Preference Files. They must, however, still contain the correct Section and Subsection definitions for each parameter defined:

```
[Section]
   < Subsection >
   ...
[Section]
   < Subsection >
   ...
...
```

2.7.2 Configuration Parameters

Configuration parameters are defined using one of the forms:

```
param_name = value
param_name = ( value,value,...,value )
```

The *param_names* are case-insensitive. The *values*, when character strings or floating point numbers with exponents, are also case-insensitive unless they are enclosed in single quotation marks as:

```
param_name = 'This is a Case-Sensitive String'
```

Only one parameter may be specified on each line of the file. Any characters that appear after *value* are treated as commentary and ignored. You may also enter comments into the file by beginning a line with any of the characters $, *, or #.

2.7.3 The Configuration Sections

The following sections provide an overview of the sections in the Preference Files.

2.7.3.1 The Host Configuration Section

This section includes parameters the define the manufacturer and model of your host computer and licensing information. A complete description of available parameters is found in Section A.1.

*2.7.3.2 The **eBase** Kernel Configuration Section*

The ***eBase*** Configuration Section includes parameters which control the ***eBase*** Engineering Database Management System kernel. These include such information as default paths were databases are stored, physical block sizes for databases, and security information. This Section is discussed in detail in Chapter 3, and a complete description of all available parameters is found in Section A.2.

*2.7.3.3 The **eBase:applib** and **eBase:matlib** Sections*

The ***eBase*:*applib*** and ***eBase*:*matlib*** Configuration Sections are described in Chapter 6. The parameters in these sections include such items as dynamic memory sizes for ***applib***, and timing constants for the ***matlib*** high-performance matrix utilities. A complete description of available parameters is found in Section A.2.

*2.7.3.4 The UAI/**NASTRAN** Configuration Section*

The **UAI/*NASTRAN*** Configuration Section includes parameters which control **UAI/*NASTRAN***. These include controls on peripheral and computing resources, model data checking, program defaults, and so forth. This section is discussed in Chapter 4. A complete description of available parameters is found in Section A.3.

2.7.4 Example Use of Configuration Overrides

The following example shows a typical preference file that might be specified by **UAI/*NASTRAN*** users to define the resources necessary to run large models:

```
[eBase]
   < I/O System Parameters >
    Temp-eBase-Data-Loc = /work1/tmp
    Temp-eBase-Data-Loc = /work2/tmp
    Temp-eBase-Data-Loc = /work3/tmp

[UAI/NASTRAN]
   < Computing Resources >
    Working-Memory = 20mw
```

If the System Support Specialist stored these configuration parameters in a file named **bignast.pref** located in the installation directory, then users could select them with the following command (this example is for Unix systems):

```
nastran -ps bignast mydata
```

The configuration parameters found in **bignast** override those in the default Configuration file and provide more memory and disk resources for **UAI/*NASTRAN***. The **-ps** parameter on the **nastran** command selects any override files. For complete descriptions of these parameters, see the ***Installation Guide and System Support Manual***.

2.7.5 How Applications Modify the Configuration

All UAI Products and user-developed ***eBase*:*applib*** applications can override the default configuration parameters at run-time in three ways.

- By default, ***eBase*:*applib*** applications search for System, User, and Local override Preference Files having the name **uai.pref**. First the installation directory is searched, then the user's home directory, and finally, the current working directory. The names of these files, and their location, may be modified for your site by changing the Default Preference Files as described in Chapter 3.
- Users may provide their own file names and file locations. The exact method for doing this differs for each UAI Product. See the individual Chapters of this manual for specific instructions.
- The commands provided by each UAI product and the Applications Programming Interface used in the development of ***eBase*:*applib*** applications allows certain configuration parameters to be overridden. See the appropriate Chapters of this manual, and individual product manuals for additional details.

2.8 INTERFACING WITH OTHER PROGRAMS

Numerous computer programs, available commercially and also found *in-house* in many engineering organizations, have been developed to enhance the productivity of engineers using **UAI/*NASTRAN***. Such programs typically provide functions including interactive model generation often using data provided by a CAD or solid modeling program. Other functions may include model display with special features such as hidden surface removal or a *shrink* option for each element to graphically locate missing elements in the model. Another major function provided allows the graphical display of deformed shapes of the structure and stress contours over the surface or over a cut through a solid section of the finite element model.

Several versions of NASTRAN are in use throughout the world. Each version has slightly different input and output data structures, and different pre- and post-processor programs have *up-to-date* interfaces for one or more of these versions. Universal Analytics, Inc. has special provisions for insuring data compatibility with the existing data structures developed for other widely used systems. Interfaces to specific products are shown in Table 2-2, Chapter 31 provides more detailed information.

Table 2-1. INTERFACES TO THIRD-PARTY SOFTWARE

VENDOR	PRODUCT	DESCRIPTION
ESP	FEMAP®	ESP and UAI have developed an interface between ESP's FEMAP pre- and post-processing system and **UAI/*NASTRAN***. This interfase uses both ASCII files and **OUTPUT2** files. The **POST** Case Control command is used to control the generation of the binary files which are required. FEMAP can create complete **UAI/*NASTRAN*** input files and performs comprehensive post-processing of solution results.
MSC	PATRAN® Version 2.5 and higher	The **UAI/*NASTRAN*** Rigid Formats have integrated support for PATRAN. The **POST** Case Control command is used to control the generation of the binary files which are required by the PATRAN program. The **POST** command allows you to specify the PATRAN version. In Multidisciplinary Rigid Formats such as Buckling, the **POST** command may be used in each discipline to select different output.
	PATRAN® Versions 5,6 and 7	UAI also provides an optional PATRAN Preference. Contact UAI directly for up-to-date information about this product.
SDRC	I-DEAS Master Series™ Simulation	The **UAI/*NASTRAN*** Rigid Formats have integrated support for I-DEAS Master Series Simulation. The **POST** Case Control command is used to control the generation of the binary files which are required by the I-DEAS™ Data Loader program. This program then creates a universal file required to as input to I-DEAS Simulation.
	I-DEAS Master Series™ System Dynamics Analysis	There is an ALTERLIB member, **SDRCSDA**, which allows you to automatically create the **OUTPUT2** files needed by this program. These files are then input to the System Dynamics Analysis program.

Chapter 3

3

ORIENTATION: THE ENGINEERING ANALYSIS SYSTEM

This chapter provides you with an orientation to the engineering capabilities of **UAI/*NASTRAN***. It also gives important information of the practical aspects of modeling your structure and validating the correctness of the model. This information will be of great use in selecting the best analytical techniques for the solution of a particular problem.

3.1 OVERVIEW

As described in Chapter 2, **UAI/*NASTRAN*** is a large-scale, general-purpose computer program. It has been designed to analyze the behavior of structures in a variety of operating environments. Its general nature makes the program usable: for structures of any size, shape, class, or configuration; for any geometric representations which can be identified by convenient coordinate systems; for elastic relations which may range from isotropic to general anisotropic; for complex as well as real matrix operations; for natural vibration frequency and mode shape determination; and for a wide variety of loading conditions. The various loading conditions may be: concentrated and distributed mechanical loads; transient, steady-state sinusoidal; static temperature profiles; enforced deformations; both time-varying and static surface and body forces; and stationary Gaussian random excitation. **UAI/*NASTRAN*** will also perform fluid-structure interaction, soil-structure interaction, and heat transfer analyses. Finally, **UAI/*NASTRAN*** may be used to perform Multidisciplinary Design Optimization of structures simultaneously for static, modal, and direct frequency response behavior.

3.1.1 Basic Approach

UAI/*NASTRAN* analyzes models by the displacement approach of the finite element method. With this approach, the distributed physical properties of a structure are represented by a model which consists of a finite number of idealized elements. These elements are defined by, and serve to interconnect, a finite number of GRID points to which loads may be applied. The GRID point is the basic framework for the structural model and all aspects of the model definition are referenced either directly or indirectly to these GRID points. The **UAI/*NASTRAN*** elements include structural elements, such as trusses and rods, and both two- and three-dimensional continuum finite elements. These elements are then used to specify many of the properties of the structure. These include material properties, physical properties, mass distribution, and some types of applied loads. In addition, various kinds of constraints can be applied to GRID points. These may be used to specify boundary support conditions and linear dependency relationships among selected displacement components, or ***Degrees of Freedom***.

3.1.2 Finite Elements

An extensive library of both structural and finite elements is provided in **UAI/*NASTRAN***. These facilitate the modeling of nearly all products. The elements include rods, beams, solids, shear panels, plates, and shells of revolution. More general types of construction may be treated using combinations of these elements and, optionally, general elements which you may define. For sophisticated dynamic and control systems analyses, non-structural features may be incorporated into the analytical model.

3.1.3 Large Problem Solving

UAI/*NASTRAN* has been specifically designed to treat large problems with many degrees of freedom. The only limitations on problem size are those imposed by practical considerations of running time and by the ultimate capacity of auxiliary storage devices provided by the computer facility being used. Computational procedures implemented in **UAI/*NASTRAN*** have been selected to provide the maximum obtainable efficiency for large problems with particular attention given to processing sparse matrices. Though **UAI/*NASTRAN*** is decidedly not constrained to in-memory processing only, in most cases, the more memory that can be made available, the more efficient that processing will be.

UAI/*NASTRAN* also provides a substructuring capability for static and dynamic structural analyses. This capability is particularly useful for handling large or complex models and for modeling structures with repetitive components. The capability may be used to realize significant processing and model development efficiencies for very large structures. It can also provide a useful tool for the project management of large-scale analyses.

3.1.4 Power and Flexibility

The needs of the structural analyst have been considered in all aspects of the design of **UAI/*NASTRAN***. However, in view of the wide range of possible applications of the program now and in the future, all the analytical and user interface needs could not be anticipated in advance. For this reason, a high degree of flexibility and generality has been incorporated into the program. For example, in addition to the usual library of structural elements used to model specific types of construction, you may define a ***General Element*** to represent part of a structure by providing deflection influence coefficients, or to represent part of a structure by its vibration modes. A number of convenience features are provided, including plotting, selective output, checkpoint/restart, and diagnostic facilities which are definite necessities for large problems.

A major difficulty that is faced in the solution of large problems is the avoidance of errors in the preparation of the input data. Data entry formats and ordering are made as simple and flexible as possible in order to avoid errors caused by trivial violations of format rules. A number of aids such as automatic data generation, Structural Plotting, X-Y Plotting, GRID point singularity tests, and GRID point force balance results have been provided to help you develop and validate your model. The checkpoint/restart capability is designed for recovery from data or system error conditions, to allow for examination of intermediate results, and to allow for retrieval of additional selective output following completion of the analysis.

3.2 STANDARD UAI/*NASTRAN* STRUCTURAL ANALYSES

UAI/*NASTRAN* currently offers a wide variety of *Rigid Formats* and an APEX Program which are used to select and control the standard analysis disciplines. These analyses, summarized in Table 3-1, may be classed into seven groups:

- ❒ Linear Static Analyses (including Design Sensitivity Analyses)
- ❒ Normal Modes Analyses (including Design Sensitivity Analyses)
- ❒ Dynamic Response Analyses (including Response Spectrum Analyses)
- ❒ Complex Eigenanalyses
- ❒ Nonlinear Geometry Static Analyses
- ❒ Nonlinear Material Static Analyses
- ❒ Heat Transfer Analysis
- ❒ Multidisciplinary Analysis and Design Optimization
- ❒ Aeroelasticity

Table 3-1. UAI/NASTRAN RIGID FORMATS

STANDARD ANALYSIS	RIGID FORMAT or APEX	ANALYTICAL CAPABILITY
STATIC ANALYSIS	1 2 14 51	Linear static analysis Statics with inertia relief (free-body analysis) Static analysis with cyclic symmetry Design Sensitivity Analysis
NORMAL MODES ANALYSIS	3 13 15 52	Normal modes analysis Normal modes with differential stiffness Normal modes with cyclic symmetry Design Sensitivity Analysis
DYNAMIC RESPONSE ANALYSIS	8 9 11 12 17	Direct frequency and random response Direct transient response Modal frequency and random response Modal transient response Response spectrum analysis
COMPLEX EIGENANALYSIS	7 10	Direct complex eigenvalue analysis Modal complex eigenvalue analysis
GEOMETRIC NONLINEAR STATIC ANALYSIS	4 5 16 1 (APP NONLINEAR)	Static analysis with differential stiffness Buckling analysis Buckling analysis with cyclic symmetry Geometric and combined nonlinear analysis
MATERIAL NONLINEAR STATIC ANALYSIS	1 (APP NONLINEAR)	Static analysis
HEAT TRANSFER ANALYSIS	1 2 9 (APP HEAT)	Linear steady-state conduction Nonlinear steady-state heat transfer Transient heat transfer
MULTIDISCIPLINARY ANALYSIS AND DESIGN OPTIMIZATION	MULTI	Performs analysis, sensitivity analysis, and optimization for any combination of linear statics (including inertia relief), normal modes, and both direct and modal frequency response analyses.
AEROELASTIC FLUTTER	MULTI	Performs aerodynamic load generation, splining of loads to the structure, and flutter analysis.

Each Rigid Format treats a single class of problem and has its own particular set of control parameters, output options and types of input data. The APEX (Advanced Programmable Executive) is a Fortran-based solution sequence introduced with Version 11.7. It represents the first portion of the Open Architecture concept which will be released in **UAI/*NASTRAN*** in the future. Both the Multidisciplinary Analysis and Design Optimization and the Aeroelasticity capabilities has been formulated using APEX, rather than DMAP, in order to support the complex interrelationships of the multidisciplinary analysis process. The following sections offer guidelines to help you select the approach or method of analysis best suited to a structural analysis problem.

3.2.1 Using the Rigid Formats

A number of solution algorithms have been included in **UAI/*NASTRAN*** as Rigid Formats in order to relieve you from the necessity of constructing a DMAP sequence for commonly used analyses. A Rigid Format consists of a DMAP sequence that is stored in **UAI/*NASTRAN*** and available to you by specifying the number of the Rigid Format using the **SOL** Executive Control command. Additionally, Rigid Formats contain a set of Restart tables that automatically modify the series of DMAP operations to account for any changes that are made in any part of the Input Data stream when making a Restart, after having previously run all, or a part of the problem. These tables insure you of a correct and efficient program execution. See Chapter 2 for more information.

If you wish to modify the DMAP sequence of a Rigid Format you may use DMAP ALTER features. Typical uses are to force an exit prior to completion, to check intermediate output, to request the printing of a table or matrix for diagnostic purposes, and to delete or add a functional or utility module to the DMAP sequence. Any DMAP instructions that are added to a Rigid Format are automatically executed when a Restart is performed. You should be familiar with the rules for DMAP programming, as described in Chapter 8 of the **UAI/*NASTRAN User's Reference Manual***, prior to making alterations to a Rigid Format. Note that the new APEX program for Multidisciplinary Analysis and Design may not be accessed.

Your may need a listing of the DMAP comprising a Rigid Format to insure that you know the location where you may ALTER these solutions in order to print additional data or solution results that are not provided by the standard output commands. You may obtain a listing of any Rigid Format by using **DIAG 14** in the Executive Control packet. For example, a complete **UAI/*NASTRAN*** input file to obtain the Normal Modes (Rigid Format 3) listing would consist of the following data:

```
SOL 3
DIAG 14
ALTER 1
EXIT $
CEND
TITLE = RIGID FORMAT 3 - NORMAL MODES - LISTING
BEGIN BULK
ENDDATA
```

3.2.2 Linear Static Analysis

There are four available Rigid Formats for the solution of linear statics problems, as shown in Table 3-1. The Linear Static Analyses provided by Rigid Formats 1, 2, 14 and 51 use most of the features available for model definition in **UAI/*NASTRAN***. These include specification of GRID point locations and coordinate systems, element connectivities, material properties, constraints, and temperature distributions as well as direct GRID point and element loadings. The solution results include weight and balance information, displacements, element forces and

stresses, and support reactions. Plots of the undeformed and the deformed structural geometry are also available.

3.2.2.1 Inertia Relief

Rigid Format 2, which provides all the features of Rigid Format 1, also treats structural models which are not fully constrained. It computes the inertia effects of unconstrained rigid body accelerations. These effects are added to the your loadings; the rigid body motions are constrained; and, finally, a normal static analysis is performed.

3.2.2.2 Cyclic Symmetry

Rigid Format 14 is a special case of Rigid Format 1 which provides for efficient modeling and analysis of structures exhibiting cyclic symmetry. Many structures, including pressure vessels, rotating machines, and antennae for space communications, are made up of virtually identical segments that are symmetrically arranged with respect to a centroidal axis. The use of Cyclic Symmetry requires that you model only one of the identical segments. This can yield a large saving of computer time, hence cost, for many problems. Chapter 22 describes this special feature.

3.2.2.3 Substructuring

The Substructuring capability can be used with either Rigid Format 1 or 2. This technique greatly enhances analysis flexibility and can accommodate very large problems. The Substructuring feature is described in detail in Chapter 19.

3.2.2.4 Design Sensitivity

Finally, Rigid Format 51 may be used to perform design sensitivity analysis for statics. This feature, described in detail in Chapter 26, allows you to define design variables and constraints. You may then determine the sensitivity of these constraints to changes in the design variables. This information is crucial when performing automated design.

3.2.3 Normal Modes Analysis

The extraction of natural frequencies and normal modes of a structural system is available in four Rigid Formats as shown in Table 3-1. Rigid Formats 3, 13, 15, and 52 provide for a normal mode analysis of undamped systems with symmetric matrices. Rigid Format 3 is for use with elastic, linear structural models and can also be used with Substructuring.

3.2.3.1 Differential Stiffness

Rigid Format 13 allows the use of differential stiffness effects. Differential stiffness effects are a linear approximation to large deflection problems. The applied loads are assumed to move with their points of application and they remain fixed in magnitude and direction. Rigid Format 13 adds the differential stiffness matrix to the original stiffness matrix in proportion to a user-specified increment of the applied load.

3.2.3.2 Cyclic Symmetry

You use Rigid Format 15 to compute normal modes for Cyclic Symmetry models.

3.2.3.3 Design Sensitivity

Rigid Format 52 may be used to perform design sensitivity analysis for normal modes analysis. See Chapter 26 for details.

3.2.4 Dynamic Response Analysis

Both modal and direct formulations are available for the solution of dynamic response problems in either the time or frequency domain. The available Rigid Formats are also summarized in Table 3-1. The **UAI*/NASTRAN*** dynamic analysis Rigid Formats provide a maximum amount of flexibility and capability. Any combination of external loads and enforced motion, including displacements, velocities, and accelerations, may be specified. Rigid Formats 8 and 11 offer Frequency and Random Response capability and Rigid formats 9 and 12 offer Transient Analysis capability. Rigid Formats 8 and 9 use a direct approach; the other two use a modal approach.

These capabilities can be used to solve problems with large numbers of degrees of freedom, time steps, or frequencies and mode shapes. Provision has been made to allow versatile special modeling techniques. Care should be taken, however, to avoid the dynamic analysis of models that may be unnecessarily large because of the computing expense required to extract the solution. You should pay special attention to the selection of the most economical method for solving your particular problem. The Guyan Reduction, Dynamic Reduction, modal reduction and residual flexibility features of **UAI*/NASTRAN*** can be used effectively to reduce problem size.

3.2.4.1 The Modal Approach

The modal approach for dynamic analyses differs from the direct approach in that it uses the undamped mode shapes as independent degrees of freedom replacing the original model. In either approach, EXTRA points may be added for special modeling purposes. The input loads for the modal approach are transformed to the modal coordinates, and the resulting modal displacements are transformed back to the physical displacements for output purposes. The choice of direct or modal approach is dependent on the specific problem. The direct approach solves a more general problem and does not require the extra steps of extracting the normal modes and of performing the transformations to form modal coordinates. For large systems, however, the direct formulation may be inefficient compared with the modal approach.

3.2.4.2 Mode Acceleration

Also available with the modal approach, Rigid Formats 11 and 12, is a mode acceleration method for improving the solution accuracy. This procedure replaces the unconstrained rigid body accelerations with equivalent inertial and damping forces. The augmented loading is then solved statically to obtain improved element forces and stresses. You are cautioned, however, that this method is costly and should be used selectively.

3.2.4.3 Residual Flexibility

A generalization of the modal reduction method is available in the APEX program that allows solution vectors from selected static loads to be used to augment the generalized degrees of freedom. This approach, called ***Residual Flexibility*** or ***Residual Stiffness***, can improve the frequency range over which the modal approximations are accurate. The selected static load cases are typically taken from the spatial components of the dynamic load (i.e. the **DAREA**), but can be any appropriate load case that results in a generalized mode shape useful in describing the system dynamics. This method is described in detail in Chapter 9.

3.2.4.4 Frequency and Random Response Analyses

You can use Frequency and Random Response analysis, Rigid Formats 8 and 11, to analyze the steady state response of systems due to sinusoidally varying loads at specified frequencies. The model may include direct input matrices and transfer functions. The assembled matrices may be non-symmetric, real or complex, and singular or non-singular. Loads or motions must be input on specific degrees of freedom and may have phase angles, time lags and frequency dependent magnitudes. The output may be sorted by frequency or by component for all frequencies and printed in either **RECTANGULAR** or **POLAR** format. Curve plotting of these results is also available for each output component as a function of frequency.

The Random Response analysis is treated as a data reduction procedure that is applied to the results of a Frequency Response analysis. To use this feature, you input a power spectral density function, which is assumed to be stationary with respect to time, and the program computes the power spectral density, mean deviation and auto correlation function for the response of the selected output components. The output from the Random Analysis must be selected one component at a time, not with Case Control options, but with X-Y Plotter commands. You must use the X-Y Plotter to recover random response solution results.

3.2.4.5 Transient Response Analyses

The Transient Analyses, Rigid Formats 9 and 12, perform a step-by-step integration of the structural response in the time domain. You provide the initial conditions, time step requirements, and time varying loads or motions. The available output, given at selected integration time intervals, includes GRID point displacements, velocities, accelerations, and support reactions. as well as element forces and stresses,

Some special features available to Transient Analysis include the capacity to incorporate direct input matrices, unsymmetric control function equations, and a limited nonlinear analysis capability. The nonlinear capability is available through displacement and velocity dependent load components applied to specific degrees of freedom. For purposes of economy, you may change the integration time step over the course of the integration.

3.2.4.6 Response Spectrum Analyses

UAI/*NASTRAN* provides an analytical capability to perform Response Spectrum Analyses using Rigid Format 17. This capability is especially suited to earthquake analysis where the the base excitation can be represented by a response spectra. This feature first computes the normal modes of the structure and, optionally, the modal participation factors for each mode. It then calculates the modal response for each mode and then transforms the results into physical displacement, velocity, and acceleration. A number of industry-standard combination rules are available to simplify response calculations.

3.2.5 Complex Eigenvalue Analysis

Two Rigid Formats, shown in Table 3-1, are available to compute the complex eigenvalues and eigenvectors of non-conservative systems. These Rigid Formats provide for a very general overview to the vibration characteristics of a system which includes the input of real or complex mass, damping and stiffness matrices. The assembled matrices may be real or complex, symmetric or unsymmetric, singular or nonsingular.

Two formulations are provided, the direct formulation of Rigid Format 7 and the modal formulation of Rigid Format 10. The direct formulation provides for the most general description of a model which may include viscous and structural damping. The unknowns are the degrees of freedom related to GRID points, SCALAR points and the special EXTRA points which are used to specify such conditions as may be encountered in control systems, or hydrodynamics.

The modal approach, in contrast, employs the free vibration modes of the undamped model as independent degrees of freedom to be assembled with the direct input matrices and EXTRA point control parameters. Proportional viscous damping in the modal coordinates is included. You may also use the Guyan or Dynamic Reduction features with Rigid Formats 7 and 10. The selection of approach depends on the particular problem to be solved and the number of roots to be extracted. The direct approach offers the greatest flexibility if only a few roots are desired. If many roots are to be extracted, the modal approach may be more economical.

Available solution results include frequencies and mode shape displacements, element forces and stresses and support reactions. These may be printed in either **SORT1** or **SORT2** in either **RECTANGULAR** or **POLAR** format. **SORT1** prints all the requested output for each frequency. **SORT2** prints each component of output for all frequencies.

3.2.6 Geometric Nonlinear Static Analysis

UAI/*NASTRAN* provides three Rigid Formats to solve static geometric nonlinear problems as shown in Table 3-1. The differential stiffness effects treated by Rigid Formats 4, 5 and 16 provide a second order approximation to the nonlinear effects of large deflections. The applied loads are assumed to move with their points of application and they remain fixed in magnitude and direction. The differential stiffness matrix is computed as a correction to the original stiffness matrix from the work done by each element in response to a preload. In Rigid Format 4 this matrix is added to the original stiffness matrix in proportion to a user-specified increment of the preload. The new matrix is then used to generate an independent static solution for the deflections due to the preload plus incremental load which is the total load applied to the model.

A complete geoetric nonlinear analysis capability is also provided. This allows you to solve problems having large deformations and rotations. These features are thoroughly described in Chapter 17 of this manual.

3.2.6.1 Linearized Buckling

Rigid Format 5 adds the differential stiffness matrix to the original stiffness matrix and then solves an eigenvalue problem to obtain the critical load factor that would produce buckling. This critical load factor, or the eigenvalue, multiplied into the preload vector gives the critical loading at which the structure would go unstable. The eigenvector obtained is the buckling mode of the structure. Finally, Rigid Format 16 provides the capabilities of Rigid Format 5 for problems which have been modeled using the Cyclic Symmetry feature.

3.2.7 Material Nonlinear Static Analysis

UAI/*NASTRAN* provides one Rigid Format to solve material nonlinear statics. A selected set of elements may be modeled with nonlinear material properties defined by load-deflection or stress-strain relations. The static solution may then be obtained using this feature. The one-dimensional elements BAR, BEAM, ELASNL, GAP, PILE, and ROD; the plate elements QUAD4, QUADR, TRIA3, and TRIAR; and the solid elements HEXA, PENTA and TETRA may be nonlinear.

Solution methods allow both elastic and plastic unloading, tangential or secant modulus stiffness updates, unbalanced force iteration solutions between stiffness updates for improved efficiency, and numerous other controls on the solution process. The nonlinear statics Rigid Format uses a database, called the NLDB, for improved data flow in the program. This database may be saved between **UAI/*NASTRAN*** executions and used to perform solution restarts.

Models may include any number of linear **UAI/*NASTRAN*** elements; their behavior will remain linear during the solution. Also, SPC, MPC and rigid element constraints may be used just as in linear static analysis.

3.3 SPECIAL UAI/*NASTRAN* CAPABILITIES

In addition to the standard **UAI/*NASTRAN*** analyses discussed in the previous section, there are a number of special, or advanced, modeling and solution techniques:

- ❒ Substructuring and Modal Synthesis
- ❒ Heat Transfer
- ❒ Medium / Structure Interaction
- ❒ Multidisciplinary Design Optimization
- ❒ Aeroelasticity

Each of them is briefly described in this section. Complete descriptions for using these capabilities are presented later in subsequent Chapters of this manual.

3.3.1 Substructuring

In cases when finite element models become too large for efficient **UAI/*NASTRAN*** processing, substructuring may be used as a method to subdivide the model into more convenient and manageable components. Each component substructure itself may be further subdivided into even more basic components. These basic components may be modeled independently and combined directly with other components using matrix operations to build the final desired model for solution analysis. This procedure is also convenient for redesign and remodeling problems: changes in one section of the structure do not necessarily require reformulation of the entire model.

The substructuring system uses a three-phase approach for modeling with substructures. Each elementary or basic substructure component model is created in a PHASE 1 execution. The only restrictions on the model are that boundary GRID points have compatible locations with adjoining components. GRID point and element identification numbers and responses are unrestricted. All required data for the models are automatically generated and permanently stored in a database that you must provide and maintain. This database is called the Substructure Operating File or SOF.

In the second phase of substructuring, the substructures may be combined; they may be reduced in size using the Guyan reduction or modal synthesis methods; and they may be combined again to form even more complex models. At each stage you maintain direct control over each operation using simple engineering-based commands such as **COMBINE**, **REDUCE**, and **SOLVE**. All data are stored on the SOF at each stage. This provides ease of restarting and recovering from data errors. Any substructure may be analyzed using standard **UAI/*NASTRAN*** Rigid Formats.

For detailed data recovery at the level of the original basic substructure, a PHASE 3 execution is made using a restart file or the original model input data itself. Again, the SOF is accessed by **UAI/*NASTRAN*** to automatically retrieve the required solution vectors needed for computing detailed deformations and element stresses.

3.3.2 Heat Transfer Analysis

Modeling procedures for heat transfer analysis using finite elements closely parallels standard structural modeling methods. Normal **UAI/*NASTRAN*** rod, plate, and solid elements may be used to model conduction and capacitance for heat transfer analysis in structures. At each

GRID point, temperature is the generalized degree of freedom rather than displacement. Heat input rates on GRID points, surfaces, and volumes are the generalized loads. Fixed temperatures at selected GRID points may also be specified.

A set of specialized ***boundary elements*** are also provided for heat transfer modeling. These elements are used to define convection between structural surfaces and fluids or other structural surfaces. They are also used in the definition of radiation ***view factor*** terms and as receivers of thermal flux inputs. Other non-convective boundaries may be specified using conventional constraints.

3.3.2.1 Steady-State Heat Transfer

UAI/*NASTRAN* provides three Rigid Formats for heat transfer analysis as summarized in Table 3.1. The linear steady-state heat transfer formulation is analogous to the structural formulation with temperatures replacing deflections as the unknowns. Rigid Format 1 is available for determining steady-state temperatures with no radiation effects. The same modeling features for specifying GRID point geometry, element connectivity and material properties that you use for structural analysis are also available for Heat Transfer analyses. In addition, special features are available for the specification of fixed temperature, convection, and heat flux boundary conditions. The available output consists of resultant GRID point temperatures and element heat fluxes throughout the model.

3.3.2.2 Nonlinear Heat Transfer

Rigid Format 3, which provides all the features of Rigid Format 1, also treats heat transfer analyses with temperature dependent conductivities of the elements and nonlinear radiation exchange. The solution of thermal equilibrium is obtained using an iterative technique that you may control.

3.3.2.3 Transient Heat Transfer

The capability of time domain analysis for models that include linear conduction, film heat transfer, nonlinear radiation, and any special effects modeled with **UAI/*NASTRAN*** nonlinear elements is provided by Rigid Format 9. Once again, you are provided with options to control the integration procedures for linearized radiation effects.

3.3.3 Medium-Structure Interaction

UAI/*NASTRAN* provides analytical capabilities for both fluid-structure and soil-structure interaction problems. These are briefly described in this section.

The fluid-structure interaction capability of **UAI/*NASTRAN*** is designed primarily to solve for small motion dynamic response of models with combined structure and fluid effects. The options include both rigid and flexible container boundaries, free surface effects and compressibility.

The fluid is modeled by 3-D finite elements, including the HEXA, PENTA and TETRA, while the structure is described using any 2-D or 3-D element in the **UAI/*NASTRAN*** finite element library. The elements which lie on the fluid-structure interface may have a non-matching mesh. In such cases, the user may define the wetted surfaces, or the program will generate them automatically. The analysis assumes small perturbations about static equilibrium. Second or-

der velocity terms are ignored. Special features of the Fluid-Structure Interaction analysis capability are described in Chapter 23 of this manual.

UAI/*NASTRAN* provides a special element, the PILE, for the analysis of structural piles embedded in soil. This element may be used in both linear statics and dynamics analysis and in nonlinear statics. The element provides for the definition of both structural and soil properties, any of which may be linear or nonlinear in any combination. The properties and material of the PILE element are defined much the same as for the BAR element. Soil properties may be defined in axial, lateral and torsional directions. Soil property input is usually load versus deflection tabular data obtained from field tests.

3.3.4 Multidisciplinary Design Optimization

UAI/*NASTRAN* provides a powerful Multidisciplinary Design Optimization capability. This feature allows you to design a structure automatically for a specific objective function such as minimum weight or volume. This is done by formulating a ***Design Model*** based on the finite element model of the structure. The design model specifies a set of Design Variables, such as plate thicknesses, member areas, or GRID point coordinates, which may be changed during the design cycle. It then defines constraints on the structural responses which correspond to the design requirements of the structure. Such constraints include stress allowables, natural frequencies, or response to frequency excitation.

The design model is then optimized across one or more analytical disciplines which include statics, normal modes, and both direct and modal frequency response analyses. Mathematical Programming methods are used to solve the simplified Approximate Problem. Details of this procedure are found in Chapter 25 of this manual.

3.3.5 Aeroelasticity

For solving many problems in aeroelastic stability, **UAI/*NASTRAN*** provides capabilities for computing aerodynamic loads and performing flutter analyses with those loads. This capability is described in detail in Chapter 31 of this manual.

3.4 PRACTICAL ASPECTS OF MODELING

Before using a system with the variety of features found in **UAI/*NASTRAN***, comprehensive planning should be done. A basic understanding of the available solution techniques and how they are used is also important. These topics are discussed in this section.

3.4.1 Planning a UAI/*NASTRAN* Analysis

Detailed modeling of complex structures can be expensive in terms of both human and computer resources. Control of these costs is usually required, and control can best be maintained by adherence to an analysis project plan. In this section, guidelines for formulating a **UAI/*NASTRAN*** analysis plan are presented.

A suggested approach to planning an analysis requires separating the activity into four phases: the problem definition phase, the data preparation phase, the analysis execution phase and the solution interpretation phase.

The problem definition phase provides the framework for the remaining planning activity. Key activities in this phase are illustrated in Table 3-2.

Table 3-2. PROBLEM DEFINITION

1.	Identify structure
2.	Define analysis objectives
3.	Identify desired results
4.	Define engineering assumptions
5.	Specify budget and schedule

The first step is to identify the structure to be modeled using available drawings, CAD models, layouts or other documents. Information relative to environmental factors such as loads and boundary conditions should also be identified. This information allows you to clearly define the purposes and objectives of the analysis, and to formulate the results that you desire to obtain. The next step is to estimate the budget, computer resource requirements and schedules for the analysis project. Finally, you should outline the engineering assumptions which will be made while performing the analysis. Once these steps have been completed, you are ready to proceed to the data preparation phase.

The data preparation phase is generally the most time consuming. Table 3-3 illustrates the steps required during this phase. You first must gather all of the detailed information for the model—the geometry, constraints, loads, and any other necessary data. Next, you determine which solution algorithm, or Rigid Format, will provide you with the desired results. It is useful to consider whether any external interfaces to **UAI/*NASTRAN*** are needed. For example, are pre- or post-processors going to be employed? The final report should be outlined to ensure that you request all of the appropriate **UAI/*NASTRAN*** solution results during the analysis.

Table 3-3. DATA PREPARATION

1.	Collect descriptive data
2.	Select solution algorithm
3.	Consider final report requirements
4.	Generate input data stream
5.	Layout cost and schedule plans

A detailed project schedule and cost estimate should be prepared. It will hopefully be consistent with the requirements established in the problem definition phase. If not, you must either request additional funding or time, or you must scale-down the analysis. Finally, you must generate the input data stream for your model. You are then prepared to execute **UAI/*NASTRAN***, following guidelines such as those presented in Table 3-4.

Validation procedures should be specified whenever possible to allow checking of your model. Such procedures might include correlation with test data, using simple boundary conditions or loadings, or extensive use of computer graphics. You should also track your progress with respect to your project cost and schedule plan. For a very large or complex analysis a ***pilot model*** analysis could be most effective. With the pilot model, you test all of the **UAI/*NASTRAN*** features that will be exercised in the final analysis. This will assure that planned analysis methods will satisfy the analysis objectives.

Table 3-4. UAI/NASTRAN EXECUTION

1.	Specify validation procedures
2.	Track cost and schedule
3.	Develop pilot model

Table 3-5. SOLUTION INTERPRETATION

1.	Is the solution intuitively correct?
2.	Have analysis objectives been met?
3.	Is project within budget and schedule?
4.	Are engineering assumptions correct?
5.	Is the design satisfying its requirements?

The final phase is the interpretation of the **UAI/*NASTRAN*** solution results. Some important questions which should be answered each time you obtain a solution are given in Table 3-5. When interpreting solution results, always use your good engineering judgment. Are the results intuitively correct? You must also verify that all of the objectives of the analysis have been met within the budget and schedule allocated. Consider these results for future analysis projects. If the correlation of your results with other independent solutions is bad, you must review your engineering assumptions. Finally, if the design under analysis has failed to perform in its operational environment, can the failure be explained and corrected?

3.4.2 Critical Thought Processes

UAI/*NASTRAN* is a very flexible and powerful analytical tool. Its complexity requires that you be alert at all times during an analysis. The larger and more costly the analysis, the more important this is. The opportunity to induce subtle errors is large. Typically, such errors include:

- ❒ Bad engineering assumptions — *I thought it was a fixed boundary.*
- ❒ Poor modeling practices — *I never expected a stress concentration there.*
- ❒ Undetected modeling errors — Thousands of data entries are difficult to check.
- ❒ Errors in judgment — *I thought that Guyan reduction would be accurate enough.*
- ❒ Known or unknown program *bugs* - Even with extensive testing, some program errors are bound to be lurking about!

The major warning is then a simple one — **when you get answers that do not make sense, there is something wrong!** Do not blindly accept the results, but keep an open mind.

3.4.3 Selecting the Right Finite Elements

UAI/*NASTRAN* provides a comprehensive library of finite elements for modeling complex arbitrary geometries. The selection of finite elements and the construction of the finite element idealization, or model, are crucial to the success of your analysis task. These topics are discussed in detail in Chapters 4 and 5.

3.4.4 Selecting a Solution Strategy

Once you understand the solution objectives of your analysis, you need to determine the best **UAI/*NASTRAN*** solution strategy to use. This section discusses the primary strategies available for static and dynamic structural analysis.

3.4.4.1 Static Analysis

There are three approaches to the solution of statics problems: a single step solution, Substructuring, and Cyclic Symmetry. The Substructuring and Cyclic Symmetry features were briefly discussed earlier in this chapter. You will find detailed descriptions of these techniques and guidelines for choosing the best approach in Chapters 10, 19, and 22 of this manual.

3.4.4.2 Dynamic Analysis — Normal Modes

The approaches to the extraction of natural frequencies and mode shapes are the same as those used in static analysis. However, there are two additional strategies that need to be considered. The first is the eigenvalue extraction method. **UAI/*NASTRAN*** provides four basic eigenextraction methods. The first of these, called the ***Givens Method*** (which includes the Modified Given's method), is used to extract all of the eigenvalues from a matrix. The next three, the ***Inverse Power Method***, the ***Lanczos Method***, and the ***Subspace Iteration Method***, are used to extract a small number of eigenvalues in a specified frequency range. Guidelines for the selection of an eigenmethod are found in Chapter 11.

The second is the determination of a reduction strategy to reduce the problem size and significantly reduce computer expense. The reduction methods offered are Guyan Reduction, Dynamic Reduction, and Modal Synthesis.

When you use the Lanczos or Inverse Power eigenmethods, reduction methods are ***not*** recommended.

The Guyan Reduction, or static condensation, is selected by the use of **ASET** or **OMIT** Bulk Data. You must determine the degrees-of-freedom in your model that define the stiffness and mass characteristics accurately. Dynamic Reduction and Modal Synthesis, which is available with substructuring, are closely related. They involve reducing the structure not to physical degrees of freedom, but to ***generalized coordinates*** that represent the model by some modest number of its natural modes. Guidelines for selecting are found in Chapters 6, 9, 11, and 19.

3.4.4.3 Dynamic Analysis — Transient and Frequency Response

The solution of both transient and frequency response problems is available in **UAI/*NASTRAN***. There are two solution approaches for each. The first is called the ***direct approach*** wherein the equations of motion are solved explicitly. The second is called the ***modal approach***. A structure is reduced to a small subset of its modes prior to solution. This technique is similar to dynamic reduction; however, the eigenvectors used are fully orthogonalized.

The direct approach should normally be used only for very small problems, on the order of several hundred degrees of freedom, or when many nonlinear elements are being used. For any larger problem, the modal approach is usually more cost effective. More information is found in Chapters 12, 13, 14 and 19.

3.4.5 Interactive Graphics

The last three phases of the **UAI/*NASTRAN*** analysis activity, as described in Section 2.8.1, are the determining factors in both the cost and schedule of a project. In recent years software has been developed to reduce the high costs associated with the data preparation phase. Such software is generally referred to as a pre-processor. For structural analysis, pre-processors usually are able to generate the highly repetitive information needed in a model, this includes:

- ❒ Element Connection Data
- ❒ GRID Point Data
- ❒ Boundary Condition Data
- ❒ Simple Load Data

Other information such as complex constraint equations and loads, element properties, material properties and miscellaneous problem descriptors are usually not generated.

Once a solution has been obtained, other software, called post-processors, may be used to aid in the interpretation of results. Typical capabilities of post-processors include:

- ❒ Deformed and Undeformed Structural Plots
- ❒ Surface Contour Plots of Responses such as Stress and Temperature
- ❒ Histograms
- ❒ Selective Output Presentation

Currently most pre- and post-processing software runs interactively letting you perform many operations quickly. Such systems usually support high-quality graphics, including color, to make interpretation easier. Interactive software has had a significant impact on total cost and schedule and can result in substantial savings.

3.4.6 Validating Your Model and Solution Results

After you have successfully created your finite element model and executed **UAI/*NASTRAN***, you should evaluate both the model and results for correctness. **UAI/*NASTRAN*** offers a number of features to assist you in this task.

The first step in model validation is usually to plot the finite element model. To do this you may use the **UAI/*NASTRAN*** plotter or some commercially available or *home-grown* pre- and post-processing software. Displaying a number of views of the model will often detect major geometry errors.

Next, you execute **UAI/*NASTRAN***. A number of consistency checks on the input data are performed and error or warning messages are issued when problems are encountered. Once the data are correct in format, **UAI/*NASTRAN*** will begin the analysis. As the various modules in the selected Rigid Formats are executed, additional checks for errors occur. Some of these checks include detection of warped element geometry and singular matrices.

When no further errors are detected, a solution is obtained. You must still determine whether this is the solution to the problem that you intended to solve! Subsequent Chapters of this manual will describe tools which will assist you in doing this.

3.5 SPECIAL ALTERs FOR UAI/*NASTRAN* ENHANCED CAPABILITIES

UAI/*NASTRAN* also provides the user with the capability to extend the normal solution procedure, by means of additional Rigid Format ALTER packets. Table 3-6 summarizes these additional ALTER packets by reference to the Rigid Format they serve as well as the function they perform.

3.5.1 ALTERs for Output of *a-set* Data Only

UAI/*NASTRAN* Alters **ASETxx** provide the user with the capability for output of only *a-set* degrees of freedom associated data quantities. Displacement requests in case control should only include degrees of freedom in the *a-set*; all other output will be zero. Element output must be limited to those elements whose GRID point motions are completely defined by *a-set* and *s-set* displacements. Any other element data will be incorrect.

3.5.2 ALTER for Checking Finite Element Models for Grounding Forces

UAI/*NASTRAN* Alter **MODELCHK** provides you with the capability of checking your finite element model for defects which can create constraint forces to ground (i.e., grounding forces). The DMAP ALTER is utilized in Rigid Format 1, Statics, at three different levels of set membership:

- *g-set* before elimination of any degrees of freedom
- *n-set* after elimination of multipoint constraints
- *f-set* after elimination of automatic constraints and SPC's

The finite element model is subjected both to unit displacements in each of the three translational degrees of freedom, and to unit radian rotations for each of the three rotational rigid body modes. The matrix **NCHECK** provides the forces which result between grounded degrees of freedom and ground for each of the rigid body motions (called "**COLUMN 1**" through "**COL-**

Table 3-6. SUMMARY OF ALTERLIB MEMBERS

GROUP	RIGID FORMAT	MEMBER NAME	DESCRIPTION
MODEL CHECKING	1	MODELCHK	Model Check for Grounding Constraints
a-set ONLY DATA RECOVERY	3	ASET03	Normal Modes Solution
	8	ASET08	Direct Frequency Response
	9	ASET09	Direct Transient Response
	11	ASET011	Modal Frequency Response
	12	ASET012	Modal Transient Response
SENSITIVITY ANALYSIS	7	CMSENS07	Complex Eigenvalue Sensitivity Analysis
	10	CMSENS10	Modal Complex Eigenvalue Sensitivity Analysis
I-DEAS™ SYSTEM DYNAMICS ANALYSIS		SDRCSDA	ALTERLIB member **SDRCSDA** allows you to automatically create the **OUTPUT2** files needed by this program.

UMN 6" in the output). Forces are printed for values which exceed a threshold value which you set with the parameter:

PARAM	**FILTER**	***thresh***							

The default value of *thresh* is 0.05. The matrix **SUMCHK** provides the strain energy which results from each of the rigid body motions.

Values of the grounding forces and moments should be less than 0.01 and model length times 0.01, respectively, for well conditioned models. The parameter **GRIDMOM** is set by default to the location of the basic coordinate system origin. It is used to calculate the grounding moments; hence, the selection of a GRID close to the center of gravity may be better for comparison purposes if the model is far removed from the basic origin. This selection can be made by specifying the GRID point identification number, *gid*, on the Bulk Data entry:

PARAM	**GRIDMOM**	***gid***							

The usual Case Control command for single point constraints should be omitted from the model during this check or large grounding forces will necessarily exist during the *s-set* check. Also, the Parameter **AUTOSPC** should either be entered as **YES** or omitted, as the default value of the option **AUTOSING** is **YES**, in order to see the effect of automatically imposed constraints on suppressed singularities.

A **SUPORT** Bulk Data entry, specifying any GRID point identification number, must be used, and all six components must be entered for the degrees of freedom.

3.5.3 Complex Eigenvalue Sensitivities

Two ALTERLIB members, **CMSENS07** and **CMSENS10** are available to allow the computation of Complex Eigenvalue Sensitivities. The former is used with the direct solution approach, and the latter with the modal approach. When using both **CMSENS07** and **CMSENS10**, you control the sensitivity computation with two parameters:

PARAM	**LMODSNS**	***lmid***							

PARAM	**HMODSNS**	***hmid***							

These parameters specify the range of modes for which sensitivities will be computed. *lmid* is the low mode identification number, and *hmid* is the high mode identification number. To see the type of results that are computed by these ALTERs, you may use your favorite text editor to view the files in the ALTERLIB directory and read the detailed commentary.

3.5.4 SDRC I-DEAS Master Series™ System Dynamics Analysis

UAI/*NASTRAN* may also create user files for input to the SDRC System Dynamics Analysis program. ALTERLIB member **SDRCSDA** allows you to automatically create the **OUTPUT2** files needed by this program. See Chapter 31 for additional details.

This page is intentionally blank.

Chapter 4

THE FINITE ELEMENT IDEALIZATION

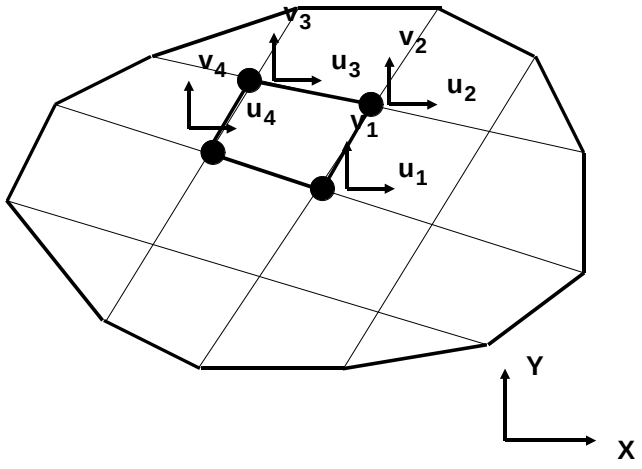
The finite element method, FEM, is an approximation technique which may be used to solve large classes of problems in mathematical physics and engineering. This chapter provides you with background information describing the way in which FEM is implemented in **UAI/*NASTRAN*** and the terminology used by the program and its user community. It also introduces you to the importance of good models in obtaining accurate solution results.

4.1 OVERVIEW

The fundamental underlying concept of the finite element method is that any continuous function over a two- or three-dimensional domain can be approximated by a finite series of piecewise continuous functions which are defined over a set of subdomains. These imaginary subdomains are separated by lines, or planes, as illustrated in Figure 4-1. These functions must be specified in such a manner that as the number of subdomains approaches infinity, the exact solution is asymptotically approached. The subdomains are called ***Finite Elements***. These elements are assumed to be connected at node points, called ***Grid Points*** in **UAI/*NASTRAN***, which are located at the intersections of their boundaries. The approximate functions may then be explicitly solved for their independent variables at each of the GRID points. In **UAI/*NASTRAN*** these variables are called ***Degrees of Freedom*** or DOF.

The actual equations solved by **UAI/*NASTRAN***, and thus the meaning of the degrees of freedom, depend on the analytical discipline which you select. For example, in structural analysis, the degrees of freedom represent the displacements, or deformations, of the structure, and in heat transfer analysis they represent the temperature. Figure 4-1, which represents a two-dimensional plane strain idealization, shows a typical element and the degrees of freedom at its GRID points. The displacements u and v represent motions in the X and Y coordinate directions.

Figure 4-1. THE FINITE ELEMENT DISCRETIZATION



4.2 GRID POINTS AND OTHER DEGREES OF FREEDOM

The first essential components of the finite element model are points in space where structural responses are computed. These GRID points not only represent the locations where solutions are obtained, but they also allow many non-structural phenomena to be modeled and coupled to the structural model.

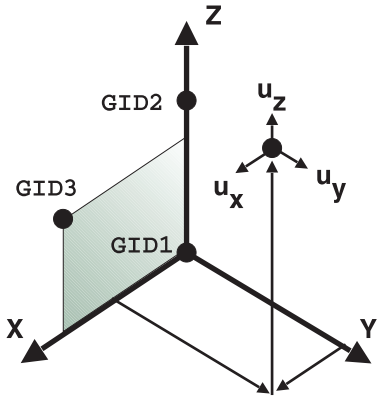
Because these phenomena may not require that the degrees of freedom be located in geometric space, other points called ***Scalar Points*** may also be defined. A GRID point is simply a point in three dimensional space at which three translational and three rotational components of motion are defined. The locations of GRID points are defined by **GRID** Bulk Data entries. A SCALAR Point is a single degree of freedom in vector space which is explicitly defined using **SPOINT** Bulk Data entries. SCALAR points may be connected with scalar elements and they may be connected to GRID points with constraint equations or scalar elements.

4.2.1 GRID Points and Coordinate Systems

GRID points are defined by specifying their coordinates in either the basic or 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. Each local system must be related directly or indirectly to the basic coordinate system. **CORD1C**, **CORD1R** and **CORD1S** Bulk Data are used to define cylindrical, rectangular and spherical local coordinate systems, respectively, by referencing the identification numbers of three GRID points in your model. **CORD2C**, **CORD2R** and **CORD2S** data are used to define local coordinate systems in terms of the coordinates of three arbitrary points. These points may be represented in a different coordinate system that you have previously defined.

Six rectangular displacement components, three translations and three rotations, are defined at each GRID point. The local coordinate systems used to define the directions of motion may be different from the local coordinate system used to locate the GRID point. Both the location coordinate system and the displacement coordinate system are specified on the **GRID** Bulk Data entry for each GRID point. The orientation of displacement components depends on the type of local coordinate system used to define the displacement components. If the local system is rectangular, the displacement system is parallel to the local system and is independent of the GRID point location as indicated in Figure 4-2. If the local system is cylindrical, the displacement components are in the radial, tangential and axial directions as indicated in Figure 4-3. If the local system is spherical, the displacement components are in the radial, meridional, and azimuthal directions as indicated in Figure 4-4. Each 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 this 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. Similarly, the symbols **R1**, **R2** and **R3** denote rotations about the three axes.

Figure 4-2. RECTANGULAR COORDINATE SYSTEM



4.2.2 Permanent Constraints

You may also specify fixed boundary conditions using the **GRID** Bulk Data entry. Any constraints specified on the **GRID** entry will be automatically used for all solutions. Because of this, it is best to use constraint Bulk Data entries for this purpose.

You may use the **GRDSET** Bulk Data entry to avoid the necessity of repeating the specification of location coordinate systems, displacement coordinate systems, and single-point constraints, when all, or many, of the **GRID** entries have the same values for these items. When any of the three items are specified on the **GRDSET** entry, these data are used to replace the corresponding blank fields on the **GRID** entry. This feature is useful in the case of problems such as space trusses where all rotational degrees of freedom are to be removed, or in the case of plane structures where either all out-of-plane or all in-plane motions may be removed.

Figure 4-3. CYLINDRICAL COORDINATE SYSTEM

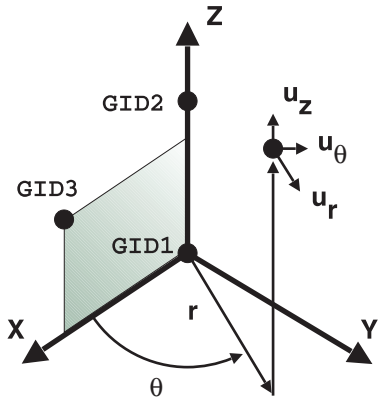
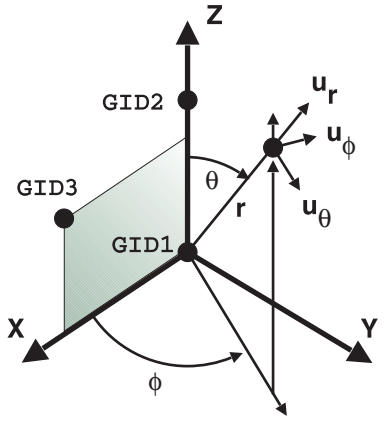


Figure 4-4. SPHERICAL COORDINATE SYSTEM



4.2.3 SCALAR Points

SCALAR points are defined either explicitly with an **SPOINT** Bulk Data entry, or implicitly 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 with **EPOINT** Bulk Data entries.

4.2.4 Applying Constraints and Loads

You may apply various kinds of constraints to the GRID points. Single-point constraints are used to specify boundary conditions, including enforced displacements. Multipoint constraints are used to specify a linear relationship among selected degrees of freedom, including the definition of infinitely rigid elements. Omitted points are used as a tool in matrix partitioning and for reducing the number of degrees of freedom used in dynamic analysis. Free-body supports are used to remove stress-free motions in static analysis and to evaluate the free-body inertia properties of the structural model. You will find detailed descriptions of these boundary conditions in Chapter 6.

Static loads may be applied directly to the structural model by concentrated loads at GRID points as well as pressure loads on surfaces. Chapter 7 describes loads in great detail.

4.2.5 Other GRID Point Properties

Some of the characteristics of the structural model are introduced as properties of GRID points, rather than as properties of structural elements. Any of the various forms of direct matrix input are considered as describing the structural model in terms of properties at GRID points. 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** Bulk Data entry is used to define direct input matrices to be added to the matrices created from finite element data. These matrices may be associated with components of GRID points, SCALAR points, or EXTRA points introduced for dynamic analysis. The **TF** Bulk Data entry is used to define transfer functions that are internally converted to direct matrix input. The **DMIGAX** Bulk Data entry is an alternate form of direct matrix input that is used for axisymmetric models.

Mass properties may be input as properties of GRID points by using the concentrated mass element. The **CONM1** Bulk Data entry is used to define a 6 x 6 matrix of mass coefficients at a GRID point in any selected coordinate system. The **CONM2** entry is used to define a concentrated mass at a 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.

Thermal fields are defined by specifying temperatures at GRID points. The **TEMP** Bulk Data entry is used to specify the temperature at GRID points both for thermal loading and temperature dependent material properties. The **TEMPD** Bulk data entry is used to specify a default temperature, in order to avoid duplicating data when the temperature is uniform over a large portion of the structure. Similarly, the **TEMPAX** entry is used for axisymmetric problems.

4.2.6 GRID Point Sequencing

The best decomposition and equation solution times are obtained if GRID points are sequenced in such a manner as to create matrices with relatively small bandwidths. While you may select any arbitrary external numbering system for your GRID points, **UAI/*NASTRAN*** will resequence them internally for maximum efficiency. The resequencing operation may be automatically performed in **UAI/*NASTRAN*** depending on your Preference File. In all cases, you may explicitly deselect it with the **SEQUENCE** Executive Control command or you use **SEQGP** Bulk Data entries to define your own sequence.

When the high-performance sparse solver is selected, it performs a special resequencing. While it is not necessary to perform resequencing before the solution step, it is still advantageous if your model contains MPC equations or Rigid Elements, or if you are performing reductions.

Three resequencing algorithms, the ***Gibbs-Poole-Stockmeyer*** (GPS), the ***Cuthill-McKee*** (CM) and the ***Minimum Wavefront*** (MWF), have been incorporated into **UAI/*NASTRAN***. A summary table, shown in Table 4-1, provides you with pertinent information relative to the performance of the sequencer algorithm. Also included in the table are a summary of the number and type of each finite element in the model, the number of GRID points in the model, and the maximum number of GRID points, **MAXIMUM NODAL DEGREE**, which are connected to one another. Note that the GPS method is the most economical and should be used with the sparse solver.

4.2.7 Standard GRID Point Output

There are two major categories of printed output which may be generated by a **UAI/*NASTRAN*** analysis. The first category includes finite element quantities such as element stress or force. The other major category includes solution results related to the degrees of freedom of the

Table 4-1. SAMPLE UAI/NASTRAN SEQUENCER RESULTS

```
***   S U M M A R Y   O F   A U T O M A T I C   R E S E Q U E N C I N G   ***

                METHOD SELECTED                          GPS
                CRITERION                        RMS WAVEFRONT


                                              BEFORE     AFTER
                BANDWIDTH                         68         9
                PROFILE                         1046       641
                MAXIMUM WAVEFRONT                 18         9
                AVERAGE WAVEFRONT             13.410     8.218
                RMS WAVEFRONT                 13.915     8.389


                NUMBER OF GRID POINTS                       78
                MAXIMUM NODAL DEGREE                         8

                ELEMENTS PROCESSED
                                              QUAD4         60
                                              TRIA3          6

                TOTAL ELEMENTS                              66
```

4

model — GRID points, SCALAR points and EXTRA points. The GRID point related output options are summarized in this section. In the remainder of this section, the term GRID point will be used, but you should remember that the reference also applies to SCALAR and EXTRA points unless specifically stated otherwise.

UAI/*NASTRAN* does not automatically print analytical results such as GRID point displacements. Automatic output is not desired for a program such as **UAI/*NASTRAN*** where very large models are frequently processed and automatic output would generate excessive computer print. Therefore, you must explicitly request all output. Most printed output is requested using Case Control commands. Three basic types of output are available. The first type includes physical quantities at degrees of freedom or GRID points. The second type includes that which is available for some dynamic analyses. This output is called ***Solution Set*** output. If the direct dynamics solution approach is used, the degrees of freedom are GRID points remaining after constraints and matrix reductions have been applied. If the modal solution approach is used, the degrees of freedom consist of generalized coordinates describing the motion of the structural modes and any EXTRA points. The third type is special output that either summarizes various quantities at GRID points, or extrapolates element solution results to the GRID points defining the element. Table 4-2 summarizes all of the available output types.

Most GRID point solution results are selected with Case Control commands of the form:

ACCELERATION $\left[\left(\left[\left\{\begin{matrix}\textbf{PRINT}\\ \text{NOPRINT}\end{matrix}\right\},\right]\text{[PUNCH ,]}\left[\left\{\begin{matrix}\textbf{RECT}\\ \text{POLAR}\end{matrix}\right\}\right]\right)\right] = \left\{\begin{matrix}\text{ALL}\\ \textit{sid}\\ \textbf{NONE}\end{matrix}\right\}$

You may select a set of GRID points for which output is requested by using the **SET** Case Control command to define the set with an identification number of *sid*. The special keyword **ALL** allows you to request output for all of the GRID points in your model. You may then request that the results be **PRINT**ed in the output file, **PUNCH**ed to a file, or both. Note that if you select the **PUNCH** option, you may use the Executive Control command **ASSIGN** to define a specific file name, or you may use the default name defined by your Preference File. If you select the file, be sure to specify **USE=PUNCH**. You use the **NOPRINT** option to request the computation of results which will later be used by a post-processor. In this case, you generally

Table 4-2. GRID POINT OUTPUT QUANTITIES

FORMAT	TYPE	COMMAND	DESCRIPTION
STANDARD	PHYSICAL SOLUTION QUANTITIES	ACCELERATION	Physical DOF accelerations
		DISPLACEMENT	Physical DOF displacements
		GPKE	Kinetic Energy
		MPCFORCE	Multipoint Forces of Constraint
		OLOAD	External loads resolved at GRID points
		PRESSURE	Physical DOF pressures (Hydroelastic or Acoustic)
		SPCFORCE	Forces of constraint
		THERMAL	Temperatures at GRID points (Heat Transfer)
		VECTOR	Synonym for **DISPLACEMENT**
		VELOCITY	Physical DOF velocities
	SOLUTION SET QUANTITIES	SACCELERATION	Solution DOF accelerations
		SDISPLACEMENT	Solution DOF displacements
		SVELOCITY	Solution DOF velocities
		NLLOAD	Nonlinear component of transient loads
SPECIAL		ELOFORCE	Element-oriented GRID point forces
		GPFORCE	Force balance at GRID points
		GPSTRAIN	Strains at GRID points
		GPSTRESS	Stresses at GRID points

must use the **UAI/*NASTRAN* POST** module, described in Chapter 31 of this manual, to create files which contain the solution results. Finally, the format of the output may be **RECT**angular or **POLAR**. The **POLAR** format, described in detail in Section 4.2.9, is used for dynamic response analyses.

Most output for GRID points includes one or more values for each degree of freedom at the GRID. In this format, called *standard* in Table 4-2, GRID point output is printed with column headings labeled: **POINT ID.**, **TYPE**, **T1**, **T2**, **T3**, **R1**, **R2** and **R3**. The values given for **T1**, **T2** and **T3** correspond to the three translations, and **R1**, **R2**, and **R3** correspond to the three rotations of that GRID point. An example of typical GRID point solution results is shown in Table 4-3. The orientation of these displacement components is defined by the output displacement coordinate system, which is either the **UAI/*NASTRAN*** basic rectangular system or any other coordinate system that you specify for that GRID point. These coordinates were defined in Figures 4-2 through 4-4. The rotations are given in units of radians and are defined as vectors with the same directions as the translations.

GRID points, SCALAR points, EXTRA points, and modal coordinates may be intermingled in the output. The points with only one degree of freedom are printed in a special format. The **POINT ID.** number is printed only for the first point in a sequential set of up to six such points. The results for the second and succeeding points are printed on the same line with the first. If more than six points occur in a numerical sequence, the seventh **POINT ID.** is printed to identify the second line of printout, and so on for every succeeding set of six points.

Table 4-3. TYPICAL GRID POINT SOLUTION RESULTS

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	GRID	0.0	0.0	0.0	0.0	0.0	0.0
2	GRID	0.0	0.0	0.0	0.0	0.0	0.0
3	GRID	0.0	0.0	0.0	0.0	0.0	0.0
4	GRID	0.0	0.0	0.0	0.0	0.0	0.0
5	GRID	0.0	0.0	0.0	0.0	0.0	0.0
11	GRID	0.0	0.0	0.0	0.0	0.0	0.0
12	GRID	0.0	0.0	-1.766039E-01	-2.118679E-01	1.210239E-01	0.0
13	GRID	0.0	0.0	-3.201588E-01	-4.030518E-01	0.0	0.0
14	GRID	0.0	0.0	-1.766039E-01	-2.118679E-01	-1.210239E-01	0.0
15	GRID	0.0	0.0	0.0	0.0	0.0	0.0
21	GRID	0.0	0.0	0.0	0.0	0.0	0.0
22	GRID	0.0	0.0	-3.215491E-01	-3.995105E-16	2.259518E-01	0.0
23	GRID	0.0	0.0	-5.938403E-01	-4.996004E-16	-2.289835E-16	0.0
24	GRID	0.0	0.0	-3.215491E-01	-1.526557E-16	-2.259518E-01	0.0
25	GRID	0.0	0.0	0.0	0.0	0.0	0.0
31	GRID	0.0	0.0	0.0	0.0	0.0	0.0
32	GRID	0.0	0.0	-1.766039E-01	2.118679E-01	1.210239E-01	0.0
33	GRID	0.0	0.0	-3.201588E-01	4.030518E-01	-4.163336E-17	0.0
34	GRID	0.0	0.0	-1.766039E-01	2.118679E-01	-1.210239E-01	0.0
35	GRID	0.0	0.0	0.0	0.0	0.0	0.0
41	GRID	0.0	0.0	0.0	0.0	0.0	0.0
42	GRID	0.0	0.0	0.0	0.0	0.0	0.0
43	GRID	0.0	0.0	0.0	0.0	0.0	0.0
44	GRID	0.0	0.0	0.0	0.0	0.0	0.0
45	GRID	0.0	0.0	0.0	0.0	0.0	0.0

4

4.2.8 Special GRID Point Output

As introduced above, there are two types of special output for GRID point solution results. The first two provide a balance of forces from all sources at each GRID point, and the second provides extrapolated element data. These are described in the following sections.

4.2.8.1 GRID Point Force Quantities

Grid Point Forces may be requested using the **GPFORCE** and **ELOFORCE** Commands, to produce GRID Point ***Force Balance*** and ***Element Internal Force*** output, respectively. The primary distinction between the two forms of GRID Point Force data is that **GPFORCE** presents output in the form of loads applied to the GRID points, while **ELOFORCE** presents the loads as applied to the elements. You may request force results with the Case Control commands:

GPFORCE $\left[\left(\left[\left\{ \begin{matrix} \textbf{PRINT} \\ \text{NOPRINT} \end{matrix} \right\}, \right] [\ \text{PUNCH}\ ] \right) \right] = \left\{ \begin{matrix} \text{ALL} \\ \textit{sid} \\ \text{NONE} \end{matrix} \right\}$

ELOFORCE $\left[\left(\left[\left\{ \begin{matrix} \textbf{PRINT} \\ \text{NOPRINT} \end{matrix} \right\}, \right] [\ \text{PUNCH}\ ] \left[\left\{ \begin{matrix} \text{SUMMATION} \\ \text{NOSUMMATION} \end{matrix} \right\}, \right] \right) \right] = \left\{ \begin{matrix} \text{ALL} \\ \textit{sid} \\ \text{NONE} \end{matrix} \right\}$

In both cases, you may select a set of GRID points for which output is requested by using the **SET** Case Control command to define the set with an identification number of *sid*. The special keyword **ALL** allows you to request forces for all of the GRID points in your model. You may then request that the results be **PRINT**ed in the output file, **PUNCH**ed to a file, or both. Note that if you select the **PUNCH** option, you may use the Executive Control command **ASSIGN** to define a specific file name, or you may use the default name defined by your Preference File. If you select the file, be sure to specify **USE=PUNCH**. You use the **NOPRINT**

Table 4-4. GRID POINT FORCE BALANCE RESULTS

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
41		APP-LOAD	-2.0445E+02	0.0	0.0	0.0	0.0	0.0
41		F-OF-SPC	4.3053E+03	-6.051844E+03	0.0	0.0	0.0	-9.1972E+03
41		F-OF-MPC	-2.4390E+03	9.711316E+03	4.4482E+03	3.2049E+03	-8.4319E+03	1.0762E+04
41	13	TRIA3	-1.7683E+03	6.866295E+03	-4.4434E+03	-7.3595E+03	1.7767E+04	1.1544E+04
41	14	TRIA3	1.5300E+03	6.337786E+02	7.7787E+03	6.2296E+03	-1.5039E+04	-2.1672E+04
41	5	TRIA6	-3.2942E+02	-1.011722E+04	-5.5917E+03	-6.2482E+03	1.7175E+04	-2.2374E+04
41	6	TRIA6	-1.0941E+03	-1.042330E+03	-2.1917E+03	4.1732E+03	-1.1471E+04	3.0936E+04
41		*TOTALS*	0.0	1.220703E-04	-7.3242E-04	0.0	0.0	0.0

option to request the computation of results which will later be used by a post-processor. In this case, you can use the **POST** module, described in Chapter 31 of this manual, to create files which contain the solution results.

The results of the **GPFORCE** command are shown in Table 4-4. This output form presents an equilibrium check at each GRID point, with the applied loads, reactions and the load contribution from each connected element expressed in the global coordinate system of the reference GRID point. This local equilibrium check is a valuable tool as part of a model checkout procedure, and the GRID oriented loads are in a convenient format for producing free-body diagrams or new applied loads in a general three dimensional space.

The results of the **ELOFORCE** command are shown in Table 4-5a. This output presents GRID point forces as internal loads applied to the connected elements, with the component directions oriented with the element edges. These element oriented GRID loads are in a convenient form for producing free-body diagrams on two-dimensional or curved subcomponents of a larger three-dimensional model. The result of the additional output option, **ELOFORCE(SUM)**, is shown in Table 4-5b. This output is the result of ***summing the loads along adjacent element edges***.

Note the convention for **ELOFORCE** is positive action toward the **LOAD-POINT** as applied to the elements.

Table 4-5. ELEMENT-ORIENTED GRID POINT FORCES

a. Element Oriented GRID Point Forces

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

LOAD POINT-ID	ELEMENT ID	ELEMENT TYPE	REFERENCE POINT-1	FORCE-1	MOMENT-1	REFERENCE POINT-2	FORCE-2	MOMENT-2	KICK-LOAD
12	1	QUAD4	11	-4.22825E-01	0.00000E+00	2	2.50000E-01	0.00000E+00	0.00000E+00
12	2	QUAD4	2	-2.50000E-01	0.00000E+00	13	-8.28099E-01	0.00000E+00	0.00000E+00
12	31	ROD	11	-2.06895E+00	0.00000E+00				
12	32	ROD	13	-1.66769E+00	0.00000E+00				

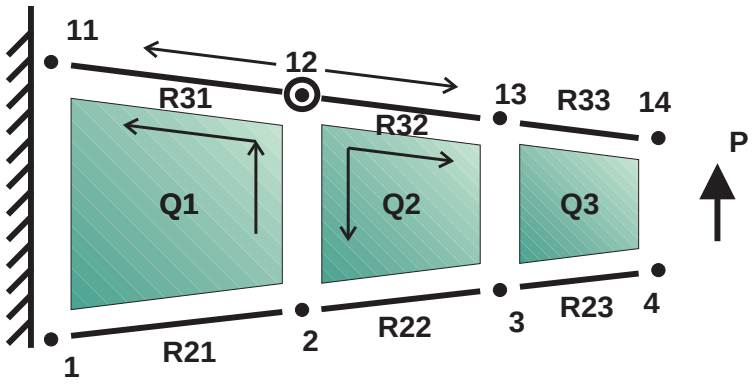
b. Summation of GRID Point Forces

SUBCASE 1

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

POINT-ID	ORIENT-ID	TENSION =(+)
12	2	-1.788E-07
12	11	-2.492E+00
12	13	-2.496E+00

Figure 4-5. ELEMENT-ORIENTED GRID POINT FORCES



To illustrate the interpretation of these forces, consider Figure 4-5, which illustrates a cantilevered beam modeled with QUAD4 elements for the web and ROD elements for the flanges. The left end has been fixed and a positive (upward) load applied to the right end, which puts the upper flange in compression. The command **ELOFORCE(SUM)** for GRID point 12 produced the results shown in Tables 4-5a and 4-5b.

Remembering the beam is in an up-bending condition, a comparison of the **ELOFORCE** results can be made. There are four elements connected to GRID point 12 giving four results entries as shown in Table 4-5a. Instead of loads in the global coordinate system, the loads reported are ***along the element edges*** positive toward the GRID point of interest. In this case, the load on the ROD flanges are both negative, or acting away from point 12, meaning that both ROD elements are in compression. The QUAD4 webs are also carrying a portion of the axial load, and this is shown as **FORCE-1** on element 1 and **FORCE-2** on element 2. These are acting in the same directions as the ROD forces. The remaining loads on the QUAD4 elements are from the shear load (0.25 lb) on the web. This is transferred from the element 2 to element 1 and can be seen by the equal and opposite forces from GRID 2 to 12.

The **ELOFORCE(SUM)** output for this problem is shown in Table 4-5b. Note that there is no external shear load at GRID point 12. If there was an external shear load, this would have shown as a non-zero load at GRID point 12 from **ORIENT-ID** 2. It can also be seen that the only remaining forces are the compression load in the upper fiber.

4.2.8.2 GRID Point Values of Element Solution Results

There may be cases in which you want element solution results, in particular stresses and strains, extrapolated to the GRID points of your model. This may be done with the Case Control commands:

GPSTRAIN
GPSTRESS [(([{ **PRINT** / NOPRINT } ,] [PUNCH]))] = { ALL / ALLFIELDS / GRIDSET *sid* / FIELDSET *fid* / **NONE** }

The keyword **ALL** allows you to request strains, or stresses, for all of the elements in your model. For irregular structures that may have stress discontinuities, such as impinging corners, you may have to define a ***Stress Field***. A stress field is simply a set of elements that form a continuous stress surface. You define these fields with **GPFIELD** Bulk Data entries. If you

define one or more fields, then you can request the calculations by selecting the **ALLFIELDS** option. Alternately, you may select a set of GRID points that defines a field — all elements connected to any of the GRID points in the set define a field — for which output is requested by using the **SET** Case Control command to define the set with an identification number of *sid*. Finally, you may select a subset of the stress field that you have defined by using the **FIELDSET** *fid* option. In this case *fid* again references an integer set defined by a **SET** Case Control command.

You may then request that the results be **PRINT**ed in the output file, **PUNCH**ed to a file, or both. Note that if you select the **PUNCH** option, you may use the Executive Control command **ASSIGN** to define a specific file name, or you may use the default name defined by your Preference File. If you select the file, be sure to specify **USE=PUNCH**. You use the **NOPRINT** option to request the computation of results which will later be used by a post-processor. In this case, you generally must use the **UAI/*NASTRAN* POST** module, described in Chapter 31 of this manual, to create files which contain the solution results. Finally, the format of the output may be **RECT**angular or **POLAR**. The **POLAR** format, described in the following section, is used for dynamic response analyses.

The **GPFIELD** Bulk Data entry defines a stress or strain field by listing a set of element identification numbers:

GPFIELD	**FID**	**TYPE**	**CIDOUT**	**AXIS**	**NORM**			**TOLER**	**-cont-**
-cont-	***ELEMENT ID LIST***								**-cont-**

You must specify a field **TYPE** of either **SHELL** or **SOLID**, because the stresses for shell elements and solid elements are considerably different and cannot be mixed. All stress values will be computed either in the Basic coordinate system, or in a system, **CIDOUT**, that you select. The fields **AXIS** and **NORM** allow you to specifiy the x-axis and z-axis of system **CIDOUT**. Finally, you may specify a **TOLER**ance for **SHELL** fields. This tolerance defines the angle beyond which two elements are considered to be in different fields.

Table 4-6. GRID POINT STRESS RESULTS

G R I D P O I N T S T R E S S E S I N 2 - D I M E N S I O N A L S T R E S S F I E L D S
F I E L D I D = 20

GRID-ID		-STRESSES OF EXTREME FIBERS IN OUTPUT C.S.- NORMAL-X	NORMAL-Y	SHEAR-XY	- PRINCIPAL STRESSES (ZERO SHEAR) - ANGLE	MAJOR	MINOR	MAX SHEAR	VON MISES
1	BOT	3.9479E+07	2.6543E+08	-2.2072E+07	-84.473	2.6757E+08	3.7343E+07	1.1511E+08	2.5099E+08
	TOP	3.9479E+07	2.6543E+08	-2.2072E+07	-84.473	2.6757E+08	3.7343E+07	1.1511E+08	2.5099E+08
2	BOT	4.7309E+06	1.5572E+08	-1.2029E+07	-85.473	1.5667E+08	3.7785E+06	7.6449E+07	1.5482E+08
	TOP	4.7309E+06	1.5572E+08	-1.2029E+07	-85.473	1.5667E+08	3.7785E+06	7.6449E+07	1.5482E+08
3	BOT	4.8760E+06	1.4430E+08	1.6024E+07	83.527	1.4612E+08	3.0580E+06	7.1532E+07	1.4461E+08
	TOP	4.8760E+06	1.4430E+08	1.6024E+07	83.527	1.4612E+08	3.0580E+06	7.1532E+07	1.4461E+08
4	BOT	3.5738E+07	2.4186E+08	1.7156E+07	85.274	2.4327E+08	3.4320E+07	1.0447E+08	2.2806E+08
	TOP	3.5738E+07	2.4186E+08	1.7156E+07	85.274	2.4327E+08	3.4320E+07	1.0447E+08	2.2806E+08
6	BOT	2.6626E+07	1.7775E+08	2.9925E+07	79.198	1.8346E+08	2.0916E+07	8.1274E+07	1.7395E+08
	TOP	2.6626E+07	1.7775E+08	2.9925E+07	79.198	1.8346E+08	2.0916E+07	8.1274E+07	1.7395E+08
7	BOT	5.7620E+06	1.1182E+08	2.8110E+07	76.036	1.1881E+08	-1.2279E+06	6.0019E+07	1.1942E+08
	TOP	5.7620E+06	1.1182E+08	2.8110E+07	76.036	1.1881E+08	-1.2279E+06	6.0019E+07	1.1942E+08
9	BOT	1.3979E+07	8.9498E+07	3.4824E+07	68.658	1.0310E+08	3.7170E+05	5.1367E+07	1.0292E+08
	TOP	1.3979E+07	8.9498E+07	3.4824E+07	68.658	1.0310E+08	3.7170E+05	5.1367E+07	1.0292E+08
10	BOT	1.1613E+06	4.1502E+05	3.0427E+07	44.649	3.1218E+07	-2.9641E+07	3.0429E+07	5.2711E+07
	TOP	1.1613E+06	4.1502E+05	3.0427E+07	44.649	3.1218E+07	-2.9641E+07	3.0429E+07	5.2711E+07
11	BOT	-7.8905E+06	-6.5155E+07	1.7720E+07	15.877	-2.8505E+06	-7.0195E+07	3.3672E+07	6.8814E+07
	TOP	-7.8905E+06	-6.5155E+07	1.7720E+07	15.877	-2.8505E+06	-7.0195E+07	3.3672E+07	6.8814E+07
12	BOT	-1.1344E+07	-8.9365E+07	-1.8566E+06	-1.362	-1.1300E+07	-8.9410E+07	3.9054E+07	8.4329E+07
	TOP	-1.1344E+07	-8.9365E+07	-1.8566E+06	-1.362	-1.1300E+07	-8.9410E+07	3.9054E+07	8.4329E+07
14	BOT	6.9422E+06	6.7212E+07	3.2575E+07	66.386	8.1453E+07	-7.2989E+06	4.4376E+07	8.5337E+07
	TOP	6.9422E+06	6.7212E+07	3.2575E+07	66.386	8.1453E+07	-7.2989E+06	4.4376E+07	8.5337E+07
15	BOT	8.1875E+06	2.2489E+07	2.8548E+07	52.031	4.4768E+07	-1.4091E+07	2.9430E+07	5.3232E+07
	TOP	8.1875E+06	2.2489E+07	2.8548E+07	52.031	4.4768E+07	-1.4091E+07	2.9430E+07	5.3232E+07
16	BOT	9.3098E+06	-1.0887E+07	1.6603E+07	29.345	1.8644E+07	-2.0222E+07	1.9433E+07	3.3668E+07
	TOP	9.3098E+06	-1.0887E+07	1.6603E+07	29.345	1.8644E+07	-2.0222E+07	1.9433E+07	3.3668E+07
17	BOT	9.6729E+06	-2.3078E+07	4.2082E+06	7.206	1.0205E+07	-2.3610E+07	1.6907E+07	3.0042E+07
	TOP	9.6729E+06	-2.3078E+07	4.2082E+06	7.206	1.0205E+07	-2.3610E+07	1.6907E+07	3.0042E+07

4.2.9 Complex Results

Complex number results occur when performing frequency response or complex eigenvalue analyses. There are two available formats for recovering complex solution results. The first is called **RECTANGULAR** and the second **POLAR**. The **RECTANGULAR** option represents the value of a complex number, *P*, using the real and imaginary parts:

$$P = a + bi$$

while the **POLAR** format represents the value as a magnitude, *r*, and the corresponding phase angle, θ, defined by:

$$r = (a^2 + b^2)^{1/2}$$

$$\theta = \tan^{-1}\left(\frac{b}{a}\right) \quad \text{where } 0° \le \theta \le 360°$$

The descriptors for **RECTANGULAR** are called Real/Imaginary and for **POLAR** are called Magnitude/Phase.

4.2.10 Output Sorting

Most of the **UAI/*NASTRAN*** solution results may be printed in either of two collating orders, or sorts. These are called **SORT1** and **SORT2**. All output for your **UAI/*NASTRAN*** execution must be in one of these sort orders. You select the sort with the command:

```
SORT = { SORT1 }
       { SORT2 }
```

Note that **SORT1** is used if you do not select a sort option except in transient analysis where the default sort order is **SORT2**. Table 4-7 illustrates the manner in which results are presented for the various Rigid Formats and **SORTi** options.

When using the standard Rigid Formats, any request for **SORT2** in Case Control will cause all output to be in that sort. When performing Analysis or Design with the Multidisciplinary solution sequence, **SOL MULTI**, the sorting of data depends on the discipline. For all disciplines except Statics, the sort request may vary from case to case. In the case of Statics analysis all consecutive cases are treated as a single group for data sorting purposes. As a result, any request for **SORT2** will cause the entire group to be processed in **SORT2**. Note that, with **SOL MULTI**, the Normal Modes results may also be in **SORT2**. Finally, note that for more general sorting and filtering operations, the optional eSHELL program may be used to query a **UAI/*NASTRAN*** ARCHIVE database.

Table 4-7. OUTPUT SORT ORDER FOR RIGID FORMATS

RIGID FORMAT	SORT1	SORT2
STATICS (1,2)	SUBCASE 1 ALL GRID RESULTS ALL ELEMENT RESULTS SUBCASE 2 ALL GRID RESULTS ALL ELEMENT RESULTS ...	GRID 1 ALL SUBCASE RESULTS GRID 2 ALL SUBCASE RESULTS ... ELEMENT 1 ALL SUBCASE RESULTS ELEMENT 2 ALL SUBCASE RESULTS ...
NORMAL MODES (3)	MODE 1 ALL GRID RESULTS ALL ELEMENT RESULTS MODE 2 ALL GRID RESULTS ALL ELEMENT RESULTS ...	**SORT2** output is not available.
FREQUENCY RESPONSE (8,11)	FREQUENCY 1 ALL GRID RESULTS ALL ELEMENT RESULTS FREQUENCY 2 ALL GRID RESULTS ALL ELEMENT RESULTS ...	GRID 1 ALL FREQUENCIES GRID 2 ALL FREQUENCIES ... ELEMENT 1 ALL FREQUENCIES ELEMENT 2 ALL FREQUENCIES ...
TRANSIENT RESPONSE (9,12)	TIME 1 ALL GRID RESULTS ALL ELEMENT RESULTS TIME 2 ALL GRID RESULTS ALL ELEMENT RESULTS ...	GRID 1 ALL TIME POINTS GRID 2 ALL TIME POINTS ... ELEMENT 1 ALL TIME POINTS ELEMENT 2 ALL TIME POINTS ...

4.3 FINITE ELEMENT CONVERGENCE

As introduced in Section 4-1, the more refined a model, the closer it approximates the exact continuum solution. The number of elements and their distribution is ideally determined by the physics of the problem. For example, in static stress analysis, the load paths and regions of stress concentration must be considered in defining the mesh. In dynamics, care must be taken to model the mass characteristics of the structure accurately. To understand the importance of the idealization, consider the two principal types of convergence. The first is called ***h-convergence*** and the second, ***p-convergence***. These are briefly described in the following sections.

4.3.1 h-Convergence

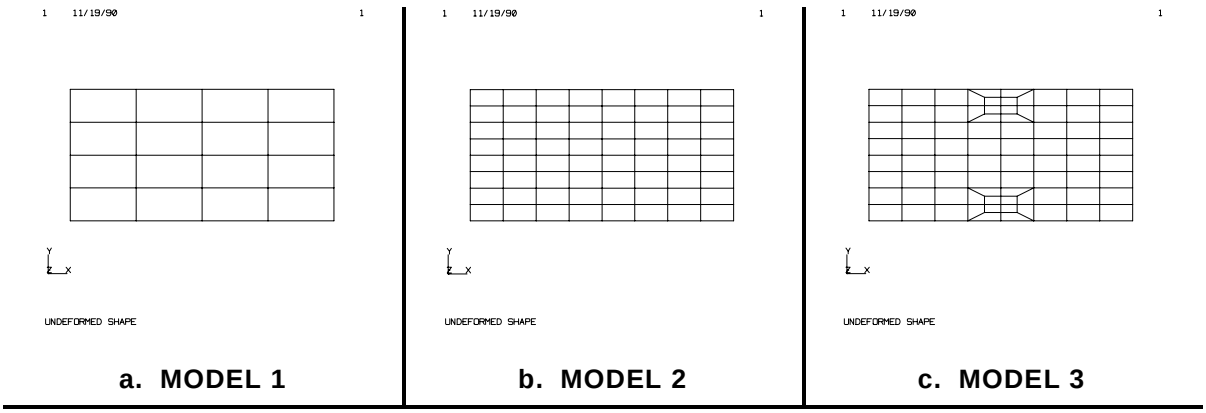
The convergence characteristics of finite elements are often measured by performing a baseline analysis with a mesh that is sequentially refined by reducing the size of the elements. This is referred to as ***h-refinement***. For example, consider a flat rectangular plate which is fixed along all of its edges and subjected to a uniform pressure load. The maximum deformation occurs at the center of the plate and the maximum stress at the midpoint of the long edge. Figure 4-6 shows three successive h-refinements of the plate ranging from 45 DOF for Model 1 to 325 DOF for Model 3. Model 2 is simply created by subdividing the elements of Model 1. Model 3 uses ***Local Refinement*** to improve the solution accuracy near the point of known stress concentration.

The ratio of the center displacement and maximum stress to the exact solution is shown in Figure 4-7. Note that the displacements rapidly approach the exact solution while the stresses converge more slowly. This is a characteristic of the displacement based finite elements used in all commercial FEA systems.

4.3.2 p-Convergence

A second type of refinement is called ***p-refinement***. In this case, the order of the finite elements is increased while their size remains constant. **UAI/*NASTRAN*** has both linear and quadratic plate and solid elements. Model 1 was executed using QUAD4 linear elements, and then a second model, called Model 1Q was created by adding new GRID points and replacing each QUAD4 element with a QUAD8 quadratic element. Figure 4-8 shows the rapid improvement in both the displacement and stress values.

Figure 4-6. MODELS FOR h-CONVERGENCE



a. MODEL 1 **b. MODEL 2** **c. MODEL 3**

Figure 4-7. RATE OF CONVERGENCE FOR h-REFINEMENT

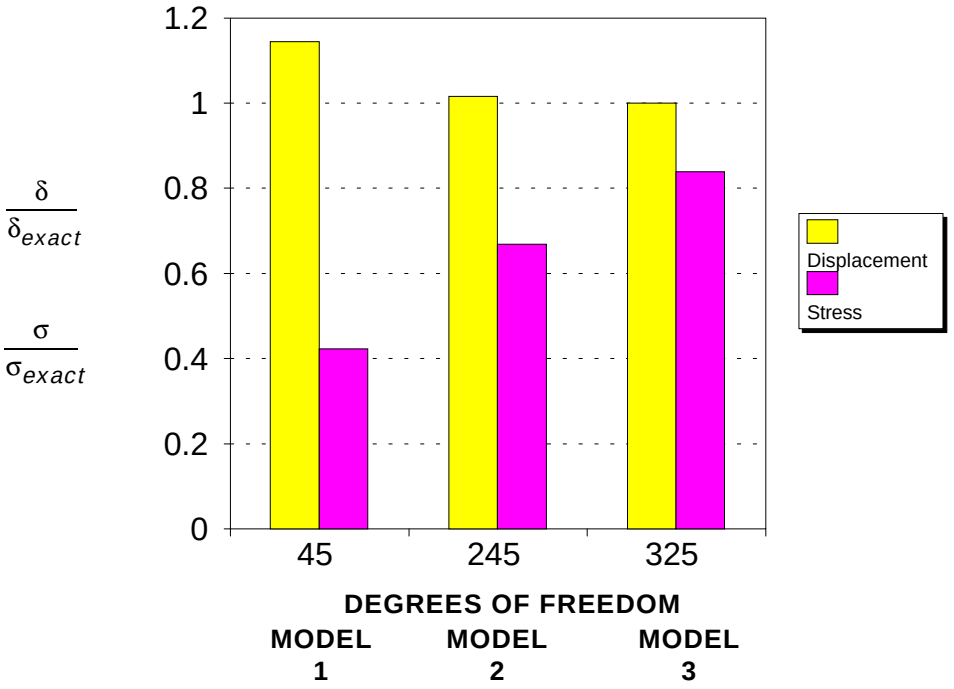
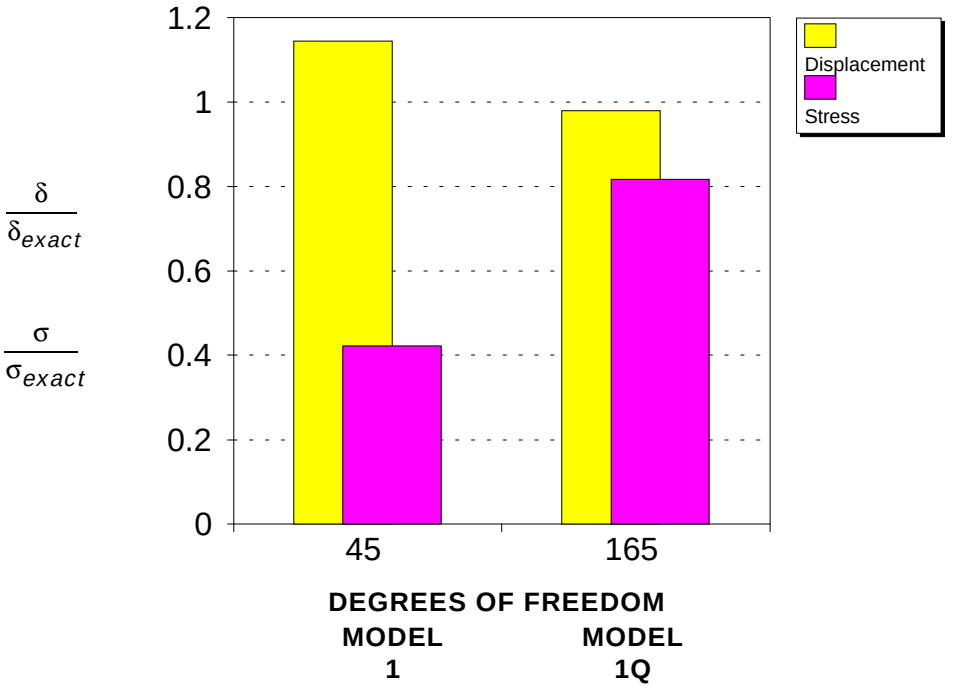


Figure 4-8. RATE OF CONVERGENCE FOR p-REFINEMENT



4.3.3 Relative Cost

The cost of performing a finite element analysis is a function primarily of the number of degrees of freedom in the model. This is natural since the most computationally intense portion of the analysis is the solution of the equations of motion. For Static Analysis, the equations of motion take the form of simultaneous linear equations. For Normal Modes Analysis, an eigenextraction is required, and transient integration is required in the case of dynamic response analyses. There is also a somewhat higher cost for quadratic elements because of their added complexity.

These observations indicate that you must make a trade-off between the accuracy required and the available analysis budget when creating your finite element model. This trade-off can only be determined by the requirements defined by your organization and your experience performing previous analyses.

4.3.4 Element Distortion

The mathematical formulation of a finite element is based on a particular geometric shape and an assumed displacement field. The functions used are designed to be well behaved when an element has a *regular* shape. They are also designed to behave reasonably well when distorted from the regular shape. However, as a practical matter, computational effects, such as round-off error and numerical integration can cause inaccuracies to develop under severe distortion. Examples of distortions that can occur include:

- Bad Aspect Ratio
- Skew and Taper
- Warping
- Curvature
- Midside Node Displacement

As a result, you will obtain the best possible results if your elements are regular and uniformly shaped. This is often the case if you are using a graphics pre-processor to create your models.

4.3.5 Impact on Multidisciplinary Design Optimization

When performing Multidisciplinary Design Optimization, described in Chapter 25, especially for designs subject to large numbers of stress and strain constraints, the accuracy of the finite element solution results may be critical. Errors in the peak stress calculations may dramatically skew the final optimum design. It is therefore recommended the Mesh Error Estimates, described in Chapter 27, be performed on a baseline model prior to performing optimization.

Similarly, if you have redesigned your geometry, you may confirm that the final mesh is accurate by performing error estimates again on the final configuration.

Chapter 5

THE FINITE ELEMENT LIBRARY

The **UAI/*NASTRAN*** finite element library contains a wide variety of elements for modeling structures of arbitrary geometry. Each element is defined by its node point connectivities, its individual physical properties, and its material description. As you saw in Chapter 4, the selection of the proper finite elements is crucial when performing structural analysis. If the appropriate elements are not selected, then inadequate representation of structural behavior may occur.

5.1 INPUT DATA REQUIREMENTS — GENERAL

Each finite element is defined by a Bulk Data entry that specifies its geometry in terms of GRID points. These entries begin with the prefix **C**, for ***Connectivity***, and they are followed by the element name, such as **CBAR** or **CROD**. An element is assigned an integer identification number which must be unique among all elements.

Similarly, the engineering properties of an element, such as the thickness of a plate or cross sectional area of a beam, are defined by Bulk Data entries prefixed with a the letter **P**, for ***Property***.

In general, an element geometric coordinate system is defined for each element based on the order of the GRID points on the connection entry. The coordinate system for each element is defined in the following sections of this Chapter. The element coordinate system is important if you are to properly interpret element stress and force results. Modeling flexibility is enhanced by the wide variety of material properties that are available. Many elements allow material coordinate systems that differ from the element geometric system. This is specified either on the connection or property data. The materials themselves are fully described in Chapter 8.

Some elements may also allow additional information requesting stress results in an arbitrary coordinate system or the calculation of margins-of-safety. These are also described for each element.

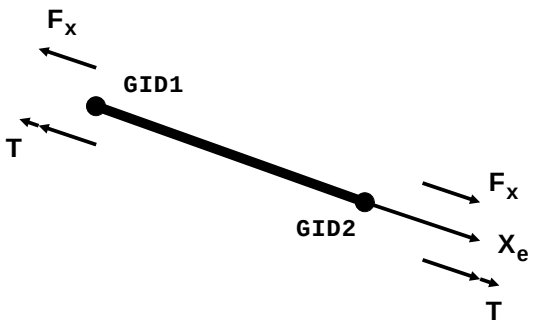
5.2 ONE-DIMENSIONAL ELEMENTS

There are five one-dimensional elements in the **UAI/*NASTRAN*** element library, these allow you to model structural members which range from simple axial force members to complex beams with varying cross-sections. These are described in this section.

5.2.1 The ROD and TUBE Elements

Two one-dimensional elements are available for modeling straight structural members which have only extensional and torsional stiffness. The first of these is the ROD element and the second is the TUBE element. The internal loads are constant for these elements. The axial material property may be dependent on strain allowing you to perform Nonlinear Material Analyses. The element geometry, coordinate system, and the sign convention for element stresses and internal forces are shown in Figure 5-1. The element forces are those forces applied by the GRID points to the element as shown. The ROD element is defined using a **CROD** Bulk Data entry and the TUBE element is defined using a **CTUBE** Bulk Data entry. For both elements, you assign the element an identification number, **EID**, and reference a property entry by specifying its identification number. You then enter the two GRID point identification numbers, **GID1** and **GID2**, which define the ends of the element. The element properties are then defined using **PROD** and **PTUBE** Bulk Data entries. In the case of the ROD element, input data includes the cross-sectional area and torsional characteristics. The TUBE element is, by definition, a circular cross-section for which you define the outside diameter and the wall thickness.

Figure 5-1. ROD ELEMENT GEOMETRY



The solution results for the ROD and TUBE are the axial force, F_x, and the torque, T. The axial and torsional stresses are also given. These results are available in both real and complex form for frequency response dynamic analysis. The CONROD is an alternate form that allows the ROD property to be defined on the connection entry.

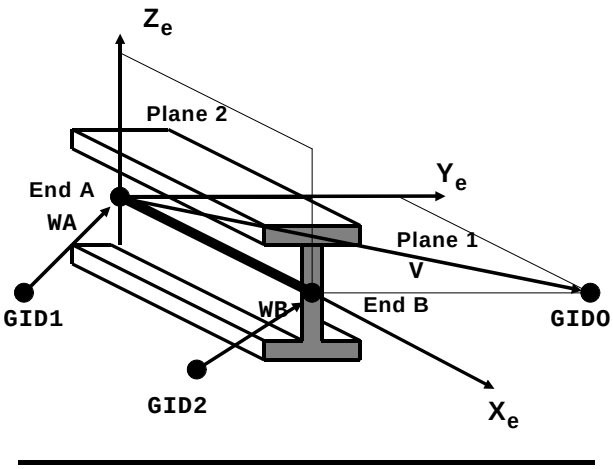
5.2.2 The BAR Element

The BAR element is a straight line structural element. It may carry bending, shear, axial, and torsional loads. The BAR element has a constant cross sectional geometry along its length. The BEAM element, described later, may be used to model members with varying cross sections.

Axial and bending material properties may include nonlinear strain relationships. The shear center is assumed to coincide with the neutral axis. Pin codes may be used to release any combination of element forces at the ends of the element. The element axis may be offset from the GRID points it connects. As shown in Figure 5-2, End A is offset from GRID point **GID1** by the vector **WA** and End B is offset from GRID point **GID2** by the vector **WB**. The vectors **WA** and **WB** are given in the output coordinate system of the corresponding GRID points. The x-axis of

the element coordinate system is defined by a line connecting End A to End B of the BAR element.

Figure 5-2. BAR ELEMENT GEOMETRY



The BAR geometry, element coordinate system, and roll axis orientation are described using the two end GRID points and an orientation vector as illustrated in Figure 5-2. Two reference planes are defined by the orientation vector **V** as illustrated. This vector may be defined directly by its three components in the output coordinate system at End A of the BAR, or by a line drawn from End A to a third GRID point, **GO**. The first reference plane, Plane 1, is defined by the x-axis of the BAR and the vector **V**. The second reference plane, Plane 2, is a plane perpendicular to Plane 1 which includes the X-axis of the element. 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, **I12**, on the property specification. If shearing deformations are included, the reference axes and the principal axes must coincide. 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 BAR, not at the GRID point. Normally, the BAR element mass matrix does not include torsional inertia terms. To include these terms, you include the following **PARAM** entry in your Bulk Data packet.

PARAM	**BARTORM**	**YES**							

5.2.2.1 BAR Element Properties

The properties of the BAR element are defined by either **PBAR** or **PBAR1** Bulk Data entries. Data specifying the BAR orientation are defined with the **CBAR** entry. If a large number of BARs have the same orientation, the **BAROR** Bulk Data entry may be used to define default values. The default values are used only when the corresponding fields of the **CBAR** entry are blank.

PBAR	**PID**	**MID**	**A**	**I1**	**I2**	**J**	**NSM**	**SHAPE**	**-cont-**
-cont-	**Y1**	**Z1**	**Y2**	**Z2**	**Y3**	**Z3**	**Y4**	**Z4**	**-cont-**
-cont-	**K1**	**K2**	**I12**						

The **PBAR** entry provides the most general input capability. It specifies the property identification number, **PID**, and the material property of the element, **MID**. The BAR element may only be composed of isotropic material. When modeling linear material behavior, **MAT1** Bulk Data entries are used. If you are modeling nonlinear behavior, then either **MAT1NL** or **MAT1NL1** entries may be used. The former is used to describe a general nonlinear stress-strain relationship, while the latter is used for a bilinear relationship. This is followed by the section properties of the BAR: its cross sectional area, **A**; its three area moments of inertia, **I1**, **I2**, and **I12**; and its torsional constant, **J**. You may select four distinct points, (**Yi**,**Zi**), on the cross section of the element at which you wish to have stress and strain computed. You specify these coordinates in the element coordinate system. The fields **K1** and **K2** specify shear area factors for the element. If you do not specify these factors, then the transverse shear flexibilities are assumed to be infinite.

The **PBAR1** entry provides a much simpler form of BAR section property definition:

PBAR1	PID	MID	CSHAPE	D1	D2	D3	D4	D5	-cont-
-cont-	NSM								

The **PBAR1** allows you to select from standard parameterized shapes such as I-Beams, T-Sections, and Tubes. These are shown in Figure 5-22 of Section 5.8.1. The **Di** fields define the dimensions of the cross section. The location of the stress recovery points are predefined for each standard cross section. **UAI/*NASTRAN*** then computes the required section properties. You may request a printed summary of the section properties generated, or you may request that equivalent **PBAR** Bulk Data entries be written on **BULK** file by using the **PARAM**eter **SECTION** as described in Section 5.8.1. Table 5-1 shows an example using the **PBAR1** Bulk Data entry. First the input data is shown. This is followed by the printed summary of calculated section properties when you select the **PRINT** option. Finally, when you select the **PUNCH** option, **PBAR** Bulk Data entries are written on the **BULK** file in High Precision format. You may use these data in subsequent jobs if you wish. Also note that both lumped and consistent mass formulations are available for the BAR element.

5.2.2.2 Bar Element Solution Results

For each BAR element, you may obtain forces, stresses and strain energy using the **FORCE**, **STRESS**, and **ESE** Case Control commands, respectively. The internal forces in BAR elements are: the bending moments at each end for each reference plane, M_{1a}, M_{1b}, M_{2a}, and M_{2b}; the shear forces, V_{1a}, V_{2a}, V_{1b}, and V_{2b}; the average axial force, F_{xa}, F_{xb}; and the torque, T_a,T_b. These forces and their sign conventions are illustrated in Figure 5-3.

The stresses computed for the BAR element are the average axial stress and the extensional stress at each end. Tensile stresses are positive and compressive stresses are negative. Stresses may be obtained at as many as four points on the cross section of the BAR. The coordinates of these recovery points are specified on the **PBAR** Bulk Data entry. They are given as pairs of **(Yi,Zi)** coordinates in the element system. Remember that these points are automatically generated when using the **PBAR1** Bulk Data entry. If you have entered stress allowables on your material Bulk Data entry, then the Margins-of-safety with respect to those allowables are also computed.

The format of the BAR element forces and stresses are shown in Table 5-2. Note that the stresses at the four recovery points are labeled **SXC**, **SXD**, **SXE** and **SXF**.

Figure 5-3. BAR ELEMENT FORCES

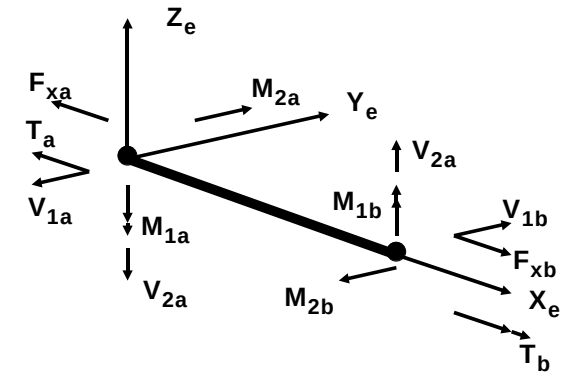


Table 5-1. PBAR1 CROSS SECTIONAL PROPERTIES

a. Input Data Specification

PBAR1	**5**	**1**	**BAR**	**1.0**	**2.0**				

b. Printed Section Properties

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


ELEMENT PROPERTY  SECTION     AREA           I1            I2            J            K1            K2
  TYPE      ID      TYPE

   BAR         5     BAR    2.00000E+00   6.66667E-01   1.66667E-01   4.57760E-01   8.33333E-01   8.33333E-01

                                        STRESS POINT LOCATIONS:         Y1 =  1.00000E+00     Z1= 0.00000E+00
```

c. Punched Section Properties

```
PBAR*                   5              1 2.000000000E+00 6.666666865E-01*
*        1.666666716E-01 4.577604234E-01 0.000000000E+00    BAR         *
*        1.000000000E+00 0.000000000E+00-1.000000000E+00 0.000000000E+00*
*        0.000000000E+00-5.000000000E-01 0.000000000E+00 5.000000000E-01*
*        8.333333135E-01 8.333333135E-01 0.000000000E+00
```

Table 5-2. BAR ELEMENT SOLUTION RESULTS

a. BAR Element Forces

```
                            F O R C E S   I N   B A R   E L E M E N T S        ( C B A R )
                  STAT DIST/  - BENDING MOMENTS -            - WEB SHEARS -            AXIAL
ELEMENT-ID   GRID   LENGTH   PLANE 1      PLANE 2         PLANE 1      PLANE 2          FORCE         TORQUE
       110
              15   0.000   0.0          -3.972531E+00   0.0           2.184291E+00  -9.059420E-14   0.0
               7   1.000   0.0          -1.489399E+01   0.0           2.184291E+00  -9.059420E-14   0.0
```

b. BAR Element Stresses

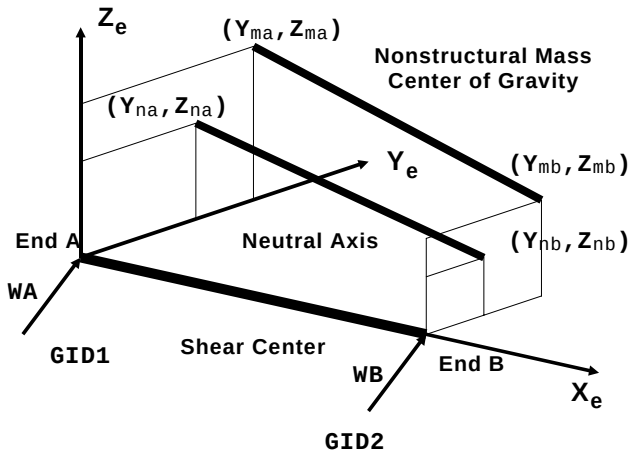
```
                              S T R E S S E S   I N   B A R   E L E M E N T S         ( C B A R )
                  STAT DIST/
ELEMENT-ID   GRID   LENGTH    SXC           SXD           SXE           SXF           S-MAX           S-MIN        M.S.-T
M.S.-C
       110
             15   0.000  -3.123938E-14 -3.123938E-14 -3.123938E-14 -3.123938E-14 -3.123938E-14   -3.123938E-14
              7   1.000  -3.123938E-14 -3.123938E-14 -3.123938E-14 -3.123938E-14 -3.123938E-14   -3.123938E-14
```

5.2.3 The BEAM Element

The BEAM element provides several extensions to the modeling features provided by the BAR including varying cross section, taper with shear relief effects, open sections with cross sectional warping, and varying nonstructural mass. The BEAM geometry, element coordinate system and roll axis orientation are defined in Figure 5-4.

The BEAM lies along its shear center between End A and End B. These ends may be offset from GRID points **GID1** and **GID2** by vectors **WA** and **WB** respectively. Also, the neutral axis may be offset from the shear center. The offset distance, which may vary linearly from End A to End B, is defined by the points **(YNA,ZNA)** and **(YNB,ZNB)**. The nonstructural mass center of gravity may be offset in a similar manner by specifying the coordinates **(YNA,ZNA)** and **(YNB,ZNB)**. Pin flags may be used to release forces or moments at either end.

Figure 5-4. BEAM ELEMENT GEOMETRY



5.2.3.1 BEAM Element Properties

The properties of the BEAM element are defined by either **PBEAM** or **PBEAM1** Bulk Data entries. Data specifying the BEAM orientation are defined with the **CBEAM** entry. If a large number of BEAMs have the same orientation, the **BEAMOR** Bulk Data entry may be used to define default values. The default values are used only when the corresponding fields of the **CBEAM** entry are blank.

The cross sectional properties including area, area moments of inertia, and polar moments of inertia may be specified at each end, and, optionally, at as many as nine intermediate stations along the length of the BEAM. Properties between each station vary linearly. For properties not specified at intermediate stations, interpolation between values at the two ends is used. If you do not specify cross sectional properties for End B, then End A properties are used throughout the beam. Additional lumped nonstructural mass may be applied at the intermediate stations. All property data are entered with the either the **PBEAM** or **PBEAM1** Bulk Data entries. The format of the **PBEAM** Bulk Data entry is:

PBEAM	PID	MID	AA	I1A	I2A	I12A	JA	NSMA	-cont-
THE NEXT TWO CONTINUATIONS ARE REPEATED FOR EACH INTERMEDIATE STATION									
-cont-	STRESS	STATN	A	I1	I2	I12	J	NSM	-cont-
-cont-	Y1	Z1	Y2	Z2	Y3	Z3	Y4	Z4	-cont-
-cont-	K1	K2	S1	S2	NSIA	NSIB	CWA	CWB	-cont-
-cont-	YMA	ZMA	YMB	ZMB	YNA	ZNA	YNB	ZNB	

Beam theory has been used to provide the BEAM element with complete generality. To accurately model the action of the BEAM under shear loads, two features are available. The first feature allows you to define shear area factors, **K1** and **K2**. These factors are used to specify

the effective area of the BEAM in shear. You use this, for example, to model a wide flange beam which is loaded in the plane of its web. In this case, only the area of the web is effective in resisting shear. The second feature allows you to define shear relief coefficients, **S1** and **S2**. You use these coefficients to represent the shear relief which occurs if the BEAM is tapered. For example, if a bending moment is carried entirely by the flange, then the vertical components of the flange force help resist the shear.

Another feature allows you to model warping behavior. For example, warping occurs when an open section BEAM undergoes torsional loading. Its cross sections do not remain plane, but instead displace longitudinally, thereby causing warping. The best example of this effect is a circular cylindrical tube with a longitudinal slot. For a closed cross section, torsion is normally resisted by shear. If a lengthwise slot is then made in the tube, the longitudinal shear at the slot must be zero. As a result, the cross section warps, and the torsional stiffness of the tube is greatly reduced. An additional degree of freedom is included to model this phenomenon. SCALAR or GRID points at each end represent the warping variable. Warping coefficients, **CWA** and **CWB**, are included on the **PBEAM** property entry.

The **PBEAM1** entry provides a much simpler form of BEAM section property definition.

PBEAM1	**PID**	**MID**	**SHAPE**	**D1A**	**D2A**	**D3A**	**D4A**	**D5A**	**-cont-**
-cont-	**NSMA**	**NSMB**	**NSTAT**	**D1B**	**D2B**	**D3B**	**D4B**	**D5B**	

The **PBEAM1**, like the **PBAR1**, allows you to select from standard parameterized shapes such as I-Beams, T-Sections, and Tubes. Again, refer to Figure 5-22 of Section 5.8.1 and the description given in Section 5.3.1. The **Di** fields define the dimensions of the cross section. **UAI/*NASTRAN*** then computes the required section properties. As in the case of the BAR element, you may use the **SECTION PARAM**eter to request a summary of the section properties computed. The results of the computations are similar to the BAR element, but properties are printed for each station, as shown in Table 5-3. This example is for a tapered BEAM with a **SHAPE** of **BAR**.

5

5.2.3.2 BEAM Element Solution Results

For each BEAM element, you may obtain forces, stresses and strain energy using the **FORCE**, **STRESS**, and **ESE** Case Control commands, respectively. The internal BEAM forces include the axial force along the neutral axis, moments about the element axis, torsion about the shear center and the shears. The sign conventions are the same as those illustrated for the BAR element in Figure 5-4. Note that for BEAMs which have intermediate stations, internal forces correctly account for any loads which span the stations of the BEAM.

The longitudinal stresses for each axial station along the BEAM are calculated. These are recovered at as many as four points on the cross section at each station. The coordinates of these points are specified on the **PBEAM** Bulk Data entry. They are given as pairs of (**Y**,**Z**) coordinates in the element system. The format of the BEAM element forces and stresses are shown in Table 5-4.

5.2.3.3 Equivalent Beam of Solid Elements

You may also model beams as assemblies of solid elements. This allows you to compute equivalent beam forces from the three-dimensional stress field. The technique for doing this is described in section 5.4.1.2.

Table 5-3. PBEAM1 CROSS SECTIONAL PROPERTIES

a. Input Data Specification

PBEAM1	6	1	BAR	0.8	1.5				cont
cont				1.2	2.0				

b. Printed Section Properties

```
                              B E A M     S E C T I O N     P R O P E R T Y     C A L C U L A T I O N S

PROPERTY  SECTION     STATION        AREA          I1            I2            J           K1           K2
   ID       TYPE       (XFR)

       6     BAR    0.00000E+00   1.20000E+00   2.25000E-01   6.40000E-02   1.70564E-01   8.33333E-01   8.33333E-01

                                                STRESS POINT LOCATIONS:         Y1 =  7.50000E-01     Z1= 0.00000E+00
                                                                                Y2 = -7.50000E-01     Z2= 0.00000E+00
                                                                                Y3 =  0.00000E+00     Z3=-4.00000E-01
                                                                                Y4 =  0.00000E+00     Z4= 4.00000E-01


                     1.00000E+00   2.40000E+00   8.00000E-01   2.88000E-01   7.21247E-01   8.33333E-01   8.33333E-01

                                                STRESS POINT LOCATIONS:         Y1 =  1.00000E+00     Z1= 0.00000E+00
                                                                                Y2 = -1.00000E+00     Z2= 0.00000E+00
                                                                                Y3 =  0.00000E+00     Z3=-6.00000E-01
                                                                                Y4 =  0.00000E+00     Z4= 6.00000E-01
```

c. Punched Section Properties

```
PBEAM*                  6               1 1.200000048E+00 2.250000089E-01*
*        6.400000304E-02                  1.705639660E-01 0.000000000E+00*
*        7.500000000E-01 0.000000000E+00-7.500000000E-01 0.000000000E+00*
*        0.000000000E+00-4.000000060E-01 0.000000000E+00 4.000000060E-01*
*       YESA            1.0              2.400000095E+00 8.000000119E-01*
*        2.880000472E-01                  7.212470174E-01 0.000000000E+00*
*        1.000000000E+00 0.000000000E+00-1.000000000E+00 0.000000000E+00*
*        0.000000000E+00-6.000000238E-01 0.000000000E+00 6.000000238E-01*
*        8.333333135E-01 8.333333135E-01
```

Table 5-4. BEAM ELEMENT SOLUTION RESULTS

a. BEAM Element Forces

```
                                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

      8702
            8702     0.000     0.0          0.0             0.0          0.0           5.000000E+00   0.0           0.0
               0     0.200     0.0          0.0             0.0          1.110223E-16  5.000000E+00   0.0           0.0
               0     0.400     0.0          0.0             0.0          0.0           5.000000E+00   0.0           0.0
               0     0.600     0.0          0.0             0.0          0.0           5.000000E+00   0.0           0.0
               0     0.800     0.0          0.0             0.0          0.0           5.000000E+00   0.0           0.0
            8703     1.000     0.0          0.0             0.0          0.0           5.000000E+00   0.0           0.0
```

b. BEAM Element Stresses

```
                                S T R E S S E S   I N   B E A M   E L E M E N T S        ( C B E A M )
                    STAT DIST/
ELEMENT-ID   GRID     LENGTH      SXC           SXD           SXE           SXF           S-MAX          S-MIN         M.S.-T   M.S.-C

      8702
            8702     0.000    1.000000E+00  1.000000E+00  1.000000E+00  1.000000E+00  1.000000E+00   1.000000E+00
               0     0.200    1.000000E+00  1.000000E+00  1.000000E+00  1.000000E+00  1.000000E+00   1.000000E+00
               0     0.400    1.000000E+00  1.000000E+00  1.000000E+00  1.000000E+00  1.000000E+00   1.000000E+00
               0     0.600    1.000000E+00  1.000000E+00  1.000000E+00  1.000000E+00  1.000000E+00   1.000000E+00
               0     0.800    1.000000E+00  1.000000E+00  1.000000E+00  1.000000E+00  1.000000E+00   1.000000E+00
            8703     1.000    1.000000E+00  1.000000E+00  1.000000E+00  1.000000E+00  1.000000E+00   1.000000E+00
            8704     1.000    1.000000E+00  1.000000E+00  1.000000E+00  1.000000E+00  1.000000E+00   1.000000E+00
```

5.2.4 BAR and BEAM Offset Data Checking

When using large numbers of offsets for BAR or BEAM elements, the possibility of an error may become great. You may request that **UAI/*NASTRAN*** check for unrealistic offset values by using the Executive Control command **DATACHECK**:

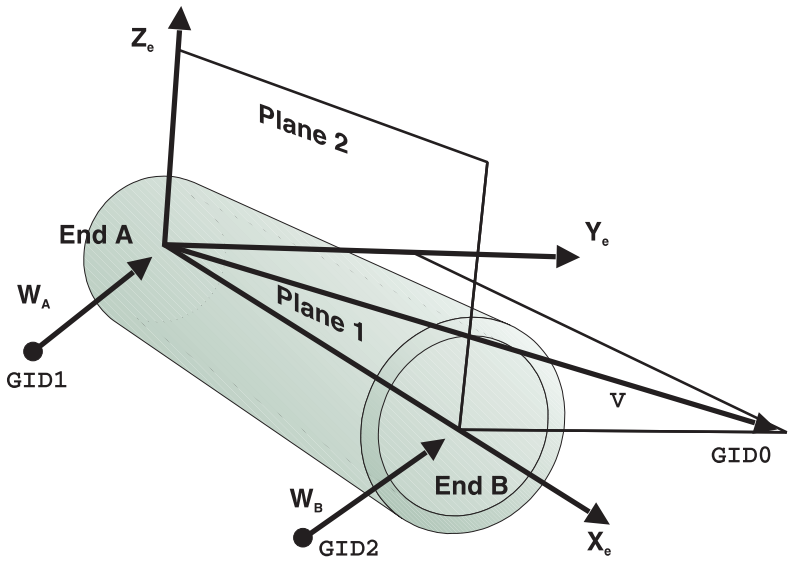
DATACHECK BEAMOFFSETS = { IGNORE / **WARNING** [MAXWARNING = *n*] / FATAL } [RATIO = *x*]

The **RATIO** argument specifies the maximum ratio of the offset of the element to its length that will be considered reasonable. You may have such elements **IGNORE**d, or you may have **WARNING** messages printed, or you may have the program terminated with the **FATAL** option. When selecting **WARNING**, you may also specify the maximum number of such messages. The default **RATIO** is given in the Preference file. You may use the Executive Control command **PRINT PREFERENCES** to confirm the current default.

5.2.5 The PILE Element

The PILE element is provided especially to solve nonlinear soil-structure interaction analyses. It combines a BAR structural element with soil characteristics which vary in the axial, torsional, and lateral directions and are distributed along the length of the element. The structural material properties for axial and bending motions may also be strain-dependent The PILE element, whose geometry is shown in Figure 5-5, is defined by a **CPILE** Bulk Data entry which is similar to the **CBAR** entry.

Figure 5-5. PILE ELEMENT GEOMETRY



The PILE element is defined by it identification number, and references two optional properties: the structural property, and the attached soil property. If no soil property is specified, then there is no soil attached to the element. The **PPILE** and **PPILE1** Bulk Data entries are available to define the structural properties. **PSOIL** Bulk Data entries define the soil properties:

PSOIL	PID	TIDA	TIDJ	TIDL					

In addition to the property identification number, **PID**, these data define tabular functions which define the axial, **TIDA**, torsional, **TIDJ**, and lateral, **TIDL**, force-deflection relationships of the soil. Note that you do not need to define soil properties in all directions if, for example, only a planar or longitudinal model is desired. Soil data may be either constant or a function of PILE displacements. In the latter case, the **TIDA**, **TIDJ**, and **TIDL** relationships are defined using **TABLENL** Bulk Data entries.

Soil data are defined in terms of soil reaction forces per unit area of the pile. This convention may differ from soil data obtained in field testing of piles. For example, if lateral soil reactions are obtained for a 24 inch diameter pile at various deflections, the reactions must be divided by 24 to provide the correct input. If the data are input using the **TABLENL** Bulk Data entries, a **SCALE** field may be set to 1/24 to provide the required scaling. Also, it must be remembered that the slope of such data curves is used to define soil stiffness. Therefore, if a linear analysis is to be performed, and you wish to avoid imputing the table data, the initial slope of the table must be computed, divided by 24 for the case described above, and the resulting value may be used directly on the **PSOIL** entry. If total soil reaction loads are available for axial or torsional tests, these data must be divided by the pile circumference. Once again, the **SCALE** field on the **TABLENL** Bulk Data entry may be used.

The PILE element formulation uses constant soil properties for the length of the element. Soil variation with depth is modeled by using a number of PILE elements with each referencing different **PSOIL** and **TABLENL** data. Shorter elements should be used in areas of rapidly changing soil properties.

Elements force and stress output for the PILE are identical to those of the BAR element. No additional output for the soil is generated.

5.2.6 Summary of Bulk Data for the One-Dimensional Elements

The following table summarizes the various Bulk Data entries frequently used when modeling with one-dimensional elements. Detailed descriptions of these Bulk Data entries are found in Chapter 7 of the ***User's Reference Manual***.

ELEMENT	BULK DATA ENTRIES		
	CONNECTION	PROPERTY	
		Arbitrary Cross-Sections	Standard Cross-Sections
ROD	CROD,CONROD	PROD	
TUBE	CTUBE	PTUBE	
BAR	CBAR	PBAR	PBAR1
BEAM	CBEAM	PBEAM	PBEAM1
PILE	CPILE	PPILE	PPILE1

5.3 TWO-DIMENSIONAL ELEMENTS

UAI/*NASTRAN* includes a library of two-dimensional elements for modeling arbitrary plate and shell structures. A family of linear and quadratic isoparametric triangular and quadrilateral elements are available. Additionally, there are quadrilateral and triangualr elements which allow the modeling of normal rotations, or drilling degrees of freedom. Finally, there are two specialty elements used for some applications. All of these are described in this Section.

5.3.1 The QUAD4 and TRIA3 Elements

The four node quadrilateral, QUAD4, and three node triangular, TRIA3 elements are constant strain isoparametric elements that may be used to model either or both membrane and bending behavior. The geometry and coordinate systems of these elements are shown in Figures 5-6 and 5-7. The elements have three distinct coordinate systems only two of which are shown in the figure: the element system, denoted by the subscript e, and the material system, denoted by the subscript m. If the material and stress systems are not specified, they are assumed to coincide with the external element system. The element system is defined for convenience to support interpretation of output and development of post-processors.

For the QUAD4 element, the Y-axis of the element system lies on the plane defined by the three GRID points **GID1**, **GID2**, and **GID3**.

These elements are defined with **CQUAD4** an **CTRIA3** Bulk Data entries respectively. Each element connection entry includes the element identification number, a property identification number, and the GRID points defining the element. Note that the element definition includes the specification of options for material angle and element reference plane. The mean plane of the element may be offset from the GRID points by the value, which is entered either on the connection data entry or on one of the property entries described in the following sections. You may also specify a bilinear variation in element thickness by entering thickness values at each GRID point. If you do not enter any or all thicknesses, then all missing values will inherit the thickness value that you place on the **PSHELL** Bulk Data entry as described below.

5

Figure 5-6. TRIA3 ELEMENT GEOMETRY

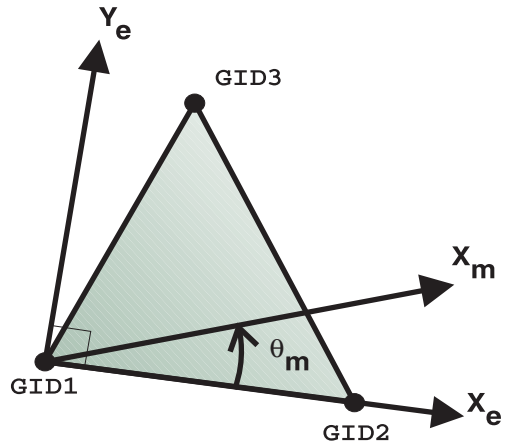
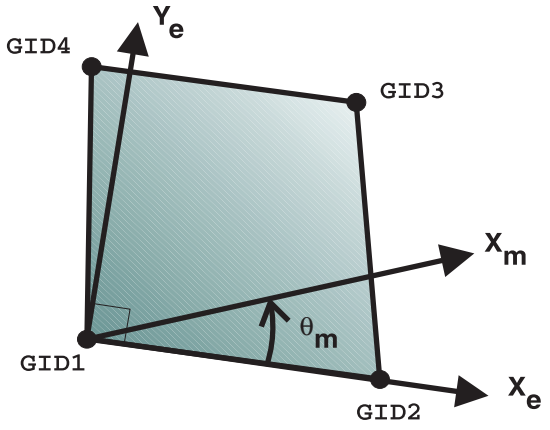


Figure 5-7. QUAD4 ELEMENT GEOMETRY



5.3.1.1 Element Properties: Homogeneous Materials and Membrane-Bending Coupling

You have several options for modeling the behavior of plate and shell structures using the QUAD4 and TRIA3 elements. These elements may be used to model homogeneous materials or materials which vary through the element thickness.

If your elements are made of a homogeneous material, or, if you are modeling a complex material such as honeycomb buildups, you use the **PSHELL** Bulk Data entry:

PSHELL	PID	MID1	T	MID2	12I/T3	MID3	TS/T	NSM	-cont-
-cont-	Z1	Z2	MID4	MCSID	SCSID	ZO			
				θm	θs				

The first two data fields, **MID1** and **T**, specify a material property identification number that will be used to represent the membrane behavior of the plate. The value of **T** represents the membrane thickness. **MID1** may reference either isotropic, **MAT1**, anisotropic, **MAT2**, or orthotropic, **MAT8**, materials. This is the case for all of the material selections on the **PSHELL** entry. The fields **MID2** and **12I/T3** define the element bending material property and bending stiffness parameter, respectively. You may also specify a material property to model transverse shear effects, **MID3**, and a normalized transverse shear thickness, **TS/T**. The purpose of this feature is to increase the element flexibility if you are analyzing beam-like models. The final material behavior is the coupling of membrane and bending behavior which is defined by **MID4**. Membrane/bending coupling allows you to model plates which have a linear variation of elastic modulii through their thickness. Such is the case when modeling reinforced concrete beams or bimetallic plates. Table 5-5 summarizes these modeling features.

The field **θm** is used to define the angle between the x-axis of the material system and the x-axis of the element coordinate system. Alternately, you may specify an **MCSID** which references a coordinate system definition. In this case, the orientation of the material x-axis is along the projection of the x-axis of the specified coordinate system on the xy-plane of the element system. The fields **θs** and **SCSID** similarly define the element stress coordinate system. A special convention, used by some post-processors allows you to request that the stress system be aligned with the element internal coordinate system, is provided. To request this feature, contact your **UAI/*NASTRAN*** Support Specialist.

Table 5-5. PLATE ELEMENT MATERIAL MODELLING

TO MODEL THIS BEHAVIOR:	USE THESE MATERIALS:			
	MID1	MID2	MID3	MID4
PURE MEMBRANE	✔			
PURE BENDING		✔		
BENDING WITH TRANSVERSE SHEAR		✔	✔	
MEMBRANE AND BENDING WITHOUT COUPLING	✔	✔	✘	
MEMBRANE AND BENDING WITH COUPLING	✔	✔	✘	✔

✔ Property Required To Model This Behavior
✘ Optional Property

5.3.1.2 Element Properties: Laminated Composite Elements

UAI/*NASTRAN* allows you to model complex laminated composites. This is done by referencing special composite property data as described in 5.8.2.

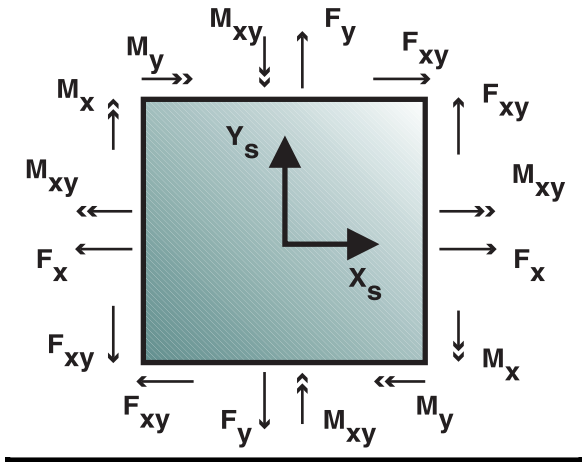
5.3.1.3 Element Solution Results

For each of these elements, you may obtain forces, stresses, strains and strain energy solution results. The element forces, which are defined in the stress coordinate system for both the QUAD4 and TRIA3, are out-of-plane bending and twisting moments M_x, M_y, M_{xy}; and out-of-plane shear forces V_x, V_y. The sign convention of these forces is shown in Figure 5-8.

Stress values are computed at two points on the cross section at the element centroid and include: the normal stresses in the x and y directions; in-plane shear stresses; the angle between the x-axis and the major principal axis; major and minor principal stresses; and either the maximum shear stress or the von Mises equivalent stress. You may select between the latter two by specifying one of the keywords **VONMISES** or **MAXSHEAR** when entering your **STRESS** Case Control command. The von Mises stress is computed using:

$$\tau_{vm} = (\sigma_x^2 - \sigma_x\sigma_y + \sigma_y^2 + 3\tau_{xy}^2)^{1/2}$$

Figure 5-8. PLATE ELEMENT FORCES



You may specify the location of the stress output points with the **PSHELL** Bulk Data entry using the fiber distance data fields. The positive direction for these distances is determined by using the right hand rule to the order of the GRID point specification on the connection definition, **CQUAD4** or **CTRIA3** Bulk Data. For complex stress and strain output, only the normal and shear data are computed. Finally, if you are using laminated composite materials, you may request layer stress output by using the keyword **LAYER**.

Strain values are also computed at two points on the cross section at the element centroid and include: the normal strains in the x and y directions; in-plane shear strains; the angle between the x-axis and the major principal axis; major and minor principal strains; and either the maximum shear strain or the von Mises equivalent strain. As in the case of stresses, you may select between the latter two by specifying one of the keyword **VONMISES** or **MAXSHEAR**. When you are using laminated composite materials, you may request layer stress output by using the keyword **LAYER**, or you may select equivalent strains and curvatures at the midplane of the element by specifying **STRCUR**. Table 5-6 presents examples of the solution output for these elements.

Table 5-6. QUAD4 and TRIA3 SOLUTION RESULTS

a. Element Forces

F O R C E S I N G E N E R A L 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 4)

ELEMENT ID	- MEMBRANE FORCES - FX	FY	FXY	- BENDING MOMENTS - MX	MY	MXY	- TRANSVERSE SHEAR FORCES - VX	VY
6801	1.00000E+00	9.32653E-02	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
6802	1.00000E+00	-2.32301E-02	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

b. Element Stresses

S T R E S S E S I N G E N E R A L 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 4)

ELEM ID	FIBER DISTANCE	STRESSES IN STRESS COORD SYSTEM NORMAL-X	NORMAL-Y	SHEAR-XY	PRINCIPAL STRESSES (ZERO SHEAR) ANGLE	MAJOR	MINOR	VON MISES
6801	-1.000E-01	1.00000E+00	9.32653E-02	0.00000E+00	0.000	1.00000E+00	9.32653E-02	9.56783E-01
	1.000E-01	1.00000E+00	9.32653E-02	0.00000E+00	0.000	1.00000E+00	9.32653E-02	9.56783E-01
6802	-2.500E-01	1.00000E+00	-2.32301E-02	0.00000E+00	0.000	1.00000E+00	-2.32301E-02	1.01182E+00
	2.500E-01	1.00000E+00	-2.32301E-02	0.00000E+00	0.000	1.00000E+00	-2.32301E-02	1.01182E+00

c. Element Strains

S T R A I N S I N G E N E R A L 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 4)

ELEM ID	FIBER DISTANCE	STRAINS IN STRESS COORD SYSTEM NORMAL-X	NORMAL-Y	SHEAR-XY	PRINCIPAL STRAINS (ZERO SHEAR) ANGLE	MAJOR	MINOR	VON MISES
6801	-1.000E-01	3.25561E-08	-5.22449E-09	0.00000E+00	0.000	3.25561E-08	-5.22449E-09	3.54582E-08
	1.000E-01	3.25561E-08	-5.22449E-09	0.00000E+00	0.000	3.25561E-08	-5.22449E-09	3.54582E-08
6802	-2.500E-01	3.35269E-08	-9.10767E-09	0.00000E+00	0.000	3.35269E-08	-9.10767E-09	3.88890E-08
	2.500E-01	3.35269E-08	-9.10767E-09	0.00000E+00	0.000	3.35269E-08	-9.10767E-09	3.88890E-08

d. Element Forces: Composite Materials

F O R C E S I N L A Y E R E D C O M P O S I T E E L E M E N T S (Q U A D 4)

ELEMENT ID	- MEMBRANE FORCES - FX	FY	FXY	- BENDING MOMENTS - MX	MY	MXY	- TRANSVERSE SHEAR FORCES - VX	VY
6803	1.00000E+00	4.46141E-03	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

e. Element Stresses: Composite Materials

S T R E S S E S I N L A Y E R E D C O M P O S I T E E L E M E N T S (T R I A 3)

ELEMENT ID	PLY ID	*STRESSES IN FIBER AND MATRIX DIRECTIONS* NORMAL-1	NORMAL-2	SHEAR-12 *	*DIRECT FIBER FAILURE INDEX*	*INTER-LAMINAR STRESSES* SHEAR-1Z	SHEAR-2Z*	* SHEAR BOND FAILURE INDEX*	*MAXIMUM* * INDEX *
6902	1	6.37819E-01	3.62181E-01	-4.80631E-01	0.000	0.00000E+00	0.00000E+00	0.000	
	2	4.65122E-01	5.34878E-01	-4.98782E-01	0.000	0.00000E+00	0.00000E+00	0.000	
	3	7.93893E-01	2.06107E-01	-4.04509E-01	0.000	0.00000E+00	0.00000E+00	0.000	
	4	1.52671E-01	8.47329E-01	-3.59670E-01	0.000	0.00000E+00	0.00000E+00	0.000	
									0.000

f. Element Strains: Composite Materials

S T R A I N S I N L A Y E R E D C O M P O S I T E E L E M E N T S (Q U A D 4)

ELEMENT ID	PLY ID	*STRAINS IN FIBER AND MATRIX DIRECTIONS* * NORMAL-1	NORMAL-2	SHEAR-12 *	*DIRECT FIBER FAILURE INDEX*	*INTER-LAMINAR STRAINS* SHEAR-1Z	SHEAR-2Z*	* SHEAR BOND FAILURE INDEX*	*MAXIMUM* * INDEX *
6803	1	3.32962E-08	-8.18462E-09	5.99706E-24	0.000	0.00000E+00	0.00000E+00	0.000	
	2	1.96494E-08	5.46214E-09	-3.89792E-08	0.000	0.00000E+00	0.00000E+00	0.000	
	3	1.96494E-08	5.46214E-09	-3.89792E-08	0.000	0.00000E+00	0.00000E+00	0.000	
	4	3.32962E-08	-8.18462E-09	5.99706E-24	0.000	0.00000E+00	0.00000E+00	0.000	

5.3.2 The QUAD8 and TRIA6 Elements

The eight node quadrilateral shape QUAD8 and six node triangular shape TRIA6 elements are linear strain isoparametric shell elements that support quadratic variation of geometry. With each of these elements any or all of the mid-side GRID points may be omitted. These elements, like the QUAD4 and TRIA3, allow independent element, material, and stress coordinate systems as shown in Figures 5-9 and 5-10.

For both elements, the GRID points **GID1**, **GID2** and **GID3** define the X-Y plane of the element coordinate system. The X direction is defined by the vector from **GID1**to **GID2**. The mean plane of the element may be offset from the GRID points by a single constant value specified on either the connection, **CQUAD8** or **CTRIA6**, Bulk Data entry or on the property, **PSHELL**, entry.

As with the QUAD4 and TRIA3 elements, complex behavior may be modeled including combinations of membrane, bending, transverse shear, and coupling of membrane and bending actions. The **PSHELL**, or for composite material elements the **PCOMP**, **PCOMP1**, or **PCOMP2** Bulk Data are used to define the properties of these elements. Either isotropic, **MAT1**, orthotropic, **MAT8**, or anisotropic, **MAT2**, material properties may be used.

5.3.2.1 Element Properties

The properties for the QUAD8 and TRIA6 elements are the same as those available for the QUAD4 and TRIA3 elements as described in Sections 5.3.1.1 and 5.3.1.2.

5

5.3.2.2 Element Solution Results

The element forces are defined in the stress coordinate system for both the QUAD8 and TRIA6. The definition of these forces, and their sign conventions, are the same as for the QUAD4 and TRIA3. The element stress and strain output options are also the same as those of the QUAD4 and TRIA3. However, the stress and strain quantities are also output at the element corner GRID points as well as the element centroid. Various examples of this are shown in Table 5-7.

Figure 5-9. QUAD8 ELEMENT GEOMETRY

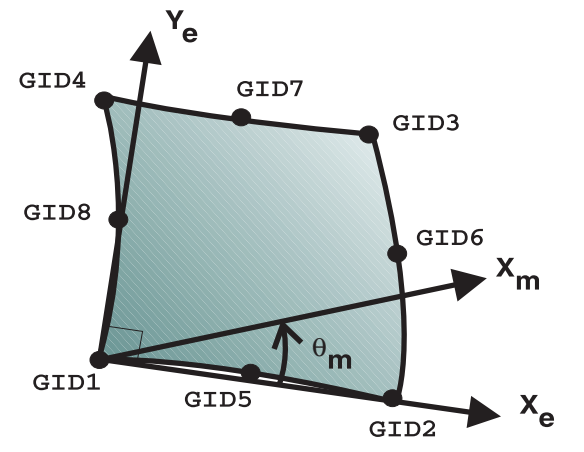


Figure 5-10. TRIA6 ELEMENT GEOMETRY

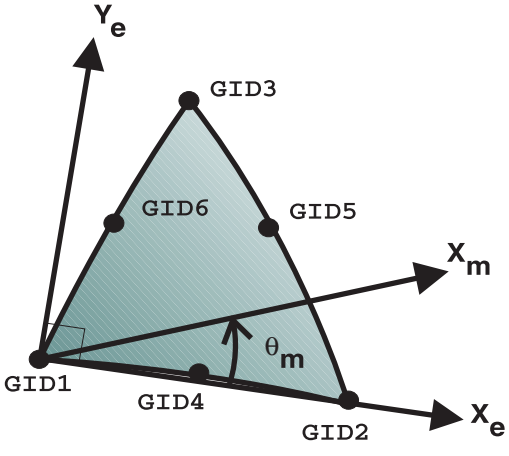


Table 5-7. QUAD8 AND TRIA6 SOLUTION RESULTS

a. Element Forces

F O R C E S I N G E N E R A L 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 8)

ELEMENT ID	GRID-ID	- MEMBRANE FORCES - FX	FY	FXY	- BENDING MOMENTS - MX	MY	MXY	- TRANSVERSE SHEAR FORCES - VX	VY
201	CNTR	9.99998E-01	1.53120E-02	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
	201	9.99998E-01	9.75944E-02	1.07900E-01	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
	206	9.99998E-01	-6.69704E-02	-7.58449E-02	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
	208	9.99998E-01	-6.69704E-02	7.58449E-02	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
	203	9.99998E-01	9.75944E-02	-1.07900E-01	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

b. Element Stresses: Homogeneous Materials

S T R E S S E S I N G E N E R A L 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 8)

ELEM ID	GRID-ID	FIBER DISTANCE	STRESSES IN STRESS COORD SYSTEM NORMAL-X	NORMAL-Y	SHEAR-XY	PRINCIPAL STRESSES (ZERO SHEAR) ANGLE	MAJOR	MINOR	VON MISES
201	CNTR	-5.000E-01	9.99998E-01	1.53120E-02	0.00000E+00	0.000	9.99998E-01	1.53120E-02	9.92431E-01
		5.000E-01	9.99998E-01	1.53120E-02	0.00000E+00	0.000	9.99998E-01	1.53120E-02	9.92431E-01
	201	-5.000E-01	9.99998E-01	9.75944E-02	1.07900E-01	6.725	1.01272E+00	8.48724E-02	9.73064E-01
		5.000E-01	9.99998E-01	9.75944E-02	1.07900E-01	6.725	1.01272E+00	8.48724E-02	9.73064E-01
	206	-5.000E-01	9.99998E-01	-6.69704E-02	-7.58449E-02	-4.046	1.00536E+00	-7.23348E-02	1.04341E+00
		5.000E-01	9.99998E-01	-6.69704E-02	-7.58449E-02	-4.046	1.00536E+00	-7.23348E-02	1.04341E+00
	208	-5.000E-01	9.99998E-01	-6.69704E-02	7.58449E-02	4.046	1.00536E+00	-7.23348E-02	1.04341E+00
		5.000E-01	9.99998E-01	-6.69704E-02	7.58449E-02	4.046	1.00536E+00	-7.23348E-02	1.04341E+00
	203	-5.000E-01	9.99998E-01	9.75944E-02	-1.07900E-01	-6.725	1.01272E+00	8.48724E-02	9.73064E-01
		5.000E-01	9.99998E-01	9.75944E-02	-1.07900E-01	-6.725	1.01272E+00	8.48724E-02	9.73064E-01

c. Element Stresses: Composite Materials

S T R E S S E S I N L A Y E R E D C O M P O S I T E E L E M E N T S (Q U A D 8)

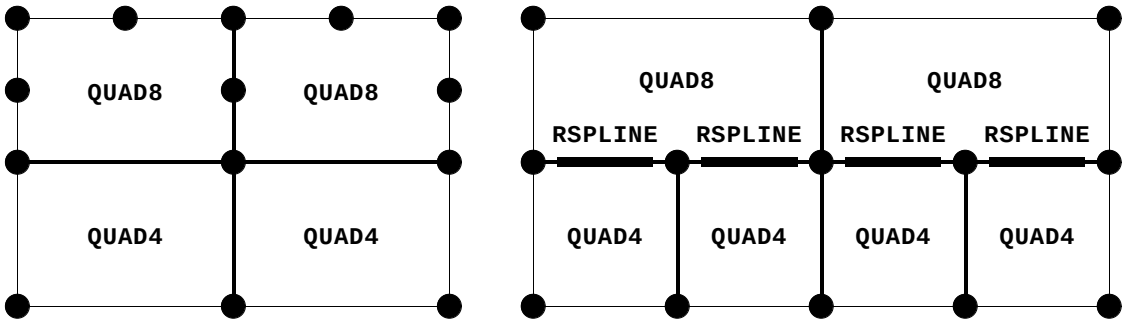
ELEMENT ID	PLY ID	*STRESSES IN FIBER AND MATRIX DIRECTIONS* NORMAL-1	NORMAL-2	SHEAR-12 *	*DIRECT FIBER * *FAILURE INDEX*	*INTER-LAMINAR STRESSES* *SHEAR-1Z	SHEAR-2Z*	* SHEAR BOND * *FAILURE INDEX*	*MAXIMUM* * INDEX *
203	1	9.99998E-01	1.87200E-03	-4.48771E-16	0.000	0.00000E+00	0.00000E+00	0.000	
	2	9.69901E-01	3.19692E-02	-1.70690E-01	0.000	0.00000E+00	0.00000E+00	0.000	
	3	9.69901E-01	3.19692E-02	1.70690E-01	0.000	0.00000E+00	0.00000E+00	0.000	
	4	7.50466E-01	2.51403E-01	-4.32201E-01	0.000	0.00000E+00	0.00000E+00	0.000	
									0.000

5.3.2.3 Compatibility with QUAD4 and TRIA3

Because both of these quadratic elements can be defined without midside GRID points, or with any combination of midside GRID points present, possible discontinuities may occur if they are coupled with the linear isoparametric elements. To insure compatibility, the element shape functions are reduced to linear order if the midside node is eliminated from any edge, as shown in Figure 5-11.

However, each time a midside node is deleted from a QUAD8 or TRIA6, the element becomes stiffer. As a result, more elements are needed for equivalent accuracy. The preferred method for transitioning from quadratic to linear elements is with **RSPLINE** connections. These connections define dependency equations between GRID points which enforce element compatibility, also as shown in Figure 5-11.

Figure 5-11. USING LINEAR AND QUADRATIC ELEMENTS

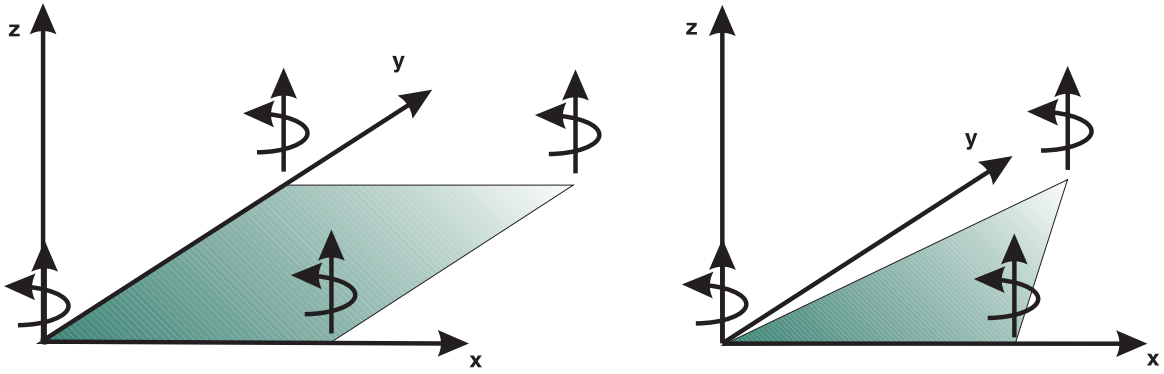


5.3.3 The QUADR and TRIAR Elements

In **UAI/*NASTRAN***, all geometric GRID points represent six degrees of freedom. However, the formulations of the elements discussed in the previous sections, the QUAD4, TRIA3, QUAD8 and TRIA6, allow for only five degrees of freedom at each of the element connection points. This leaves the normal rotation, sometimes called the *drilling degree of freedom*, with zero stiffness. As a result, these normal rotations must often be removed by using **SPC** Bulk Data entries or the AUTOSPC capability. The QUADR and TRIAR elements eliminate this requirement because they provide for all six degrees of freedom at their connection points. This extra freedom allows these elements to better model inplane bending loads.

These elements are defined with **CQUADR** and **CTRIAR** Bulk Data entries. All data on these entries are the same as those used for the QUAD4 and TRIA3 elements. The input data for element properties and the allowable materials are also the same. The only difference is that these elements have stiffness at all six degrees of freedom, rather than only five degrees of freedom. The additional stiffness is supplied in the drilling direction as shown in Figure 5-12.

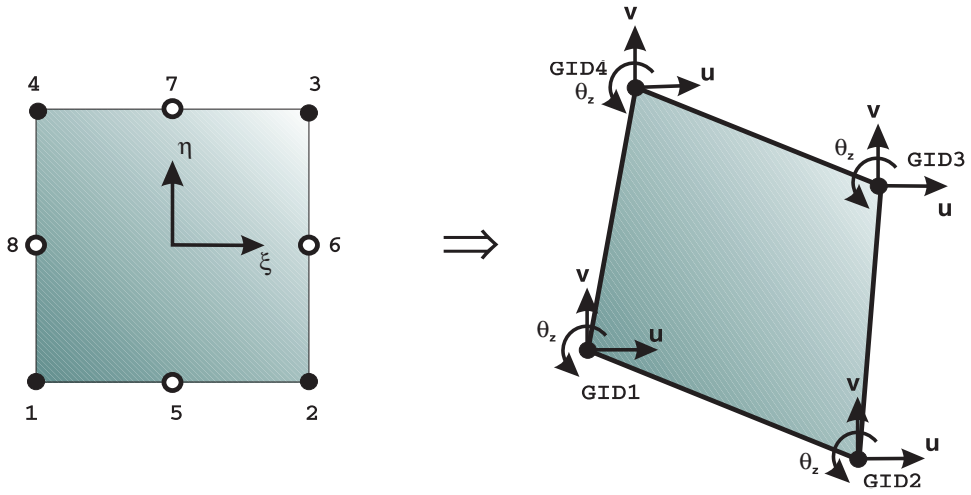
Figure 5-12. "DRILLING" DEGREES OF FREEDOM



Internally, these elements are hybrids of the linear and quadratic plate elements. For membrane behavior, they behave primarily like the QUAD8 and TRIA6. In fact, the displacement field for the QUADR/TRIAR are constructed from those of the QUAD8/TRIA6 as shown in Figure 5-13. The midside nodes of the QUAD8/TRIA6 are eliminated in favor of the normal rotations at the corner nodes. On the other hand, for bending behavior these elements are very similar to the QUAD4/TRIA3. This design allows the QUADR/TRIAR to have accuracy that is similar to, or better than, that of the QUAD8/TRIA6 while having the modeling simplicity of the QUAD4/TRIA3.

5

Figure 5-13. DEGREES OF FREEDOM FOR THE QUADR AND TRIAR



5.3.4 Special Modeling Features

There are two special features which you may use when modeling with the QUAD4, QUAD8, TRIA3, and TRIA6 isoparametric elements. These are described in the following sections.

5.3.4.1 Rotational Stiffness Option

In some cases, it is possible for singularities to remain in your model after the AUTOSPC feature has been executed. This might occur, for example, when a BAR element is attached normal to the surface of a plate. In such cases, you may elstically support this singularity by using the parameter:

PARAM	**K6ROT**	*kval*							

If you specify a *kval* which is less than zero, then the rotational stiffness for each GRID point which connects to one of the isoparametric elements will be given the value:

$$K = 10^{-6} \cdot AtG \cdot kval$$

where A is the element area, t its thickness, and G the shear modulus.

5.3.4.2 QUAD4 and QUADR Warping Check

Warped QUAD4 or QUADR elements, those for which the GRID points are non-coplanar, may result in poor solution results. You may request that **UAI/*NASTRAN*** check for specific warping values by using the Executive Control command **DATACHECK**:

```
DATACHECK  QUADWARP = { IGNORE                                    }
                      { FATAL , RATIO = warp                      }
                      { WARNING , RATIO = warp , MAXWARN = nwarn  }
```

The *warp* value specifies the ratio of warping that you will allow. The warping value is defined as:

$$warp = \frac{d}{\sqrt{A}}$$

where d is the distance of the GRID points from the mean plane of the element, and A is the element area as projected on the mean plane. If you select **FATAL**, then elements whose *warp* exceeds the specified value will result in the program terminating. For example, to cause warping which exceeds 0.25 to result in fatal errors, you would use the command:

```
DATACHECK QUAD4WARP = FATAL , RATIO = 0.25
```

You may use the **WARNING** option to diagnose the problem without terminating execution. In this case, **MAXWARN** indicates the maximum number of warning messages that will be printed. The default selection, warping value, and maximum number of warning messages are given in the Preference file. You may use the Executive Control command **PRINT PREFERENCES** to confirm the current default.

5.3.4.3 Negative Volume Check

Certain poorly shaped isoparametric elements may result in the computation of negative volume during element matrix generation. You may allow such elements to complete their generation by again using the Executive Control command **DATACHECK**:

```
DATACHECK { PLATEGEOM } = { IGNORE                                          }
          { SOLIDGEOM }   { FATAL , PERCENT = negvol                        }
                          { WARNING , PERCENT = negvol , MAXWARN = nwarn    }
```

In both cases, *negvol* represents the percentage of negative volume (**0.0-100.0**) allowed for an element. For example, to request that warning messages be issued for all elements with 20% or more negative volume, you use the command:

```
DATACHECK PLATEGEOM = WARNING,PERCENT=20. SOLIDGEOM = WARNING,PERCENT = 20.
```

The default *negvol*s are given in the Preference file. You may use the Executive Control command **PRINT PREFERENCES** to confirm the current default.

5.3.4.4 Plate Offset Data Check

When using large numbers of offsets for QUAD or TRIA elements, the possibility of an error may become great. The Executive Control command **DATACHECK** may once again be used to request that **UAI/*NASTRAN*** check for unrealistic offset values:

```
DATACHECK PLATEOFFSETS = { IGNORE                                   }
                         { FATAL , RATIO = ratio                    }
                         { WARNING , RATIO = ratio , MAXWARN = nwarn }
```

The *ratio* argument specifies the maximum ratio of the offset of the element to its thickness that will be considered reasonable. You may have elements whose *ratio* exceeds the specified value **IGNORE**d, or you may have **WARNING** messages printed, or you may have the program terminated with the **FATAL** option. When selecting **WARNING**, you may also specify the maximum number of such messages. For example, to cause ratios which exceed 10.0 to result in fatal errors, you would use the command:

```
DATACHECK PLATEOFFSETS = FATAL RATIO = 10.0
```

The default selection, *ratio* value, and maximum number of warning messages are given in the Preference file. You may use the Executive Control command **PRINT PREFERENCES** to confirm the current default.

5.3.4.5 Plate Element Aspect Ratio Check

Badly distorted elements may often be identified by a measure of their aspect ratio. You may trap such elements with the Executive Control command **DATACHECK**:

```
DATACHECK ASPECTRATIO = { IGNORE                                   }
                        { FATAL , RATIO = ratio                    }
                        { WARNING , RATIO = ratio , MAXWARN = nwarn }
```

The *ratio* argument specifies the maximum aspect ratio that will be considered reasonable. For the QUAD4 element, two calculations are performed. The first is the ratio of the average lengths of the opposite sides as measured in the element coordinate system. The second is the ratio of the diagonals of the element. The maximum of these ratios is taken as the error measure. For the TRIA3 element, the measure is defined as the ratio of the longest side to the perpendicular distance from the longest side to the opposite GRID point. You may have elements whose *ratio* exceeds the specified value **IGNORE**d, or you may have **WARNING** messages printed, or you may have the program terminated with the **FATAL** option. When selecting **WARNING**, you may also specify the maximum number of such messages. For example, to cause ratios which exceed 20.0 to result in fatal errors, you would use the command:

```
DATACHECK ASPECTRATIO = FATAL RATIO = 20.0
```

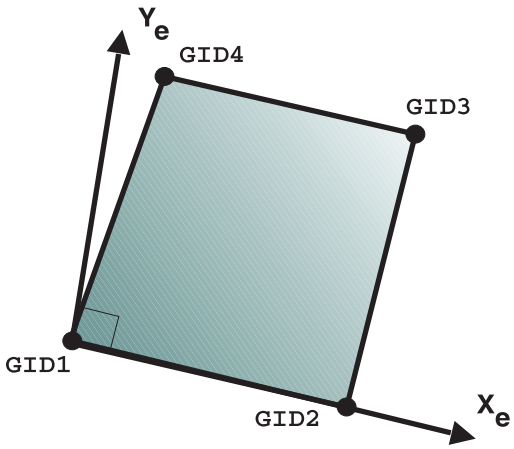
The default selection, *ratio* value, and maximum number of warning messages are given in the Preference file. You may use the Executive Control command **PRINT PREFERENCES** to confirm the current default.

5.3.5 The SHEAR and TWIST Panels

There are two specialty plate elements, the SHEAR and the TWIST. The SHEAR panel element is designed to resist tangential shearing along its edges. In normal circumstances it does not resist any membrane forces or out-of-plane loads. The TWIST panel resists pure moments.

Both elements are defined by four GRID points. Their connections are specified by **CSHEAR** and **CTWIST** Bulk Data entries, respectively. These connections reference corresponding **PSHEAR** and **PTWIST** property entries. Figure 5-14 illustrates the element geometry and coordinate system for both of these elements.

Figure 5-14. SHEAR AND TWIST GEOMETRY



5.3.5.1 Element Solution Results

For each of these elements you may obtain element stresses and forces. These are requested in the normal manner using the **STRESS** and **FORCE** Case Control commands. The element forces for these elements are shown in Figures 5-15 and 5-16. The force at each corner of the SHEAR panel is resolved into its three components. Note that the *kick* forces, K_1, K_2, K_3, and K_4 occur only for non-planar elements. For the TWIST panel, moments are given at each corner. Not that these are couples having the same magnitude at opposite corners. Stresses for the SHEAR element include the average and maximum shear, and those for the TWIST the average and maximum bending stress. Sample results are shown in Table 5-8.

Figure 5-16. SHEAR PANEL FORCES

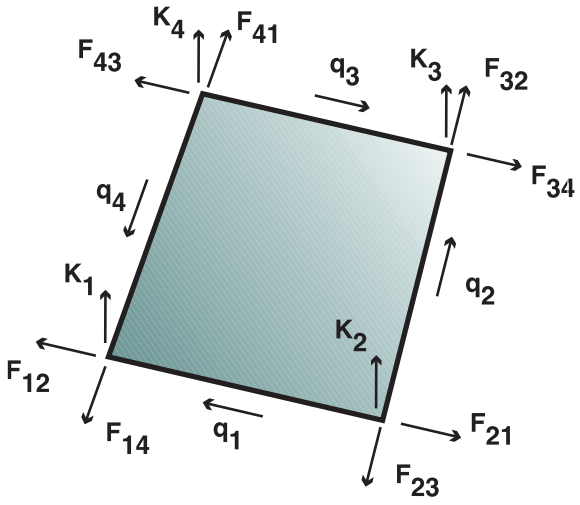


Figure 5-15. TWIST PANEL FORCES

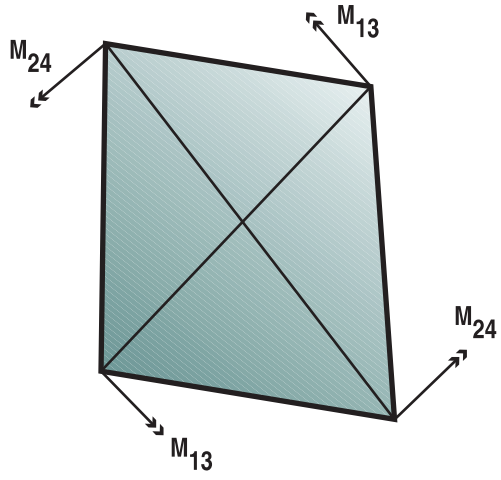


Table 5-8. SHEAR AND TWIST SOLUTION RESULTS

a. SHEAR Element Forces

```
          F O R C E S   A C T I N G   O N   S H E A R   P A N E L   E L E M E N T S   ( C S H E A R )

          ====== POINT  1 ======      ====== POINT  2 ======      ====== POINT  3 ======      ====== POINT  4 ======
ELEMENT    F-FROM-4      F-FROM-2      F-FROM-1      F-FROM-3      F-FROM-2      F-FROM-4      F-FROM-3      F-FROM-1
  ID              KICK-1       SHEAR-12       KICK-2       SHEAR-23       KICK-3       SHEAR-34       KICK-4       SHEAR-41

    401 -5.55112E-16  -1.11022E-15   1.11022E-15   5.55112E-16  -5.55112E-16  -1.11022E-15   1.11022E-15   5.55112E-16
              0.0          -2.22045E-16   0.0          -2.22045E-16   0.0          -2.22045E-16   0.0          -2.22045E-16

    402 -6.93889E-16  -1.38778E-15   1.38778E-15   6.93889E-16  -6.93889E-16  -1.38778E-15   1.38778E-15   6.93889E-16
              0.0          -2.77556E-16   0.0          -2.77556E-16   0.0          -2.77556E-16   0.0          -2.77556E-16
```

b. TWIST Element Forces

```
                              F O R C E S   I N   T W I S T   P A N E L S      ( C T W I S T )
ELEMENT          MOMENT         MOMENT                      ELEMENT          MOMENT         MOMENT
  ID.            PTS 1,3        PTS 2,4                       ID.            PTS 1,3        PTS 2,4
    501        0.0            0.0                             502        0.0            0.0
    503        0.0            0.0
```

c. SHEAR Element Stresses

```
                              S T R E S S E S   I N   S H E A R   P A N E L S      ( C S H E A R )
ELEMENT            MAX            AVG        SAFETY         ELEMENT            MAX            AVG        SAFETY
  ID.            SHEAR          SHEAR       MARGIN            ID.            SHEAR          SHEAR       MARGIN
    401        2.220446E-16  -2.220446E-16                     402        2.775558E-16  -2.775558E-16
    403        3.330669E-16   3.330669E-16
```

d. TWIST Element Stresses

```
                              S T R E S S E S   I N   T W I S T   P A N E L S      ( C T W I S T )
ELEMENT            MAX            AVG        SAFETY         ELEMENT            MAX            AVG        SAFETY
  ID.                                         MARGIN            ID.                                      MARGIN
    501        0.0            0.0                             502        0.0            0.0
    503        0.0            0.0
```

5

5.3.6 Summary of Bulk Data for the Two-Dimensional Elements

The following table summarizes the various Bulk Data entries frequently used when modeling with two-dimensional elements. Detailed descriptions of these Bulk Data entries are found in Chapter 7 of the ***User's Reference Manual***.

ELEMENT	BULK DATA ENTRIES			
	CONNECTION	PROPERTY		MATERIALS
		HOMOGENEOUS MATERIALS	COMPOSITE MATERIALS	
TRIA3	CTRIA3	PSHELL	PCOMP PCOMP1 PCOMP2	MAT1 MAT2 MAT8
QUAD4	CQUAD4			
TRIA6	CTRIA6			
QUAD8	CQUAD8			
TRIAR	CTRIAR			
QUADR	CQUADR			
SHEAR	CSHEAR	PSHEAR		MAT1
TWIST	CTWIST	PTWIST		

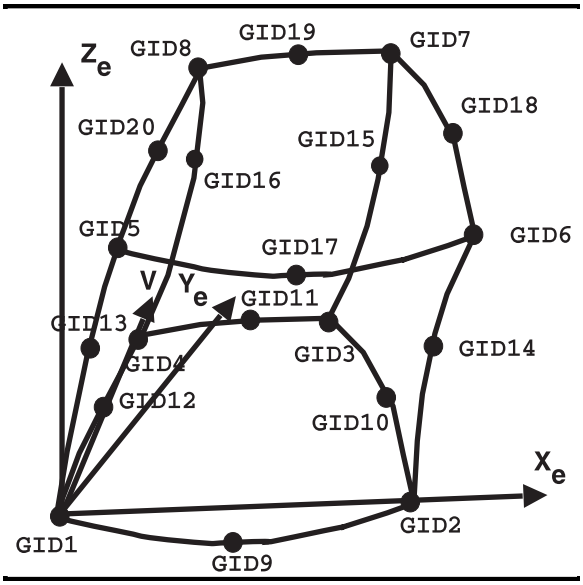
5.4 THREE-DIMENSIONAL ELEMENTS

General three-dimensional structures may be modeled using the family of quadratic isoparametric solid elements described in this Section.

5.4.1 The HEXA, PENTA and TETRA Elements

The modeling of complex three-dimensional solid structures is facilitated by three **UAI/*NASTRAN*** elements: the twenty node hexahedron, HEXA; the fifteen node pentahedron PENTA; and the ten node tetrahedron, TETRA. These elements are quadratic isoparametric finite elements. The connection Bulk Data entries for these elements are **CHEXA**, **CPENTA**, and **CTETRA**, respectively. You may omit any or all of the mid-side GRID points when using these elements. As in the case of other isoparametric quadratic elements, the geometry and stiffness characteristics of these elements vary depending upon the number of GRID points defining them.

Figure 5-17. HEXA ELEMENT GEOMETRY



The element coordinate systems for the solid elements are shown in Figures 5-17, 5-18, and 5-19. In each case, the element X-axis is defined by the vector from **GID1** to **GID2**. The Z-axis is then formed by the cross product of the x-axis with the vector from **GID1** to **GID4**, in the case of HEXA, or from **GID1** to **GID3** in the case of PENTA and TETRA. The cross product of the X-axis and the Z-axis then gives the Y-axis.

Figure 5-18. PENTA ELEMENT GEOMETRY

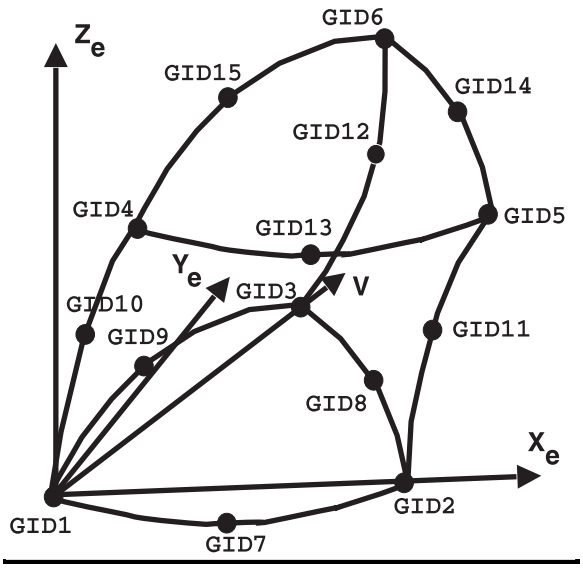
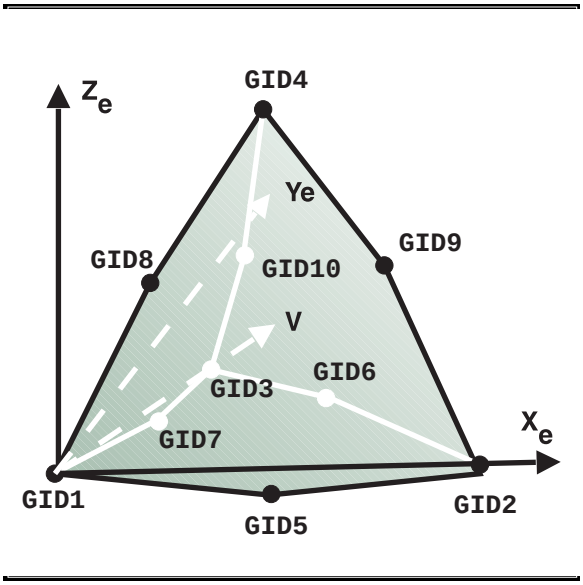


Figure 5-19. TETRA ELEMENT GEOMETRY



Each solid element references a **PSOLID** Bulk Data entry:

PSOLID	PID	MID	MCSID		SCSID				

This entry, which includes the property identification number **PID**, selects the material property, **MID**, of the element and may optionally specify independently defined material, **MCSID**, and stress, **SCSID**, coordinate systems. The material identifier, **MID**, may reference either isotropic, **MAT1**, or anisotropic, **MAT9**, material properties.

The solid elements perform best for moderately thick to thick continua. Where bending is important, midside nodes are recommended. Generally, the accuracy of solution is better facilitated by using midside modes rather than by increasing the number of elements. Several elements with reasonable aspect ratios should be used through the thickness of a structure if you are modeling bending and shear effects. Proper modeling will produce cross sectional warping under shear, as predicted by theory. If plate or beam theoretical results are the objective, then the appropriate elements should be used.

5.4.1.1 Element Stress Results

With the exception of the equivalent beam forces described in the next section, there are no element forces, available for the HEXA, PENTA or TETRA. The element stresses, which are selected using the **STRESS** Case Control command, are given at the vertex points and the centroid of each element. The three-dimensional solution results are normal and shear stresses, principal stresses, the direction cosine from the element coordinate system to the principal stress system, and the mean (hydrostatic) pressure. You may request either the von Mises equivalent stress or the octahedral shear stress. This latter option is selected with the keyword **MAXSHEAR** of the **STRESS** command as in the case of the plate elements. The stresses are available in complex form for Frequency Response analyses.

The three principal stresses, σ_p, are found by solving the cubic stress equation:

$$\sigma_p^3 - C_1\sigma_p^2 + C_2\sigma_p - C_3 = 0$$

where the constants C_i are functions of the normal and shear stress components and are given by:

$$C_1 = \sigma_x + \sigma_y + \sigma_z$$

$$C_2 = \sigma_x\sigma_y + \sigma_y\sigma_z + \sigma_x\sigma_z - \tau_{xy}^2 - \tau_{yz}^2 - \tau_{xz}^2$$

$$C_3 = \sigma_x\sigma_y\sigma_z + 2\tau_{xy}\tau_{yz}\tau_{xz} - \sigma_x\tau_{yz}^2 - \sigma_y\tau_{xz}^2 - \sigma_z\tau xy^2$$

Then, for each value of principal stress, σ_p, computed, the direction cosines l_x, l_y, and l_z are calculated using the relation:

$$\begin{bmatrix} \sigma_x - \sigma_p & \tau_{xy} & \tau_{xy} \\ \tau_{xy} & \sigma_y - \sigma_p & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \sigma_z - \sigma_p \end{bmatrix} \begin{Bmatrix} l_x \\ l_y \\ l_z \end{Bmatrix} = \mathbf{0}$$

Table 5-9. STRESS RESULTS FOR SOLID ELEMENTS

S T R E S S E S I N H E X A H E D R O N S O L I D E L E M E N T S (H E X A)

ELEMENT-ID	CORNER GRID-ID	------CENTER AND CORNER POINT STRESSES--------- NORMAL	SHEAR	PRINCIPAL	DIR. COSINES -A- -B- -C-	MEAN PRESSURE	VON MISES
8401		OUTPUT STRESS COORDINATE SYSTEM IS ELEMENT. ELEMENT HAS 8 GRID POINTS.					
	CNTR	X 9.134616E-03	XY 5.349738E-05	A 7.500000E-02	LX 0.00 0.71 0.71	-3.108974E-02	6.586545E-02
		Y 9.134616E-03	YZ 0.000000E+00	B 9.081118E-03	LY 0.00-0.71 0.71		
		Z 7.500000E-02	ZX 0.000000E+00	C 9.188113E-03	LZ 1.00 0.00 0.00		
	8401	X 4.375000E-02	XY 0.000000E+00	A 2.246604E-01	LX 0.01 0.53 0.85	-1.040318E-01	1.809429E-01
		Y 4.375000E-02	YZ 2.907764E-03	B 4.368501E-02	LY 0.02 0.85-0.53		
		Z 2.245954E-01	ZX 1.817352E-03	C 4.375000E-02	LZ 1.00-0.02 0.00		
	8402	X 4.375000E-02	XY 0.000000E+00	A 7.563009E-02	LX 0.06 0.00 1.00	-4.180154E-02	6.025436E-02
		Y 6.250000E-03	YZ -2.907764E-03	B 6.127798E-03	LY -0.04-1.00 0.00		
		Z 7.540463E-02	ZX 1.817352E-03	C 4.364674E-02	LZ 1.00-0.04-0.06		
	8404	X 6.250000E-03	XY 0.000000E+00	A 6.393742E-03	LX 0.53 0.02 0.85	2.096821E-02	8.187033E-02
		Y 6.250000E-03	YZ -2.907764E-03	B -7.554837E-02	LY 0.85 0.04-0.53		
		Z -7.540463E-02	ZX -1.817352E-03	C 6.250000E-03	LZ -0.04 1.00 0.00		
	8403	X 6.250000E-03	XY 0.000000E+00	A 7.571667E-02	LX 0.03 1.00 0.00	-4.180154E-02	6.025436E-02
		Y 4.375000E-02	YZ 2.907764E-03	B 6.202118E-03	LY -0.09 0.00 1.00		
		Z 7.540463E-02	ZX -1.817352E-03	C 4.348584E-02	LZ -1.00 0.03-0.09		
	8405	X -1.177885E-02	XY 1.069948E-04	A 2.246451E-01	LX 0.01 0.66 0.75	-6.701256E-02	2.364489E-01
		Y -1.177885E-02	YZ 2.907764E-03	B -1.188905E-02	LY 0.01-0.75 0.66		
		Z 2.245954E-01	ZX 1.817352E-03	C -1.171839E-02	LZ 1.00 0.00-0.01		
	8406	X -1.177885E-02	XY 1.069948E-04	A 7.555176E-02	LX 0.02 1.00 0.02	-2.064770E-02	8.281293E-02
		Y -1.682692E-03	YZ -2.907764E-03	B -1.181951E-02	LY -0.04-0.02 1.00		
		Z 7.540463E-02	ZX 1.817352E-03	C -1.789155E-03	LZ 1.00-0.02 0.04		
	8408	X -1.682692E-03	XY 1.069948E-04	A -1.421441E-03	LX 0.64 0.02 0.77	2.625667E-02	7.396101E-02
		Y -1.682692E-03	YZ -2.907764E-03	B -7.556356E-02	LY 0.77 0.04-0.64		
		Z -7.540463E-02	ZX -1.817352E-03	C -1.785005E-03	LZ -0.05 1.00 0.01		
	8407	X -1.682692E-03	XY 1.069948E-04	A 7.554405E-02	LX 0.02 0.02 1.00	-2.064770E-02	8.281293E-02
		Y -1.177885E-02	YZ 2.907764E-03	B -1.187848E-02	LY -0.03-1.00 0.02		
		Z 7.540463E-02	ZX -1.817352E-03	C -1.722485E-03	LZ -1.00 0.03 0.02		

The mean pressure is computed using:

$$\bar{\sigma} = -\frac{\sigma_x + \sigma_y + \sigma_z}{3}$$

The octahedral stress, if requested, is computed from:

$$\sigma_o = \frac{1}{3}\left[(\bar{\sigma}+\sigma_1)^2 + (\bar{\sigma}+\sigma_2)^2 + (\bar{\sigma}+\sigma_3)^2 \right]^{1/2}$$

Finally, the von Mises stress, the default output, is computed from the octahedral stress using:

$$\sigma_{vm} = \frac{1}{\sqrt{2}}\left[(\sigma_1-\sigma_2)^2 + (\sigma_2-\sigma_3)^2 + (\sigma_3-\sigma_1)^2 \right]^{1/2}$$

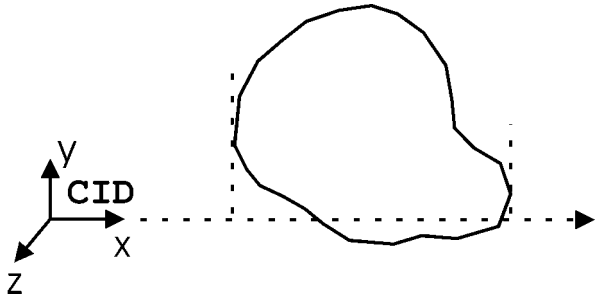
Example stress output for the solid elements is shown in Table 5-9.

5.4.1.2 Equivalent Beam Forces

You may define equivalent beam elements composed of collections of solid elements for which beam forces (moments, shears, axial loads, and torques) may be computed. This is done by defining one or more special elements using the Bulk Data entries **BMFORCE** or **BMFORC1**. The format of the **BMFORCE** entry is:

BMFORCE	EID	CID							-cont-
-cont-	X1	X2	X3	CONTINUES WITH LIST OF VALUES					-cont-
-cont-	EID1	EID2	EID3	CONTINUES WITH LIST OF VALUES					-cont-

The pseudo-beam element is created by a group of solid elements specified by the **EIDi** fields. The element coordinate system is given by **CID**. The data recovery stations are defined by a set of real values defined **Xi**. The three-dimensional stress field of the element set given by **EIDi** is then converted into equivalent beam forces at normal sections defined by the station values. For stations outside the projection of the **EIDi** elements on the **CID** x-axis, no beam forces can be, or will be, computed. This is illustrated in the following figure:



The element identification numbers are defined by a list of integer values. At least one **EIDi** must be input, and the continuation is repeated until all values have been entered. Note that the first **EIDi** must begin in Field 2 of the continuation entry following the one which contains the last **Xi** value.

5

Th second Bulk Data entry, **BMFORC1**, for defining an equivalent beam has the format:

BMFORC1	EID	CID	SSID	ESID					

Rather than specifying lists explicitly, this entry allows you to select a set of stations, **SSID**, that are in turn defined by **SETR** Bulk Data entries, and a set of element identification numbers, **ESID**, which are defined by **SETI** or **SETOP** Bulk Data entries.

The request for computation of the forces is made using the Case Control command **BMFORCE**, which is described in section 5.9.1.2. An example of the force output, which is given at each specified station, is shown in Table 5.10.

Table 5-10. EQUIVALENT BEAM FORCES

F O R C E S I N E Q U I V A L E N T B E A M S (B M F O R C E)

ELEMENT-ID	STAT DIST	- BENDING MOMENTS - PLANE 1	PLANE 2	- WEB SHEARS - PLANE 1	PLANE 2	AXIAL FORCE	TORQUE
111							
	0.00000E+00	9.53674E-07	-3.00000E+01	0.00000E+00	-2.00000E+01	0.00000E+00	0.00000E+00
	1.00000E+00	1.90735E-06	-2.66667E+01	0.00000E+00	-2.00000E+01	0.00000E+00	0.00000E+00
	1.50000E+00	9.53674E-07	-2.33333E+01	0.00000E+00	-2.00000E+01	0.00000E+00	0.00000E+00
	2.50000E+00	9.53674E-07	-1.66667E+01	0.00000E+00	-2.00000E+01	0.00000E+00	0.00000E+00
	3.30000E+00	0.00000E+00	-1.13333E+01	0.00000E+00	-2.00000E+01	0.00000E+00	0.00000E+00
	3.70000E+00	2.38419E-07	-8.66667E+00	0.00000E+00	-2.00000E+01	0.00000E+00	0.00000E+00
	4.10000E+00	2.38419E-07	-6.33333E+00	0.00000E+00	-2.00000E+01	0.00000E+00	0.00000E+00
	4.40000E+00	2.38419E-07	-5.33333E+00	0.00000E+00	-2.00000E+01	0.00000E+00	0.00000E+00
	4.50000E+00	3.57628E-07	-5.00000E+00	0.00000E+00	-2.00000E+01	0.00000E+00	0.00000E+00
	4.70000E+00	-1.19209E-07	-4.33333E+00	0.00000E+00	-2.00000E+01	0.00000E+00	-2.98023E-08
222							
	0.00000E+00	9.53674E-07	-3.00000E+01	0.00000E+00	-2.00000E+01	0.00000E+00	0.00000E+00
	3.00000E-01	1.90735E-06	-2.90000E+01	0.00000E+00	-2.00000E+01	0.00000E+00	0.00000E+00
	6.00000E-01	9.53674E-07	-2.80000E+01	0.00000E+00	-2.00000E+01	0.00000E+00	0.00000E+00
	9.00000E-01	1.90735E-06	-2.70000E+01	0.00000E+00	-2.00000E+01	0.00000E+00	0.00000E+00
	1.00000E+00	1.90735E-06	-2.66667E+01	0.00000E+00	-2.00000E+01	0.00000E+00	0.00000E+00
	3.30000E+00	0.00000E+00	-1.13333E+01	0.00000E+00	-2.00000E+01	0.00000E+00	0.00000E+00
	3.70000E+00	2.38419E-07	-8.66667E+00	0.00000E+00	-2.00000E+01	0.00000E+00	0.00000E+00
	4.10000E+00	2.38419E-07	-6.33333E+00	0.00000E+00	-2.00000E+01	0.00000E+00	0.00000E+00
	4.40000E+00	2.38419E-07	-5.33333E+00	0.00000E+00	-2.00000E+01	0.00000E+00	0.00000E+00
	4.60000E+00	1.19209E-07	-4.66667E+00	0.00000E+00	-2.00000E+01	0.00000E+00	0.00000E+00
	4.70000E+00	-1.19209E-07	-4.33333E+00	0.00000E+00	-2.00000E+01	0.00000E+00	-2.98023E-08

5.4.2 Summary of Bulk Data for the Three-Dimensional Elements

The following table summarizes the various Bulk Data entries frequently used when modeling with three-dimensional elements. Detailed descriptions of these Bulk Data entries are found in Chapter 7 of the ***User's Reference Manual***.

ELEMENT	BULK DATA ENTRIES		
	CONNECTION	PROPERTY	MATERIALS
HEXA	CHEXA	PSOLID	MAT1 MAT1NL[1] MAT1NL1[1] MAT4[2] MAT5[2] MAT9 MATF[3]
PENTA	CPENTA		
TETRA	CTETRA		
EQUIVALENT BEAM	BMFORCE BMFORC1		

1. For nonlinear material analyses only.
2. For heat transfer analyses only.
3. For fluid-structure interaction analyses only.

5.5 THE SCALAR ELEMENTS

Scalar elements are connected between pairs of degrees of freedom defined at either SCALAR or GRID points, or between one degree of freedom and ground. Available scalar elements include springs, masses and viscous dampers. Scalar spring elements are useful for representing elastic properties that cannot be conveniently modeled with the usual structural elements. Scalar masses are useful for the selective representation of inertia properties, for example, to model a concentrated mass which is effectively isolated for motion in one direction only. Viscous damping is described in detail in Chapter 9. It is expected that you will use these elements only when the normal finite element library can not model a specific behavior.

Scalar elements may be connected to ground without the use of constraint specification. Grounded connections are indicated on the connection entry by leaving the appropriate SCALAR point identification number blank.

5.5.1 Scalar Springs

There are several forms of data entry for scalar springs. The first is the **CELAS1** Bulk Data entry. The properties include the spring constant, or stiffness value, a nonuniform structural damping coefficient, and a stress coefficient. You may include the properties of the spring directly on the connection entry by using the **CELAS2**. The **CELAS3** and **CELAS4** entries perform the same function when the elements connect only SCALAR points. However, no damping or stress coefficients are available with the ELAS4 element.

To understand what effect the scalar spring element has on the model, consider the definition of an ELAS1 element:

CELAS1	EID	PID	GID1	DOF1	GID2	DOF2			

The element connects two individual degrees of freedom, **DOF1** and **DOF2**, or, alternately, one degree of freedom which is grounded. The element stiffness matrix for the first case is:

$$\boldsymbol{K}_{EID} = \begin{array}{c} \\ dof_1 \\ dof_2 \end{array} \begin{array}{c} \begin{array}{cc} dof_1 & dof_2 \end{array} \\ \begin{bmatrix} K & -K \\ -K & K \end{bmatrix} \end{array}$$

where K is the spring constant specified on the **PELAS** entry. This element matrix is then assembled into the global stiffness matrix in the normal manner. If the element is grounded, then the value of K is simply added to the row and column position represented by the single degree of freedom. This is, if fact, true for the scalar damping and mass elements as well.

If you are performing nonlinear analyses, you may use a nonlinear spring element, the ELASNL. This element is defined using **CELASNL** Bulk Data entries The ELASNL spring element allows the specification of a constant stiffness, or it selects a table which defines a force deflection curve. This table is input with **TABLENL** Bulk Data entries. If the ELASNL is used in a linear analysis, then the initial slope of the table data will define the element stiffness.

5.5.2 Scalar Dampers

Viscous dampers are modeled using the family of DAMPi elements. These elements are defined with **CDAMPi** Bulk Data entries in a manner similar to the scalar spring definitions. The associated **PDAMP** entry contains only a value for the Scalar damper. The element damping matrix

defined by these elements has the same form as those for the spring elements. The **VISC** element, defined with **CVISC** and **PVISC** Bulk Data entries, defines damping between two distinct GRID points, rather than scalar degrees of freedom. It allows you to define damping in the three translational DOF, the three rotational DOF, or both. For example, consider the **PVISC** Bulk Data entry:

PDAMP	PID	B1	B2						

The element damping matrix is assembled as:

$$\boldsymbol{B}_{EID} = \begin{bmatrix} \boldsymbol{B}_{11} & 0 \\ 0 & \boldsymbol{B}_{22} \end{bmatrix}$$

where each of the partitions $\boldsymbol{B}_{ii}$ have the form:

$$\boldsymbol{B}_{EID} = \begin{bmatrix} B1 & 0 & 0 & 0 & 0 & 0 \\ 0 & B1 & 0 & 0 & 0 & 0 \\ 0 & 0 & B1 & 0 & 0 & 0 \\ 0 & 0 & 0 & B2 & 0 & 0 \\ 0 & 0 & 0 & 0 & B2 & 0 \\ 0 & 0 & 0 & 0 & 0 & B2 \end{bmatrix}$$

B1 and **B2** are the damping coefficients that you enter on the Bulk Data entry. This matrix is then transformed to the global coordinate system.

5

5.5.3 Scalar Masses

There are two types of scalar mass elements. The first is the MASSi family, and the second is the CONMi family. A scalar mass may be defined using any of the **CMASSi** and, where applicable, PMASS Bulk Data. Again, the element mass matrix defined by these elements has the same form as the stiffness and damping matrices generated by the corresponding ELASi and DAMPi elements. While scalar springs and dampers function in an obvious manner, this is not always the case with the scalar mass elements. The equation below illustrates the matrix equations of motion generated by **UAI*/NASTRAN*** for a model with two SCALAR points and a single scalar spring, damper and mass element.

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{B}\dot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{F}$$

which, for the indicated example, may be written:

$$\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} \ddot{u}_1 \\ \ddot{u}_2 \end{Bmatrix} + \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} \dot{u}_1 \\ \dot{u}_2 \end{Bmatrix} + \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix}$$

You might not expect the mass matrix to be coupled. If you wish to insure that the mass matrix is diagonal, then two scalar mass elements should be used. Each scalar mass must reference only one of the SCALAR points — the second point must be grounded.

CONM1 and **CONM2** Bulk Data are used to define lumped masses at GRID points. The **CONM1** allows you to define a complete mass matrix of the form:

$$M = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} & M_{15} & M_{16} \\ & M_{22} & M_{23} & M_{24} & M_{25} & M_{26} \\ & & M_{33} & M_{34} & M_{35} & M_{36} \\ & & & M_{44} & M_{45} & M_{46} \\ & & -Sym- & & M_{55} & M_{56} \\ & & & & & M_{66} \end{bmatrix}$$

While **CONM2** is used to define a mass matrix of the form:

$$M = \begin{bmatrix} M & & & 0 & V_3 M & -V_2 M \\ & M & & -V_1 M & 0 & V_1 M \\ & & M & V_2 M & -V_1 M & 0 \\ & & & I_{11} & -I_{12} & -I_{13} \\ & & -Sym- & & I_{22} & -I_{32} \\ & & & & & I_{33} \end{bmatrix}$$

where the **Vi** are components of an offset vector and the **Iij** are mass moments of inertia.

5

5.5.4 Summary of Bulk Data for the Scalar Elements

The following table summarizes the various Bulk Data entries frequently used when modeling with scalar elements. Detailed descriptions of these Bulk Data entries are found in Chapter 7 of the ***User's Reference Manual***.

TO MODEL SCALAR:	USE ELEMENTS:	BULK DATA ENTRIES	
		CONNECTION	PROPERTY
STIFFNESS	ELAS1, ELAS3	CELAS1,CELAS3	PELAS
	ELAS2, ELAS4	CELAS2, CELAS4	
	ELASNL[1]	CELASNL	TABLENL[2]
DAMPING	DAMP1, DAMP3	CDAMP1,CDAMP3	PDMAP
	DAMP2, DAMP4	CDAMP2, CDAMP4	
	CVISC	CVISC	PVISC
MASS	MASS1, MASS3	CMASS1, CMASS3	PMASS
	MASS2, MASS4	CMASS2, CMASS4	
	CONM1,CONM2	CONM1, CONM2	

1.The ELASNL element may only be used in nonlinear analyses.

2.**TABLENL** data are used to define the element force-deflection curve.

5.6 SPECIALTY ELEMENTS

The previous sections of this Chapter have described the general purpose finite elements used to model most structures. There are, however, instances in which other modeling techniques may be more effective for you. **UAI/*NASTRAN*** provides a number of such features which are briefly discussed in this Section. Reference is made to other Chapters of this manual where detailed descriptions of the element and capabilities are found.

5.6.1 Axisymmetric Elements

Axisymmetric structures, and in limited cases fluids, may be modeled with **UAI/*NASTRAN***. There are two approaches available for this. The first is called Axisymmetric Harmonic analysis, and the second, Axisymmetric Ring analysis. These are briefly described in the following sections.

5.6.1.1 Axisymmetric Harmonic Elements

These special elements are not defined by GRID points, like the usual structural elements, but by *rings*. The solution is facilitated by the expansion of element shape functions as Fourier series. The available elements are a conical shell, the CONEAX, and triangular and quadrilateral solids, TRIAAX and TRAPAX. These elements may also be combined with other structural elements. Details on using the Axisymmetric Harmonic modeling techniques are described in Chapter 20.

5.6.1.2 Axisymmetric Ring Elements

The more traditional finite element approach defines axisymmetric rings by their connection GRID points. However, these elements only allow axisymmetric loading and are less general than the harmonic elements. Those available are a toroidal shell, TORDRG, that can also be used as a shell cap, and triangular and quadrilateral solid rings, TRIARG and TRAPRG. Axisymmetric Ring analysis is described in Chapter 21.

5.6.2 Rigid Elements

A number of so-called *rigid* elements are also available. These elements allow you to model infinitely rigid members by means of constraint equations. More appropriately, they are a tool that you use to define dependency relationships between motions within the model. As such they generate *multi-point constraint* equations. These are fully discussed in Chapter 6.

5.6.3 The General Element

The general element is a stiffness element connected to any number of degrees of freedom. It is most often used to model nonstructural components such as servomechanisms for which laboratory testing has provided stiffness or flexibility data. Two options are provided to define the characteristics of such an element.

The first option requires specifying the deflection influence coefficients for the element when it is supported in a non-redundant manner. The associated matrix of the restrained rigid body motions may be specified or it may be generated by **UAI/*NASTRAN***. The second option requires the stiffness matrix of the element to be defined directly. This stiffness matrix may represent the unsupported body, containing all the rigid body modes, or it may represent a

portion of the degrees of freedom from which some or all of the rigid body motions are deleted. In the latter case, an option is available to reintroduce the restrained rigid body terms, providing that the original support conditions were not redundant. 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 to high precision in order to preserve the rigid body properties of the element.

The force-displacement relationship for the general element, when written in the flexibility form, is:

$$\begin{Bmatrix} \boldsymbol{u}_i \\ \boldsymbol{f}_d \end{Bmatrix} = \begin{bmatrix} \boldsymbol{Z} & \boldsymbol{S} \\ -\boldsymbol{S}^T & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{f}_i \\ \boldsymbol{u}_d \end{Bmatrix}$$

where $\boldsymbol{Z}$ is the matrix of deflection influence coefficients for coordinates $\boldsymbol{u}_i$ when coordinates $\boldsymbol{u}_d$ are rigidly restrained. $\boldsymbol{S}$ is a rigid body matrix whose terms are the displacements $\boldsymbol{u}_i$ due to unit motions of the coordinates $\boldsymbol{u}_i$, when all $\boldsymbol{f}_i = \boldsymbol{0}$. $\boldsymbol{f}_i$ are the forces applied to the element at the $\boldsymbol{u}_i$ coordinates. $\boldsymbol{f}_d$ are the forces applied to the element at the $\boldsymbol{u}_d$ coordinates. They are assumed to be statically related to the $\boldsymbol{f}_i$ forces, so that they constitute a non-redundant set of reactions for the element.

In terms of stiffness coefficients, this relationship is:

$$\begin{Bmatrix} \boldsymbol{f}_i \\ \boldsymbol{f}_d \end{Bmatrix} = \begin{bmatrix} \boldsymbol{k} & -\boldsymbol{kS} \\ -\boldsymbol{S}^T\boldsymbol{k} & \boldsymbol{S}^T\boldsymbol{kS} \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_d \end{Bmatrix}$$

where all symbols have the same meaning as above and $\boldsymbol{k} = \boldsymbol{Z}^{-1}$, if $\boldsymbol{k}$ is nonsingular. It is permissible for $\boldsymbol{k}$ to be singular.

Data for the general element consists of: the $\boldsymbol{u}_i$ and $\boldsymbol{u}_d$ components, which may occur at either GRID points or SCALAR points; the terms of the $\boldsymbol{Z}$ matrix, or the term of the $\boldsymbol{k}$ matrix; and, optionally the terms of the $\boldsymbol{S}$ matrix.

The $\boldsymbol{S}$ matrix may be automatically generated. If so, the $\boldsymbol{u}_i$ and $\boldsymbol{u}_d$ components may occur only at GRID points, and there must be six or fewer $\boldsymbol{u}_d$ components that provide a non-redundant set of reactions for the element as a three-dimensional body. General elements are defined by the **GENEL** Bulk Data entry. No internal forces or other solution results are produced for the general element.

5.6.4 The GAP Element

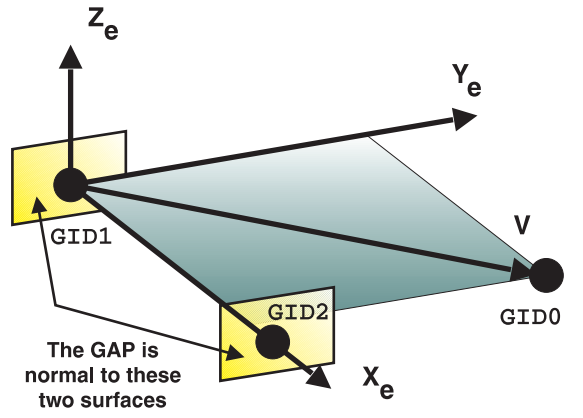
The GAP element is designed for the analysis of structures with *contact* problems. The GAP will carry load with essentially no relative deflection when a gap opening is closed, and it will not restrict motion when the gap is opened. Also, when closed, the GAP may exhibit friction effects.

The nonlinear behavior of the GAP is available when you use the **STATICS** Rigid Format with **APP NONLINEAR**. The GAP may also be used in linear analyses. In this case, the element stiffness properties are those that are defined for the initial state. The GAP has only stiffness properties. There are no mass or damping data generated by the GAP.

A GAP is defined between two GRID points, which may be coincident, using **CGAP** and **PGAP** Bulk Data entries. The GAP element orientation can be defined in several ways that depend on

the location of the GRID points. Consider Figure 5-20. When **GID1** and **GID2** are coincident, the element orientation and coordinate system may be defined in either of two ways. First, if no **CID** is specified, the basic coordinate system is used. If a **CID** is specified, then that coordinate system is used as the element coordinate system. In both cases the element X-axis is aligned with the coordinate system X-axis.

Figure 5-20. GAP ELEMENT GEOMETRY



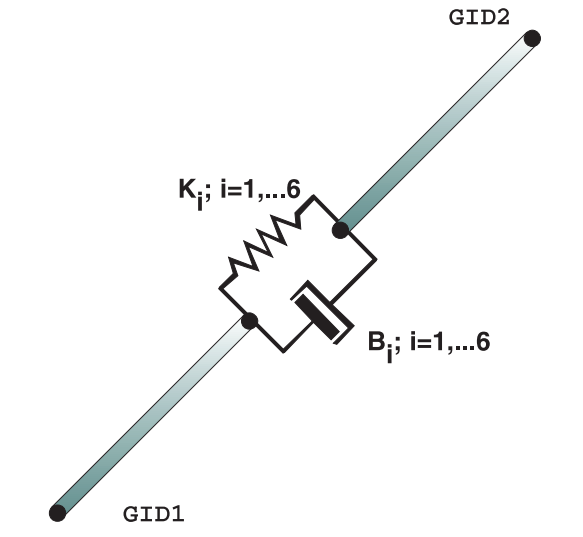
If **GID1** and **GID2** are not coincident, then the element coordinate system can be defined either by reference to the GRID point **GO**, as with the BEAM element, or with the vector **V**, defined by specified on the **CGAP** data entry.

GAP element data recovery is requested with the **FORCE** Case Control command. Element forces are output in the element coordinate system. Deformations of the GAP are also output in the element coordinate system. Deformations include the total relative motion in the GAP axial and shear directions. Additionally, the amount of shear motion after friction effects have been overcome is output under the heading of **SLIP**. More information about the GAP element and its use in nonlinear analyses is found in Chapter 18.

5.6.5 The BUSHing Element

The BUSH element is a combination scalar element which consists of six independent springs and six independent dampers. The springs and dampers are located between two inner GRID points which are coincident and are at the middle of the the bushing element, as shown in Figure 5-21. These inner GRID points are automatically connected to outer GRID points by rigid links. The outer GRID points may also be coincident. The springs and dampers are located at the center of the element. This idealization technique, coupled with the automatically generated Rigid elements, avoids any free-body modeling errors. The BUSHing element may be used to model such components as bushings, shock absorbers, and hydraulic springs.

Figure 5-21. BUSH ELEMENT GEOMETRY



In Direct Frequency analysis, the BUSHing element stiffness and damping may be modelled as either constant or frequency-dependent functions. In nonlinear static analysis, the spring stiffnesses may be modelled with nonlinear force-deflection relationships in the manner of the ELASNL element. Details of these special features are given in Chapters 13 and 18. The BUSH element behaves linearly when used in all other analyses.

The BUSH element is defined with:

CBUSH	EID	PID	GID1	GID2	V1	V2	V3	CID	
					GID0				

The element orientation can be defined in several ways depending upon the location of the GRID points. When **GID1** and **GID2** are coincident, the element orientation and coordinate system may be defined in ome of two ways. First, if no **CID** is specified, the Basic Coordinate system is used an the element coordinate system. If **CID** is specified, the coordinate system **CID** is used to define the element coordinate system. With either of the two cases, the element x, y and z axes are the same as the x, y and z axes of the **CID** (or Basic) coordinate system.

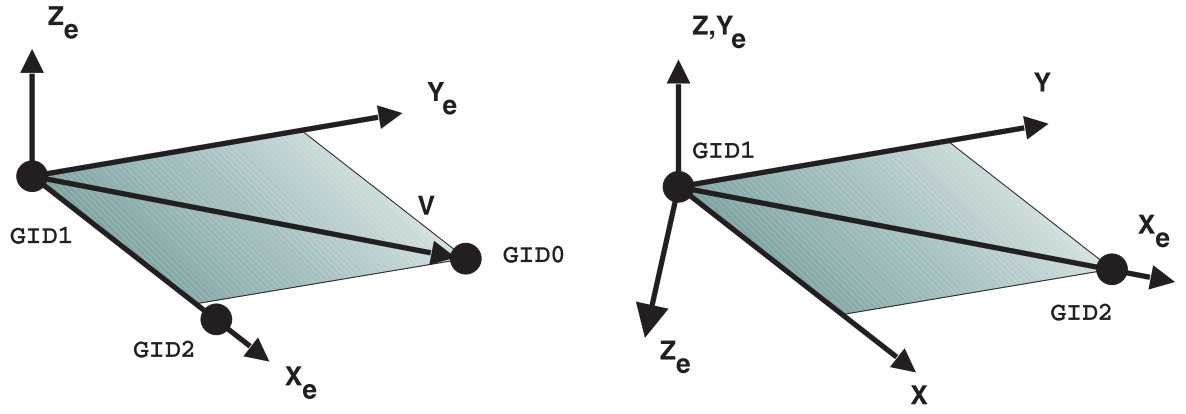
When **GID1** and **GID2** are not coincident, and if the **CID**≥ 0, then fields 6-8 (**V1,V2,V3**) are ignored. The coordinate system **CID** is used to define the element coordinate system as before.

If **GID1** and **GID2** are not coincident, and **CID** is not specified, then the element coordinate system is defined in the same manner as is done for the BAR, BEAM and GAP elements. This procedure requires the definition of a vector **V**, either by (1) specification of its components (**V1,V2,V3**) relative to **GID1** or by (2) defining the vector from **GID1** to **GID0**, as shown in Figure 5-22a.

If **CID** is not specified and neither (**V1,V2,V3**) nor **GID0** are input by the user, the element coordinate system is established as follows: The element x axis is defined as the line between **GID1** and **GID2**. The element y axis is the vector perpendicular to the element x axis and in the plane of the element x axis and one of the basic x,y,z axes which is close to being normal to the element x axis. The element z axis is then determined as the vector perpendicular to both the element x and y axes. This is shown in Figure 5-22b.

5

Figure 5-22. BUSH ELEMENT COORDINATE SYSTEMS



a. BUSH Element Orientation

b. Default Element Orientation

The element stiffness and damping matrices are calculated first by computing the 12x12 matrices at the middle of the element. For example, the stiffness matrix at the inner GRIDs is given by:

$$\boldsymbol{K}_e^{inner} = \begin{bmatrix} K1 & 0 & 0 & 0 & 0 & 0 & -K1 & 0 & 0 & 0 & 0 & 0 \\ 0 & K2 & 0 & 0 & 0 & 0 & 0 & -K2 & 0 & 0 & 0 & 0 \\ 0 & 0 & K3 & 0 & 0 & 0 & 0 & 0 & -K3 & 0 & 0 & 0 \\ 0 & 0 & 0 & K4 & 0 & 0 & 0 & 0 & 0 & -K4 & 0 & 0 \\ 0 & 0 & 0 & 0 & K5 & 0 & 0 & 0 & 0 & 0 & -K5 & 0 \\ 0 & 0 & 0 & 0 & 0 & K6 & 0 & 0 & 0 & 0 & 0 & -K6 \\ -K1 & 0 & 0 & 0 & 0 & 0 & K1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -K2 & 0 & 0 & 0 & 0 & 0 & K2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -K3 & 0 & 0 & 0 & 0 & 0 & K3 & 0 & 0 & 0 \\ 0 & 0 & 0 & -K4 & 0 & 0 & 0 & 0 & 0 & K4 & 0 & 0 \\ 0 & 0 & 0 & 0 & -K5 & 0 & 0 & 0 & 0 & 0 & K5 & 0 \\ 0 & 0 & 0 & 0 & 0 & -K6 & 0 & 0 & 0 & 0 & 0 & K6 \end{bmatrix}$$

The matrices are then transformed to the outer GRID points by using a rigid body transformation relation, $\boldsymbol{T}$, between displacements at the inner and outer GRIDs through:

$$\boldsymbol{K}_e = \boldsymbol{T}^T \boldsymbol{K}_e^{inner} \boldsymbol{T} \text{ and } \boldsymbol{B}_e = \boldsymbol{T}^T \boldsymbol{B}_e^{inner} \boldsymbol{T} \text{ and}$$

The BUSH element forces, stresses and strains are output in the element coordinate system at the middle of the element.

5.6.6 Fluid Elements

UAI/*NASTRAN* provides the capability to solve fluid-structure interaction problems. You may analyze both compressible and incompressible fluids. Chapter 23 describes which structural elements may be used, how fluids are modeled, and special elements used in Acoustic analyses.

5.7 THE ELEMENT MASS MATRIX FORMULATION

UAI/*NASTRAN* provides two formulations of the mass matrix for many finite elements in the library. The first of these is the ***lumped mass formulation***. This method places translational mass components at each GRID point defining the element. The second method is the ***consistent,*** or ***coupled, mass formulation***. In this method, the shape functions used by the elements for determining stiffness are also used to generate the mass matrix. This results in off-diagonal mass coupling terms.

Lumped masses are used unless you request consistent mass with the Bulk Data entry:

PARAM	**COUPMASS**	**1**							

Whenever you select consistent mass with this Bulk Data entry, all finite elements in the model which support this feature will produce a consistent mass matrix. Some elements do not include a consistent mass formulation, in which cases they only produce a lumped mass matrix. This is shown in the table below.

A second important **PARAM**eter affecting the mass matrix is:

PARAM	**WTMASS**	**FACTOR**							

The **FACTOR** is used as a multiplier to convert weight units to mass. You should use the **WTMASS PARAM**eter when you specify weight density on your material property Bulk Data entries.

Finally, there is also a special parameter, **BARTORM**, which allows you to include torsional mass terms for BAR elements.

TYPE	ELEMENT	LUMPED MASS	CONSISTENT MASS
ONE-DIMENSIONAL	ROD, CONROD	✓	
	TUBE	✓	
	BAR	✓	✓
	BEAM	✓	✓
	GAP		
TWO-DIMENSIONAL	SHEAR	✓	
	TWIST	✓	
	TRIA3, TRIA6	✓	✓
	QUAD4, QUAD8	✓	✓
THREE-DIMENSIONAL	TETRA	✓	✓
	PENTA	✓	✓
	HEXA	✓	✓
AXISYMMETRIC HARMONIC	CONEAX	✓	
	TRIAAX	✓	✓
	TRAPAX	✓	✓

5.8 AUTOMATICALLY GENERATED DATA

There are some special **UAI/*NASTRAN*** features that provide you with simplified input data entry. These include special entries for BAR and BEAM elements which have standard cross-sections, and entries for the definition of laminated composites which may be used with the plate and shell elements TRIA3, QUAD4, TRIA6, and QUAD8. These are described in the following sections.

5.8.1 Section Properties

The one-dimensional elements BAR and BEAM require the definition of section properties based on their cross-sectional characteristics. **UAI/*NASTRAN*** includes special input features that automatically compute these section properties for standard, often encountered, cross-sections. These cross-sections, shown in Figure 5-24, have been parameterized based on key dimensions. Specific input requirements and output results for the elements are given in Sections 5-2 and 5-5. You may request a printed summary of the section properties generated, and you may also request that equivalent **PBAR** Bulk Data entries be written on the **BULK** file by including the following **PARAM** Bulk Data entry in your input stream:

PARAM	**SECTION**	**PRINT**							
		PUNCH							
		BOTH							

Where the available options allow you to **PRINT** the properties, **PUNCH** them as **PBAR** Bulk Data entries, or **BOTH**. If you select the **PUNCH** option, then you must **ASSIGN** a file with **USE=BULK** in your Executive Control packet.

5.8.2 Laminated Composites

UAI/*NASTRAN* allows you to model complex laminated composites when using the plate and shell elements TRIA3, QUAD4, TRIA6, and QUAD8. Figure 5-23 illustrates the numbering scheme and coordinate systems used for such materials. The material layers are numbered consecutively from 1 to n beginning at the bottom of the element as shown in the Figure. The ply orientation angles, θ_n, are measured relative to the element material axis.

To model laminated composite materials, you may use **PCOMP**, **PCOMP1** or **PCOMP2** Bulk Data entries. The basic data required in all cases is the definition of the thickness, material property, and fiber orientation angle of each layer. The different data forms are provided for convenience of data entry.

Figure 5-23. LAMINATED COMPOSITE ORIENTATION

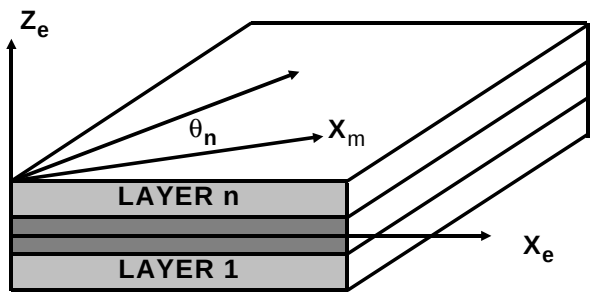
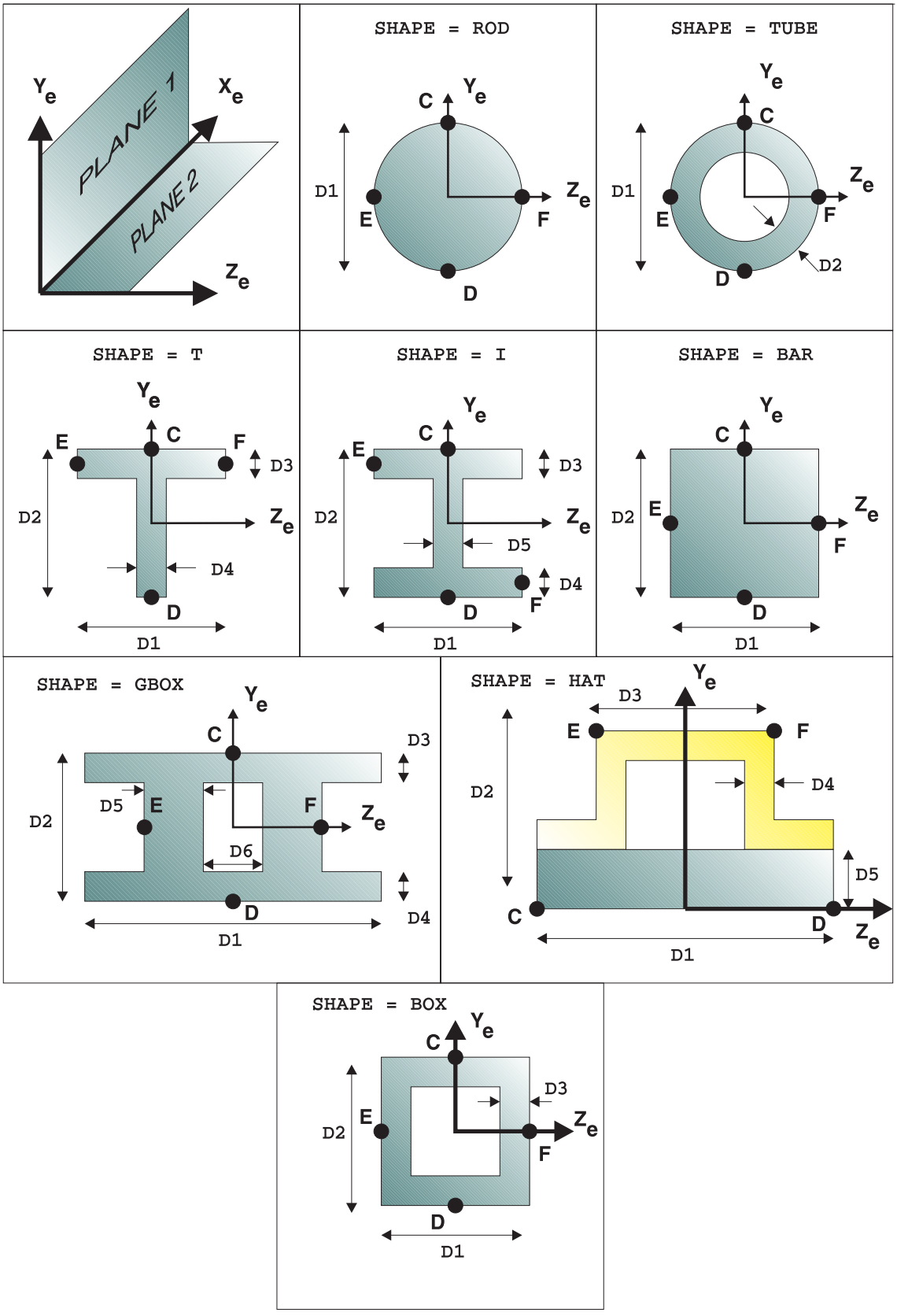


Figure 5-24. STANDARD CROSS SECTIONS



5

The most general composite definition is provided by the **PCOMP** data entry:

PCOMP	PID	ZO	NSM	SBOND	FT	TREF	GE	LAM	-cont-
-cont-	MID1	T1	θ1	SOUT1	MID2	T2	θ2	SOUT2	-cont-
-cont-	MID3	T3	θ3	SOUT3	*CONTINUES IN GROUPS OF 4*				

The **PID** field defines this particular property and it is referenced by any element using it. **ZO** defines the offset from the reference plane of the element, discussed earlier, which is used only when you do not place an offset value on the **CQUAD4** or **CTRIA3** Bulk Data entry. The **SBOND** field defines the maximum allowable shear stress of the composite bonding material. If this field is specified, the interlaminar shear failure indices are calculated. Similarly, you may request ply failure indices using the **FT** field. The **FT** field allows you to select from five failure theories which are used to compute failure indices on a ply-by-ply basis. These theories are: **HILL** Theory, **HOFF**man Theory, **TSAI**-Wu Theory, Maximum **STRESS** Theory, and Maximum **STRAIN** Theory. The failure index output is presented with the element layer stress output. As a result, you must request stress output in order to get the failure index results. If you are performing thermal structural analysis, you enter the reference temperature of the element as **TREF**. When performing dynamic response analyses you may enter an optional structural damping coefficient, **GE**. The final field of the parent data entry, **LAM**, allows you to enter symmetric plies more quickly. If you are defining all composite layers, you enter the keyword **ALL**. If your composite is made of layers which are symmetric about the center ply, then you enter **SYM**. Finally, if you want the laminate to model only membrane behavior, then you enter **MEM**. You may also combine these last two by requesting **SYMMEM**.

The composite layup is then defined using one or more continuation entries. Each entry contains the definition of two layers. These data include the material identification number, **MIDi**, the layer thickness, **Ti**, the layer fiber angle, θ**i**, and an optional request for layer stress output, **SOUTi**. If **SOUTi** is blank or the keyword **NO**, then layer stresses will not be printed. To select this option, you enter **YES**.

You may use the **PCOMP1** Bulk Data entry when you have a composite whose plies are all composed of the same material, the same ply thickness, and differ only by fiber angle. Similarly, **PCOMP2** Bulk Data entries are used when the plies are of the same material, but their thicknesses and fiber angles differ.

UAI/*NASTRAN* converts the physical and material data for a composite property into an equivalent property represented by a **PSHELL** Bulk Data entry and a set of **MAT2** entries. You may request a printed summary of the converted data, you may write it to the **BULK** file, or both, using the Bulk Data entry:

PARAM	PLYDATA	PRINT								
		PUNCH								
		BOTH								

Where the available options allow you to **PRINT** or **PUNCH** them as **PSHELL** and **MAT2** Bulk Data entries, or **BOTH**. If you select the **PUNCH** option, then you must **ASSIGN** a file with **USE=BULK** in your Executive Control packet.

5.9 OUTPUT REQUESTS — GENERAL

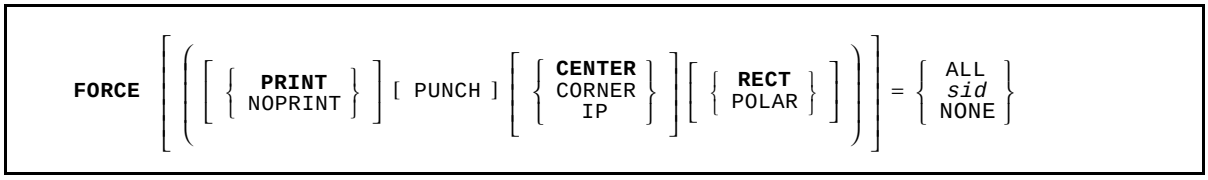
The results of any analysis include solution quantities for each finite element in your model. Although these quantities depend on the specific element type, you request the element output in the same manner. The sections below describe how this is done.

5.9.1 Linear Analyses

When performing linear analyses, element solution results generally include forces, stresses, strains, and strain energy. To request the output of these results, you use Case Control commands, described below. Specific examples of the resulting output for each element are shown in later sections of this Chapter.

5.9.1.1 Element Forces

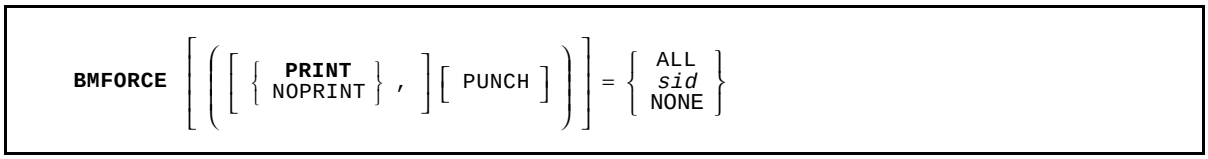
Element forces are selected with the Case Control command:



You may select a set of elements for which output is requested by using the **SET** Case Control command to define the set with an identification number of *sid*. The special keyword **ALL** allows you to request forces for all of the elements in your model. You may then request that the results be **PRINT**ed in the output file, **PUNCH**ed to a file, or both. Note that if you select the **PUNCH** option, you may use the Executive Control command **ASSIGN** to define a specific file name, or you may use the default name defined by your Preference File. If you select the file, be sure to specify **USE=PUNCH**. You use the **NOPRINT** option to request the computation of results which will later be used by a post-processor. In this case, you generally must use the **UAI/*NASTRAN* POST** module, described in Chapter 31 of this manual, to create files which contain the solution results. For TRIA3, TRIAR, QUAD4, and QUADR4 elements, you may request recovery of the forces at the corner GRID points in addition to the usual values by using the **CORNER** option, or at the integration points by using the **IP** option for all two- and three-dimensional elements. When using the **IP** option, the nearest GRID point is reported in the print. Finally, the format of the output may be **RECT**angular or **POLAR**. The **POLAR** format, described in detail in the section on GRID point output in Chapter 4 of the ***User's Reference Manual***, is used for dynamic response analyses.

5.9.1.2 Equivalent Beam Forces

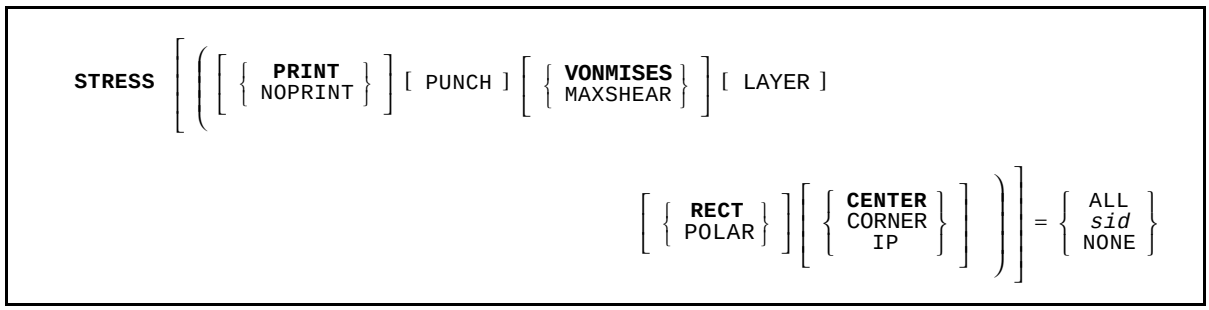
There is a special type of output recovery that allows you to compute equivalent beam forces (moments, shears, axial loads, and torques) for an equivalent beam element composed of solid elements. This option is requested with the Case Control command **BMFORCE**:



You may specify that equivalent beam forces are to be computed for **ALL** equivalent beams defined with **BMFORCE** and **BMFORC1** Bulk Data entries, or for those specified in the integer **SET** with identification number *sid*. **NONE** disables the request. You may then request that the results be **PRINT**ed in the output file, **PUNCH**ed to a file, or both. Again, if you select the **PUNCH** option, you may use the Executive Control command **ASSIGN** to define a specific file name, or you may use the default name defined by your Preference File. If you select the file, be sure to specify **USE=PUNCH**. You use the **NOPRINT** option to request the computation of results which will later be used by a post-processor. In this case, you generally must use the **UAI/*NASTRAN* POST** module, described in Chapter 31 of this manual, to create files which contain the solution results. Note that **BMFORCE** computations may be performed only in Statics and Normal Modes analyses.

5.9.1.3 Element Stress

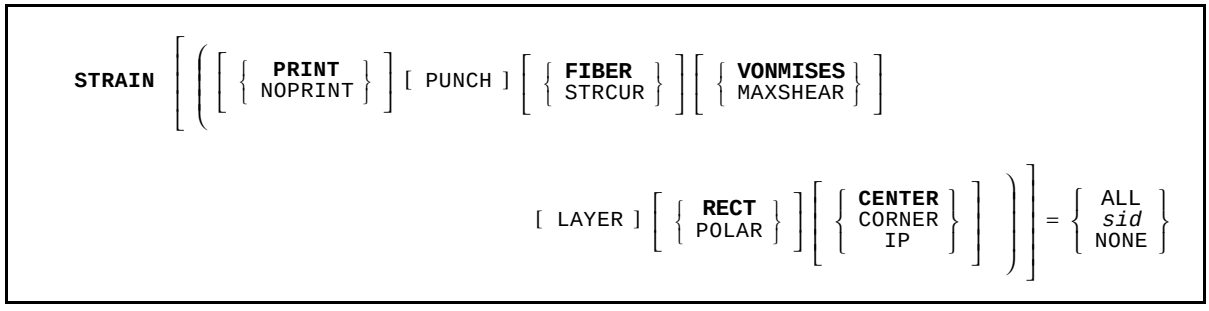
The element stress results are requested with the Case Control command:



Most of the parameters for this command are the same as for the **FORCE** command. The major difference relates only to the TRIA and QUAD families of plate and shell elements. First, you must select between the **VONMISES** stress resultant or the maximum shear, **MAXSHEAR** for output. For homogeneous elements, the stresses are recovered at the extreme fibers of the element. If your model includes laminated composite elements, then you may request **LAYER** to recover the layer stresses and failure indices for those elements. Again, for TRIA3, TRIAR, QUAD4 and QUADR elements, you may request recovery of the forces at the corner GRID points in addition to the usual values by using the **CORNER** option, or at the integration points by using the **IP** option for all two- and three-dimensional elements. When using the **IP** option, the nearest GRID point is reported in the print.

5.9.1.4 Element Strain

The element strain request is made with the Case Control command:



This command is identical to the **STRESS** command with one exception: you may select the recovery of strains at the extreme **FIBER**s of the element, or you may select the strains and curvatures at the midsurface of the elements by using the **STRCUR** option.

5.9.1.5 Element Strain Energy

For Statics, Normal Modes, and Frequency Response analyses, element strain energy is selected with the Case Control command:

```
ESE [ ( [ { PRINT } ] [ PUNCH ] [ THRESH = x ] ) ] = { ALL }
    [     { NOPRINT }                              ]   { sid }
                                                       { NONE }
```

The **THRESH**old allows you to specify a filter so that only selected elements whose strain energy exceeds **THRESH** percent of the total strain energy of the model will print. Results are sorted by element type and the contribution of each element to the total strain energy is given.

5.9.1.6 Element Kinetic Energy

When performing Normal Modes and Frequency Response analyses, you may request element kinetic energies using the command:

```
EKE [ ( [ { PRINT } ] [ PUNCH ] [ THRESH = x ] ) ] = { ALL }
    [     { NOPRINT }                              ]   { sid }
                                                       { NONE }
```

As with strain energy, the **THRESH**old allows you to specify a filter so that only selected elements whose kinetic energy is greater than **THRESH** percent of the total kinetic energy of the model will print. Results are again sorted by element type and the contribution of each element to the total kinetic energy is given.

5.9.2 Nonlinear Analyses

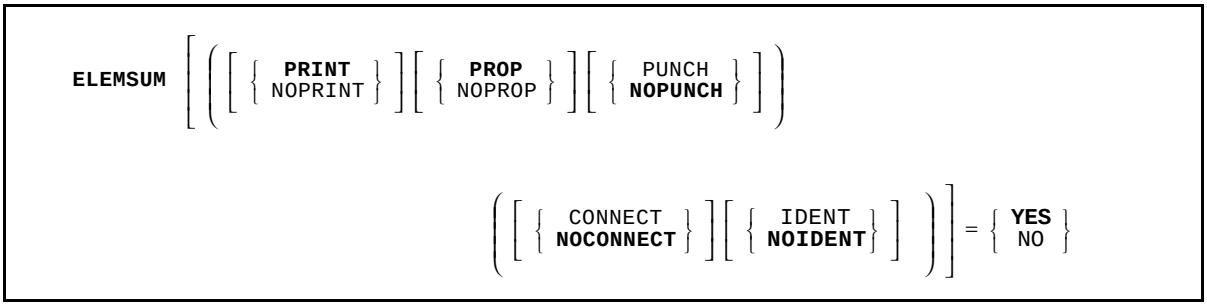
When you perform nonlinear analyses, there are three special Case Control commands which are used to select element output:

```
NLFORCE [ ( [ { PRINT } , ] [ PUNCH ] ) ] = { ALL }
        [     { NOPRINT }                 ]   { sid }
                                              { NONE }

NLSTRAIN [ ( [ { PRINT } , ] [ PUNCH ,] [ { EFFECTIVE } , ] [ LAYER ] ) ] = { ALL }
         [     { NOPRINT }                [   { VONMISES  }               ]   { sid }
                                            { MAXSHEAR  }                   { NONE }

NLSTRESS [ ( [ { PRINT } , ] [ PUNCH ,] [ { FIBER } , ] [ { EFFECTIVE } , ] [ LAYER ] ) ] = { ALL }
         [     { NOPRINT }               [   { STRCUR }               [   { VONMISES  }               ]   { sid }
                                                                        { MAXSHEAR  }                   { NONE }
```

These commands are used in the same manner as the corresponding commands used in linear analysis. The only important difference is that when you are performing plasticity analysis, you may select the **EFFECTIVE** stress or strain as well as the usual **VONMISES** and **MAXSHEAR**. The **LAYER** stresses and strains are available only for Geometric Nonlinear analyses. More details are found in Chapters 16 and 17.

5.9.3 Element Summary Data

An element summary table may be printed using the command:



First, you may request the **PROP**erties of each element. These data, shown in Table 5-10a, include: the element type and identification number; the length, area, or volume of the element (depending on its geometric type); and the mass and weight of the element. Note that the summary is sorted by element type, and the contributions of each type to the properties are summed. There is also a final total for each characteristic. You may also request a summary of the element **CONNECT**ions. These data, shown in Table 5-10b, indicates the elements which are connected to each GRID point in the model. Finally, the **IDENTS** option requests a simple list of all elements, sorted by the element identification number, and their types. This is shown in Table 5-10c.

5.9.4 Other Element-Related Output

In addition to the solution results discussed in the previous two sections, there are also several other types of results that relate to elements. The computation of GRID point stresses involves interpolating the values of stresses for selected elements to the GRID points in a field. This feature is selected with the **GPSTRESS** Case Control command. Similarly, when you perform Design Sensitivity analyses, described in detail in Chapter 25, you may selectively request the output of sensitivities for constrained elements.

Table 5-10. ELEMENT SUMMARY RESULTS

a. The PROP Option Results

E L E M E N T P R O P E R T Y S U M M A R Y

TYPE	ID	LENGTH	AREA	VOLUME	MASS	WEIGHT
QUAD4	111	0.000000E+00	4.320000E+04	4.320000E+00	4.320000E+00	4.320000E+00
QUAD4	SUBTOTAL	0.000000E+00	4.320000E+04	4.320000E+00	4.320000E+00	4.320000E+00
ROD	1	1.800000E+02	7.970000E+00	1.434600E+03	1.434600E+03	1.434600E+03
ROD	2	3.000000E+02	2.116000E+01	6.348000E+03	6.348000E+03	6.348000E+03
ROD	3	2.400000E+02	1.177000E+01	2.824800E+03	2.824800E+03	2.824800E+03
ROD	4	1.800000E+02	7.970000E+00	1.434600E+03	1.434600E+03	1.434600E+03
ROD	5	3.000000E+02	7.970000E+00	2.391000E+03	2.391000E+03	2.391000E+03
ROD	6	1.800000E+02	1.177000E+01	2.118600E+03	2.118600E+03	2.118600E+03
ROD	7	2.400000E+02	1.177000E+01	2.824800E+03	2.824800E+03	2.824800E+03
ROD	8	3.000000E+02	2.116000E+01	6.348000E+03	6.348000E+03	6.348000E+03
ROD	9	1.800000E+02	7.970000E+00	1.434600E+03	1.434600E+03	1.434600E+03
ROD	SUBTOTAL	2.100000E+03	1.095100E+02	2.715900E+04	2.715900E+04	2.715900E+04
SHEAR	112	0.000000E+00	4.320000E+04	4.320000E+00	4.320000E+00	4.320000E+00
SHEAR	SUBTOTAL	0.000000E+00	4.320000E+04	4.320000E+00	4.320000E+00	4.320000E+00
TRIA3	101	0.000000E+00	2.160000E+04	2.160000E+00	2.160000E+00	2.160000E+00
TRIA3	102	0.000000E+00	2.160000E+04	2.160000E+00	2.160000E+00	2.160000E+00
TRIA3	103	0.000000E+00	2.160000E+04	2.160000E+00	2.160000E+00	2.160000E+00
TRIA3	104	0.000000E+00	2.160000E+04	2.160000E+00	2.160000E+00	2.160000E+00
TRIA3	SUBTOTAL	0.000000E+00	8.640000E+04	8.639999E+00	8.639999E+00	8.639999E+00
	TOTAL	2.100000E+03	1.729095E+05	2.717628E+04	2.717628E+04	2.717628E+04

b. The CONNECT Option Results

G R I D P O I N T C O N N E C T I O N S U M M A R Y

GRID ID	ELEMENT	TYPE	ELEMENT	TYPE	ELEMENT	TYPE	ELEMENT	TYPE	ELEMENT	TYPE
1	1	ROD	2	ROD	101	TRIA3				
2	111	QUAD4	1	ROD	3	ROD	4	ROD	112	SHEAR
	101	TRIA3	102	TRIA3						
3	111	QUAD4	2	ROD	3	ROD	5	ROD	6	ROD
	112	SHEAR	101	TRIA3	102	TRIA3	103	TRIA3		
4	111	QUAD4	4	ROD	5	ROD	7	ROD	9	ROD
	112	SHEAR	102	TRIA3	103	TRIA3	104	TRIA3		
5	111	QUAD4	6	ROD	7	ROD	8	ROD	112	SHEAR
	103	TRIA3	104	TRIA3						
6	8	ROD	9	ROD	104	TRIA3				

c. The IDENT Option Results

E L E M E N T I D E N T I F I C A T I O N S U M M A R Y

ID	TYPE	ID	TYPE	ID	TYPE	ID	TYPE	ID	TYPE	ID	TYPE
1	ROD	2	ROD	3	ROD	4	ROD	5	ROD	6	ROD
7	ROD	8	ROD	9	ROD	101	TRIA3	102	TRIA3	103	TRIA3
104	TRIA3	111	QUAD4	112	SHEAR						

Chapter 6

BOUNDARY CONDITIONS AND EQUATION REDUCTION

As you have seen in Chapters 4 and 5, GRID points and elements are used to define your finite element model. The characteristics of these GRID points and elements as well as environmental loading conditions and boundary conditions are then used to formulate the equations of motion for a specific analytical discipline. In the finite element method, this system of equations is written in matrix form. The variables in these equations can represent structural displacements, velocities, accelerations, temperatures, or other physical responses. These variables are called ***Degrees of Freedom***. The complete set of degrees of freedom defined by the GRID and SCALAR points in your model make up what is called the *g-set*. Each GRID point in your model contributes six degrees of freedom to the *g-set*, and each SCALAR point one degree of freedom. The *g-set* defines the equations of motion for your model before the application of any boundary conditions, rigid links, reduction techniques, etc. **UAI/*NASTRAN*** uses a number of different sets to describe the procedures for application of boundary conditions and other reduction techniques, and all of these sets are defined below in Section 6.3

For each element of the model, one or more element mass, stiffness or damping matrix may be generated. Also, each element may produce body load vectors from gravity, centrifugal or thermal effects.

These element matrices are then assembled into *g-set* size system matrices. Additionally, any external loads on your model are added to the load vectors. After assembly, the matrices are partitioned by a sequence of operations that account for boundary conditions, or fixities, in the model; dependencies between two of more degrees of freedom; and matrix reductions, which are typically used to reduce the problem size in many analysis disciplines.

This chapter provides you with the mathematical background of these operations, information on selecting the various options, and guidelines for using these features.

Rev: V20.1

6.1 SYSTEM MATRIX ASSEMBLY

The first step in each of the **UAI/*NASTRAN*** analysis disciplines is the assembly of the system, *g-set* or *g-size*, matrices. The specific matrices which are assembled depends on the analytic discipline which you are using. The following sections describe the matrices which are assembled for the various disciplines.

6.1.1 The Stiffness Matrix

Each finite element in your model is represented by an element stiffness matrix $\boldsymbol{k}^e$. Regardless of the analysis discipline, all of the element stiffness matrices are assembled into the *g-size* stiffness matrix[1]:

$$\boldsymbol{K}_{gg}^{E} = \mathbf{A}\, \boldsymbol{k}^{e} \tag{6-1}$$

6.1.2 The Mass Matrix

When you perform any dynamics analysis or a static analysis which has inertial loadings such as gravity, a request for weight calculation, or is being solved as a free-body, the g-size mass matrix generated by the finite elements in your model is formed by assembling the contributions of each element mass matrix $\boldsymbol{m}^e$:

$$\boldsymbol{M}_{gg}^{E} = \mathbf{A}\, \boldsymbol{m}^{e} \tag{6-2}$$

6.1.3 The Damping Matrices

Finally, when you perform a dynamics analysis with **UAI/*NASTRAN***, there are a number of ways in which damping phenomena may be modeled. These are thoroughly described in Chapter 9. If you have specified viscous damping, then the *g-size* viscous damping matrix generated by the finite elements in your model is formed by assembling the contributions of each element damping matrix $\boldsymbol{b}^e$:

$$\boldsymbol{B}_{gg}^{E} = \mathbf{A}\, \boldsymbol{b}^{e} \tag{6-3}$$

If you have specified that one or more elements have structural damping, g_e, then a structural damping matrix is assembled as:

$$\boldsymbol{K}_{gg}^{4} = \mathbf{A}\, g_e \boldsymbol{b}^{e} \tag{6-4}$$

1 $\mathbf{A}$ is an operator denoting matrix assembly.

6.1.4 The Load Vectors

Loads for static analysis are of two types: the first type are those applied at individual degrees of freedom such as forces and moments; and the second are those associated with elements, such a surface pressures and thermal gradients. The element load contributions $\boldsymbol{p}^e$ are computed, and their contributions are assembled. These are then added to $\boldsymbol{p}^g$, the external applied loads:

$$\boldsymbol{P}_g = \boldsymbol{p}^g + \mathbf{A}\boldsymbol{p}^e \tag{6-5}$$

When performing transient response analyses, the loads are time-dependent. In this case the dynamic loads are computed for each time step required by the transient integration:

$$\boldsymbol{P}_g(t) = \boldsymbol{p}^g(t) + \mathbf{A}\,\boldsymbol{p}^e(t) \tag{6-6}$$

Similarly, when performing frequency response analyses, any frequency-dependent loads that you specify are assembled:

$$\boldsymbol{P}_g(f) = \boldsymbol{p}^g(f) + \mathbf{A}\,\boldsymbol{p}^e(f) \tag{6-7}$$

6.2 DIRECT MATRIX INPUT

You may define *g-size* stiffness, damping, or mass matrices, independently with respect to your finite element data, which are called $\boldsymbol{K}^2_{gg}$, $\boldsymbol{B}^2_{gg}$ and $\boldsymbol{M}^2_{gg}$ respectively. You define these matrices using **DMIG** Bulk Data entries and select them with the Case Control commands:

```
K2GG = matname
B2GG = matname
M2GG = matname
```

where *matname* is the matrix name used on the corresponding **DMIG** Bulk Data entries. These matrices must be symmetric. Since the matrices are *g-size*, they are simply added to the assembled structural matrices defined in previous sections to result in the final solution matrices:

$$\boldsymbol{K}_{gg} = \boldsymbol{K}^E_{gg} + \boldsymbol{K}^2_{gg}; \quad \boldsymbol{M}_{gg} = \boldsymbol{M}^E_{gg} + \boldsymbol{M}^2_{gg}; \quad \boldsymbol{B}_{gg} = \boldsymbol{B}^E_{gg} + \boldsymbol{B}^2_{gg} \tag{6-8}$$

Naturally, if you do not provide *g-size* direct input matrices, then the solution matrices are those assembled from the finite element model.

6.3 THE REPRESENTATION OF DISPLACEMENT SETS

As described above, when constructing the structural matrices used in **UAI/*NASTRAN***, each row and column of a matrix is associated with a GRID point, a SCALAR point, or an EXTRA point. Every GRID point is represented by six degrees of freedom, and hence six rows and columns of each of the system response coefficient matrices. SCALAR and EXTRA points only have one degree of freedom. At each point these degrees of freedom are further classified into sets, depending on the constraints, boundary conditions, or other handling required for each particular degree of freedom. For example, for a two-dimensional model in the x-y plane, all degrees of freedom in the z direction must be constrained to zero motion and hence belong to a class of degrees of freedom called the ***single-point constraint set,*** or, *s-set* .

All degrees of freedom in **UAI/*NASTRAN*** are assigned to one or more sets, and these assignments, along with the definitions of treatment for each set, control all the matrix operations which ultimately lead to a solution of the equations of motion. Table 6-1 presents a brief definition of each set. Many of the matrix operations in **UAI/*NASTRAN*** are concerned with partitioning, merging, and transforming the coefficient matrices from one set of displacement components to another. All the components of displacement of a given type of set, such as all points constrained by single-point constraints, form a set. A given component of displacement may, and usually does, belong to several sets.

Several sets are mutually exclusive with respect to one another. All sets which define ***dependent*** degrees of freedom are of this type. A dependent degree of freedom may therefore belong to only one dependent set and its dependency may only be defined in one manner. For example, a degree of freedom may not be both constrained to zero motion, such as in the definition of a model boundary condition (*s-set*), and also assigned a rigid link dependency using either a rigid element or assigned as a dependent degree of freedom with a multipoint constraint equation (*m-set*).

Table 6-1. DEFINITION OF DISPLACEMENT SETS

DISPLACEMENT SET	DESCRIPTION
$u_g = u_n + u_m$	All structural DOF including scalar DOF
u_m	DOF eliminated by multipoint constraints and rigid elements
$u_n = u_f + u_s$	All structural DOF not constrained by multipoint constraints
u_s	DOF eliminated by single-point constraints
$u_f = u_a + u_o$	Unconstrained (free) structural DOF
u_o	DOF omitted by static condensation (Guyan Reduction)
$u_a = u_r + u_l$	DOF used in real eigenvalue analysis
u_r	DOF to which determinate reactions are applied for the solution of free body models
u_l	The remaining structural DOF used in static analysis (points left over)
u_e	Extra DOF introduced in dynamic analysis
$u_d = u_{a+} u_e$	DOF used in dynamic analysis by the direct method
$u_p = u_g + u_e$	The g-set plus EXTRA points for dynamic analysis
u_z	DOF representing modal coordinates
$u_h = u_e + u_z$	DOF used in dynamic analysis by the modal method

Most other sets define the independent equations of motion. For example, the *a-set* describes the degree of freedom assigned to the ***analysis*** set, that set which contains the independent equations of motion in most analyses.

A nomenclature using the variable *u* for the unknowns of the **UAI/*NASTRAN*** solution, and using subscript characters for the description of the set, are used throughout this chapter to describe how boundary conditions are defined, applied and processed, and how other dependency relationships, such as rigid links, are handled. The relationship between the displacement sets may also be illustrated graphically, and this is done in Figure 6-1.

In **UAI/*NASTRAN***, the definition of constraints causes the original equations to be partitioned and processed in special ways depending on the particular constraints specified. The partitioning concept is further extended in **UAI/*NASTRAN*** to provide a convenient means for *reduction*. Reduction is the process of transforming the original equations of motion to a smaller number of equations. When this is done, some of the original degrees of freedom in the model are made dependent on those degrees of freedom not reduced. Reduction techniques may be used to improve efficiency in dynamic analyses, to allow the analysis of extremely large models using substructuring techniques, and to improve numerical solution procedures when very stiff or rigid sections of structure are included.

Certain classes of Bulk Data entries are used in specifying constraints. These include:

- ❒ Single-Point Constraint Entries
- ❒ Multipoint Constraint Entries
- ❒ Rigid Element Entries
- ❒ Entries that define reaction points on free bodies
- ❒ Entries used for matrix reduction

Figure 6-1. RELATIONSHIP AMONG DISPLACEMENT SETS

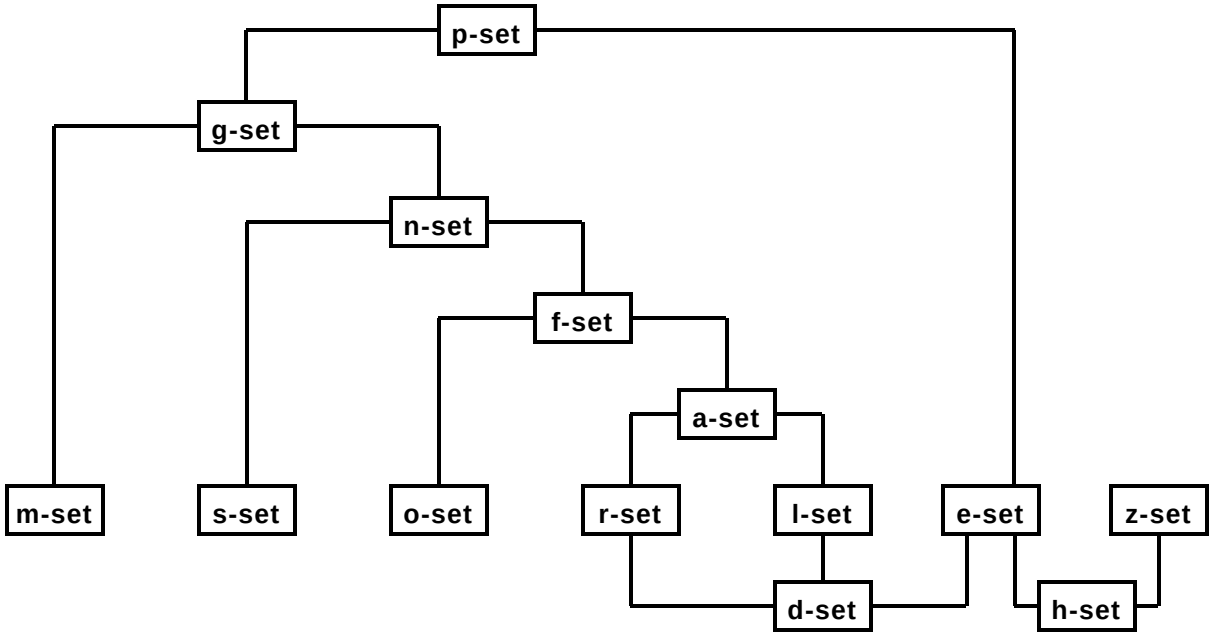


Table 6-2 presents the relationships between the various Case Control commands and Bulk Data entries used to define constraints, partitioning and reduction.

The procedures you use to assign each degree of freedom to its respective set are described in the next several sections of this Chapter.

Table 6-2. SELECTING CONSTRAINT AND REDUCTION DATA

CASE CONTROL COMMAND	BULK DATA ENTRIES USED ARE:		DOF ARE PLACED IN:
AUTOOMIT	*None*		**o-set**
AUTOREDUCE	*None*		**o-set**
AUTOSPC	*None*		**s- or** *m-set*
BOUNDARY	**BDYS,BDYS1**		*a-set*
DYNRED	**DYNRED**		*a-set*
INERTIA	**BDYS,BDYS1**		*a-set*
MPC	**MPCADD**	**MPC**	*m-set*
		MPCS[2]**,MPCAX**[3]	
NLREDUCE	*None*[1]		*o-set*
SPC	**SPCADD**	**SPC,SPC1**	*s-set*
		SPCS[2]**,SPCS1**[2]**,SPCAX**[3]	
Reduction	**ASET,ASET1,ASETAX**[3]**,ASETAX1**[3]		*a-set*
	OMIT,OMIT1,OMITAX[3]		*o-set*
	BDYC	**BDYS**[2]**, BDYS1**[2]	*a-set*
Rigid Body Supports	**SUPORT,SUPORTS**[2]**,SUPAX**[3]		*r-set*
Rigid Elements	**RBAR,RROD,RTRPLT**		*m-set*
	RBE1,RBE2,RBE3		
	RSPLINE		
Retaining *a-set* Degrees of Freedom	**BDYS,BDYS1**[4]		*a-set*

Notes:
1 Used for Material Nonlinear Analysis
2 Used for Automated Substructuring
3 Used for Axisymmetric Harmonic Analysis
4 Can be used with **AUTOSPC**, **AUTOOMIT**, **AUTOREDUCE** and **NLREDUCE**

6

6.4 DEPENDENT DEGREES OF FREEDOM

You may define degrees of freedom within your model to be linear combinations of the values of one or more other degrees of freedom. You may define these dependent degrees of freedom by explicitly defining their relationships using **MPC** Bulk Data entries, or you may have **UAI/*NASTRAN*** determine the exact relationships from the model geometry by using rigid elements.

This section describes the several different methods available for defining these dependent relationships. These methods include directly writing constraint equations (multipoint constraints), using rigid elements, or using other features which create the effect of multipoint constraint equations.

After describing procedures for using these features, the equations which are used by **UAI/*NASTRAN*** to implement the procedures are presented.

6.4.1 Multipoint Constraints

Each multipoint constraint is described by a single equation that specifies a linear relationship between two or more degrees of freedom. You may include any number of sets of multipoint constraint data in the Bulk Data packet which you then select at execution time. Some uses of multipoint constraints are:

- To fix motion in directions other than those corresponding to components of the global coordinate system. In such cases, the multipoint constraint involves only the degrees of freedom at a single GRID point. The constraint equation defines displacement in the direction of zero motion using the displacement components in global system at the GRID point.
- To describe rigid bodies and mechanisms such as levers, pulleys and gear trains. In applications, the degrees of freedom associated with the rigid body motion are with multipoint constraint equations. Treatment of very stiff members as rigid elements eliminates ill-conditioning of the stiffness matrix associated with their treatment ordinary elastic elements.
- To generate nonstandard structural elements and other special effects.
- To represent parts of a structure by local vibration modes by using the matrix of eigenvectors to represent a set of constraints relating physical coordinates to coordinates.

Bulk Data entries which you may use to define multipoint constraints include **MPC**, **MPCADD**, **MPCS** and **MPCAX**. The **MPC** entry is the basic entry for defining multipoint constraints. When you use **MPC** data, you must select that data in the Case Control packet with the command

```
MPC = mpcid
```

The format of an **MPC** Bulk Data entry is:

MPC	**SID**	**GIDD**	**DOFD**	**B**	**GID1**	**DOF1**	**A1**		**-cont-**
-cont-		**GID2**	**DOF2**	**A2**	***CONTINUES IN GROUPS OF 3***				**-cont-**

The **SID** is the identification number that you use to select the **MPC** Bulk Data entries. You may use as many entries as are required. The next field specifies a GRID or SCALAR point identification number, **GIDD**, and one of its components to be the *dependent* degree of freedom, **DOFD**. The **DOFD** value must be a single digit degree of freedom Code. You may apply a constant factor, **B**, to the displacement of **DOFD**. Next, you define one or more groups of three fields of data which specify the independent degrees of freedom **DOFi** for additional GRID or SCALAR points, **GIDi**, and their associated scale factors, **Ai**. Each **MPC** Bulk Data entry then defines a constraint equation of the form:

$$DOF_d = -\frac{1}{B}\sum_i A_i DOF_i \qquad (6\text{-}9)$$

Dependent degrees of freedom may appear as independent terms in other MPC equations. However, they may appear as dependent terms in only a single equation. Also, dependent components may not be referenced on any other Bulk Data entry which would cause the component to be placed in a mutually exclusive set, such as the *o-set* or *s-set*. The **MPCADD** Bulk Data entry defines a union of two or more multipoint constraint sets. The **MPCS** and **MPCAX** entries are used only for specifying multipoint constraints in problems using substructuring and axisymmetric harmonic elements, respectively.

6.4.2 Rigid Body Connections

UAI***NASTRAN*** provides you with seven special *rigid elements* that may be used to conveniently model rigid body structural connections, to define motions of points as an *average* of the motion of other points, and to define constraint relationships between points using spline functions. Table 6-3 summarizes the Bulk Data entries that define these rigid connections.

The elements **RBAR**, **RBE1**, **RBE2**, **RROD** and **RTRPLT** are used to model rigid mechanical connections between GRID points. This variety of connections allows simple modeling of special connections involving as many as six degrees of freedoms at each GRID point. This allows complete control over the selection of which GRID points and degrees of freedoms are placed in the *m-set*. The use of the rigid elements greatly reduces the work necessary to write explicit MPC equations containing geometric terms which would otherwise be necessary.

A rigid connection between coincident GRID points may also be modeled. This feature is useful if the motion of a point is desired in two or more coordinate systems. For example, suppose that **GA** and **GB** are both located at the same point in the basic coordinate system.

Table 6-3. SUMMARY OF RIGID ELEMENTS

ELEMENT	DESCRIPTION
RBAR **RBE1** **RBE2**	Define rigid connections between GRID points using different specifications of the dependent DOF.
RBE3	Used to average the motions at a GRID point or to distribute loads among GRID points.
RROD	Defines a rigid extensional connection between two GRID points.
RSPLINE	Defines a spline function used for interpolating the motions of dependent DOF
RTRPLT	Defines a rigid connection between various DOF relating three GRID points

The output of **GA** is defined in the basic system, and the output of **GB** is defined in a local, cylindrical coordinate system. The Bulk Data entry:

RBE2	**EID**	**GA**	**123456**	**GB**					

will cause the output motions of **GB** to reflect the motions of **GA** with the proper coordinate system transformations applied.

6.4.3 The RBE3 Connections

The RBE3 element can be used to define the motion of one GRID point, at one or more degrees of freedom, as a weighted average of one or more other GRID points, and it can be used to distribute loads or mass from one reference point to other GRID points using one of several techniques. The RBE3 equations are controlled by the data fields on the **RBE3** Bulk Data entry, shown below. The controls are illustrated in the following sections.

RBE3	**EID**	**OPTION**	**REFG**	**REFC**	**WTi**	**Ci**	**Gij**	**Gij+1**	

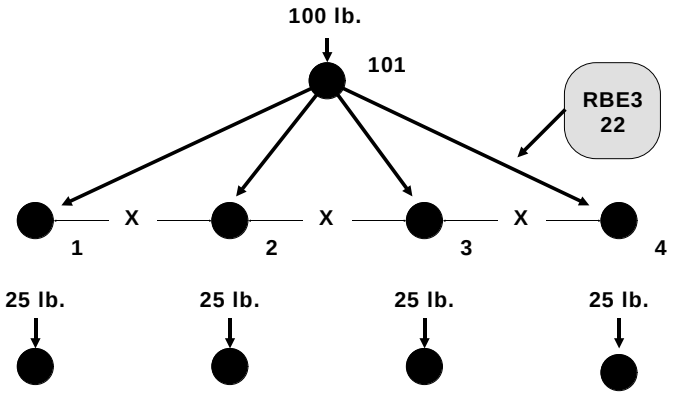
6.4.3.1 Selecting the Algorithm - The **OPTION** *Field.*

This field selects the basic algorithm to be used. A value of blank or 0 will cause the coupling between the reference GRID and the independent points to be dependent only upon translations. A static equilibrium solution is then computed. For example, the reference point 101 is located mid-way between the uniformly spaced independent points 1 through 4. A 100 pound force or translational mass can be distributed equally to the independent points by using the Bulk Data entry shown in Figure 6-2.

Now, suppose that a reference point 201 is not located mid-way between the independent points. With **OPTION=0**, the same 100 lb. load will be approximately distributed as illustrated in Figure 6-3.

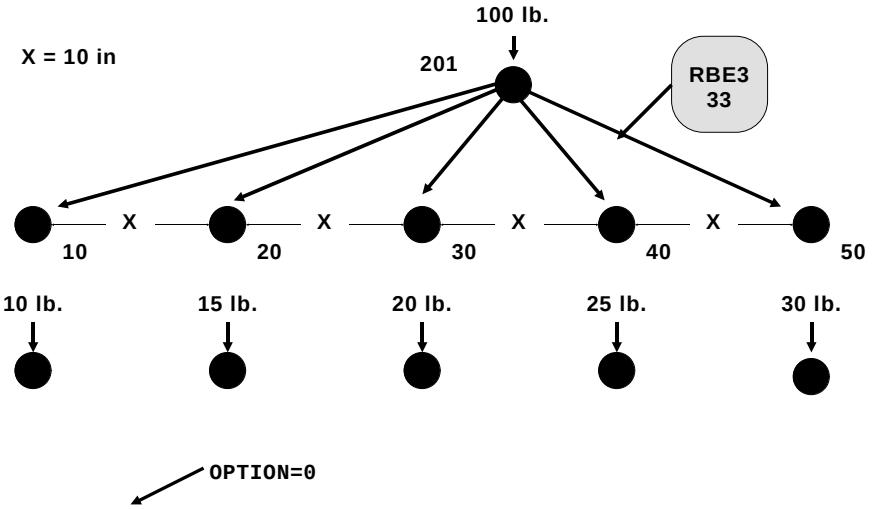
6

Figure 6-2. USING THE RBE3: EXAMPLE 1



RBE3	**22**		**101**	**123**	**1.0**	**123456**	**1**	**2**	**+A**
+A	**3**	**4**							

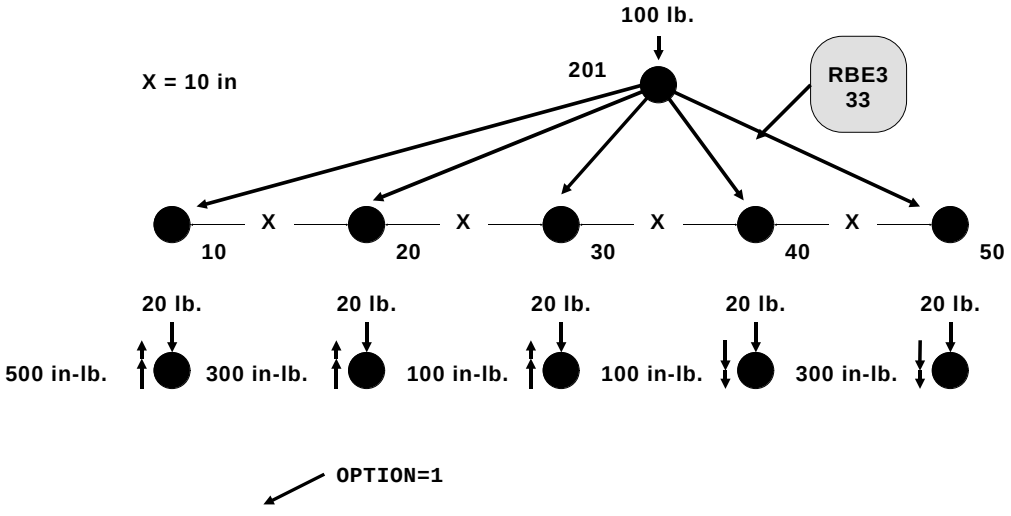
Figure 6-3. USING THE RBE3: EXAMPLE 2



RBE3	33	0	201	123456	1.0	123456	10	20	+A
+A	30	40	50						

The distribution shown above maintains static equilibrium using translational forces. Only insignificant moments are created by the algorithm. If **OPTION=1** is selected with the same data as above, an entirely different load distribution results, as seen in Figure 6-4. In this case the translational loads are distributed exactly as requested by the defined weighting factors (1.0

Figure 6-4. USING THE RBE3: EXAMPLE 3



RBE3	33	1	201	123456	1.0	123456	10	20	+A
+A	30	40	50						

for all points in this case), and static equilibrium is maintained by applying moments to the independent points.

6.4.3.2 Static Equilibrium Rules.

These **REFC** and **Ci** fields may be used to control the rule of static equilibrium discussed above for **OPTION=1**. If static equilibrium is not required or desired, and **OPTION=1**, these fields should be equal to one another and contain any combination of the digits 1-3. Otherwise, if **OPTION=0** or if static equilibrium is desired, these fields should contain all the digits 1-6.

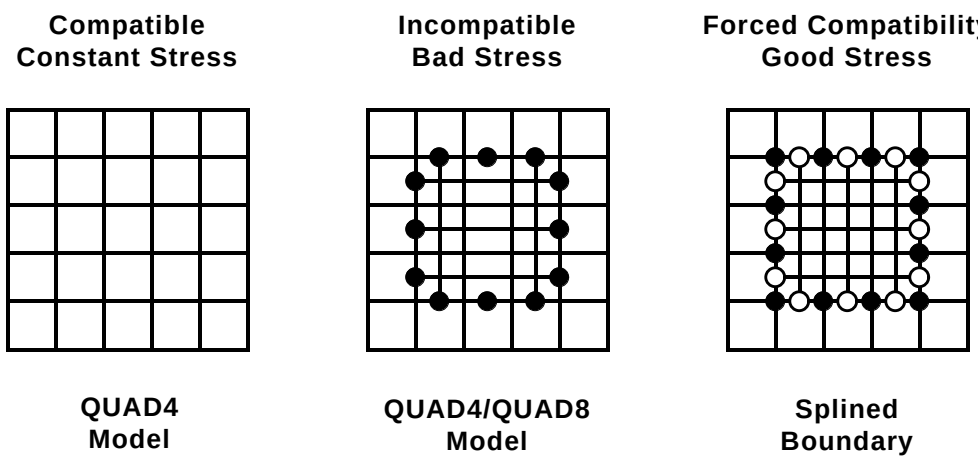
6.4.3.3 Weighting Factors.

You may modify the distribution of the effects of RBE3 connections by using values other than 1.0 in the **WTi** fields. Complete control of these weighting factors is available only with **OPTION=1**. With **OPTION=1** and components 1-6 involved, the translational coupling will be exactly as specified by the **WTi** factors, and rotational coupling will be adjusted to provide static equilibrium. If only components 1-3 are used, static equilibrium is not maintained and no rotational coupling is introduced.

6.4.4 The RSPLINE Connection

The RSPLINE is a special, high-order interpolating connection. Its purpose is to enforce interelement continuity when modeling with both linear and quadratic isoparametric elements. Consider, for example, the three models shown in Figure 6-5. This simple flat plate was solved using a closed form solution for the boundary displacement that leads to a constant stress state. The first model is a five-by-five mesh of QUAD4 elements. The solution is in exact agreement with theory and all stresses are equal. The same plate was then modeled using a combination of QUAD8 and QUAD4 elements. This solution resulted in a 10% stress distortion in the QUAD4 elements connected to midside modes of QUAD8 elements. This model was then modified by adding four RSPLINE elements along the QUAD4/QUAD8 boundary. The solid

Figure 6-5. USING THE RSPLINE ELEMENT



GRIDs represent independent degrees of freedom and the open GRIDs represent dependent degrees of freedom. The result was again in exact agreement with the theory.

It is generally a poor modeling practice to use high order elements for mesh transition regions. If these regions are areas of high stress gradient, then erroneous answers will result. The RSPLINE can alleviate this problem.

6.4.5 Automatically Generated MPC Equations

If you select the AUTOSPC option, which is described in detail in the next section, this feature may automatically generate MPC relationships to remove singularities from your model. As you will see, this option results in the correct constraints to the model, rather than the approximation which is introduced by fully constraining the degree of freedom closest to the singularity. You should note, however, that this will cause the MPC reductions (6-10 through 6-19) to be performed, thereby incurring a modest additional cost. You may, however, override the default mode of operation of the AUTOSPC function. To do this, you select the **SPC** option when using the **AUTOSPC** Case Control command.

6.4.6 Equations for Dependent Reduction

The relationships defined by (6-9) are assembled into matrix $\boldsymbol{R}_{mg}$ where each row represents an individual dependent degree of freedom. The *g-set* matrices are then partitioned into the *m-set*, which represents the dependent degrees of freedom, and the *n-set*, which represents the independent degrees of freedom, using the following sequence of operations:

$$\boldsymbol{R}_{mg} \rightarrow [\, \boldsymbol{R}_{mm} \mid \boldsymbol{R}_{mn} \,] \tag{6-10}$$

$$\boldsymbol{G}_{mn} = -\boldsymbol{R}_{mm}^{-1}\boldsymbol{R}_{mn} \tag{6-11}$$

$$\boldsymbol{K}_{gg} \rightarrow \begin{bmatrix} \overline{\boldsymbol{K}}_{nn} & \boldsymbol{K}_{nm} \\ \boldsymbol{K}_{mn} & \boldsymbol{K}_{mm} \end{bmatrix} \text{ and } \quad \boldsymbol{M}_{gg} \rightarrow \begin{bmatrix} \overline{\boldsymbol{M}}_{nn} & \boldsymbol{M}_{nm} \\ \boldsymbol{M}_{mn} & \boldsymbol{M}_{mm} \end{bmatrix} \tag{6-12}$$

$$\boldsymbol{B}_{gg} \rightarrow \begin{bmatrix} \overline{\boldsymbol{B}}_{nn} & \boldsymbol{B}_{nm} \\ \boldsymbol{B}_{mn} & \boldsymbol{B}_{mm} \end{bmatrix} \text{ and } \quad \boldsymbol{K}^{4}_{gg} \rightarrow \begin{bmatrix} \overline{\boldsymbol{K}}^{4}_{nn} & \boldsymbol{K}^{4}_{nm} \\ \boldsymbol{M}^{4}_{mn} & \boldsymbol{K}^{4}_{mm} \end{bmatrix} \tag{6-13}$$

The following reductions are then performed:

$$\boldsymbol{K}_{nn} = \overline{\boldsymbol{K}}_{nn} + \boldsymbol{G}_{mn}^{T}\boldsymbol{K}_{mn} + \boldsymbol{K}_{mn}^{T}\boldsymbol{G}_{mn} + \boldsymbol{G}_{mn}^{T}\boldsymbol{K}_{mm}\boldsymbol{G}_{mn} \tag{6-14}$$

$$\boldsymbol{M}_{nn} = \overline{\boldsymbol{M}}_{nn} + \boldsymbol{G}_{mn}^{T}\boldsymbol{M}_{mn} + \boldsymbol{M}_{mn}^{T}\boldsymbol{G}_{mn} + \boldsymbol{G}_{mn}^{T}\boldsymbol{M}_{mm}\boldsymbol{G}_{mn} \tag{6-15}$$

$$\boldsymbol{B}_{nn} = \overline{\boldsymbol{B}}_{nn} + \boldsymbol{G}_{mn}^{T}\boldsymbol{B}_{mn} + \boldsymbol{B}_{mn}^{T}\boldsymbol{G}_{mn} + \boldsymbol{G}_{mn}^{T}\boldsymbol{B}_{mm}\boldsymbol{G}_{mn} \tag{6-16}$$

$$\boldsymbol{K}^{4}_{nn} = \overline{\boldsymbol{K}}^{4}_{nn} + \boldsymbol{G}_{mn}^{T}\boldsymbol{K}^{4}_{mn} + \boldsymbol{K}^{4T}_{mn}\boldsymbol{G}_{mn} + \boldsymbol{G}_{mn}^{T}\boldsymbol{K}^{4}_{mm}\boldsymbol{G}_{mn} \tag{6-17}$$

Load vectors are similarly transformed:

$$\boldsymbol{P}_g \rightarrow \left\{ \begin{matrix} \overline{\boldsymbol{P}}_n \\ \boldsymbol{P}_m \end{matrix} \right\} \tag{6-18}$$

$$\boldsymbol{P}_n = \overline{\boldsymbol{P}}_n + \boldsymbol{G}_{mn}^T \boldsymbol{P}_m \tag{6-19}$$

This sequence of operations results in reducing the equations of motion to the *n-set* which represents all independent degrees of freedom.

6.4.7 Forces of Multipoint Constraint

The GRID point forces required to enforce the multipoint constraint equations may be recovered during Static and Normal Modes analyses. You request these results with the Case Control command:

MPCFORCES $\left(\left[\left\{ \begin{matrix} \textbf{PRINT} \\ \text{NOPRINT} \end{matrix} \right\} , \right] [\ \text{PUNCH}\ ] \right) = \left\{ \begin{matrix} \text{ALL} \\ \textit{sid} \\ \text{NONE} \end{matrix} \right\}$

where *sid* specifies a set of previously defined GRID points for which the forces will be recovered.

To compute these forces, first the *g-set* stiffness matrix is partitioned:

$$\boldsymbol{K}_{gg} \rightarrow \left[\begin{matrix} \boldsymbol{K}_{ng} \\ \boldsymbol{K}_{mg} \end{matrix} \right] \tag{6-20}$$

then, using the load partition, $\boldsymbol{P}_m$, from (6-18), the forces are computed using:

$$\boldsymbol{Q}_g = \left[\begin{matrix} \boldsymbol{Q}_{ng} \\ \boldsymbol{Q}_{mg} \end{matrix} \right] = \left[\begin{matrix} \boldsymbol{G}_{mn}^T \\ \boldsymbol{I} \end{matrix} \right] \left[\boldsymbol{K}_{mg} \boldsymbol{u}_g - \boldsymbol{P}_m \right] \tag{6-21}$$

6.5 FIXED AND ENFORCED MOTION

A single-point constraint, or SPC, applies a fixed value, which may be zero, to a translational or rotational degree of freedom at a GRID point or to a SCALAR point. The most common use of single-point constraints is to specify the fixed boundary conditions of a structural model. Multiple boundary condition problems can be solved easily in a single **UAI/*NASTRAN*** execution using the SUBCASE feature. This feature is particularly useful, for example, in the solution of problems having one or more planes of symmetry. In this case, different SUBCASEs would be used for each of the symmetric and unsymmetric solutions for each plane of symmetry.

A second use of SPCs is needed when the finite element formulation does not provide resistance to motion in a certain direction thus causing the stiffness matrix to be singular. Single-point constraints are used to remove these degrees of freedom from the matrix. An 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. A special feature, called AUTOSPC, is provided to automatically remove singularities of this type and is described below.

If all singularities have not been removed, or are removed by the AUTOSPC feature, **UAI/*NASTRAN*** will print a GRID Point Singularity summary table. This table identifies singularities in the *f-set*. A vector is printed which defines the direction of the singularity in the output coordinate system for the GRID point.

Single-point constraints are defined using the Bulk Data entries **SPC, SPC1, SPCADD,** and **SPCD**. For substructuring, the additional entries **SPCS, SPCS1** and **SPCSD** are also used. Axisymmetric harmonic analyses also have a special Bulk Data entry, **SPCAX**.

Single-point constraint data defined using **SPC** Bulk Data entries are only used when requested by a Case Control command of the form:

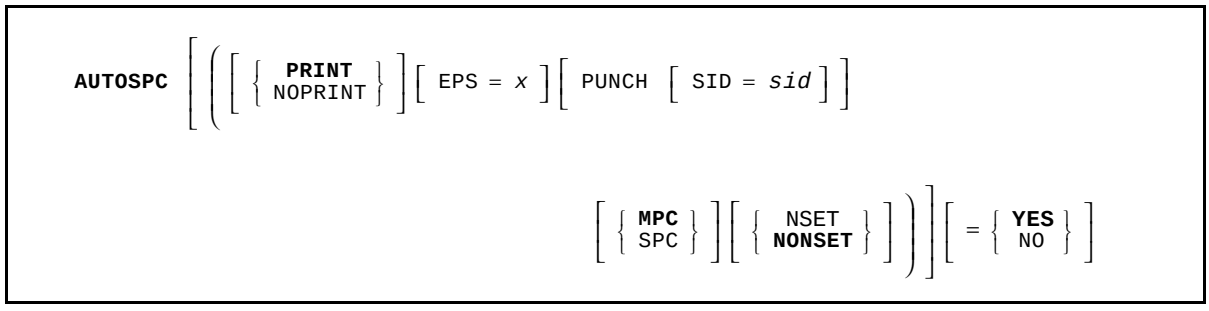
```
SPC = sid
```

Single-point constraints may also be specified on the **GRID** Bulk Data entry. When this feature is used, these single-point constraints are always used, and they apply to all SUBCASEs in the analysis.

Procedures for automatically generating single-point constraints, and for specifying non-zero values for boundary conditions are described next. Then, the equations used by **UAI/*NASTRAN*** for implementing fixed and prescribed motions are presented to conclude this section.

6.5.1 The AUTOSPC and AUTOSING Features

UAI/***NASTRAN*** provides two Case Control commands for conveniently controlling the handling of singularities. The first command is **AUTOSPC**:



This command requests that all GRID point singularities not previously removed by means of SPCs, MPCs or rigid elements be removed automatically. **UAI/*NASTRAN*** identifies the singularities and the direction of the singularity for each GRID point. For situations where the singularity is at an angle to the GRID point output coordinate system, constraint equations (MPCs) which describe the singularity in terms of its related coordinates are automatically created. An optional technique, which you may select, applies SPCs to remove the singularities. This method, which is only an approximation, can introduce small *springs-to-ground* into the model. In both cases, the AUTOSPC feature actually changes the definition of the displacement sets as originally defined in the model.

In linear analysis, the AUTOSPC operation is applied twice. First on the g-set metrices, and then on the n-set matrices. This insures that all singularities are removed. When performing nonlinear analyses, the second application of AUTOSPC to the *n-set* may be very expensive for large models. Since the application of AUTOSPC to the *g-set* is generally sufficient to remove the singularities, by default it is not performed on the *n-set*, i.e. **NONSET** is selected. You may request that it be performed by using the **NSET** option.

The second command, **AUTOSING**, provides a method for treating singularities identified during symmetric decomposition operations.

```
AUTOSING [ ( [ { PRINT   } ] [ { NULL   } ] [ { ZERO   } ] [ { NOCOMP  } ]
           [ { NOPRINT } ] [ { NONULL } ] [ { NOZERO } ] [ { COMPUTE } ]

                                  [ MAXRATIO = x ] [ { GO   } ] ) ] [ = { YES } ]
                                                   [ { NOGO } ]      [   { NO  } ]
```

This command is particularly useful in special modeling situations where AUTOSPC can not completely remove all singularities.

6

6.5.2 Enforced Displacements

UAI/*NASTRAN* allows you to specify enforced displacements for degrees of freedom in your model. Two approaches are provided for defining enforced displacements in statics analysis, and a third approach is provided for defining enforced motions (accelerations, velocities and displacements) in dynamic response analysis.

For static analysis, two methods are provided for reasons of computational efficiency in different modeling situations. In each case, the degrees of freedom with defined, non-zero motion are placed in the *s-set*.

With the first approach, the non-zero values are specified using **SPC** Bulk Data entries which are requested in the normal fashion with an **SPC** Case Control command. This procedure is efficient if the same SPC condition is used for all SUBCASEs in an analysis. It is not efficient if there are multiple SUBCASEs and only the non-zero values are changing from one SUBCASE to another, because a separate decomposition of the stiffness matrix will be performed for each unique SPC condition.

In the second approach, the degrees of freedom with enforced motion are placed in the *s-set* with zero motion assigned, using either **SPC** or **SPC1** Bulk Data entries. The non-zero values of enforced motion are then defined with **SPCD** Bulk Data entries. Enforced displacements are then requested using both the **SPC** Case Control command and the static load request with the Case Control commands:

```
SPC = spc_sid
LOAD = load_sid
```

The third approach for defining enforced motions is for use in dynamic response analysis. This approach is described in detail in Chapters 13 and 14 for each discipline of frequency and transient analysis.

6.5.3 Equations for Fixed and Prescribed Motion

After multipoint constraints have been eliminated as described earlier in Section 6.4, a partitioning of each *n-set* matrix is performed to account for the single point constraints:

$$\boldsymbol{K}_{nn} \rightarrow \begin{bmatrix} \boldsymbol{K}_{ff} & \boldsymbol{K}_{fs} \\ \boldsymbol{K}_{sf} & \boldsymbol{K}_{ss} \end{bmatrix} \text{ and } \boldsymbol{M}_{nn} \rightarrow \begin{bmatrix} \boldsymbol{M}_{ff} & \boldsymbol{M}_{fs} \\ \boldsymbol{M}_{sf} & \boldsymbol{M}_{ss} \end{bmatrix} \tag{6-22}$$

$$\boldsymbol{B}_{nn} \rightarrow \begin{bmatrix} \boldsymbol{B}_{ff} & \boldsymbol{B}_{fs} \\ \boldsymbol{B}_{sf} & \boldsymbol{B}_{ss} \end{bmatrix} \text{ and } \boldsymbol{K}^4_{nn} \rightarrow \begin{bmatrix} \boldsymbol{K}^4_{ff} & \boldsymbol{K}^4_{fs} \\ \boldsymbol{K}^4_{sf} & \boldsymbol{K}^4_{ss} \end{bmatrix} \tag{6-23}$$

6.5.3.1 Static Analysis

The equations of motion for static analysis may then be written:

$$\begin{bmatrix} \boldsymbol{K}_{ff} & \boldsymbol{K}_{fs} \\ \boldsymbol{K}_{sf} & \boldsymbol{K}_{ss} \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}_f \\ \boldsymbol{u}_s \end{Bmatrix} = \begin{Bmatrix} \overline{\boldsymbol{P}}_f \\ \boldsymbol{P}_s \end{Bmatrix} \tag{6-24}$$

Moving the prescribed motions, $\boldsymbol{u}_s$, to the right hand side results in:

$$\boldsymbol{K}_{ff}\,\boldsymbol{u}_f = \boldsymbol{P}_f \text{ and } \boldsymbol{P}_f = \overline{\boldsymbol{P}}_f - \boldsymbol{K}_{fs}\,\boldsymbol{u}_s \tag{6-25}$$

6.5.3.2 Dynamic Analysis

Traditionally, enforced motion in Dynamic analysis is implemented by using the large-mass approach. This method lacks robustness in that solution accuracy is poor when the large mass is too small, and singularities may occur when the large mass is too large. **UAI/*NASTRAN*** provides a more reliable method based on relative motion. Consider the following partitions of the system matrices:

$$\boldsymbol{K}_{nn} \rightarrow \begin{bmatrix} \boldsymbol{K}_{ff} & \boldsymbol{K}_{fs} \\ \boldsymbol{K}_{sf} & \boldsymbol{K}_{ss} \end{bmatrix} \rightarrow \begin{bmatrix} \overline{\boldsymbol{K}}_{aa} & \boldsymbol{K}_{ao} & \overline{\boldsymbol{K}}_{as} \\ \boldsymbol{K}_{oa} & \boldsymbol{K}_{oo} & \boldsymbol{K}_{os} \\ \overline{\boldsymbol{K}}_{sa} & \boldsymbol{K}_{so} & \overline{\boldsymbol{K}}_{ss} \end{bmatrix} \tag{6-26}$$

$$\boldsymbol{M}_{nn} \rightarrow \begin{bmatrix} \boldsymbol{M}_{ff} & \boldsymbol{M}_{fs} \\ \boldsymbol{M}_{sf} & \boldsymbol{M}_{ss} \end{bmatrix} \rightarrow \begin{bmatrix} \overline{\boldsymbol{M}}_{aa} & \boldsymbol{M}_{ao} & \overline{\boldsymbol{M}}_{as} \\ \boldsymbol{M}_{oa} & \boldsymbol{M}_{oo} & \boldsymbol{M}_{os} \\ \overline{\boldsymbol{M}}_{sa} & \boldsymbol{M}_{so} & \overline{\boldsymbol{M}}_{ss} \end{bmatrix} \tag{6-27}$$

$$\boldsymbol{B}_{nn} \rightarrow \begin{bmatrix} \boldsymbol{B}_{ff} & \boldsymbol{B}_{fs} \\ \boldsymbol{B}_{sf} & \boldsymbol{B}_{ss} \end{bmatrix} \rightarrow \begin{bmatrix} \overline{\boldsymbol{B}}_{aa} & \boldsymbol{B}_{ao} & \overline{\boldsymbol{B}}_{as} \\ \boldsymbol{B}_{oa} & \boldsymbol{B}_{oo} & \boldsymbol{B}_{os} \\ \overline{\boldsymbol{B}}_{sa} & \boldsymbol{B}_{so} & \overline{\boldsymbol{B}}_{ss} \end{bmatrix} \tag{6-28}$$

$$\boldsymbol{K}^4_{nn} \rightarrow \begin{bmatrix} \boldsymbol{K}^4_{ff} & \boldsymbol{K}^4_{fs} \\ \boldsymbol{K}^4_{sf} & \boldsymbol{K}^4_{ss} \end{bmatrix} \rightarrow \begin{bmatrix} \overline{\boldsymbol{K}}^4_{aa} & \boldsymbol{K}^4_{ao} & \overline{\boldsymbol{K}}^4_{as} \\ \boldsymbol{K}^4_{oa} & \boldsymbol{K}^4_{oo} & \boldsymbol{K}^4_{os} \\ \overline{\boldsymbol{K}}^4_{sa} & \boldsymbol{K}^4_{so} & \overline{\boldsymbol{K}}^4_{ss} \end{bmatrix} \tag{6-29}$$

$$\boldsymbol{P}_n \rightarrow \begin{Bmatrix} \boldsymbol{P}_f \\ \overline{\boldsymbol{P}}_s \end{Bmatrix} \rightarrow \begin{Bmatrix} \overline{\boldsymbol{P}}_a \\ \boldsymbol{P}_o \\ \overline{\boldsymbol{P}}_s \end{Bmatrix} \tag{6-30}$$

Define the following transformations:

$$\boldsymbol{G}_{os} = -\boldsymbol{K}_{oo}^{-1}\boldsymbol{K}_{os} \quad \text{and} \quad \boldsymbol{G}_{oa} = -\boldsymbol{K}_{oo}^{-1}\boldsymbol{K}_{oa} \tag{6-31}$$

and write:

$$\boldsymbol{u}_o = [\,\boldsymbol{G}_{oa}\,\boldsymbol{G}_{os}\,] \begin{Bmatrix} \boldsymbol{u}_a \\ \boldsymbol{u}_s \end{Bmatrix} \tag{6-32}$$

then:

$$\boldsymbol{u}_n = \begin{Bmatrix} \boldsymbol{u}_f \\ \boldsymbol{u}_s \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{u}_a \\ \boldsymbol{u}_o \\ \boldsymbol{u}_s \end{Bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{G}_{oa} & \boldsymbol{G}_{os} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}_a \\ \boldsymbol{u}_s \end{Bmatrix} \tag{6-33}$$

Transforming the equations of motion yields:

$$\begin{bmatrix} \boldsymbol{M}_{aa} & \boldsymbol{M}_{as} \\ \boldsymbol{M}_{sa} & \boldsymbol{M}_{ss} \end{bmatrix} \begin{Bmatrix} \ddot{\boldsymbol{u}}_a \\ \ddot{\boldsymbol{u}}_s \end{Bmatrix} + \begin{bmatrix} \boldsymbol{B}_{aa} & \boldsymbol{B}_{as} \\ \boldsymbol{B}_{sa} & \boldsymbol{B}_{ss} \end{bmatrix} \begin{Bmatrix} \dot{\boldsymbol{u}}_a \\ \dot{\boldsymbol{u}}_s \end{Bmatrix} + \begin{bmatrix} \boldsymbol{K}_{aa} & \boldsymbol{K}_{as} \\ \boldsymbol{K}_{sa} & \boldsymbol{K}_{ss} \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}_a \\ \boldsymbol{u}_s \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{P}_a \\ \boldsymbol{P}_s \end{Bmatrix} \tag{6-34}$$

In which the partitions of the $\boldsymbol{K}$, $\boldsymbol{M}$, and $\boldsymbol{B}$, may be written (using $\boldsymbol{K}$ as an example):

$$\boldsymbol{K}_{aa} = \overline{\boldsymbol{K}}_{aa} + \boldsymbol{G}_{oa}^T\boldsymbol{K}_{oa} + \boldsymbol{K}_{oa}^T\boldsymbol{G}_{oa} + \boldsymbol{G}_{oa}^T\boldsymbol{K}_{oo}\boldsymbol{G}_{oa} \tag{6-35}$$

$$\boldsymbol{K}_{as} = \overline{\boldsymbol{K}}_{as} + \boldsymbol{G}_{oa}^T\boldsymbol{K}_{os} + \boldsymbol{K}_{oa}^T\boldsymbol{G}_{os} + \boldsymbol{G}_{oa}^T\boldsymbol{K}_{oo}\boldsymbol{G}_{os} \tag{6-36}$$

$$\boldsymbol{K}_{ss} = \overline{\boldsymbol{K}}_{ss} + \boldsymbol{G}_{os}^T\boldsymbol{K}_{os} + \boldsymbol{K}_{os}^T\boldsymbol{G}_{os} + \boldsymbol{G}_{os}^T\boldsymbol{K}_{oo}\boldsymbol{G}_{os} \tag{6-37}$$

This same set of equations can be written for $\boldsymbol{M}$, $\boldsymbol{B}$, and $\boldsymbol{K}^4$, as well.

$$\boldsymbol{P}_a = \overline{\boldsymbol{P}}_a + \boldsymbol{G}_{oa}^T \boldsymbol{P}_o \quad \text{and} \quad \boldsymbol{P}_s = \overline{\boldsymbol{P}}_s + \boldsymbol{G}_{os}^T \boldsymbol{P}_o \tag{6-38}$$

Now, let:

$$\boldsymbol{D} = -\boldsymbol{K}_{aa}^{-1} \boldsymbol{K}_{as} \tag{6-39}$$

and define the transformation

$$\begin{Bmatrix} \boldsymbol{u}_a \\ \boldsymbol{u}_s \end{Bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{D} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}_a^r \\ \boldsymbol{u}_s \end{Bmatrix} \tag{6-40}$$

Then:

$$\boldsymbol{M}_{aa} \ddot{\boldsymbol{u}}_a^r + \boldsymbol{B}_{aa} \dot{\boldsymbol{u}}_a^r + \boldsymbol{K}_{aa} \boldsymbol{u}_a^r = \boldsymbol{P}_a - \hat{\boldsymbol{M}}_{as} \ddot{\boldsymbol{u}}_s - \hat{\boldsymbol{B}}_{as} \dot{\boldsymbol{u}}_s = \boldsymbol{P}_a^* \tag{6-41}$$

in which:

$$\hat{\boldsymbol{M}}_{as} = \boldsymbol{M}_{aa} \boldsymbol{D}_{as} + \boldsymbol{M}_{as} \quad \text{and} \quad \hat{\boldsymbol{B}}_{as} = \boldsymbol{B}_{aa} \boldsymbol{D}_{as} + \boldsymbol{B}_{as} \tag{6-42}$$

For damping which is proportional to stiffness, the matrix $\hat{\boldsymbol{B}}_{as}$ is null. Similarly, for proportional damping, $\hat{\boldsymbol{B}}_{as} = k \hat{\boldsymbol{M}}_{as}$, where k is a constant. In most applications, the former condition is true, and thus:

$$\boldsymbol{P}_a^* = \boldsymbol{P}_a - \hat{\boldsymbol{M}}_{as} \ddot{\boldsymbol{u}}_s \tag{6-43}$$

6.5.4 Forces of Single Point Constraint

The single-point constraint forces (SPC Forces) are computed for each solution vector — whether a static loading condition, and eigenvector, or an output time or frequency step in dynamic response. You request these results with the Case Control command:

SPCFORCES = { ALL / *sid* / NONE }

In static analysis, the forces are computed using:

$$\boldsymbol{q}_s = \boldsymbol{K}_{fs}\boldsymbol{u}_f + \boldsymbol{K}_{ss}\boldsymbol{Y}_s \tag{6-44}$$

where $\boldsymbol{Y}_s$ are the enforced displacements defined by **SPCD** Bulk Data entries.

When you are using enforced motion in dynamic response analysis, (6-44) is modified to be:

$$\boldsymbol{q}_s = \boldsymbol{K}_{fs}\boldsymbol{u}_f + \boldsymbol{K}_{ss}\left[\boldsymbol{Y}_s + \int \dot{\boldsymbol{Y}}_s\, dt + \int\int \ddot{\boldsymbol{Y}}_s\, dt\, dt\right] \tag{6-45}$$

where $\dot{\boldsymbol{Y}}_s$ and $\ddot{\boldsymbol{Y}}_s$ are the enforced velocities and accelerations. The integrations are performed automatically if you have applied enforced motions.

6.6 REDUCTION OF EQUATIONS OF MOTION

When solving large analyses, particularly in the normal modes and dynamic response disciplines, it is highly recommended that you perform one or more reduction processes which reduce the number of equations of motion while still resulting in the accuracy that you require. **UAI/*NASTRAN*** provides four methods which may be selected individually or in combination:

- Static Condensation or Guyan Reduction
- Dynamic Reduction
- Modal Transformation
- Component Mode Synthesis

Static Condensation and Dynamic Reduction are techniques available to transform the equations of motion for more efficient dynamics analysis. These procedures allows the removal of massless degrees of freedom from a dynamic analysis while maintaining stiffness fidelity. The procedures are especially useful when using the Givens or modified Givens eigenvalue extraction methods because these techniques require or work best with non-zero mass data and with relatively small sized matrices.

Static Condensation techniques are also often used in substructuring statics analysis, wherein a typical objective is to reduce many substructures to a relatively small sized set of equations before combining the substructures to construct a large system model.

The Modal Transformation feature is provided for use in dynamic response analysis. This feature performs a transformation from physical degrees of freedom to generalized, normal mode degrees of freedom for a more computationally efficient solution of many dynamics problems. This technique is often used following either a static or dynamic reduction.

The Component Mode Synthesis feature is an integral part of the automated substructuring analysis capability of **UAI/*NASTRAN***. This feature is described in Chapter 19 of this manual.

In the same manner as previously done for multipoint and single-point constraints, this section presents both descriptive material describing how to use the features and summaries of the equations which are used.

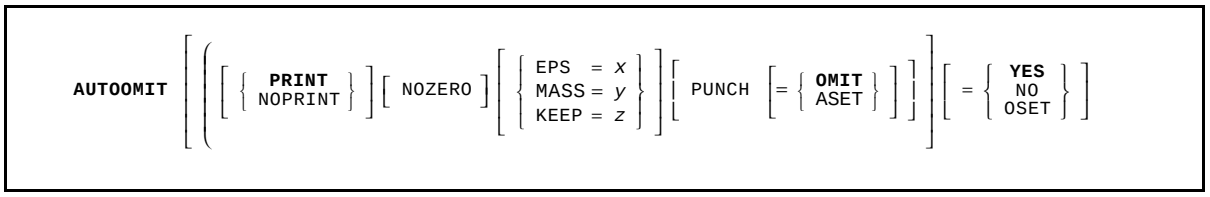
6.6.1 Static Condensation

You select static condensation in **UAI/*NASTRAN*** in one of three ways:

- By defining the retained, or *a-set*, degrees of freedom
- By defining the omitted, or *o-set*, degrees of freedom
- By requesting that **UAI/*NASTRAN*** automatically select one of these sets

In the first two cases, you define the retained set with the **ASET** family of Bulk Data entries or the omitted set with the **OMIT** family of Bulk Data entries. Only one of these types of data may be present in your Bulk Data packet. The data is then automatically used to perform the static condensation. These Bulk Data are simply lists of GRID point identification numbers and component degrees of freedom which will be placed in either the *a-set* or the *o-set*. The third method is to use the Case Control command **AUTOOMIT** to automatically select degrees of freedom based on one of several optional criteria.

The format of this command is:



When this command is used, **UAI/*NASTRAN*** automatically defines the *a-set* and *o-set* data based on selected options. This command must be placed above the SUBCASE level in the Case Control packet. **UAI/*NASTRAN*** will then process the *a-set* as defined by the Bulk Data, identify components to be omitted based on the specified options, and move those components to the *o-set*.

6.6.1.1 Standard Uses of AUTOOMIT

The standard use of this feature is invoked with the commands.

```
AUTOOMIT (option_list)
   or
AUTOOMIT (option_list) = YES
```

If no options are specified, the defaults are used. This means that all massless components not otherwise constrained are moved to the *o-set* and that a summary table showing which components that have changed sets will be printed.

If the **NOPRINT** option is specified, then the summary table is not printed. The table will print only those non-zero mass components which are assigned to the *o-set* if the **NOZERO** option is specified. If the **PUNCH** option is specified, then **OMIT** Bulk Data entry images are written to file BULK, which must be requested using an **ASSIGN** Executive Control command.

For most engineering problems, it is not efficient to reduce only the massless components which usually represent about half the degrees of freedom. For this reason the special options **KEEP**, **EPS** and **MASS** are available.

The **KEEP** option specifies the percentage of the component degrees of freedom with the highest mass to stiffness ratios that will be retained in the *a-set*. For example:

```
AUTOOMIT(KEEP=10.0) = YES
```

results in components (the first 10 percent) whose ratios of their diagonal mass terms to stiffness terms is the highest positive number. These components are then *kept* in the *a-set* and the remaining components not otherwise constrained are moved to the *o-set*.

The **EPS** option allows you to define a specific mass to stiffness ratio below which components will be moved from the *a-set* to the *o-set*. The default option is **EPS=0.0**. This causes all null mass components to be omitted. Similarly, the **MASS** option allows you to define a specific value of diagonal term mass below which components are moved from the *a-set* to the *o-set*.

Finally, the **KEEP** option is designed for general use, whereas the **EPS** and **MASS** options require a greater knowledge of the model including the specific values of terms in the mass and stiffness matrices. When using this form of AUTOOMIT, no **ASET** Bulk Data entries should be present.

6.6.1.2 Automatic Use of AUTOOMIT

UAI***/NASTRAN*** automatically activates the AUTOOMIT function in either normal modes analysis or dynamic response analysis using the modal transformation approach whenever the following are all true:

- ❒ A Givens or modified Givens eigenvalue extraction is requested
- ❒ No dynamic reduction is requested
- ❒ No AUTOOMIT Command is provided
- ❒ No **ASET** or **OMIT** Bulk Data entries are present

6.6.1.3 Alternate Uses of AUTOOMIT

Certain modeling situations arise, as with automated substructuring, where you desire specific components with zero mass to remain in the *a-set*, while the remaining components of the *a-set* are to be automatically computed. For example, some *boundary* components for a substructure may have zero mass components, but these components must not be omitted during a substructuring Phase 1 operation or their stiffness contributions to a boundary with another structure would be reduced into the interior degrees of freedom. This would be necessary, for example, to avoid *pinned* connections between substructures. This alternate use of AUTOOMIT is activated by two items. First, **ASET** Bulk Data are used to indicate which components are to be placed in the *a-set*, second the Case Control command **AUTOOMIT=OSET** is used to indicate that the otherwise *o-set* components are to be moved to the *a-set* based on the other options (**MASS, KEEP**, or **EPS**). The default option is that all non-zero mass components are moved from the *o-set* to join the any *a-set* components that you have defined.

6.6.2 Dynamic Reduction

UAI***/NASTRAN*** includes a feature called dynamic reduction, this feature is designed to reduce the cost of most dynamics analyses while retaining good accuracy.

The major advantages of dynamic reduction compared to the Guyan reduction techniques are:

- ❒ The accuracy of results is neither dependent upon a *good* manual selection of *a-set* degrees of freedom, which is error-prone, nor on an automatic selection of *a-set* degrees of freedom which is less accurate for structures with nearly uniform mass distributions.
- ❒ Accuracy is good because the generalized coordinates represent the flexible shapes of the structure in a frequency range that you specify. **UAI/*NASTRAN*** automatically selects the number of coordinates needed for an accurate solution.
- ❒ You may retain *a-set* degrees of freedom in the reduced model. This is necessary when using automated substructuring because *a-set* degrees of freedom are required to combine two or more substructures.

The dynamic reduction feature involves transforming a larger set of equations, typically describing the motions of all the unconstrained degrees of freedom of the model, to a much smaller set of equations which describe the behavior of one or more types of generalized coordinates. These coordinates are selected to provide high accuracy over a frequency range that you select for a particular problem or model. Traditional alternatives to the dynamic

reduction method have included static condensation, often called Guyan reduction, or direct solution of the larger set of equations if that is possible or economically feasible.

For the purpose of describing the Dynamic Reduction procedure, a displacement set, called the *k-set* denotes the generalized coordinates which define the flexible shapes of the structure which are computed using a general Krylov subspace iteration technique. These shapes are approximate eigenvectors of the model.

Dynamic reduction is requested using the Case Control command:

```
DYNRED = sidk
```

and a Bulk Data entry of the form:

```
DYNRED , sidk , fmax
```

where *sidk* is referenced in Case Control command, *fmax* is the maximum frequency of interest, that is, the same as the **F2** field on an **EIGR** Bulk Data entry.

Several additional parameters may be defined on the **DYNRED** Bulk Data entry, however it is recommended that the default values be normally used. The UAI staff should be consulted for advice on use of these additional parameters. The **DYNRED** Bulk Data with the fmax definition causes the automatic definition of SCALAR points representing the *k-set*. The identification numbers begin with one greater than the largest identification number that you have defined for any point.

Additionally, **ASET, ASET1** and **SUPORT** Bulk Data entries may be used to define points to remain in the expanded *a-set* as indicated by the transformation equation presented above. These Bulk Data have no Case Control reference.

Dynamic reduction is not supported in statics analysis. It is designed to replace the use of large and sometimes difficult to determine *a-set* data for normal or complex modes and frequency and transient response analyses.

6

6.6.3 Basic Theory for Reduction Procedures

The following discussion presents the equations used by **UAI/*NASTRAN*** to reduce the equations of motion from the *f-set* to the *a-set* using either the Guyan reduction or dynamic reduction procedure. Both techniques use the same general form of equations, as presented next. Only the creation of the matrix $\boldsymbol{G}_{oa}$ differs between the two techniques. Different procedures for creating $\boldsymbol{G}_{oa}$ are presented following the summary of reduction procedure equations.

First, the *f-set* system coefficient matrices are partitioned:

$$\boldsymbol{K}_{ff} \rightarrow \begin{bmatrix} \overline{\boldsymbol{K}}_{aa} & \boldsymbol{K}_{ao} \\ \boldsymbol{K}_{oa} & \boldsymbol{K}_{oo} \end{bmatrix} \tag{6-46}$$

$$\boldsymbol{M}_{ff} \rightarrow \begin{bmatrix} \overline{\boldsymbol{M}}_{aa} & \boldsymbol{M}_{ao} \\ \boldsymbol{M}_{oa} & \boldsymbol{M}_{oo} \end{bmatrix} \tag{6-47}$$

$$\boldsymbol{B}_{ff} \rightarrow \begin{bmatrix} \overline{\boldsymbol{B}}_{aa} & \boldsymbol{B}_{ao} \\ \boldsymbol{B}_{oa} & \boldsymbol{B}_{oo} \end{bmatrix} \tag{6-48}$$

$$K^4_{ff} \rightarrow \begin{bmatrix} \overline{K}^4_{aa} & K_{ao} \\ M^4_{oa} & K^4_{oo} \end{bmatrix} \tag{6-49}$$

$$P_f \rightarrow \begin{Bmatrix} \overline{P}_a \\ P_o \end{Bmatrix} \tag{6-50}$$

The following transformation is defined between the *f-set* and *a-set* degrees of freedom.

$$\{u_f\} \rightarrow \begin{Bmatrix} u_a \\ u_o \end{Bmatrix} = \begin{bmatrix} I \\ G_{oa} \end{bmatrix} u_a \tag{6-51}$$

This transformation matrix is next applied to the equations of motion resulting in the following calculations:

$$K_{aa} = \overline{K}_{aa} + K_{ao} G_{oa} \tag{6-52}$$

$$M_{aa} = \overline{M}_{aa} + G^T_{oa} M_{oa} + M^T_{oa} G_{oa} + G^T_{oa} M_{oo} G_{oa} \tag{6-53}$$

$$B_{aa} = \overline{B}_{aa} + G^T_{oa} B_{oa} + B^T_{oa} G_{oa} + G^T_{oa} B_{oo} G_{oa} \tag{6-54}$$

$$K^4_{aa} = \overline{K}^4_{aa} + G^T_{oa} K^4_{oa} + K^{4T}_{oa} G_{oa} + G^T_{oa} K^4_{oo} G_{oa} \tag{6-55}$$

$$P_a = \overline{P}_a + G^T_{oa} P_o \tag{6-56}$$

6.6.4 Static Condensation Transformation Matrix

The concept behind development of the static condensation transformation is that inertia effects of the *o-set* degrees of freedom may be ignored. Also, it is assumed that damping effects on these *o-set* degrees of freedom may also be ignored. If this is the case, then we can write the equation of free vibration as:

$$\begin{bmatrix} M_{aa} & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \ddot{u}_a \\ \ddot{u}_o \end{Bmatrix} + \begin{bmatrix} K_{aa} & K_{ao} \\ K_{oa} & K_{oo} \end{bmatrix} \begin{Bmatrix} u_a \\ u_o \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{6-57}$$

The second row of this equation can be easily rewritten, leaving only u_o on the left side to yield:

$$u_o = -K^{-1}_{oo} K_{oa} u_a \tag{6-58}$$

Then, rewriting (6-41) as:

$$u_o = G_{oa} u_a \tag{6-59}$$

results in:

$$\boldsymbol{G}_{oa} = -\boldsymbol{K}_{oo}^{-1}\,\boldsymbol{K}_{oa} \tag{6-60}$$

6.6.5 Dynamic Reduction Transformation Matrix

Dynamic reduction of the equations of motion to generalized coordinates is based upon a discrete Rayleigh-Ritz reduction. The assumption is made that the physical motions of the *f-set* degrees of freedom may be redefined in terms of some combination of the following degrees of freedom:

- Approximate eigenvector shapes
- Static displacement shapes

As developed above for Guyan reduction, a transformation will be developed between the *o-set* and *a-set*. For the purposes of this description, we will call the matrix $\overline{\boldsymbol{G}}_{oa}$. It has the general form as shown below:

or,

$$u_o = \overline{\boldsymbol{G}}_{oa}\,\overline{\boldsymbol{u}}_a \tag{6-61}$$

$$\left\{\boldsymbol{u}_f\right\} \rightarrow \begin{Bmatrix}\overline{\boldsymbol{u}}_a \\ \boldsymbol{u}_o\end{Bmatrix} = \begin{bmatrix}\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{G}_{oa} & \boldsymbol{G}_{ok}\end{bmatrix}\begin{Bmatrix}\boldsymbol{u}_a \\ \boldsymbol{u}_k\end{Bmatrix} \tag{6-62}$$

The equation above illustrates that the original *a-set* is expanded to include new degrees of freedom to accommodate the *k-set*. The *k-set* DOF represent the approximate eigenvector shapes.

6.6.5.1 Approximate Eigenvector Shapes

The *k-set* generalized coordinates are developed using a subspace iteration technique which yields a set of approximate, nearly orthogonal eigenvectors

$$\Phi_{ok} = \left\{\varphi_1, \varphi_2, \ldots, \varphi_k\right\}$$

The *o-set* degrees of freedom are related to these eigenvectors by:

$$\boldsymbol{u}_o = \Phi_{ok}\,\boldsymbol{u}_k \tag{6-63}$$

These vectors are used directly to create the partition $\boldsymbol{G}_{ok}$ of (6-62).

6.6.5.2 Static Displacement Shapes

The static displacement shapes are computed using the procedures of static condensation described earlier. The matrix partition $\boldsymbol{G}_{oa}$ is computed as shown in (6-59). This partition defines the displacement of all *o-set* degrees of freedom given unit displacement of each *a-set* degrees of freedom.

6.7 THE AUTOMATIC REDUCTION FEATURES

Large order, high fidelity finite element models often describe the dynamic behavior of structural components. In order to alleviate demands for computer resources and reduce analysis time, reduced order models may be defined. Reduced order models must satisfy two primary requirements; they must maintain fidelity within a user-specified frequency band, and they must retain a complete and consistent boundary degree of freedom description. The reduced order structural component model may be exported using **DMIG** Bulk Data entries. This facilitates integration with a complete system model, or another user application.

Two methods are offered for automatic reduction of a component finite element model, static condensation and modal reduction. The first method, static condensation, also known as Guyan reduction, produces a reduced order model described in terms of physical displacement degrees of freedom. Those degrees of freedom corresponding to the component boundary are not statically condensed. The second method, modal reduction, develops a reduced order model described in terms of generalized modal degrees of freedom and the complete set of boundary displacement degrees of freedom. A variety of modal reduction options (fixed, free or mixed boundary), corresponding to accepted engineering practice, are available.

Automatic reduction has been implemented in the multidisciplinary solution sequence, **SOL MULTI**. The available reduction methods are described in the following sections.

6.7.1 Static Reduction

In order to develop a reduced model by the static condensation method, two solution **CASE**s are required. Constraints imposed in both of these **CASE**s must be identical. The first **CASE** establishes baseline modes for the unreduced component, for example:

```
SOL MULTI
CEND
...
CASE 1 MODES
   METHOD = 100
   SPC    = 10
   ...
```

The user should specify the frequency band of interest, rather than the number of modes sought, on the **EIGR** Bulk Data entry. Because this **CASE** is intended to result in accurate modes, the Guyan reduction approximation should not be imposed by using **ASET** or **OMIT** Bulk Data entries. Moreover, for large-order models, **LANCZOS** should be used for computational efficiency.

The second **CASE** establishes physical degrees of freedom required by the statically condensed model to accurately describe component modes in the specified frequency band, and to include the complete set of boundary degrees of freedom. For example:

```
CASE 2 REDUCE STATICS
   USING MODES = 1
   METHOD = 1
   SPC = 10
   BOUNDARY = 10
   AUTOREDUCE(SUMMARY,MAXITER=5,TOLER=1.0,KEEP=10.0,INCREMENT=10.0)
      ...
```

This second **CASE USE**s the modal frequencies from **CASE 1** as the baseline values. It then iterates on a set of degrees of freedom, excluding those on the boundary, that will be retained

in the *a-set*. This iteration is specified by the Case Control command **AUTOREDUCE**. The general format format of this command is:

AUTOREDUCE [([{ NONE / **SUMMARY** / DETAIL }] [{ **KEEP** = *x* , *delx* / EPS = *y* , *dely* / MASS = *z* , *delz* }] [TOLER [= { *tid* / *toler* }]]) [{ **ABSOLUTE** / PERCENT }] [**MAXITER** = *niter*]] [= { **YES** / NO }]

The iteration procedure begins by setting the iteration counter to zero, and then selecting the initial set of retained degrees of freedom, $\boldsymbol{u}_a^o$, using the value of *delx*, *dely* or *delz* and the option specified by **EPS**, **MASS** or **KEEP**. These sets are defined as:

$$\boldsymbol{u}_a^i = \left\{ u \;\middle|\; \frac{\boldsymbol{M}_{uu}}{\boldsymbol{K}_{uu}} \geq x \right\} \qquad \text{for } \textbf{EPS}$$

$$\boldsymbol{u}_a^i = \left\{ u \;\middle|\; \boldsymbol{M}_{uu} \geq y \right\} \qquad \text{for } \textbf{MASS}$$

$$\boldsymbol{u}_a^i = \left\{ u \;\middle|\; u \text{ is in the top } z \text{ percent of the sorted list of } \frac{\boldsymbol{M}_{uu}}{\boldsymbol{K}_{uu}} \right\} \qquad \text{for } \textbf{KEEP}$$

where the superscript *i* denotes the iteration number.

The initial reduction matrix, $\boldsymbol{G}_{oa}^o$, is then computed, and the stiffness and mass matrices, $\boldsymbol{K}_{aa}^o$ and $\boldsymbol{M}_{aa}^o$ are then reduced to this set using(6-35) through (6-37). The standard eigenproblem is then formulated:

$$\left[\boldsymbol{K}_{aa}^o - \lambda^o \boldsymbol{M}_{aa}^o \right] \boldsymbol{\Phi}^o = \boldsymbol{0} \qquad (6\text{-}64)$$

where λ^o is an eigenvalue and $\boldsymbol{\Phi}^o$ the eigenvector associated with λ^o. Each eigenvalue represents a natural frequency (in radians), ω^o ,of the system, and a cyclic frequency, f^o, which are related by:

$$f^o = \frac{\omega^o}{2\pi} = \frac{\sqrt{\lambda^o}}{2\pi} \qquad (6\text{-}65)$$

The iteration counter is then set to 1 and the set $\boldsymbol{u}_a^i$ is computed by the appropriate computation depending on the selected options:

$$\boldsymbol{u}_a^i = \left\{ u \;\middle|\; \frac{\boldsymbol{M}_{uu}}{\boldsymbol{K}_{uu}} \geq x - i\,\Delta x \;\; or \;\; \frac{\boldsymbol{M}_{uu}}{\boldsymbol{K}_{uu}} \geq (1 - i\,\Delta x)\, x \right\} \qquad \text{for } \textbf{EPS}$$

$$\boldsymbol{u}_a^i = \left\{ u \;\middle|\; \boldsymbol{M}_{uu} \geq y - i\,\Delta y \text{ or } \boldsymbol{M}_{uu} \geq (1 - i\,\Delta y)\,y \right\} \qquad \text{for } \textbf{MASS}$$

$$\boldsymbol{u}_a^i = \left\{ u \;\middle|\; \begin{array}{l} u \text{ is in the top } (z + i\,\Delta z) \\ \qquad \text{percent of the sorted list of } \dfrac{\boldsymbol{M}_{uu}}{\boldsymbol{K}_{uu}} b \\ \quad or \\ u \text{ is in the top } (1 + i\,\Delta z)\,z \\ \qquad \text{percent of the sorted list of } \dfrac{\boldsymbol{M}_{uu}}{\boldsymbol{K}_{uu}} b \end{array} \right\} \qquad \text{for } \textbf{KEEP}$$

The new reduction matrix, $\boldsymbol{G}_{oa}^i$, is then computed, and the dynamic equations of motion are then reduced to this set using (6-64) to (6-67) giving $\boldsymbol{K}_{aa}^i$ and $\boldsymbol{M}_{aa}^i$. The new eigenproblem is then solved:

$$\left[\boldsymbol{K}_{aa}^i - \lambda^i \boldsymbol{M}_{aa}^i \right] \boldsymbol{\Phi}^i = \boldsymbol{0} \tag{6-66}$$

Convergence is then checked by comparing the cyclic frequencies for the current iteration, *i*, with the initial frequencies:

$$\Delta \boldsymbol{f}^i = \boldsymbol{f}^i - \boldsymbol{f}^o \leq \mathit{toler}$$

If each frequency satisfies this requirement, then the iteration terminates. There are two ways to specify the convergence criterion. The first is with an explicit tolerance, *toler*. The second way is to reference **TABLED1** Bulk Data having the identification number *tid*. This table gives the *toler* value as a function of natural frequency. Thus, you can make the convergence tighter for low frequency modes, and looser for the less important high frequency modes.

The final displacement set, $\boldsymbol{u}_a^i$ is used to perform the reduction of the stiffness, mass and damping matrices of the initial system. These matrices may then be written to the Bulk file in the form of **DMIG** Bulk Data entries using the **EXPORT REDUCED MODEL** command described below. The **MAXITER** option of the **AUTOREDUCE** command provides an absolute limit for the number of iterations performed. A **BOUNDARY** Case Control command is required to define the physical boundary degrees of freedom to be retained in the *a-set* during the reduction process, and an **SPC** command is required to remove singularities.

6.7.2 Modal Reduction

The reduction of a model to a set of generalized modal coordinates does not require an iterative procedure. Instead, a single **CASE** is executed. This **CASE** has the form:

```
SOL MULTI
...
CASE 1 MREDUCE
   METHOD = 100
   SPC    = 10
   BOUNDARY = 10
   ...
```

Three different classes of reduced modal models may be defined by the **SPC** and **BOUNDARY** set specifications, namely

- If all **BOUNDARY** degrees of freedom coincide with **SPC** degrees of freedom, a fixed boundary (popularly called ***Craig-Bampton*** [Craig68] reduction) component model is formed
- If no **BOUNDARY** degrees of freedom coincide with **SPC** degrees of freedom, a free boundary (equivalent to the MacNeal-Rubin method [MacNeal71], [Rubin75]) component model is formed
- If not all **BOUNDARY** degrees of freedom coincide with **SPC** degrees of freedom, a mixed boundary (equivalent to the Herting method [Herting79]) model is formed

The reduced mass and stiffness matrices associated with each of these modal reduction methods are of the form:

$$\begin{bmatrix} \boldsymbol{M}_{bb} & \boldsymbol{M}_{bq} \\ \boldsymbol{M}_{qb} & \boldsymbol{M}_{qq} \end{bmatrix}; \begin{bmatrix} \boldsymbol{K}_{bb} & \boldsymbol{0}_{bq} \\ \boldsymbol{0}_{qb} & \boldsymbol{K}_{qq} \end{bmatrix} \tag{6-67}$$

where the subscript b designates physical boundary degrees of freedom, and the subscript q designates generalized modal degrees of freedom. For the fixed boundary case, $\boldsymbol{M}_{qq}$ and $\boldsymbol{K}_{qq}$ are diagonal matrices.

6.7.3 Exporting the Reduced Model

Matrices resulting from etiher the static condensation or the modal reduction method may be exported using the Case Control command:

```
EXPORT REDUCED MODEL [TO logical_name] [STIFFNESS [= stiff_name]]
                     [MASS [= mass_name]] [DAMP [= damp_name]]
                     [GRID [({OFFSET} = n)]] [SCALAR [({OFFSET} = n)]]
                                {START}
```

The matrices are written to the Bulk Data file selected by *logical_name* in the form of **DMIG** Bulk Data entries. You may give each of the a name, e.g. *stiff_name*, or use the default database entity name used within **UAI/*NASTRAN***. In addition to the matrices, the corresponding **GRID** point and **SCALAR** point Bulk Data entries are also written to the file. In each case, you may set a **START**ing identification number, or define an **OFFSET** from the maximum identification number used in the original model. Thus, upon completion of the reduction procedure you have a self contained matrix representation of the reduced model.

These matrices may be used to couple reduced component models to other models by using direct matrix input.

6.8 THE NLREDUCE FEATURE

UAI*/NASTRAN* provides the capability of solving problems which have nonlinear material behavior. Such models generally contain both linear and nonlinear elements. The NLREDUCE feature is activated using the **NLREDUCE** Case Control command, and causes the degrees of freedom in the model to be automatically divided into a linear and a nonlinear set. This feature is installed only in nonlinear statics analysis, which is requested with an **APPROACH NONLINEAR** Executive Control command. The purpose of this feature is to provide improved computational efficiency. When NLREDUCE is requested, all degrees of freedom whose stiffness terms are affected by any nonlinear element are placed in the nonlinear set.

The partitioning to the linear and nonlinear sets takes place at the *g-set* level. The linear set is processed first for MPC constraints, then for SPC constraints, and finally a static condensation is performed to reduce the linear set to the boundary degrees of freedom common to the nonlinear set. This linear, boundary stiffness matrix is then a constant for the current subcase. When stiffness updates are performed during the nonlinear analysis, all data blocks and matrices are of the reduced size of the nonlinear set. After each nonlinear stiffness update the system stiffness is made complete by adding the linear matrix to the current nonlinear stiffness matrix.

Because of the nature and purpose of these operations, the NLREDUCE feature should not be used for nonlinear models where all elements have nonlinear properties. This procedure was developed for those models where a large number of the finite elements are linear, and the nonlinear behavior is limited to a relatively small region of the model.

6.9 RETAINING *a-set* DEGREES OF FREEDOM

If you use any of the **AUTOOMIT**, **AUTOSPC**, **AUTOREDUCE**, **NLREDUCE**, or Dynamic Reduction features, some of the *a-set* degrees of freedom in your model are automatically moved to other displacement sets. For example, **AUTOOMIT** will move massless degrees of freedom from the *a-set* to the *o-set*. There are situations where you may want to retain some of these degrees of freedom in the *a-set*. This is accomplished by defining the degrees of freeom with **BDYS** or **BDYS1** Bulk Data entries.

The format of these entries is:

BDYS	**SID**	**GID1**	**DOF1**	**GID2**	**DOF2**	**GID3**	**DOF3**		

BDYS1	**SID**	**DOF**	***GRID ID LIST***	**-cont-**

The **GIDi** values specify GRID point identification numbers, and the **DOFi** values are the derees of freedom that will be placed in the *a-set*. You then select these entries with the Case Control command:

```
BOUNDARY = sid
```

By doing this, the selected degrees of freedom will:

- ❒ Be unaffected by the **AUTOOMIT** and **AUTOSPC** operations.
- ❒ Be placed in the *a-set* if **AUTOREDUCE** or Dynamic Reduction is used.
- ❒ Be placed in the nonlinear set for static nonlinear analysis.

6.10 SOLVING FREE BODY MODELS

A free body is defined as a structure that is capable of motion without internal stress, that is, it has one or more rigid body degrees of freedom. The stiffness matrix for a free body is singular and the number of singularities equals 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 add more rigid body modes to a structure. **UAI/*NASTRAN*** places no restriction on the number of stress-free modes and hence, permits the analysis of complex mechanisms.

Static solution techniques for free body structures, provided by Rigid Format 2, utilize a temporary, restraint that you define. This restraint allows the decomposition of the stiffness matrix to proceed. The restrained degrees of freedom are given a zero displacement value in the solution and other displacements are then relative to these points. For static analyses, you must specify the restraints sufficient to constrain all rigid body motion.

Free body supports may also be specified for normal modes and other dynamic analyses of free bodies. In these cases, the rigid body mode shapes will be computed using the stiffness matrix and will be more accurate than if they are computed by the eigenvalue extraction method.

If the mode acceleration method of improved data recovery is used for dynamics problems having rigid body degrees of freedom, then free body supports must be specified. This solution improvement technique involves a static solution. Thus, 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.

SUPORT Bulk Data entries are used to define free body supports. In the case of problems using axisymmetric harmonic elements, the **SUPAX** entry is used, and for substructuring, the **SUPORTS** is used. In all cases, only a single set can be specified, and if such entries appear in the Bulk Data packet, they are automatically used in the solution. Free body supports must always be defined in the global coordinate system.

6.11 DYNAMIC RESPONSE MATRICES

When performing dynamic response analyses, additional computations may be required to formulate the final equations of motion. These include considerations for:

- ❒ Extra Points
- ❒ Transfer Functions
- ❒ Direct Matrix Input
- ❒ Modal coordinate transformation
- ❒ Residual Flexibility Vectors

Modal coordinate transformations are available in three Rigid Formats. The capability is provided to transform the structural portion of a model from the physical, *a-set* degrees of freedom to a model expressed in terms of generalized coordinates. These coordinates may arise from a Normal Modes analysis and may be augmented by residual flexibility vectors. This feature is available for complex eigenvalue extraction, frequency response and transient response.

UAI***/NASTRAN*** provides a special capability to couple non-structural phenomena to the structural model. This capability requires specification of EXTRA points to describe the unknowns of the non-structural quantities. The coefficients of any EXTRA point equations are defined using transfer functions and direct matrix input. All EXTRA point equations are excluded from the calculation of the normal modes of the structure, and are later coupled to the new modal coordinates prior to solution of dynamic response.

Details of how to use these features and transformation to modal coordinates are documented in detail in Chapters 12, 13 and 14 of this guide.

6.12 REFERENCES

[Craig68]: Craig, R.R., Jr. and Bampton, M.C.C., "Coupling of Substructures for Dynamic Analysis," *AIAA Journal*, Vol. 6, No. 7, July 1968, pp. 1313-1319.

[Herting79]: Herting, D.N., "A General-Purpose, Multi-Stage Component Mode Synthesis Method," AIAA/ASME/ASCE/AHS 20th Structures, Structural Dynamics and Materials Conference, St. Louis, MO, 1979.

[MacNeal71]: MacNeal, R.H., "A Hybrid Method of Component Mode Synthesis," *Computers and Structures*, Vol. 1, No. 4, Dec. 1971, pp. 581-600.

[Rubin75]: Rubin, S., "Improved Component-Mode Representation for Structural Dynamic Analysis," *AIAA Journal*, Vol. 13, No. 8, August 1975, pp. 995-1006.

This page is intentionally blank.

Chapter 7

STRUCTURAL LOADS

UAI/*NASTRAN* provides the capability for applying many different external structural loads to the finite element model. These loads are defined in a simple and consistent manner whether solving problems in static structural analysis or dynamics problems in the time and frequency domain. Specialized load definitions are used in Substructuring Analysis, Heat Transfer Analysis, Axisymmetric Harmonic Analysis, and Cyclic Symmetry Analysis. Although summarized in this Chapter, you will find details of their use under those topics.

7.1 SELECTING STATIC LOADS

In general, you select static loads with specific Case Control commands. These, in turn, point to Bulk Data entries which define the loads themselves. The available loads may be classified as mechanical loads, initial strain, and thermal loads. These are summarized in Table 7-1.

Table 7-1. SELECTION OF STATIC LOADS

TO SPECIFY	CASE CONTROL COMMAND USED:	BULK DATA ENTRIES USED:	
MECHANICAL LOADS	LOAD	LOAD LOADC[4]	FORCE, FORCE1, FORCE2 FORCEAX[1] MOMENT, MOMENT1, MOMENT2 MOMAX[1] PLOAD, PLOAD2, PLOAD4 PLOAD1[2] PRESAX[1] SPCD[3], SPCSD[3,4] DAREA[5]
	SPC		SPCD[3]
INERTIAL LOADS	LOAD	LOAD LOADC[4]	ACCEL, ACCEL1 GRAV RFORCE, RFORCE1
INITIAL STRAINS	DEFORM		DEFORM[2]
THERMAL LOADS	TEMP(LOAD)		TEMP, TEMPD TEMPAX[1] TEMPP1, TEMPP2, TEMPP3 TEMPRB[2]

Notes:
1. For Axisymmetric Harmonic Analysis Only
2. For One-dimensional Elements Only
3. For Enforced Deformations
4. For Substructuring Only
5. For Residual Flexibility Feature Only

7.2 MECHANICAL LOADS

This section describes the various static mechanical loads that may be applied to the structure. These include concentrated loads, pressure loads, inertial and enforced displacement loads.

7.2.1 Concentrated Loads

Each GRID or SCALAR point in a model may be subjected to a concentrated, or point, load. **UAI/*NASTRAN*** provides great flexibility in defining such loads. There are four methods for describing applied forces and moments that define the magnitude and direction of the load.

The first method defines the concentrated load at a GRID point in a direction given explicitly by the components of the vector. This vector may be specified in any coordinate system. The second method defines the direction as the vector defined by two GRID points. The third method defines the direction as the cross product of two vectors defined by four GRID points. These methods, defined by **FORCEi** and **MOMENTi** Bulk Data entries, are pictured in Figure 7-1. The fourth method, the **DAREA** Bulk Data entry, defines the load magnitude in a particular component direction at a GRID or SCALAR point. The component itself defines the direction of the load. **FORCEAX** and **MOMAX** Bulk Data entries define generalized harmonic loads for axi-symmetric harmonic analysis.

7.2.2 Distributed Loads on Bars and Beams

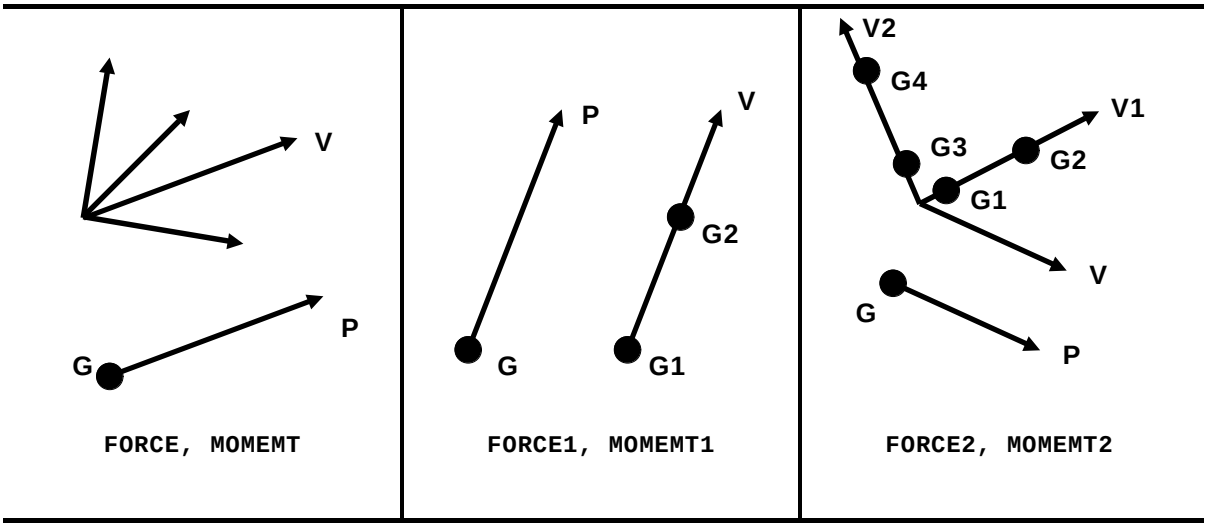
Distributed and point loads acting in any direction and at any location along BAR and BEAM elements are specified with **PLOAD1** Bulk Data. For the BEAM element, such load data are also used to obtain accurate element stress and force distributions along the length of the BEAM. This correction is not made for the BAR element.

The general **PLOAD1** Bulk Data entry is:

PLOAD1	LID	EID	TYPE	SCALE	X1	P1	X2	P2	

where **LID** is the load set identification number and **EID** is a BAR or BEAM element identification number. **TYPE** is used to define the direction of the load, whether the load is a force or

Figure 7-1. TYPES OF CONCENTRATED LOADS



7

moment, and whether the direction is given in the basic coordinate system of the model or in the element coordinate system.

The **TYPE**s are specified by strings of two or three characters which have the general form:

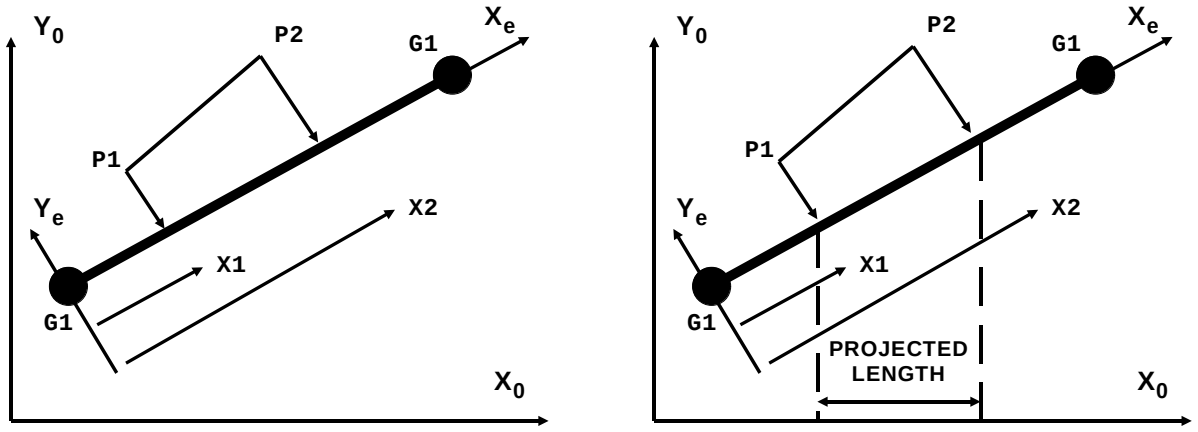
$$\left\{ \begin{matrix} \mathbf{F} \\ \mathbf{M} \end{matrix} \right\} \left\{ \begin{matrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{matrix} \right\} [\ \mathbf{E}\ ]$$

A first character of **F** indicates a force while an **M** indicates a moment. The next character must specify a coordinate direction selected from **X**, **Y** or **Z**. If the optional third character **E** is not present, then the direction is taken to be the basic coordinate system. If **E** is present, then the direction refers to the element coordinate system. For example, **FXE** on a BAR means a force acting in the direction **GID1** to **GID2** along the bar. Similarly, **MZE** indicates applied moments to the element in the direction of the z-axis of the element coordinate system.

Concentrated loads are specified by defining the location of the load as a distance, **X1**, from End 1 towards End 2 of the element. **X2** must be blank, or equal to **X1**, for concentrated loads. **P1** is the value of the load. **X1** may be specified as the ***actual*** length, by using a **SCALE** of **LE**, or as a ***fraction*** of the total length, by using a **SCALE** of **FR**. When **SCALE** is **FR**, **X1** must be greater than 0.0 and less than 1.0. Distributed loads are defined by using **X1**, **X2**, **P1** and **P2** as illustrated in Figure 7-2.

For this case, the **SCALE** declaration of *fractional*, **FR**, is extended to *projected*, **FRPR**, so that the total load will be the distributed load integrated over the projected distance as shown. This application is useful when elements are at an angle with respect to a directional load such as gravity.

Figure 7-2. DISTRIBUTED LOADS ON BARS AND BEAMS



7.2.3 Pressure Loads on Surfaces

In addition to applying concentrated point loads at specific points in the model, **UAI/*NASTRAN*** has several facilities to define pressure loads over areas of the model or individual elements within the model.

7.2.3.1 General Surface Pressure

You may apply a uniform pressure loading to an arbitrary quadrilateral or triangular surface area of your model with **PLOAD** Bulk Data. The region is defined by either three or four GRID points lying on the surface. The positive direction is determined by treating the area as an *element* and then using the usual right hand rule.

PLOAD	**LID**	**P**	**GID1**	**GID2**	**GID3**	**GID4**			

The pressure is distributed to each of the referenced GRID points defining the area so that static equilibrium is maintained. For axisymmetric harmonic models, the Bulk Data **PRESAX** is used for this purpose.

7.2.3.2 Pressure on Plate Elements

You may define pressure loads which are applied to individual plate elements as constant pressure acting normal to the surface of the element, or as a varying pressure which acts in a specified direction. **PLOAD2** Bulk Data entries are used to specify a constant pressure acting on a set of elements. The format of this entry is:

PLOAD2	**SID**	**P**	**EID1**	**EID2**	**EID3**	**EID4**	**EID5**	**EID6**	

The direction of the pressure is defined for each of the elements by the normal to the element mid-surface at each point with the right hand rule. Uniform or linearly varying pressures are defined over plates, and solid element faces, using **PLOAD4** Bulk Data. The format of the **PLOAD4** Bulk Data entry for the plate elements is:

PLOAD4	**LID**	**EID** **EIDF**	**P1**	**P2**	**P3**	**P4**	 **"THRU"**	 **EIDL**	**-cont-**
-cont-	**CID**	**V1**	**V2**	**V3**					

The pressure may act normal to the surface, as with **PLOAD2**, or may act in any other direction defined by a vector specified on the **PLOAD4** entry.

7.2.3.3 Pressure on Solid Element Faces

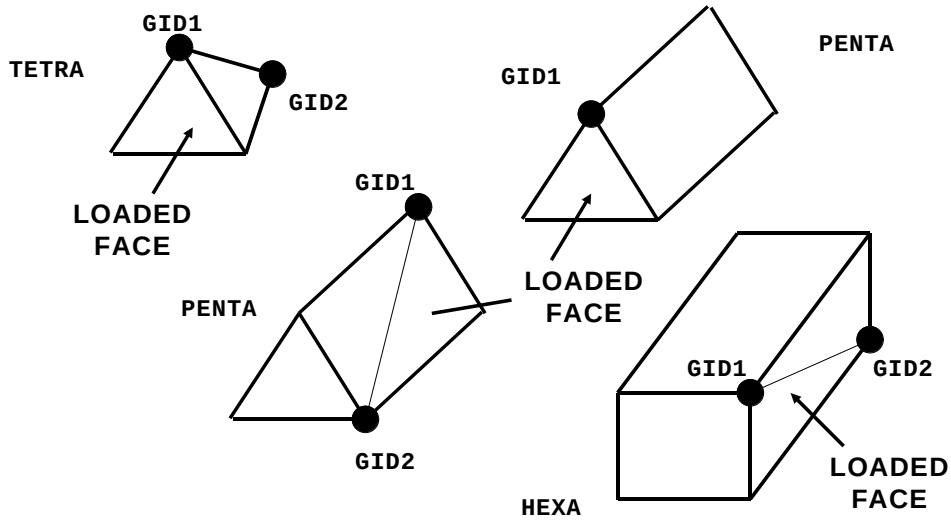
For ease of use, **PLOAD4** data are also used to specify pressure on the face of PENTA, TETRA and HEXA solid elements. The required data are slightly different from the plate element case because the face of the element upon which the pressure is acting must be identified. The minimal **PLOAD4** data specifications for solid elements are:

PLOAD4	**LID**	**EID** **EIDF**	**P1**	**P2**	**P3**	**P4**	**GID1** **"THRU"**	**GID2** **EIDL**	**-cont-**
-cont-	**CID**	**V1**	**V2**	**V3**					

GID1 and **GID2**, are the GRID point identification numbers of two points located diagonally on a quadrilateral face as shown in Figure 7-3.

For a triangular surface of the pentahedral element, only **GID1** is required since no ambiguity will result when identifying the triangular surface. On the other hand, to identify the face of a tetrahedron element, you must enter the identification number of a GRID point on the loaded face, and the identification number of the GRID point opposite to that face in **GID1** and **GID2** fields, respectively. The positive pressure on a solid element face, by default, is normal to the surface and directed inward. This may be modified by using the direction vector option.

Figure 7-3. DEFINING PRESSURE LOADS ON SOLID ELEMENTS



7.3 INERTIAL LOADS

There are three inertial loads available. The first is a uniform acceleration field, most usually representing gravity, which is specified with the **GRAV** Bulk Data entry:

GRAV	LID	CID	G	V1	V2	V3			

The acceleration, **G**, is applied to each GRID point in the direction of a vector, ***V*** whose components **V1**, **V2**, and **V3** are defined in any coordinate system **CID**. If the coordinate system is not specified, *V* is assumed to be defined in the basic coordinate system.

A more general acceleration field may also be defined using the **ACCEL** and **ACCEL1** Bulk Data entries. The **ACCEL** entry:

ACCEL	LID	CID	V1	V2	V3	DIR			-cont-
-cont-	LOC1	VAL1	LOC2	VAL2	*CONTINUES IN GROUPS OF TWO*				-cont-

is used to define an arbitrary variation of acceleration based on locations within your model. In addition to the vector ***V***, you also specify a direction, **DIR**, in which the acceleration is varying. The continuation entries then define at least two locations in this direction, **LOCi**, and the acceleration at each location. The actual acceleration is applied to all GRID points and interpolated from the table entered. For values outside the range of the table, the end-point values are used. The second form, **ACCEL1**, defines the same loading, but it selects the specific GRID points which are loaded:

ACCEL1	LID	CID	G	V1	V2	V3			-cont-
-cont-	*GRID ID LIST*								-cont-

Typical applications of the variable acceleration load include seismic loads on a highrise building, where the lateral accelerations increase with height, or the modeling of an automobile driving over a bump in the road which may be modeled as an acceleration on the front axle. The resultant static load for each of the above cases is obtained by premultiplying the assembled acceleration vector by the global mass matrix:

$$\boldsymbol{P} = \boldsymbol{M}_{gg}\,\boldsymbol{a}$$

The third inertial load, defined with **RFORCE** and **RFORCE1** Bulk Data entries, is is used to generate centrifugal forces. The **RFORCE** entry allows you to specify a direction and magnitude for the rotational velocity, ω, and acceleration, α. By definition, they both act about the same axis defined by the vector ***V*** passing through the point **GID**. The format of this entry is:

RFORCE	LID	GID	CID	RV	V1	V2	V3	METHOD	-cont-
-cont-	RA								

The **RFORCE1** entry allows you to specify a rotational velocity and acceleration which act about different axes. This is done with:

RFORCE1	LID	GIDV	CID	RV	V1	V2	V3	METHOD	-cont-
-cont-		GIDA	CID	RA	A1	A2	A3		

In both cases, the components of a spin axis, **V1**, **V2**, and **V3**, or **A1**, **A2**, and **A3**, the magnitude of the rotational velocity, **RV**, or the rotational acceleration, **RA**, and the point through which the axes act, or all three, are specified. As in the case of the uniform acceleration load, the

vectors may be defined in any coordinate system **CID**, and if the coordinate system is not specified, it is assumed to be in the Basic coordinate system. The centrifugal forces are computed for each GRID point using:

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \\ M_x \\ M_y \\ M_z \end{Bmatrix} = \begin{bmatrix} \boldsymbol{M}_t & \boldsymbol{M}_{tr} \\ \boldsymbol{M}_{rt} & \boldsymbol{M}_r \end{bmatrix} \begin{Bmatrix} \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}) \\ \boldsymbol{\omega} \times \boldsymbol{\omega} \end{Bmatrix} + \begin{bmatrix} \boldsymbol{M}_t & \boldsymbol{M}_{tr} \\ \boldsymbol{M}_{rt} & \boldsymbol{M}_r \end{bmatrix} \begin{Bmatrix} (\boldsymbol{\alpha} \times \boldsymbol{r}) \\ \boldsymbol{\alpha} \end{Bmatrix}$$

where $\boldsymbol{r}$ is the vector perpendicular to $\boldsymbol{V}$ connecting the GRID point to the rotation axis, $\boldsymbol{\omega} = \mathbf{RV} \cdot \boldsymbol{V}$, $\boldsymbol{\alpha} = \mathbf{RA} \cdot \boldsymbol{A}$, and the subscripts *t* and *r* represent translational and rotational inertia components, respectively.

7.4 ENFORCED DISPLACEMENTS

For static analysis, an enforced displacement may be defined using either of two methods shown below:

METHOD	CASE CONTROL COMMANDS	BULK DATA ENTRIES
1	**SPC** = *sid*	**SPC,sid,gid,dof,x**
2	**LOAD** = *lid* **SPC** = *sid*	**LOAD,lid,...** **SPCD,lid,gid,dof,x** **SPC,sid,gid,dof,0.0**

In Method 1, the displacements are specified using constraint data. This is the most convenient way to specify the data. Method 2 is economical if there are multiple subcases and the enforced displacements change from subcase to subcase. This method is more complex because it requires a static load entry with the same set identification as the enforced displacement *load* in the Bulk Data packet. This requirement may be satisfied with any **LOAD**, **MOMENT** or **FORCE** entry. Note that you need not specify an actual load on these entries. The components of motion referenced with **SPCD** entries must also be referenced on **SPC** or **SPC1** Bulk Data entries.

SPCDS Bulk Data entries are used for enforced displacements in substructuring analyses. These are used in a PHASE 2 execution with the **SOLVE** command. Case Control commands are as illustrated above.

7.5 INITIAL STRAINS

UAI/*NASTRAN* makes limited provision for applying pre-strain conditions within your model for the one-dimensional elements ROD, CONROD, BAR, and BEAM. To request the use of initial strain data, the command:

```
DEFORM = sid
```

must appear in your Case Control packet. The deformation set identification number, *sid*, must be distinct from that used by other kinds of loads. The deformation loads are then combined with the other mechanical and thermal loads that have been requested.

7.6 THERMAL LOADS

Thermal loads may be applied to your model either alone or in combination with mechanical loadings. As with the other external loads, temperature data may be specified either at individual GRID points or for each element. For the plate and shell elements, a temperature variation through the thickness may also be modeled.

To select thermal loads in your analysis, you must use the Case Control command:

```
TEMP(LOAD) = sid
```

The load set identification number, *sid*, must be distinct from any used for other types of loads.

7.6.1 Temperatures at GRID Points

You may define a temperature field over your model by defining the temperature at each GRID point. This is done using **TEMP** Bulk Data entries:

TEMP	SID	GID1	T1	GID2	T2	GID3	T3		

Default temperature for GRID points not otherwise defined can be given by using the **TEMPD** Bulk Data entry:

TEMPD	SID1	T1	SID2	T2	SID3	T3	SID4	T4	

When all temperatures in your model are defined at GRID points, the element temperatures are determined by simple averaging. If temperature dependent material properties are defined, then this average temperature is used for property determination. Also, note that if an element temperature is defined it will be used rather than the GRID point temperatures. When using the axisymmetric harmonic analysis capability, the **TEMPAX** Bulk Data is used. This entry defines a point by its ring identification number and an azimuthal angle.

7.6.2 Temperatures for One-dimensional Elements

A temperature field may be defined for each of the one-dimensional elements, BAR, BEAM, PIPE, ROD, CONROD, and TUBE. The **TEMPRB** Bulk Data entry is used to define a linear temperature variation along the length of the element by specifying the temperature at each end. When using temperature dependent material properties, the material properties are evaluated at the average temperature of the element. In addition, temperature gradients which induce bending may be defined for BAR and BEAM elements.

7.6.3 Temperature Gradients through Plate Elements

The plate and shell elements may have variation in temperature through their thickness. This variation is defined by **TEMPP1**, **TEMPP2**, and **TEMPP3** Bulk Data entries. All of these are described in detail in the **UAI/*NASTRAN Reference Manual*. The TEMPP1** Bulk Data entry defines a linear variation through the thickness by specifying the neutral surface temperature and the temperature gradient through the thickness. You may define temperature induced moments directly by using the **TEMPP2** Bulk Data entry. To allow for the geometry of built-up plates, you may define actual temperatures which are used at the specified fiber distances in element stress recovery. These temperatures may be different than the one predicted by the variation of temperature specified on **TEMPP1** and **TEMPP2** Bulk Data entries.

Finally, the **TEMPP3** Bulk Data entry allows the definition of nonlinear temperature variation through the thickness of the element by entering a table of temperatures as a function of the distance through the thickness from an arbitrary reference point. A maximum of ten tabular entries may be defined, and temperature is calculated by linear interpolation within the table. If the material is temperature dependent, the average temperature is used for property calculations.

7.7 COMBINING LOADING CONDITIONS

You may use the Bulk Data entry **LOAD** to combine, or superimpose, two or more loads.

LOAD	LID	S	S1	LID1	S2	LID2	S3	LID3	-cont-
-cont-	S4	LID4	*CONTINUES IN GROUPS OF 2*						-cont-

It defines a linear combination of other loads:

$$\boldsymbol{P}_{lid} = S \sum_{i} S_i \boldsymbol{P}_{lid_i}$$

If a **GRAV** or **RFORCE** loads are being combined with other loads, the **LOAD** entry must be used. The **GRAV** and **RFORCE** load identifications must be unique. When performing static analysis, loads may also be combined using the **SUBCOM** feature, as described in Chapter 10.

When performing substructuring analyses, the Bulk Data entry **LOADC** is needed because each load may be selected from a different basic substructure:

LOADC	SID	S	SNAME1	LID1	S1	SNAME2	LID2	S2	-cont-
-cont-			SNAME3	LID3	S3	*CONTINUES IN GROUPS OF 3*			-cont-

7.8 NONLINEAR STATIC LOADS

The application of static loads in a material or geometric nonlinear analysis requires some knowledge of the expected behavior of your model. In order to achieve convergence to a correct solution, the loads are typically applied in steps within a subcase. This is done to better guide the solution along the nonlinear behavior curve.

The load in each step represents the total load applied to the model up to that point. In each step, the differential load with respect to the previous step is applied in increments. The increment sizes are best determined by the intelligent algorithm within the solution process. However, by using the parameters on the **NLSOLVE** Bulk Data entry, you may also intervene manually in order to improve convergence.

You may apply any type of mechanical loads to your nonlinear model. These include **FORCEi**, **MOMENTi**, **PLOADi**, **GRAV**, **ACCELi** and **RFORCE**. In addition, these loads may be applied as follower forces when performing Geometric Nonlinear analysis. This is described in detail in Chapter 17 of this manual.

7.9 DYNAMIC LOADS

The extension of static loads to dynamic response analysis in both the time and frequency domains is described in this Section.

7.9.1 Time Domain Loads

The **DLOAD** Bulk Data entry defines linear combinations of time dependent loads that are defined on **TLOAD1** and **TLOAD2** entries. A **DLOAD** Case Control command may reference a **DLOAD** Bulk Data entry or the **TLOADi** entries directly. **TLOAD1** defines a time dependent load of the form:

$$P(t) = AF(t-\tau)$$

where A is defined by a **DAREA** entry, τ is defined on a **DELAY** entry, and $F(t-\tau)$ is defined using **TABLEDi** data. A, which is both a load factor for $F(t-\tau)$ and a reference to a specific degree of freedom identified by GRID point and component, may also be defined using any static load entries, such as **FORCE** or **PLOAD4** instead of using the **DAREA** entry, just as **DAREA** can be used to define a static load when performing Statics analysis. A constitutes a general spatial load distribution, while F represents the time-dependency.

The **TLOAD2** Bulk Data entry defines a time dependent load of the form:

$$P(t) = \begin{cases} 0 & \text{for } \bar{t} < 0 \text{ or } \bar{t} > T_2 - T_1 \\ A\,\bar{t}^{B} E^{C\,\bar{t}} \cos(2\pi F \bar{t} + \theta) & \text{for } 0 \le \bar{t} \le T_2 - T_1 \end{cases}$$

where $\bar{t} = t - T_1 - \tau$, and θ is the phase angle in degrees. The coefficients on the **DAREA** and **DELAY** entries may be different for each degree of freedom subjected to a load. The loads are applied to the specified components in the global coordinate system and assembled into the load vector.

7.9.2 Frequency Domain Loads

You must select the desired frequency domain load by using a **DLOAD** Case Control command which references either a **DLOAD** or **RLOAD1** and **RLOAD2** Bulk Data entries. A **DLOAD** Bulk Data entry is used to combine sets of **RLOADi** entries.

The **RLOAD1** entry defines a frequency dependent load of the form:

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

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 and τ is defined on a **DELAY** entry. The **RLOAD2** entry defines a frequency dependent load of the form:

$$P(f) = AB(f)\, e^{i(\varphi(f) + \theta - 2\pi f\tau)}$$

where A is defined using **DAREA** entry, $B(f)$ and $\varphi(f)$ are defined using **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.

7.9.3 Random Vibration Loads

To apply loads in a Random analysis, you must use the **RANDOM** Case Control command, which references a **RANDPS** Bulk Data entry. The **RANDPS** Bulk Data 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 **TABRND1** entry. The subscripts *j* and *k* define the subcase numbers of the load definitions. If the applied loads are independent, only the diagonal terms are defined. The **RANDT1** Bulk Data entry is used to specify the time lag constants for use in the computation of the autocorrelation functions.

7.9.4 Enforced Dynamic Motion

UAI/*NASTRAN* provides a robust mechanism to enforce motion on selected degrees of freedom in a dynamic analysis model. These degrees of freedom must appear in the *s-set*.

You may select the enforced motion at each degree of freedom to be of either displacement, velocity, or acceleration. This selection is made in the **DYNEX** field of **RLOADi** and **TLOADi** Bulk Data entries.

If you enter a value of **1** in this field, then the function is representative of a frequency- or time-dependent displacement at the given degrees of freedom. A value of **2** or **3** specifies velocity or acceleration, respectively. In these cases, the **ADEF** field on the **RLOADi** and **TLOADi** entries may only reference **DAREA** or **DAREAS** entries which define the appropriate degree of freedom and scaling factor.

If the **DYNEX** field contains a zero, which is the default case, it indicates that there is no enforced motion. Rather, the function represents a frequency- or time-dependent load. This can be accomplished by referencing **DAREA** and **DAREAS** entries, or, optionally, static load entries in the **ADEF** field.

7.9.5 Dynamic Loads For Substructures

Dynamic loads for substructuring are defined using a combination of static load vectors declared in PHASE 1 substructuring runs and dynamic load data defined in a PHASE 2 solution execution. The PHASE 1 static load data perform the role of the **DAREA** data. These data are expanded in the time or frequency domain during the PHASE 2 execution using the referencing techniques described earlier for normal dynamics analysis.

The major difference in applying dynamic loads to substructures is that you must use **DAREAS**, **DPHASES**, and **DELAYS** Bulk Data entries. This allows you to select the appropriate loads from any substructures.

7.9.6 Nonlinear Dynamic Loads

Several types of nonlinear loads are available with transient response analyses in **UAI/*NASTRAN***. These effects are treated as an additional applied load vector, where the actual loads are functions of the response of degrees of freedom. This additional load vector is then added to the applied load vector. 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., 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 EPOINTs introduced for dynamic analysis.

UAI/*NASTRAN* includes four different nonlinear loads for transient response. The **NOLIN1** Bulk Data entry defines a nonlinear load of the form:

$$P_i(t) = S\,T(u_j(t))$$

where P_i is the displacement-dependent load applied to point *i*, *S* is a scale factor, and $T(u_j(t))$ references a **TABLEDi** Bulk Data entry which provides a tabular function of displacements at point j.

NOLIN1	**NLLID**	**GIDI**	**DOFI**	**S**	**GIDJ**	**DOFJ**	**TID**		
NOLIN2	**NLLID**	**GIDI**	**DOFI**	**S**	**GIDJ**	**DOFJ**	**GIDK**	**DOFK**	
NOLIN3	**NLLID**	**GIDI**	**DOFI**	**S**	**GIDJ**	**DOFJ**	**A**		
NOLIN4	**NLLID**	**GIDI**	**DOFI**	**S**	**GIDJ**	**DOFJ**	**A**		

The **NOLIN2** entry defines a nonlinear load of the form:

$$P_i(t) = S u_j(t)\,u_k(t)$$

where u_j and u_k are any permissible pair of displacement components which may be the same. The **NOLIN3** entry defines a nonlinear load of the form:

$$P_i(t) = \begin{cases} S\,u_j(t)^A & \text{for } u_j(t) > 0 \\ 0 & \text{for } u_j(t) \le 0 \end{cases}$$

where *A* is an exponent. The **NOLIN4** entry defines a nonlinear load of the form:

$$P_i(t) = \begin{cases} -S\,(-u_j(t))^A) & \text{for } u_j(t) < 0 \\ 0 & \text{for } u_j(t) \ge 0 \end{cases}$$

Nonlinear loads applied to a massless system without damping will not converge to a steady state solution. The **DIAG 10** Executive Control command will cause the nonlinear term P_{n+1} to be replaced by:

$$\frac{P_{n+1} + P_n + P_{n-1}}{3}$$

where P_{n+1}, P_n and P_{n-1} are the values of the nonlinear loads at the time steps $n+1$, n, and $n-1$, respectively.

7

This page is intentionally blank.

Chapter 8

MATERIAL PROPERTIES

UAI/*NASTRAN* provides for the definition of various material properties for the modeling of structures. These material models range from one-dimensional isotropic to full three-dimensional anisotropic behavior. Composite material models for plate and shell elements are handled in either of two ways: using the various combinations of membrane, bending, transverse shear and coupling properties available to model complex unsymmetric composites such as honeycombs; or using special property definition data for multilayer composites. The use of these techniques is described in Chapter 5.

You may also use two special materials for heat transfer analysis, these are summarized in this section and fully described in Chapter 24. Similarly, if you are performing nonlinear static structural analysis, you may specify nonlinear stress-strain relationships for isotropic materials. Finally, if you are performing thermal-structural analyses, most of the materials may have temperature dependent properties, and, for frequency dependent materials, a special viscoelastic property is available.

8.1 FINITE ELEMENTS AND MATERIAL PROPERTIES

Each finite element in the structural model has specific properties. These element properties are normally referenced from the connection entry which in turn, references a material property. Some connection entries do not require a property entry in which case they reference material data directly. You assign an identification number to all material property Bulk Data entries. Any number of finite elements may select a material by referencing its identification number.

The material property definitions include the modulii of elasticity, shear module, and Poisson's ratios. They also include the density, structural damping, thermal expansion coefficients, and maximum allowable stresses. The density is only used when the analysis discipline requires mass data. This includes statics analysis when inertial loading is applied, normal modes analysis, and all forms of dynamic response analyses. Structural damping is only applicable to dynamic response problems. See Chapter 9, Dynamics Modeling, for more detailed information. Thermal expansion coefficients are used for models subjected to thermal loadings and for Heat Transfer analyses. Finally, the stress/strain limits, or allowables, are used for the computation of layer failure when composite materials are used and for margin-of-safety calculations for line elements. Details of describing the use of these features are found in Chapter 5, Finite Element Library. When performing Design Sensitivity analyses, you must use the same stress or strain allowables on the material property definition and stress or strain constraints.

8.1.1 Importance of Material Models

You should remember that finite element analysis is an idealization and approximation of the actual problem. The elements *per se* describe the geometry of the structure while mathematical models are used to describe the material properties. The success of a finite element analysis depends very much on the accuracy with which you can describe the geometry and material properties of your model.

The mathematical models for describing the deformational properties of the materials are called ***Material Models***, or ***Constitutive Models***, or simply ***Stress-Strain Relations***. The simplest material model is ***Hooke's Law***, which relates stress with strain by a constant elastic modulus. While almost all types of structural and fluid geometries can be described using different types of elements, the same is not true for material models. There are many materials for which the mathematical description is a gross approximation at best. Therefore, the accuracy of finite element analysis depends to a large extent on the accuracy of the material model used.

The types of materials used in different applications range from heterogeneous materials such as concrete to homogeneous material such as steel. Their properties for many applications can be approximated as linear, while for some applications nonlinear material models need to be used. The material properties may also be direction dependent. An ***isotropic*** material is the one which has identical properties in all directions. There is a distinct direction of strength in ***orthotropic*** materials. The material properties may vary differently in each coordinate direction for ***anisotropic*** materials.

UAI/*NASTRAN* provides a comprehensive set of material models to suit your modeling needs for practical problems. The available material models include:

- ❒ Linear elastic isotropic models.
- ❒ Linear elastic orthotropic models.
- ❒ Linear elastic anisotropic models.
- ❒ Nonlinear material models.
- ❒ Isotropic and anisotropic thermal material models.
- ❒ Linear viscoelastic materials

Table 8-1 shows a summary of linear material models available for different types of elements. The thermoelastic materials complement regular elastic materials by allowing most of their properties to be temperature-dependent.

For nonlinear materials, **UAI/*NASTRAN*** provides a variety of isotropic material models through **MAT1NL**, **MAT1NL1**, and **TABLENL** Bulk Data entries. They are explained in Section 8.4. For heat transfer analysiss, both isotropic and anisotropic material models are provided using **MAT4** and **MAT5** Bulk Data entries. These are explained in Section 8.3.

Table 8-1. LINEAR MATERIAL PROPERTY SUMMARY

ELEMENT	ELASTIC MATERIALS			THERMOELASTIC MATERIALS		
	ISOTROPIC	ANISOTROPIC	ORTHOTROPIC	ISOTROPIC	ANISOTROPIC	ORTHOTROPIC
BAR	**MAT1**			**MATT1**		
BEAM	**MAT1**			**MATT1**		
CONEAX	**MAT1**			**MATT1**		
CONROD	**MAT1**			**MATT1**		
HEXA	**MAT1**	**MAT9**		**MATT1**	**MATT9**	
PENTA	**MAT1**	**MAT9**		**MATT1**	**MATT9**	
PILE	**MAT1**			**MATT1**		
PIPE	**MAT1**			**MATT1**		
QUAD4	**MAT1**	**MAT2**	**MAT8**	**MATT1**	**MATT2**	
QUAD8	**MAT1**	**MAT2**	**MAT8**	**MATT1**	**MATT2**	
ROD	**MAT1**			**MATT1**		
SHEAR	**MAT1**			**MATT1**		
TETRA	**MAT1**	**MAT9**		**MATT1**	**MATT9**	
TORDRG	**MAT1**		**MAT3**	**MATT1**		**MATT3**
TRAPAX	**MAT1**		**MAT3**	**MATT1**		**MATT3**
TRAPRG	**MAT1**		**MAT3**	**MATT1**		**MATT3**
TRIA3	**MAT1**	**MAT2**	**MAT8**	**MATT1**	**MATT2**	
TRIA6	**MAT1**	**MAT2**	**MAT8**	**MATT1**	**MATT2**	
TRIAAX	**MAT1**		**MAT3**	**MATT1**		**MATT3**
TRIARG	**MAT1**		**MAT3**	**MATT1**		**MATT3**
TUBE	**MAT1**			**MATT1**		
TWIST	**MAT1**			**MATT1**		

8.2 NOMENCLATURE

The descriptions of the material property relationships in the following sections use a number of common symbols shown below:

SYMBOL	DESCRIPTION
E	Young's Modulus
ν	Poisson's ratio
G	Shear modulus
ρ	Mass density
σ_x , σ_y , σ_z , σ_r , σ_θ	Normal stresses in the x, y, x, r, and θ directions
τ_{xy} , τ_{yz} , τ_{zx} , τ_{zr}	Shear stresses
ε_x , ε_y , ε_z , ε_r , ε_θ	Normal stresses in the x, y, x, r, and θ directions
ε_{zr} , γ_{xy} , γ_{yz} , γ_{zx}	Shear strains
α_i	Coefficients of thermal expansion
T_0	Reference Temperature
g_e	Structural damping coefficient

8.3 LINEAR MATERIALS — HOOKE'S LAW

Linear elastic materials are governed by Hooke's Law. The form of this law used in **UAI/*NASTRAN*** is:

$$\sigma_m = \boldsymbol{D}\,(\varepsilon_m - \varepsilon_T) \tag{8-1}$$

which defines a linear relationship between mechanical stress, σ_m, and total strain, ε_m. The mechanical stresses are corrected when thermal loading is present to account for the effects of thermal strain, ε_T. When thermal loading is applied to an isotropic material, it generates a thermal strain in all directions which is proportional to the change in temperature:

$$\varepsilon = \alpha\,\Delta T \tag{8-2}$$

Such changes in strains do not cause any shear strains isotropic materials. The ***D*** matrix of (8-1) is often called the ***Elasticity Matrix***. The terms of this matrix depend upon the material model that is selected. **UAI/*NASTRAN*** also allows you to compute the elasticity matrix for materials which are temperature dependent. The details of the models available in **UAI/*NASTRAN*** are presented in the following sections.

8.3.1 Isotropic Material Properties

The simplest materials are homogeneous isotropic materials which are those which have the same properties in all directions. There are three stress-strain relationships. For one-dimensional stress-strain, the relationship is given by:

$$\sigma = E\left(\varepsilon - (T - T_0)\,\alpha\right) \tag{8-3}$$

The three-dimensional stress-strain relationship is:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{Bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} \end{bmatrix} \left[\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} - (T - T_0) \begin{Bmatrix} \alpha \\ \alpha \\ \alpha \\ 0 \\ 0 \\ 0 \end{Bmatrix} \right] \tag{8-4}$$

This material model is used for three-dimensional element formulations such as those found in the HEXA, PENTA and TETRA elements. For two-dimensional plane stress problems, the isotropic material relationship is then given by:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \frac{E}{(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{(1-\nu)}{2} \end{bmatrix} \left[\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} - (T - T_0) \begin{Bmatrix} \alpha \\ \alpha \\ 0 \end{Bmatrix} \right] \tag{8-5}$$

similarly, for plane strain problems, it is:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{(1-2\nu)}{2} \end{bmatrix} \left[\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} - (T - T_0) \begin{Bmatrix} \alpha \\ \alpha \\ 0 \end{Bmatrix} \right] \tag{8-6}$$

These isotropic materials are defined in **UAI/*NASTRAN*** by using **MAT1** Bulk Data entries. The format of this Bulk Data entry is:

MAT1	MID	E	G	ν	ρ	α	T0	GE	-cont-
-cont-	ST	SC	SS						

Note that the data entry has fields for **E**, ν and **G**. However, for linear isotropic materials, the following relationship between the three elastic constants holds:

$$G = \frac{E}{2(1+\nu)} \tag{8-7}$$

Therefore, you may enter any two of the elastic constants and **UAI/*NASTRAN*** will compute the third. The additional data fields specified on the **MAT1** Bulk Data entry are **ST**, **SC**, **SS**, the stress/strain limits. These are used for the computation of failure indices if the material is used as a layer in a composite element property, and for *margin-of-safety* calculations for one-dimensional elements.

8.3.2 Two-Dimensional Anisotropic Material Properties

The **UAI/*NASTRAN*** plate and shell elements may reference two-dimensional anisotropic material properties. Anisotropic properties are those which may define arbitrary symmetric stress-strain relationships defined by:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} G_{11} & G_{12} & G_{13} \\ G_{12} & G_{22} & G_{23} \\ G_{13} & G_{23} & G_{33} \end{bmatrix} \left[\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} - (T - T_0) \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_{12} \end{Bmatrix} \right] \tag{8-8}$$

You may model transverse shear effects for the plate elements by specifying an **MID3** field on the **PSHELL** Bulk Data entry. In this case, the anisotropic relationship between the transverse shear stresses and strains is taken to be:

$$\begin{Bmatrix} \tau_{xz} \\ \tau_{yz} \end{Bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{12} & G_{22} \end{bmatrix} \begin{Bmatrix} \varepsilon_{xz} \\ \varepsilon_{yz} \end{Bmatrix} \tag{8-9}$$

G_{33} in this case must be zero. Note that it is only necessary for you to enter one half of this symmetric ***D*** matrix. As in the case of **MAT1**, descriptions of the structural damping as well as stress and strain limits are found in other chapters of this manual. To define anisotropic properties you ***must*** enter explicitly the material property matrix using **MAT2** Bulk Data entries which have the format:

MAT2	MID	G11	G12	G13	G22	G23	G33	ρ	-cont-
-cont-	α1	α2	α12	T0	GE	ST	SC	SS	

8.3.3 Two-Dimensional Orthotropic Material Properties

Plate and shell elements may also reference two-dimensional orthotropic material properties by use of the **MAT8** Bulk Data. Orthotropic materials are those whose properties vary differently in each of the material coordinate directions. The stress-strain relationship defined by these data is:

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} \dfrac{E_1}{1-\nu_{12}\nu_{21}} & \dfrac{\nu_{12}E_2}{1-\nu_{12}\nu_{21}} & 0 \\ \dfrac{\nu_{12}E_2}{1-\nu_{12}\nu_{21}} & \dfrac{E_2}{1-\nu_{12}\nu_{21}} & 0 \\ 0 & 0 & \dfrac{1}{G_{12}} \end{bmatrix} \left[\begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{Bmatrix} + (T - T_0) \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ 0 \end{Bmatrix} \right] \tag{8-10}$$

where $\nu_{21} = \nu_{12}\dfrac{E_2}{E_1}$. As with anisotropic materials, the transverse stress-strain relationship may be represented. In this case, it is defined by:

$$\begin{Bmatrix} \tau_{xz} \\ \tau_{yz} \end{Bmatrix} = \begin{bmatrix} G_{1z} & 0 \\ 0 & G_{2z} \end{bmatrix} \begin{Bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} \tag{8-11}$$

The input data for this case is the elastic modulii and Poisson's ratio in both directions. The format of the **MAT8** Bulk Data entry is:

MAT8	MID	E1	E2	ν12	G12	G1Z	G2Z	ρ	-cont-
-cont-	α1	α2	T0	XT	XC	YT	YC	S	-cont-
-cont-	GE	F12							

As in the case of the previous materials, descriptions of the structural damping and stress/strain limits are found in other chapters of this manual. Note, however, that the

8

stress/strain limits are now provided in each coordinate direction. For composite materials with Tsai-Wu failure theory, a special interaction term, **F12**, has been added.

8.3.4 Three-Dimensional Anisotropic Material Properties

The solid elements HEXA, PENTA, and TETRA support full three-dimensional anisotropic material properties. The stress-strain relationship for these materials is given by:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{Bmatrix} = \begin{bmatrix} G_{11} & & & & & \\ G_{12} & G_{22} & & \textbf{SYM} & & \\ G_{13} & G_{23} & G_{33} & & & \\ G_{14} & G_{24} & G_{34} & G_{44} & & \\ G_{15} & G_{25} & G_{35} & G_{45} & G_{55} & \\ G_{16} & G_{26} & G_{36} & G_{46} & G_{56} & G_{66} \end{bmatrix} \left[\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} - (T - T_0) \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \end{Bmatrix} \right] \tag{8-12}$$

The lower triangle of the symmetric six-by-six elasticity matrix is entered explicitly. using the **MAT9** Bulk Data entry:

MAT9	**MID**	**G11**	**G12**	**G13**	**G14**	**G15**	**G16**	**G22**	**-cont-**
-cont-	**G23**	**G24**	**G25**	**G26**	**G33**	**G34**	**G35**	**G36**	**-cont-**
-cont-	**G44**	**G45**	**G46**	**G55**	**G56**	**G66**	ρ	α1	**-cont-**
-cont-	α2	α3	α4	α5	α6	**T0**	**GE**		

Again, descriptions of the structural damping and stress/strain limits are found in other chapters of this manual.

8.3.5 Axisymmetric Orthotropic Material Properties

For axisymmetric elements, an additional material property specification is available. These data, specified with **MAT3** Bulk Data, define full three-dimensional orthotropic properties in terms of the cylindrical coordinate system. The stress-strain relationship for this material is given by:

$$\begin{Bmatrix} \varepsilon_r \\ \varepsilon_\theta \\ \varepsilon_z \\ \gamma_{zr} \end{Bmatrix} = \begin{bmatrix} \frac{1}{E_r} & -\frac{\nu_{r\theta}}{E_\theta} & -\frac{\nu_{zr}}{E_z} & 0 \\ -\frac{\nu_{r\theta}}{E_r} & \frac{1}{E_\theta} & -\frac{\nu_{\theta z}}{E_z} & 0 \\ -\frac{\nu_{zr}}{E_r} & -\frac{\nu_{\theta z}}{E_\theta} & \frac{1}{E_z} & 0 \\ 0 & 0 & 0 & G_{zr} \end{bmatrix} \left[\begin{Bmatrix} \sigma_r \\ \sigma_\theta \\ \sigma_z \\ \tau_{zr} \end{Bmatrix} - (T - T_0) \begin{Bmatrix} \alpha_r \\ \alpha_\theta \\ \alpha_z \\ 0 \end{Bmatrix} \right] \tag{8-13}$$

The format of the **MAT3** Bulk Data entry is:

MAT3	MID	Er	Eθ	Ez	νrθ	νθz	νzr	ρ	-cont-
-cont-			Gzr	αr	αθ	αz	T0	GE	

The descriptions of the structural damping and stress/strain limits are found in other chapters of this manual.

8.3.6 Temperature Dependent Material Properties

UAI/*NASTRAN* provides you the capability to model thermoelastic problems. As indicated in Table 8-1, any or all of the material properties specified by **MAT1**, **MAT2**, **MAT3**, **MAT4**, **MAT5**, or **MAT9** Bulk Data entries may be defined as temperature-dependent. This is done by using Bulk Data entries that are analogous to the **MATi** data. These are called **MATTi** data. The fields of a **MATTi** entry correspond one-for-one to those on the related **MATi** entry.

For example, consider a simple isotropic property defined by the Bulk Data entry:

MAT1	100	3.+7		0.3			100.0		
		⇑		⇑			⇑		
		E		ν			**T0**		

The accompanying **MATT1** entry for this data could be:

MATT1	100	101							
		⇑							
		TID1							

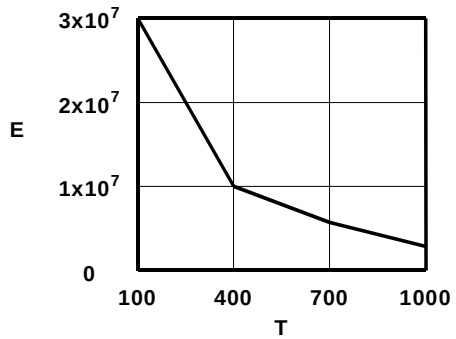
Since the third field of the **MATT1** entry is defined, this means that the third field of the **MAT1** entry, **E**, will be temperature dependent. The identification number **TID1**, refers to one of several tables that may be used to define the temperature relationship. These tables are defined by **TABLEMi** Bulk Data entries. They differ in the manner in which the property value is calculated. **TABLEM1** data provides a simple linear interpolation within the table.

Suppose, for the above example, that the temperature dependency of **E** is defined by the tabular function of temperature as illustrated in Figure 8-1. The **TABLEM1** definition of this dependence, given in the form of a free field Bulk Data entry, is:

TABLEM1	101								+A
+A	100.	3.+7	400.	1.+7	700.	0.6+7	1000.	0.3+7	+A
+A	ENDT								

The material properties are adjusted based on the current temperature value at each evaluation point.

Figure 8-1. TEMPERATURE DEPENDENT ELASTIC MODULUS



8

8.4 HEAT TRANSFER MATERIAL PROPERTIES

The materials described before allow specification of thermal expansion coefficients used in thermal-structural analysis. When using the **UAI/*NASTRAN*** heat transfer analysis capability, fully described in Chapter 24, additional properties must be specified.

8.4.1 Isotropic Materials

When using isotropic materials for Heat Transfer analyses, you must use **MAT4** Bulk Data entries to define the thermal conductivity, **K**, and the thermal capacity per unit volume, **CP**:

MAT4	**MID**	**K**	**CP**						

If a Heat Transfer boundary element, HBDY, references a **MAT4** entry, then the interpretation of **K** and **CP** is different: they represent the convective film coefficient and film capacity per unit area.

8.4.2 Anisotropic Materials

For anisotropic materials, you may define a three-by-three conductivity matrix ***K***, that provides coupled conductivities as shown below:

$$\boldsymbol{K} = \begin{bmatrix} K_{xx} & K_{xy} & K_{xz} \\ K_{xy} & K_{yy} & K_{yz} \\ K_{xz} & K_{yz} & K_{zz} \end{bmatrix} \tag{8-13}$$

This material is defined with the **MAT5** Bulk Data entry:

MAT5	**MID**	**KXX**	**KXY**	**KXZ**	**KYY**	**KYZ**	**KZZ**	**CP**	

As was the case for isotropic properties, **CP** is the thermal capacity per unit volume.

8.5 NONLINEAR MATERIAL PROPERTIES

A variety of nonlinear material models for Material Nonlinear analysis is available. Isotropic materials may be defined using nonlinear stress-strain relationships. The elements that may use nonlinear material properties are the BAR, PILE, ROD, HEXA, PENTA and TETRA. The nonlinear material behavior types modeled with the following Bulk Data entries are elasto-plastic, nonlinear elastic, and combined nonlinear elastic and plastic. You select these properties with the **MAT1NL** Bulk Data entry:

MAT1NL	MID	TNLID	TYPE	YC					

The **TNLID** field references a special **TABLENL** Bulk Data entry which defines the stress-strain relationship using a tabular listing:

TABLENL	TNLID	KU	SCALE	HYSTYP	YIELD	φ	β		-cont-
-cont-	X1	Y1	X2	Y2	*CONTINUES IN GROUPS OF TWO*				
	"SYM"		X1	Y1					

The BAR, ROD, PILE, HEXA, PENTA, and TETRA elements may also use with bilinear isotropic material properties defined with **MAT1NL1** Bulk Data entry. This entry modes elasto-plastic and nonlinear elastic types of material behavior. The format of this entry is:

MAT1NL1	MID	TYPE	YC	YIELD	ET	HYSTYP	φ	β	

The ELASNL and GAP elements may also be used in nonlinear material static analysis, but these elements do not reference stress-strain material properties. Material Nonlinear analysis is described in detail in Chapter 17.

8.6 COMPOSITE MATERIAL PROPERTIES

Laminated composite material properties may be specified for the QUAD4, QUAD8, QUADR, TRIA3, TRIA6, and TRIAR6 elements. This specification is made using either **PCOMP**, **PCOMP1** or **PCOMP2** Bulk Data entries in place of the **PSHELL** property definition. The different forms of **PCOMPi** input allow ease of modeling from a very general composite layup to a symmetric membrane layup, minimizing the amount of your input. Using **PCOMPi** entries, the ply thicknesses, relative orientations and the material identification numbers of each ply are defined. The material properties referenced on these entries may be isotropic, orthotropic, or anisotropic, specified on **MAT1**, **MAT8**, or **MAT2** Bulk Data entries respectively. Because composite materials are specified as a property of the finite elements, the use of **PCOMPi** Bulk Data entries is described in Chapter 5.

8.7 VISCOELASTIC MATERIAL PROPERTIES

Viscoelastic, or frequency dependent, materials may also be modeled in **UAI/*NASTRAN*** . The shear modulus, $G(f)$, for linear viscoelastic materials may be written as:

$$G(f) = G'(f) + i\,G''(f) \tag{8-14}$$

where $G'(f)$ and $G''(f)$ are the shear storage modulus and the shear loss modulus, respectively, and f the circular frequency. The frequency dependent Young's modulus is then computed using:

$$E(f) = 2(1+\nu)\,G(f) \tag{8-15}$$

To define an element with this type of property, you must reference a **MAT1** Bulk Data entry. The value of **G** that you specify is taken to be the reference value of the shear modulus, G_{ref}, and the value of **GE** that you specify is taken to be the reference structural damping coefficient, $g_{e_{ref}}$. Because the effects of viscous damping are included in the equations of motion using the structural damping matrix, $\boldsymbol{K}^4_{dd}$, all linear material elements must have **GE** values of zero. A detailed description of how viscoelastic materials are used is found in Chapter 9 of this manual.

The actual specification of the shear modulii is accomplished with **TABLED1** Bulk Data entries. There must be two separate tables, one which defines $G'(f)$ and another which defines $G''(f)$. The general form is then:

TABLED1	***tid***								**-cont-**
-cont-	**f(1)**	**G'(1)**	**f(2)**	**G'(2)**	**f(3)**	**G'(3)**	**ENDT**		
TABLED1	***tid+1***								**-cont-**
-cont-	**f(1)**	**G"(1)**	**f(2)**	**G"(2)**	**f(3)**	**G"(3)**	**ENDT**		

As discussed in Chapter 13, these tables are selected using the **SDAMPING** Case Control command. Note that it is required that the table identification numbers must be ***tid*** and ***tid+1*** and that the **SDAMPING** command selects ***tid***. For compatibility with other NASTRAN variants, there is an alternate input form which is detailed in Chapter 13.

8.8 FREQUENCY DEPENDENT STIFFNESS AND DAMPING PROPERTIES

Frequency dependent stiffness and damping properties may be modeled in **UAI/*NASTRAN*** by using the BUSHing element. These elements are defined by **CBUSH** Bulk Data entries. The properties of these elements are defined by **PBUSH** data which have the form:

PBUSH	PID	K1	K2	K3	K4	K5	K6		-cont-
-cont-		B1	B2	B3	B4	B5	B6		-cont-
-cont-		TIDK1	TIDK2	TIDK3	TIDK4	TIDK5	TIDK6		-cont-
-cont-		TIDB1	TIDB2	TIDB3	TIDB4	TIDB5	TIDB6		-cont-
-cont-		SA	ST	EA	ET				

The nominal stiffness and damping values, **Ki** and **Bi**, are used to compute the initial **KDD** and **BDD** matrices. The identification numbers **TIDKi** and **TIDBi** reference **TABLED1** and **TABDMP1** Bulk Data entries, respectively. These entries define the frequency dependency relationships:

TABLED1	TIDK1								-cont-
-cont-	f(1)	K1(1)	f(2)	K1(2)	ENDT				
TABLED1	TIDB1								-cont-
-cont-	f(1)	B1(1)	f(2)	B1(2)	f(3)	B1(3)	ENDT		

8

Chapter 9

DYNAMICS MODELING

UAI/*NASTRAN* provides extensive finite element modeling and analysis capabilities for structural and heat transfer dynamic analysis. Dynamic analyses which may be performed include eigenvalue extraction, frequency and random response analyses, and transient response analyses. The eigenvalue extraction analyses and frequency response calculations for structures assume linear behavior; however, nonlinear forcing functions for structures and nonlinear radiation effects for heat transfer are available in transient analysis. For structural models, dynamic analyses can be performed on either complete models or substructure models.

This chapter presents an outline of the capabilities and requirements for properly using the dynamics modeling features of **UAI/*NASTRAN*** for structural analysis. Such topics as mass modeling, damping, reduction techniques, transfer functions and solution strategies are discussed.

Other chapters of this guide present specific details for using the dynamic analysis Rigid Formats. The analysis of real eigenvalue problems is presented in Chapter 11. Complex eigenvalue problems are discussed in Chapter 12. The analysis of frequency and random response problems by direct and modal approaches is presented in Chapter 13. In Chapter 14 transient analysis by the direct and modal methods is presented. The computation of normal modes with differential stiffness effects is discussed in Chapter 16. Substructuring techniques, including component mode synthesis procedures especially useful for dynamics analysis, are presented in Chapter 19. In Chapter 22 the normal modes analysis for structures with cyclic symmetry is presented. All aspects of heat transfer modeling and analysis are presented in Chapter 24. you may perform Multidisciplinary Design Optimization (MDO) which may include both normal modes analyses and direct frequency response analyses. This is described in Chapter 25. Finally, **UAI/*NASTRAN*** also provides you the capability to compute the sensitivities of various structural responses. With this capability, you can compute frequency and eigenvector sensitivity. These features are described in detail in Chapter 26.

9.1 SUMMARY OF CAPABILITIES

Although this chapter presents modeling techniques for dynamic analysis, frequent reference is made to the analyses which may be performed on your models. These analyses capabilities are summarized in Table 9-1.

UAI/*NASTRAN* provides two major solution procedures for dynamic response analysis. These are called the ***direct method*** and the ***modal method***. With the direct method, the complete, coupled equations of motion are solved. With this technique no approximations are made with respect to the frequency fidelity of the model, however the solution of the coupled equations may become relatively CPU intensive if there are many frequencies or time steps in the analysis.

With the modal method, the coupled equations of motion are transformed to uncoupled equations which describe the behavior of normal mode, generalized coordinates. This approach usually results in much smaller and uncoupled equations which are solved in either the frequency or time domain. These solutions are much less CPU intensive than the direct approach and are therefore well suited for problems with many frequencies of excitation or many time steps of transient response. However, the modal method does have several disadvantages. First, there is an additional CPU cost associated with computing the normal modes. Second, there is an approximation introduced when less than all the modes of the structure (the normal situation) are used in the response analysis.

Table 9-1. SUMMARY OF DYNAMICS SOLUTION CAPABILITIES

ANALYSIS TYPE	Rigid Format	Solution Capability	Chapter No.
Eigenvalue Extractions	3	Normal Modes Analysis	11
	7	Direct Complex Eigenvalue Analysis	12
	10	Modal Complex Eigenvalue Analysis	12
	13	Normal Modes With Differential Stiffness Effects	16
	15	Normal Modes For Cyclic Symmetry Structures	22
	52	Normal Modes Sensitivity Analysis	26
Time Domain Analyses	9	Direct Transient Response Analysis	14
	12	Modal Transient Response Analysis	14
Frequency Domain Analyses	8	Direct Frequency and Random Response Analysis	13
	11	Modal Frequency and Random Response Analysis	13
Response Spectrum Analyses	17	Response Spectrum Analysis	15
MDO	MULTI	Multidisciplinary Design Optimization including Normal Modes Analyses and Direct Frequency Response Analyses.	25

Additional guidelines for selecting the direct versus modal approach to structural dynamic analyses are presented later in the respective chapters describing each specific type of analysis.

Input data for structural dynamic analysis have been organized to provide an easy transition from the static analysis model. All GRID and SCALAR point modeling features, elements, constraints, loads, and other options from static analysis are also available for dynamic analysis. Additionally, there are dynamics only modeling options which are described later in this chapter.

Data preparation for dynamic analysis requires consideration of the following additional factors as compared to statics analysis:

- Mass data are required.
- Damping may be included using elements or proportionality factors.
- Loads must be defined as functions of time or frequency.
- Data appropriate to the method of solution for time steps, frequencies, and eigenvalue ranges are required.

In addition to damping and mass modeling features provided for structural dynamic analysis, **UAI/*NASTRAN*** also provides general capabilities for modeling control systems and other nonstructural effects through the use of EXTRA points, scalar elements, transfer functions and direct matrix input. These data may contribute terms in any combination to the mass, damping and stiffness matrices.

9.2 MASS MODELING

Structural mass data for dynamic analyses are defined by structural elements, lumped mass elements, or direct input matrix terms and transfer function data. This section will describe the methods for defining these data.

9.2.1 Mass Properties of Structural Elements

All structural elements may have both structural and non-structural mass. Structural mass is calculated from material and geometric properties. The mass is assumed to be concentrated at the middle surface of plates and along the neutral axis of one-dimensional elements. In the case of solid elements, the mass is assumed to have a uniform density throughout the volume of the solid.

For solid elements and plate elements, rotational inertia terms are not computed. For the one-dimensional elements the total mass of the elements is lumped at the two end points. The ROD element has no bending stiffness or rotational or torsional inertia terms. The BAR element normally has no rotational or torsional inertia terms. However, if you use the coupled mass formulation, bending inertia terms are introduced. To produce torsional inertia terms for the BAR element, you may include the parameter:

PARAM	**BARTORM**	**YES**							

in you Bulk Data packet. The BEAM element ***always*** produces torsional inertia terms, and it will also produce bending inertia terms with the coupled mass formulation.

The mass of structural elements is distributed to the GRID points that define the element by either of two methods. You may choose the lumped mass method or the consistent mass method.

In the lumped mass method, the mass of an element is divided and assigned to the connected GRID points. Thus, for ROD, BAR and uniform BEAM elements, one-half of the mass is placed at each end. Second mass moments are not computed with the lumped mass method.

In the consistent, or coupled, mass method the mass matrix is computed for each structural element and includes off-diagonal coefficients that couple adjacent GRID points. This method uses the same assumed deformation shapes for both the stiffness matrix generation and the computation of the mass matrix terms.

You select the use of consistent mass modeling with the Bulk Data entry:

PARAM	**COUPMASS**	**1**							

If this Bulk Data entry is not present the lumped mass method is used.

Mass data are defined in two ways for structural elements. The first involves specification of the material density on the appropriate material Bulk Data entry. The density may be either weight or mass per unit volume. If weight is used, the Bulk Data entry:

PARAM	**WTMASS**	*scale*							

must be specified to convert weight to mass. The real number *scale* is the ratio of mass to weight. Thus, it is usually the reciprocal of the acceleration of gravity in the units of the model.

A second method is available wherein non-structural mass is specified on the element property entries directly. As is the case with density, these data may be either mass or weight per unit length or area. The units of non-structural mass must be consistent with the units of material density because the **WTMASS** scale factor will also be applied to these data. A typical example of the use of non-structural mass is the modeling of paint, wire harnesses, and other miscellaneous items attached to the structure.

9.2.2 Defining Concentrated Mass Elements

For modeling concentrated or lumped masses, **UAI/*NASTRAN*** provides six scalar elements: **CONM1**, **CONM2**, **CMASS1**, **CMASS2**, **CMASS3**, and **CMASS4**. The **CMASSi** Bulk Data entries define scalar masses. These entries define connections between a pair of degrees of freedom (at either SCALAR or GRID points) or between one degree of freedom and ground.

The **CONM1** and **CONM2** Bulk Data entries are used to specify the mass properties of a GRID point as a 6 by 6 matrix. **CONM1** requires the explicit definition of all terms in the matrix, where as **CONM2** input consists of mass and moments of inertia terms and an offset of the mass point with respect to a GRID point.

9.2.3 Weight and Balance Summary

You may request the computation and printing of weight, or mass, and center-of-gravity data for your structural model. This feature is frequently called the ***Grid Point Weight Generator*** by experienced users. Rigid body mass properties of the structure, with respect to any specified point and also with respect to the center of mass are computed. For standard analyses, these computations are requested with the Bulk Data entry:

PARAM	**GRDPNT**	*gid*							

where *gid* is a reference GRID point identification number. The rigid body mass will be computed with respect to this point. If *gid* is omitted, the origin of the BASIC coordinate system will be used as the reference point. An additional feature which may be used is the **WTMASS** parameter, described above.

Mass and balance data may also be obtained for any number of substructures when performing Substructuring Analysis by using the Substructure Command **MASSBAL** as shown below:

```
MASSBAL name
```

The substructure *name* may be any substructure. Two optional subcommands are also available, **RNAME** and **RGRID**:

```
RNAME refname
RGRID pointid
```

where *pointid* is a GRID point which defines the location of the mass reference point. The BASIC substructure *refname* must contain the GRID point given in the **RGRID** subcommand. If the reference GRID point is omitted, the origin of the BASIC coordinate system of the reference substructure will be used as the reference point. If a valid reference substructure name is not provided, the origin of the coordinate system of the named substructure will be used as the reference point.

A typical weight and balance summary table is shown in Figure 9-1. The information provided defines the important mass characteristics of the model as is described next.

9.2.3.1 Rigid Body Mass Matrix

This matrix defines the mass of the entire structure with respect to the reference point selected on the **PARAM**, **GRDPNT** Bulk Data entry.

9.2.3.2 Transformation Matrix, Basic To Principal

This is the transformation from the BASIC coordinate system to the set of principal axes, called the S-system, for the 3 by 3 scalar mass partition of the 6 by 6 mass matrix.

Figure 9-1. GRID POINT WEIGHT AND BALANCE SUMMARY

```
      G R I D   P O I N T   W E I G H T   A N D   B A L A N C E   S U M M A R Y
                              REFERENCE POINT =        0
          RIGID BODY MASS MATRIX AT REFERENCE POINT IN BASIC COORDINATE SYSTEM
***                                                                           ***
*  1.000505E+01  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00 *
*  0.000000E+00  1.000505E+01  0.000000E+00  0.000000E+00  0.000000E+00  5.046000E-02 *
*  0.000000E+00  0.000000E+00  1.000505E+01  0.000000E+00 -5.046000E-02  0.000000E+00 *
*  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00 *
*  0.000000E+00  0.000000E+00 -5.046000E-02  0.000000E+00  1.667333E+03  0.000000E+00 *
*  0.000000E+00  5.046000E-02  0.000000E+00  0.000000E+00  0.000000E+00  6.728336E-01 *
***                                                                           ***
                TRANSFORMATION MATRIX, BASIC TO PRINCIPAL MASS SYSTEM
                      ***                                   ***
                      *  1.000000E+00  0.000000E+00  0.000000E+00 *
                      *  0.000000E+00  1.000000E+00  0.000000E+00 *
                      *  0.000000E+00  0.000000E+00  1.000000E+00 *
                      ***                                   ***
                                 CENTER OF GRAVITY SUMMARY
        DIRECTION
      MASS AXIS SYSTEM       MASS              X-C.G.        Y-C.G.        Z-C.G.
            X            1.000505E+01      0.000000E+00  0.000000E+00  0.000000E+00
            Y            1.000505E+01      5.043455E-03  0.000000E+00  0.000000E+00
            Z            1.000505E+01      5.043455E-03  0.000000E+00  0.000000E+00
                      INERTIA MATRIX IN THE PRINCIPAL MASS SYSTEM
                      ***                                   ***
                      *  0.000000E+00  0.000000E+00  0.000000E+00 *
                      *  0.000000E+00  1.667333E+03  0.000000E+00 *
                      *  0.000000E+00  0.000000E+00  6.725791E-01 *
                      ***                                   ***
                               PRINCIPAL MOMENTS OF INERTIA
                      ***                                   ***
                      *  0.000000E+00                         *
                      *                1.667333E+03           *
                      *                             6.725791E-01 *
                      ***                                   ***

               TRANSFORMATION MATRIX, PRINCIPAL INERTIA TO PRINCIPAL MASS
                      ***                                   ***
                      *  1.000000E+00  0.000000E+00  0.000000E+00 *
                      *  0.000000E+00  1.000000E+00  0.000000E+00 *
                      *  0.000000E+00  0.000000E+00  1.000000E+00 *
                      ***                                   ***
```

9.2.3.3 Center Of Gravity Summary

A structural model may have different values of mass in each coordinate direction at a GRID point. This can arise, for example, when using scalar mass components. Consequently, three distinct mass systems are assembled, one in each of the three directions of the S-system. This summary has five columns. The first column gives the S coordinate axis. The second column, the mass associated with the specified direction, and finally, the x, y, and z distances from the reference point to the center of mass for each of the three mass systems.

9.2.3.4 Inertia Matrix In The Principal Mass System

This is the 3 by 3 mass moment of inertia partition, I(S), with respect to the center of gravity in the S system. This is not necessarily a diagonal matrix because the determination of the S system does not involve second moments. The values of the inertias at the center of gravity are found from the values at the reference point by employing the parallel axes theorem.

9.2.3.5 Principal Moments Of Inertia

The principal moments of inertia at the center of gravity are given in matrix form with reference to the Q system of axes. The Q system is obtained from an eigenvalue analysis of the I(S) matrix.

9.2.3.6 Transformation Matrix, Principal Inertia To Principal Mass

This is the coordinate transformation matrix between the S and the Q systems described above.

9.3 DAMPING MODELING

Damping effects may be modeled in all dynamic analyses except real eigenvalue extraction (Rigid Format 3). The actual type of damping data which may be used depends on the method of analysis formulation and solution defined by the Rigid Format. For example, in a modal formulation (Rigid Formats 10, 11, and 12), the equations of motion are normally uncoupled for solution of modal coordinate response. Therefore, uncoupled modal damping functions may be specified.

The damping options available in **UAI/*NASTRAN*** are shown in Table 9-2 along with the manner in which each type is applied to your model. Following is a brief introduction to these damping features. Detailed instructions and effects on the equations of motion for each type of damping are presented later in each chapter describing the particular response analysis procedure.

9.3.1 Viscous Damping

Viscous damping is defined with **CVISC** and **CDAMPi** Bulk Data entries. The VISC element connects two GRID points with both extensional and torsional damping terms. The DAMPi elements connect individual degrees of freedom between GRID points and SCALAR points.

In the general equation of motion for dynamics shown below, all viscous damping elements contribute terms to the $\boldsymbol{B}$ matrix.

$$\boldsymbol{M}\ddot{u} + \boldsymbol{B}\dot{u} + \boldsymbol{K}u = \boldsymbol{P} \qquad (9\text{-}1)$$

9.3.2 Structural Damping

Damping mechanisms such as Coulomb friction and hysteresis are commonly included as structural damping. Structural damping refers to terms that produce a force which is in phase with velocity and proportional to displacement. This allows structural damping to be conveniently defined as proportionality constants normalized to stiffness. The damping matrix is

Table 9-2. DAMPING OPTIONS AVAILABLE IN UAI/NASTRAN

TYPE OF DAMPING	APPLY WITH:	RIGID FORMAT						
		7	8	9	10	11	12	MULTI
Viscous	Scalar Elements **DAMPi** and **VISCi**	✔	✔	✔	✔	✔	✔	✔
Structural — Material	Bulk Data Entry **MATi** (**GE** Field)	✔	✔	✔	✔	✔	✔	✔
Structural — Global	Bulk Data Entry **PARAM G**	✔	✔	✔	✔	✔	✔	✔
Direct Input	Case Control Commands **B2GG** and **B2PP**	✔	✔	✔	✔	✔	✔	✔
Transfer Functions	Bulk Data Entry **TF**	✔	✔	✔	✔	✔	✔	✔
Modal Damping Factors	Case Control Command **SDAMP**	N/A	N/A	N/A	✔	✔	✔	N/A
4 Available N/A Not Applicable								

then produced by simply applying these scale factors to the stiffness matrix. In **UAI/*NASTRAN***, structural damping may be defined in two ways.

First, the **GE** field of any material (**MATi**) Bulk Data entry may be used to define a scale factor which is used to compute element structural damping matrices from the element stiffness matrices generated for that material.

Secondly, a global factor, *g*, may be defined with a **PARAM** Bulk Data entry as shown below:

PARAM	**G**	*g*							

In this case, a structural damping matrix of the form $g\,\boldsymbol{K}$ is created and added to the equations of motion where the factor *g* has the value defined by *g*. Note that $\boldsymbol{K}$ is the assembled stiffness for the entire model. The factor *g* is not applied to any stiffness terms that you have introduced through **DMI** Bulk Data entries, i.e. **K2GG** or **K2PP** direct matrix input.

In the case where material damping factors, **GE**, have been defined, a new and separate stiffness matrix, $\boldsymbol{K}^4$, is generated by assembling all of the scaled element stiffness matrices. Note that **GE** may be defined for all, some, or none of the materials, and that these may be used in combination with the global damping.

The two structural damping terms are then combined to form the complex stiffness matrix and it associated force-displacement relationship:

$$\boldsymbol{F} = \left[\boldsymbol{K}(1 + i\,\mathrm{g}) + i\,\boldsymbol{K}^4 \right]\boldsymbol{u} \tag{9-2}$$

This complex form of the stiffness matrix is used without modification for Frequency Response and Complex Eigenvalue analyses.

In **UAI/*NASTRAN***, transient analysis does not use complex matrices. Therefore, a set of equivalent viscous damping matrices, which are real, are defined. The displacement-dependent imaginary terms of (9-2) are converted to velocity-dependent real terms using the relationships:

$$ig\,\boldsymbol{K}\boldsymbol{u} \Rightarrow \frac{1}{\omega_3}\,g\,\boldsymbol{K}\dot{\boldsymbol{u}} \tag{9-3a}$$

$$i\,\boldsymbol{K}^4\boldsymbol{u} \Rightarrow \frac{1}{\omega_4}\,\boldsymbol{K}^4\dot{\boldsymbol{u}} \tag{9-3b}$$

A new damping matrix of the form:

$$\boldsymbol{B} + \frac{\mathrm{g}}{\omega_3}\boldsymbol{K} + \frac{1}{\omega_4}\boldsymbol{K}^4 \tag{9-4}$$

results. Note that the equivalent damping forces match at a given angular frequency, $i\,\omega_3\boldsymbol{u} = \dot{\boldsymbol{u}}$ and $i\,\omega_4\boldsymbol{u} = \dot{\boldsymbol{u}}$. To accomplish this, you must define the frequency scale factors ω_3 and ω_4 with the **PARAM** Bulk Data entries:

PARAM	**W3**	ω3							
PARAM	**W4**	ω4							

As shown in , ω_3 scales the global structural damping $g\boldsymbol{K}$, and ω_4 scales the material-dependent structural damping $\boldsymbol{K}^4$.

For Modal Transient Response analyses, an improved alternative to ω_4 is available as described in the following section.

9.3.3 Damping Modeling for the Modal Approach

The discussion of viscous and structural damping presented in earlier sections applies to both the direct and modal solution approaches. **UAI/*NASTRAN*** provides two additional damping options when you use the modal approach. This section discusses these options.

The first method allows you to define the damping as a fraction of the critical damping as if each mode were an uncoupled second order system. This type of damping is called *modal damping*.

You select modal damping in **UAI/*NASTRAN*** with the Case Control command:

```
SDAMPING = setid
```

and define the damping values as a function of frequency with a **TABDMP1** Bulk Data entry:

TABDMP1	TID	XAXIS	YAXIS	DMPTYP					-cont-
-cont-	f1	g1	f2	g2	*CONTINUES IN GROUPS OF TWO*				-cont-

The **DMPTYP** field specifies the type of damping values **gi**. The allowable values are **FRACTION**, **G**, **Q**, and **PERCENT**. The default value is **G**. The interpretation of these options is summarized in Table 9-3.

The fields **XAXIS** and **YAXIS** are used to specify that the table entries (in either or both coordinate directions) are to be interpolated using **LINEAR** or **LOG**arithmic rules. The default value is **LINEAR**. The values **gi** are input in this table as a function of frequency (Hz). The data **gi** are *twice* the value of critical damping. Thus, if 3% critical damping is desired at 30 Hz, then the data pair (30.0, 0.06) would be included in your input.

Table 9-3. SELECTING DAMPING VALUES

If DMPTYP is:	Then the g_i input is:
FRACTION	ζ
G	2ζ
PERCENT	100ζ
Q	$\frac{1}{2\zeta}$
Where the terms of the damping matrix are given by: $b_i = (2\pi f_i)(2\zeta(f_i))m_i$	

The second method, used only in the modal approach to Transient Response analysis, is a procedure wherein the equivalent viscous damping may be set independently for each modal degree-of-freedom. This method is provided to convert the structural damping from imaginary stiffness terms to real viscous terms, as shown in section 9.3.2. You select this option by using the Bulk Data entry:

PARAM	W4MODAL	YES							

When you select this option, instead of using the single value ω_4, the frequency for each normal mode is used to scale its respective damping terms. First the matrix $\mathbf{\Omega}$ is assembled:

$$\mathbf{\Omega}_{ii} = \frac{1}{\sqrt{2\pi f_i}} \text{ and } \mathbf{\Omega}_{ij} = 0.0 \text{ for } i \neq j$$

This matrix is then used to scale the modal material damping matrix, $\boldsymbol{K}^4_{hh}$:

$$\mathbf{\Omega}^T \boldsymbol{K}^4_{hh} \mathbf{\Omega}$$

This has the desired effect of scaling the modal $\boldsymbol{K}^4$ matrix such that the viscous damping forces match the structural damping forces at each modal frequency. As a result, this method is preferable to using a single ω_4 value.

When only modal damping is defined, the system equations are uncoupled. This results in a very efficient solution. However, if you use viscous dampers, or material damping, the resulting damping matrix will be coupled. If you take no further action, a coupled response solution is required. Such solutions are often significantly more expensive than the uncoupled case.

UAI/*NASTRAN* allows you to request that only the diagonal terms of the damping matrix be retained. This results in the more efficient solution process. This treatment of damping in the modal approach is selected with the Bulk Data entry:

PARAM	DIAGNLPP	YES							

9.4 DIRECT MATRIX INPUT

DMIG Bulk Data allow the definition of a matrix by GRID point and component degree of freedom. These entries can be used to define mass, stiffness, or damping matrices that are added to the *g-size* matrices before any reductions are performed. One or more **DMIG** matrices *namei* are then selected by one of the Case Control commands:

```
K2GG = name1,name2,...
M2GG = name1,name2,...
B2GG = name1,name2,...
```

These may be used in all structural Rigid Formats and must be symmetric.

DMIG matrices can also be added to the *p-set* matrices after the reduction to *a-set*, if you use the direct method, or after the normal modes are computed if you use the modal method. When using the modal approach to dynamic response analysis, direct input stiffness, mass and damping data may be input as coefficients for coupling the structural modal coordinates with any EXTRA points you have defined. This is accomplished by using the Case Control commands:

```
K2PP = name1,name2,...
M2PP = name1,name2,...
B2PP = name1,name2,...
```

These commands may reference matrices that are symmetric or unsymmetric.

9.4.1 Transfer Functions

Transfer function data, defined with the **TF** Bulk Data entries, may be used to provide additional matrix coefficient terms on dynamic degrees of freedom in any complex eigenvalue and dynamic response analysis. All Transfer Function coefficients are added to any other existing coefficient data created by structural or scalar element modeling or modal reduction.

Transfer functions have been designed to facilitate the definition of non-structural phenomena, i.e. a missile control system consisting of sensors, voltages, amplifiers, servos, etc. which are then coupled to the structural model. Often the objective of such an analytical model is the prediction of the system's stability from a controls system analysis frame of reference. In such cases, EXTRA Points are often introduced to represent the non-structural components. The Transfer Function data then contain the matrix equations which defined the non-structural components and their relationships to structural displacement, velocity and/or acceleration.

The general form of a Transfer Function is :

$$\left(\boldsymbol{B2}\, p^2 + \boldsymbol{B1}\, p + \boldsymbol{B0} \right) u_d + \sum_i \left(\boldsymbol{A2}(i)\, p^2 + \boldsymbol{A1}(i)\, p + \boldsymbol{A0}(i) \right) u(i) = 0 \qquad (9\text{-}8)$$

where u_d is a dependent coordinate and $u(i)$ are independent coordinates. Transfer functions are selected with the Case Control command:

```
TFL = setid
```

where *setid* is the identification number specified on one or more **TF** Bulk Data entries.

Demonstration problem **D100110** in the **UAI/*NASTRAN Demonstration Problem Manual*** presents an example of the use of Transfer Functions in a model of a rocket and it's attitude control system.

9.5 REDUCTIONS IN DYNAMIC ANALYSIS

Computations associated with dynamics are generally more CPU intensive than those for static analysis of the same model. For this reason, and also because certain dynamic analysis solution techniques have restrictions with respect to the conditioning of the mass and stiffness matrices, techniques have been developed to transform the original equations of motion to a smaller set of equations. The unknowns in the transformed equations may be a subset of the original physical degrees of freedom or a new set of generalized coordinates or a combination of the two sets. Such a transformation is called a reduction because the number of equations which are ultimately solved is usually smaller than the original set.

Several reduction techniques are available in **UAI/*NASTRAN***. These include *Static Condensation*, also known by other names including *Guyan Reduction* and *Mass Condensation*, *Dynamic Reduction*, and *Component Mode Synthesis*. Component mode synthesis techniques are only available when using substructure techniques, dynamic reduction is available in all normal modes and dynamic response Rigid Formats, and static condensation is available in all Rigid Formats.

9.5.1 Static Condensation

Static condensation techniques require you to specifically select all degrees of freedom which you deem *important* to the analysis. These degrees of freedom are placed in the *a-set* using **ASETi** Bulk Data. For dynamic analysis, this means degrees of freedom with significant mass and sufficient degrees of freedom to spatially describe the expected motion of the structure are selected. Normally, all massless degrees of freedom are excluded from this set as well as many other degrees of freedom which you assume as not critical to good results.

Obviously, the accuracy of analysis results using this technique is highly dependent on the skill and experience of the engineer who selects the *a-set*. As an alternative to your manual selection of the *a-set*, there are two other options. First, you may use the **AUTOOMIT** Case Control command to request **UAI/*NASTRAN*** to automatically select the *a-set* using criteria you specify. Second, you may use the **AUTOREDUCE** Case Control command to request **UAI/*NASTRAN*** to automatically perform a static condensation based on frequency accuracy criteria that you select.

Guidelines for properly performing static condensation, including the manual selection of the *a-set* or use of the AUTOOMIT and AUTOREDUCE featurse are described in detail in Chapter 6 of this guide.

9.5.2 Dynamic Reduction

The dynamic reduction technique requires that you specify an upper limit to the important frequencies of the analysis, or an upper limit to the number of generalized dynamic degrees of freedom which will be created and used in the analysis. Normally, the frequency limit is the method of control of the procedure since the number of generalized degrees of freedom which should be used may not be well defined. This procedure is activated by using the **DYNRED** Case Control command which selects a **DYNRED** Bulk Data entry.

This procedure automatically generates SCALAR points, one for each generalized dynamic degree of freedom (which will be automatically introduced and added to your model). A transformation matrix is created which defines the relationship between the physical degrees of freedom being reduced to the new set of generalized coordinates. In addition to the new SCALAR point generalize coordinates, you may also retain any number of physical degrees of freedom, identified using **ASETi** Bulk Data, to be included in the final *a-set* degrees of free-

dom for solution. For maximum accuracy, it is recommended you use this technique for degrees of freedom carrying point loads in dynamic response analyses when using dynamic reduction.

Detailed instructions for using the dynamic reduction procedure, including a theoretical summary, are presented in Chapter 6.

9.5.3 Component Mode Synthesis and Craig-Bampton Reduction

Component mode synthesis techniques are available when using the Substructuring feature of **UAI/*NASTRAN***. Component mode synthesis is similar to dynamic reduction described above, except that the generalized dynamic coordinates of dynamic reduction are replaced by true normal mode coordinates. Substructuring and component mode synthesis techniques are especially suited for large, complex models with a large number of modes in the frequency range of interest. Instructions for using this feature are presented in Chapter 19.

For standard dynamic analyses, a Craig-Bampton Reduction is available and is selected using the **AUTOREDUCE** Case Control command. This is described in Chapter 6 of this guide.

9.5.4 Guidelines For Reduction In Dynamic Analysis

Only those degrees of freedom which are sufficient to characterize the dynamic behavior or response of the structure accurately must be retained in a typical dynamic analysis. However, the selection of the dynamic degrees of freedom is not always an easy task. Many sets of reduced degrees of freedom are possible and will provide acceptable results in the analysis. In some respects, the selection of dynamic degrees of freedom is analogous to the selection of lumped masses for a dynamic model.

The following guidelines can be used, in addition to your engineering judgment, in the selection of dynamic degrees of freedom:

- One basic criterion in the selection of the dynamic degrees of freedom is that they should sufficiently describe the anticipated mode shapes of the structure.
- The dynamic degrees of freedom should be placed at locations of large masses and rotary inertia. These locations tend to define the mode shapes of the structure. You also need to place dynamic degrees of freedom at locations of lower structural rigidities for the same reason.
- It is good practice to place dynamic degrees of freedom at each point of dynamic force application.
- For structures with a high degree of planar geometry, the out-of-plane motions need to be modeled more accurately than the in-plane motions. This is because the bending frequencies will normally be much lower than tension-compression mode frequencies for such structures.
- You may reduce most or all rotational degrees of freedom in dynamic analyses. This is because the rotational inertia is often unknown or undefined and assumed to have a value of zero. All degrees of freedom with a mass value of zero have absolutely no effect on the dynamic response of the structure, and they may be reduced with a static condensation technique with no loss of accuracy.

This page is intentionally blank.

Chapter 10

LINEAR STATIC ANALYSIS

UAI/*NASTRAN* provides five Rigid Formats, shown in Table 10-1, to perform Linear Static Structural Analysis. Cyclic Symmetry, which is implemented in Rigid Format 14, is fully described in Chapter 22 and Design Sensitivity Analysis in Chapter 25. In addition, static analyses may be performed as part of Multidisciplinary Analysis and Design, described in Chapter 26. This chapter provides information on the use of Rigid Formats 1 and 2. Rigid Format 1 is used when the model is constrained from rigid body motion, and Rigid Format 2 is used for a free-body model or one which contains mechanisms.

Table 10-1. SUMMARY OF STATICS RIGID FORMATS

RIGID FORMAT	DESCRIPTION
1	Static Structural Analysis
2	Static Analysis with Inertia Relief
14	Statics using Cyclic Symmetry
51	Design Sensitivity Analysis
MULTI	Multidisciplinary Analysis and Design

10

10.1 MATHEMATICAL BACKGROUND

Static Analysis is the most widely used capability of the finite element method. It is used to determine the structural response of a model subject to steady-state external loads. This is done by solving the equations of static equilibrium:

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{P} \tag{10-1}$$

where $\boldsymbol{K}$ is the stiffness matrix, $\boldsymbol{P}$ the external loads, and $\boldsymbol{u}$ the response, or static deformations of the system. Once the deformations have been determined, the strains may be computed and then, from these and the material relationships, the stresses. Note that the solution of (10-1) requires that the stiffness matrix, $\boldsymbol{K}$, is nonsingular. This usually requires that your model, taken as a whole, be constrained against free-body motion in its translational and rotational directions. This may not always be true. In such cases, you may still solve the problem as described in the next section.

10.1.1 Solving Free-Body Models

A free body is defined as a structure that is capable of motion without internal stress, that is, it has one or more rigid body degrees of freedom. The stiffness matrix for a free-body is singular and the number of singularities equals 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 add more rigid body modes to a structure. **UAI/*NASTRAN*** places no restriction on the number of stress-free modes which permits the analysis of complex mechanisms.

Static solution techniques for free body structures, provided by Rigid Format 2, use a temporary restraint, called a SUPORT, that you define. This restraint allows the decomposition of the stiffness matrix to proceed. The restrained degrees of freedom are given a zero displacement value in the solution, and other displacements are then computed relative to these points. For Static Analyses you must specify sufficient restraints to constrain all rigid body motion.

SUPORT Bulk Data entries are used to define free-body supports. In the case of problems using substructuring, the **SUPORTS** Bulk Data entry is used. In all cases, only a single set can be specified, and if such entries appear in the Bulk Data packet, they are automatically used in the solution. Free-body supports are defined in the global coordinate system of the specified GRID point. The following section describes the manner in which the supports are used.

10.1.2 Solution Compatibility Check for Free Bodies

For the free-body problem, a subset, $\boldsymbol{u}_r$, of the solution displacement vector, $\boldsymbol{u}_a$, must be constrained during the static analysis. This subset must include degrees of freedom that are just sufficient to eliminate the stress-free motion without introducing redundant constraints. The complete static equilibrium equations may be partitioned as:

$$\begin{bmatrix} \boldsymbol{K}_{ll} & \boldsymbol{K}_{lr} \\ \boldsymbol{K}_{ll}^T & \boldsymbol{K}_{rr} \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}_l \\ \boldsymbol{u}_r \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{P}_l \\ \boldsymbol{P}_r \end{Bmatrix} \tag{10-2}$$

Equation (10-2) allows a comparison of the compatibility between the single point and multipoint constraints previously applied and the constraints placed on the reaction points. By eliminating the $\boldsymbol{u}_l$ degrees of freedom from (10-2), the reduced stiffness matrix relating to the $\boldsymbol{u}_r$ partition should be null. This reduced stiffness matrix, $\boldsymbol{X}$ is defined by:

$$\boldsymbol{X} = \left[\boldsymbol{K}_{rr} - \boldsymbol{K}_{lr}^{T} \boldsymbol{K}_{ll}^{-1} \boldsymbol{K}_{lr} \right]$$

There are several reasons that $\boldsymbol{X}$ is not identically zero. These include:

- Normal computational round-off
- That $\boldsymbol{u}_r$ is overdetermined (i.e. there are redundant supports) or underdetermined (i.e. $\boldsymbol{K}_{ll}$ is singular)
- That previously defined single point constraints or multipoint constraints are restraining the rigid body motion.

The solution error for each degree of freedom in the *r-set*, $\varepsilon_{(r)}$, is defined as:

$$\varepsilon_{(r)} = \frac{||\boldsymbol{X}_{(r)}||}{||\boldsymbol{K}_{(r)}||}$$

where the subscript (r) denotes the r^{th} column of the specified matrix. The total solution error, ε, is similarly defined as:

$$\varepsilon = \frac{||\boldsymbol{X}||}{||\boldsymbol{K}_{rr}||}$$

The values of ε_r, when determinate, are printed along with the values $||\boldsymbol{X}||^2$, $||\boldsymbol{K}_{rr}||^2$ and ε. An example is presented in Section 10.4.3, Table 10-4.

10

10.2 THE STATIC ANALYSIS MODEL

The finite element representation of a structure for Static Analysis, and indeed for most other types of analyses, is composed of the three components as discussed below.

10.2.1 The Finite Element Idealization

The mathematical idealization, or finite element model, of the structure is composed of a set of elements connected to GRID points defined in some coordinate system. **UAI/*NASTRAN*** allows independent coordinate systems for each GRID point both for geometric specification and output displacements. These systems may be rectangular, cylindrical, or spherical. The collection of all these systems is called the ***Global Coordinate System***. All **UAI/*NASTRAN*** output is defined in the global system. Each of the elements has a set of properties that define its physical behavior. You will recall that the idealization process is described in Chapter 4, the **UAI/*NASTRAN*** element library in Chapter 5, and material properties are discussed in Chapter 8.

10.2.2 Boundary Conditions

Prior to solution, boundary conditions are generally imposed on the finite element model. The constraints can be defined in several ways to represent fixity, symmetry conditions, or dependency relationships between various degrees of freedom within the model. These are detailed in Chapter 6.

10.2.3 External Loads

Finally the external environmental loads are applied to the system. Many types of mechanical, gravity, thermal, pre-strain and enforced deformation loads are available as described in Chapter 7.

10.3 INPUT DATA REQUIREMENTS

This section describes the input data requirements for Statics Analysis using **UAI/*NASTRAN***.

10.3.1 Executive Control Commands

In **UAI/*NASTRAN***, Statics Analysis is requested by using Rigid Formats 1 or 2, or by using the Multidisciplinary solution sequence. The Executive Control commands are specified as:

```
SOL 1   (Statics)
SOL 2   (Inertia Relief)
    or
SOL MULTI
   CASE STATICS
```

Depending upon the features that you select, other commands may be required. For example, if you are using the Checkpoint/Restart capability or if you are creating plots.

10.3.2 Substructuring Commands

Although there are no Substructuring commands that are used only for Static Analysis, the **OPTIONS** command for appending load vectors is very useful for Statics. This option allows you to define new loads by re-executing the PHASE 1 job using the command:

```
OPTIONS = PA
```

Only minimal computations are performed to add the new loads. See Chapter 19 for more details.

10.3.3 Case Control Commands

The majority of the available Case Control commands are used when you perform Statics Analyses. Table 10-2 summarizes these commands and groups them in seven classes. The first class is commands used to describe one or more SUBCASEs. The general structure of the Case Control command packet for Static Analysis is shown in Figure 10-1. The packet is composed of blocks of commands that terminate with the **BEGIN BULK** command. The following sections will describe how SUBCASEs are used in more detail.

Figure 10-1. CASE CONTROL ORGANIZATION

DEFAULT SELECTIONS
FIRST SUBCASE COMMANDS
SECOND SUBCASE COMMANDS
...
LAST SUBCASE COMMANDS
BEGIN BULK

The second class of commands is used to select loads. The available loads and methods that you use to select them are described in Chapter 7. Similarly, the third class of commands, used to select constraints, is described in Chapter 6. The fourth class of commands is the **SET** definition. This is also described below. The next two classes, commands used to request GRID point and element solution results, were described in Chapters 4 and 5. Finally,

Table 10-2. CASE CONTROL COMMANDS FOR STATICS

COMMAND CLASS	COMMAND	DESCRIPTION
SUBCASE DEFINITION	REPCASE	Defines an output request SUBCASE.
	SUBCASE	Defines the start of a SUBCASE.
	SUBCOM	Defines a linear combination of SUBCASEs.
	SUBSEQ	Defines the coefficients for a SUBCOM.
LOAD SELECTION	DEFORM	Selects preloading due to element deformation.
	LOAD	Selects static loading conditions.
	TEMP(LOAD)	Selects temperature set for loads.
	TEMP(MAT)	Selects temperature set for material properties.
CONSTRAINT SELECTION	AUTOSING	Selects singular matrix decomposition options.
	AUTOSPC	Selects automatic matrix singularity removal.
	MPC	Selects the multipoint constraint set.
	SPC	Selects the single-point constraint set.
SET DEFINITION	SET	Defines a list of identifiers referenced by other input data.
GRID POINT SOLUTION RESULTS	DISPLACEMENT	Requests displacements in the physical set.
	EQUILIBRIUM	Requests static equilibrium summary.
	GPFORCE	Requests GRID point forces.
	GPSTRESS	Requests GRID point stresses.
	MPCFORCE	Request multipoint constraint and rigid element forces.
	OLOAD	Requests echo of GRID point applied loads.
	SPCFORCE	Requests single-point constraint forces.
ELEMENT SOLUTION RESULTS	ESE	Requests element strain energies.
	FORCE	Requests element forces.
	STRAIN	Requests element strains.
	STRESS	Requests element stresses.
PLOTTING	OUTPUT(PLOT)	Start of structural plotter subpacket.
	OUTPUT(XYPLOT)	Start of X-Y plotter subpacket.

the last class, which allow you to define plotting commands, are discussed in Chapters 29 and 30.

10.3.3.1 Hierarchic Organization

The Case Control command packet is organized so that you are required to enter the minimal number of commands. This is facilitated by a hierarchical organization of the commands. Referring to Figure 10-1, you will note the block labeled **DEFAULT SELECTIONS**. This block contains commands that you wish to use often within the SUBCASEs which follow.

10.3.3.2 SUBCASE Definition

For efficiency, the solution algorithm for Static Analysis solves (10-1) by assembling any number of loading conditions which share the same boundary conditions into a single ***P*** matrix. However, each time that you change the boundary conditions, **UAI/*NASTRAN*** must re-partition the global matrices, as described in Chapter 6, and perform a new solution. Each unique combination of a physical load and a boundary condition is called a SUBCASE. Within each SUBCASE you may specify constraints, loads and output selections. You identify the beginning of a SUBCASE with the command:

```
SUBCASE id
```

where *id* is a unique identification number that you assign to the SUBCASE. The only restriction on these identification numbers is that they be increasing. Example 1 illustrates the use of multiple load and constraint cases. This packet defines three SUBCASEs. The displacements at all GRID points will be printed for each SUBCASE, because the command appears in the default block of the packet. Similarly, the default **MPC = 101** will be used until **SUBCASE 3** when it is overridden by the command **MPC = 102**. SUBCASEs 1 and 3 use **SPC = 201**, but **SUBCASE 2** selects **SPC = 202**. Each SUBCASE selects a different load. This particular example would require three matrix solutions because there are three distinct sets of boundary conditions.

EXAMPLE 1

```
DISPLACEMENT = ALL
MPC = 101
SPC = 201
SUBCASE 1
   LOAD = 101
SUBCASE 2
   SPC = 202
   LOAD = 102
SUBCASE 3
   MPC = 102
   LOAD = 103
BEGIN BULK
```

10.3.3.3 Superimposing Solutions — SUBCOM

There may be cases in which you wish to superimpose two or more of the loading conditions applied to your model. There is a special Case Control command which allows you to do this:

```
SUBCOM id
   SUBSEQ = c1,c2 [,c3,...,cn]
```

As before, *id* is a unique identification number for the SUBCASE. The **SUBSEQ** subcommand defines a set of coefficients, *ci* which are multipliers of the loads that were used in the immediately preceding *n* SUBCASEs. Example 2 shows how SUBCASEs may be superimposed. Two static loading conditions are defined in **SUBCASE 1** and **SUBCASE 2**. **SUBCOM 51** defines the sum of **SUBCASE 1** and **SUBCASE 2**. **SUBCOM 52** defines a linear combination

10

EXAMPLE 2

```
SPC = 2
OUTPUT
   SET 1 = 1 THRU 10,20,30
   DISPLACEMENT = ALL
   STRESS = 1
SUBCASE 1
   LOAD = 101
   OLOAD = ALL
SUBCASE 2
   LOAD = 102
   OLOAD = ALL
SUBCOM 51
   SUBSEQ = 1.0,1.0
SUBCOM 52
   SUBSEQ = 2.5,1.5
BEGIN BULK
```

consisting of 2.5 times **SUBCASE 1** plus 1.5 times **SUBCASE 2**. The displacements at all GRID points and the stresses for the elements defined in **SET 1** will be printed for all four SUBCASEs. In addition, non-zero components of the static load vectors will be printed for **SUBCASE 1** and **SUBCASE 2**.

10.3.3.4 Selecting Additional Solution Results

Sometimes you may find it useful to request different output for the same solution. To do this, the command:

```
REPCASE id
```

may be used. Again, *id* is a unique identification number for the SUBCASE. Example 3 illustrates how you may use this command. Here, one SUBCASE is defined for solution and two SUBCASEs for output control. The displacements at all GRID points and the non-zero 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**. Only the data recovery step will be performed for the **REPCASES**.

Example 3

```
SET 1 = 1 THRU 10
SET 2 = 21 THRU 30
SET 3 = 31 THRU 40
SUBCASE 1
   LOAD = 10
   SPC = 11
   DISP = ALL
   SPCF = ALL
   FORCE = 1
REPCASE 2
   FORCE = 2
REPCASE 3
   FORCE = 3
BEGIN BULK
```

10.3.3.5 Output Sets

The output request commands may be used to select only a portion of all of the solution results computed during the **UAI/*NASTRAN*** execution. This is done by defining one or more **SET**s of GRID point or element identification numbers. The general form of this command is:

```
SET sid   = { term1 [term2,...] }
            { ALL               }

      termi = id1 [ THRU id2 [ EXCEPT idlist ] ]
```

The **SET** is given an identification number, *sid*, followed by one or more *term*s. Each of the *term*s may specify a single integer value, *id1*, or an inclusive range of integer values:

```
id1 THRU id2
```

Finally, you may delete one or more of the elements of the inclusive range by using the **EXCEPT** clause. In this case, you enter an *idlist* which specifies the members of the **THRU** range to be excluded. You will find more details in Chapter 4 of the ***User's Reference Manual***.

Note that there is no distinction between a set of GRID points and a set of elements. The usage of the **SET** in output commands determines the meaning of the **SET**.

10.3.3.6 Multidisciplinary Analysis and Design

Statics analysis is one of the disciplines that may be used when performing Multidisciplinary Analysis and Design (MDO). Special constructs are available to define the staic loading conditions that are combined with other disciplines such as Normal Modes and Frequency Response analyses. See Chapter 25 of this manual for details.

10.3.4 Bulk Data

There are no Bulk Data entries which are specifically required for performing Static Analyses. Previous Chapters have described many of the entries required for defining loads, boundary conditions, finite elements, and all of the other components of the analysis model. One Bulk Data entry that is useful for Static Analyses is the **LOAD** entry:

LOAD	**LID**	**S**	**S1**	**LID1**	**S2**	**LID2**	**S3**	**LID3**	**-cont-**
-cont-	**S4**	**LID4**	***CONTINUES IN GROUPS OF 2***						

This entry defines a loading condition as a linear combination of other loads:

$$L = \mathrm{S}\sum_{i} \mathrm{Si} \bullet \mathrm{LIDi}$$

You must use this entry to combine a gravity load, defined by the **GRAV** Bulk Data entry, with any other loads. When you are performing Substructuring Analyses, the **LOADC** entry performs a similar function.

10

10.4 SOLUTION RESULTS

The primary results of a Static Analysis are the structural deformations and the stress state within each element. Table 10-2 summarizes the solution results that may be obtained and the Case Control commands used to request them. These results may be obtained for any selected sets of GRID points or elements. The following sections briefly describe the specific results that you will get when selecting these options.

10.4.1 GRID Point Solution Results

As indicated in Table 10-2, the GRID point output quantities include displacements and forces, selected with the Case Control commands **DISP** and **GPFORCE**. The force output contains, for each GRID point, all the externally applied loads, the forces calculated internally due to element reactions and single-point constraints, and the balanced total of all these forces. Forces from multipoint constraint equations and rigid elements are not included.

You may obtain reactions in the form of forces of single-point and multipoint constraints by using the **SPCF** and **MPCF** commands. Finally, to assist you in validating your model, you may request the applied loads at GRID points. All the load types which are not applied directly to the GRID points, such as gravity, pressure, or thermal loads, are processed by the program and replaced by equivalent loads applied at the GRID points.

The form of the output for all of these results is the same as that shown in Chapter 4. All values are output in the output coordinate system assigned to each GRID point in field 7 of **GRID** or **GRDSET** Bulk Data entries.

10.4.2 Static Equilibrium Summary

You may also request a static equilibrium summary for the entire model by using the **EQUILIBRIUM** Case Control command. This results in a summary of the total applied loads and forces in each of the basic coordinate directions. A sample of the resulting output is shown in Table 10-3.

10.4.3 Rigid Body Solution Summary

When you apply rigid body constraints to your model, you automatically get the Rigid Body Solution Summary shown in Table 10-4. The values in this table are described in Section 10.1.2.

Table 10-3. STATIC EQUILIBRIUM SUMMARY

E Q U I L I B R I U M C H E C K

RESULTANT LOADS IN BASIC COORDINATE SYSTEM

SUBCASE NO.	REFERENCE POINT	LOAD TYPE	T1	T2	T3	R1	R2	R3
1	ORIGIN	APP-LOAD	-3.669012E+02	-1.334959E+02	-9.296061E+02	4.696584E+00	-8.153752E+00	1.420677E-03
		F-OF-SPC	3.665487E+02	1.344192E+02	9.294927E+02	-1.080880E+02	-3.190502E+01	6.158063E+00
		F-OF-MPC	3.524538E-01	-9.232935E-01	1.134114E-01	1.033915E+02	4.005878E+01	-6.159484E+00
		TOTALS	-2.422929E-05	5.960464E-07	-5.295873E-05	7.629395E-06	0.000000E+00	-4.768372E-07

Table 10-4. Rigid Body Solution Summary

R I G I D B O D Y S O L U T I O N S U M M A R Y

POINT ID	TYPE	COMPONENT	EPSILON	XNORM	KRR NORM
1	GRID	1	1.70419E-14	2.01088E-19	6.92389E+08

THE TOTAL RIGID BODY SOLUTION ERROR IS 5.39218E-16

10.4.4 Element Solution Results

Table 10-2 also indicates the element solution results that are available from a Statics Analysis. The actual format and interpretation of these quantities in described in Chapter 5.

10.4.5 Plots

Two types of plotted output may be obtained for Statics Analysis. The first type consists of plots of all or portions of the structural model in either the deformed or undeformed configurations. The second type is an X-Y plot. Such plots may be of either a selected output quantity versus SUBCASE or one selected output quantity versus another for different SUBCASEs. Plotting is discussed in detail in Chapters 29 and 30.

10.4.6 Transient Initial Conditions

You may use the displacement results from your Statics Analysis as initial conditions when performing either Direct or Modal Transient Response Analyses. To do this, you add a special option to your **DISP** Case Control command:

```
DISP(... , TIC=ticid , ...) = sid
```

This requests that the resulting displacements in the **SET** specified by ***sid*** be written to the **BULK** file in the form of high precision **TIC** Bulk Data entries. You must **ASSIGN** the **BULK** file if you use this option. The identification number *ticid* is assigned to these entries. This initial condition data may then be included in your Transient Analysis data and selected with the Case Control command:

```
TIC = ticid
```

This page is intentionally blank.

Chapter 11

NORMAL MODES ANALYSIS

UAI/*NASTRAN* allows you to determine the free undamped natural frequencies and mode shapes of a finite element model using the Normal Modes Analysis discipline. Real Normal Modes analysis is related to Static Analysis in that only undamped, real, symmetric matrices are used. This chapter provides you with the methods and options available for performing these analyses and guidelines for their use.

Normal modes analyses can also be performed in various Rigid Formats as summarized in Table 11-1. using the differential stiffness approach, described in Chapter 17, and for models which take advantage of Cyclic Symmetry, as described in Chapter 22. If you are performing substructuring analyses, you may also perform Normal Modes analyses, as described in Chapter 19. You may design structures subject to frequency and mode shape constraints using the Multidisciplinary Analysis and Design Optimization capability detailed in Chapter 25. Finally, you may compute sensitivities of real eigenvalues and their mode shapes with respect to design variables that you define using the Design Sensitivity feature described in Chapter 26.

Table 11-1. SUMMARY OF MODES RIGID FORMATS

RIGID FORMAT	DESCRIPTION
3	Normal Modes Analysis
13	Normal Modes with Differential Stiffness
15	Normal Modes using Cyclic Symmetry
52	Design Sensitivity Analysis
MULTI	Multidisciplinary Analysis and Design

11.1 MATHEMATICAL BACKGROUND

This section describes the various mathematical approaches available in **UAI/*NASTRAN*** for solving Normal Modes Analysis.

11.1.1 The Eigenvalue Problem

Normal modes analysis is used to compute the free undamped vibration characteristics of the finite element model. The solution to this type of analysis is performed by solving the eigenvalue problem of the form:

$$[\,\boldsymbol{K} - \lambda\,\boldsymbol{M}\,]\,\boldsymbol{\Phi} = \boldsymbol{0} \tag{11-1}$$

where $\boldsymbol{K}$ is the stiffness matrix, $\boldsymbol{M}$ the mass matrix, λ an eigenvalue and Φ the eigenvector associated with λ. Each eigenvalue represents a natural frequency, ω, of the system where:

$$\omega = \sqrt{\lambda} \tag{11-2}$$

The solution of the eigenproblem has been the subject of intense study. There are several methods available in **UAI/*NASTRAN*** for solving (11-1). These are described in the following subsections.

11.1.2 Eigenvalue Extraction Methods

There are four basic eigenextraction methods. The first of these, called the ***Givens Method***, is used to extract all of the eigenvalues from a matrix. The next three, the ***Inverse Power Method***, the ***Lanczos Method***, and the ***Subspace Iteration Method***, are used to extract a portion of the eigenvalues from a specified frequency range. A brief description of these methods is given in the following sections.

11.1.2.1 The Givens Method

The Givens-Householder, or **`GIV`** method, is a tridiagonalization technique used for extracting all of the eigenvalues from the standard eigenproblem:

$$[\,\boldsymbol{A} - \lambda\,\boldsymbol{I}\,]\boldsymbol{z} = \boldsymbol{0} \tag{11-3}$$

where $\boldsymbol{A}$ is a positive definite matrix. A general overview of the steps performed by the Givens method is given in this section. First, a Cholesky factorization of the mass matrix, $\boldsymbol{M} = \boldsymbol{C}\,\boldsymbol{C}^T$, is performed. The left hand side of (11-1) is then transformed:

$$\boldsymbol{K}\Phi - \lambda\,\boldsymbol{C}\,\boldsymbol{C}^T\boldsymbol{\Phi} = \boldsymbol{0} \tag{11-4}$$

(11-4) is then premultiplied by $\boldsymbol{C}^{-1}$ yielding:

$$\boldsymbol{C}^{-1}\,\boldsymbol{K}\boldsymbol{\Phi} - \lambda\,\boldsymbol{C}^T\boldsymbol{\Phi} = \boldsymbol{0} \tag{11-5}$$

Then, making the substitution $\boldsymbol{z} = \boldsymbol{C}^T\boldsymbol{\Phi}$ and rearranging terms yields:

$$[\,\boldsymbol{A} - \lambda\,\boldsymbol{I}\,]\,\boldsymbol{z} = \boldsymbol{0} \tag{11-6}$$

which is the standard form with $\boldsymbol{A} = \boldsymbol{C}^T\boldsymbol{K}\boldsymbol{C}^{-T}$. The $\boldsymbol{A}$ matrix is then tridiagonalized using the Givens method. A modified Q-R algorithm is used to extract the eigenvalues from the tridiagonal system. Finally, the eigenvectors in a specified range are recovered using:

$$\boldsymbol{\Phi} = \boldsymbol{C}^{-T}\boldsymbol{z} \tag{11-7}$$

The Givens method extracts all eigenvalues beginning with the highest frequency and proceeding to the lowest frequency. This is efficient when the matrices are small enough to be held in memory. However, as problem size gets larger, the CPU requirements increase dramatically. In such cases, either the alternate Lanczos eigenmethod should be used, or one of the reduction methods, described in Section 11.1.3, should be selected.

11.1.2.2 The Modified Givens Method

A modification of the Givens method, known as Modified Givens, **MGIV**, is also available. In this method, the Cholesky factorization is performed on a combination of the mass and stiffness matrices, rather than on the mass matrix alone. This decomposition yields:

$$\left[\,\boldsymbol{K} + \lambda_s\,\boldsymbol{M}\,\right] = \boldsymbol{C}\,\boldsymbol{C}^T \tag{11-8}$$

where λ_s is the *shift* frequency. (11-1) is then transformed into:

$$\left[\,\boldsymbol{K} + \lambda_s\,\boldsymbol{M} - (\,\lambda + \lambda_s\,)\,\boldsymbol{M}\,\right]\boldsymbol{\Phi} = \boldsymbol{0} \tag{11-9}$$

Substituting (11-8) into (11-9), and premultiplying by $\boldsymbol{C}^{-1}$ and:

$$\lambda_n = \frac{1}{\lambda + \lambda_s} \tag{11-10}$$

yields:

$$\boldsymbol{C}^{-1}\,\boldsymbol{M}\,\boldsymbol{\Phi} - \lambda_n\,\boldsymbol{C}^T\boldsymbol{\Phi} = \boldsymbol{0} \tag{11-11}$$

Then, making the substitution $\boldsymbol{z} = \boldsymbol{C}^T\boldsymbol{\Phi}$, and rearranging the terms yields another standard form:

$$\left[\,\boldsymbol{A} - \lambda_n\,\boldsymbol{I}\,\right]\boldsymbol{z} = \boldsymbol{0} \tag{11-12}$$

In this case, $\boldsymbol{A} = \boldsymbol{C}^{-1}\,\boldsymbol{M}\,\boldsymbol{C}^{-T}$. The standard Givens formulation is then followed, and the real eigenvalues are obtained from (11-10).

Two of the advantages associated with this method are that the lowest frequency eigenvalues are computed with greater accuracy, and that the formulation may be used with singular mass matrices. This method produces very large eigenvalues for each massless degree of freedom.

You must remove the massless degrees of freedom using one of the available options, for example, **AUTOOMIT**, to avoid the computation of these eigenvalues.

11.1.2.3 The Lanczos Method

In comparison to the Inverse Power Method, described below, a much more robust and efficient method for obtaining the highest or lowest eigenvalues of a system is called the Lanczos Method. This method, developed in the 1950s, is an iterative technique, about which many papers and books have been written. A detailed technical description of this method is well beyond the scope of this manual. Those interested in general background should consider reading, for example, [Parlett80] and [Hughes87]. The **UAI/*NASTRAN*** algorithm is based on work with high-performance eigenextraction algorithms described in [Grimes86] and [Grimes91]. In general, this is the best method for extracting a small set of eigenvalues from a large, sparse system. See Section 11.4 for additional guidelines for using this method.

11.1.2.4 The Inverse Power Method

The inverse power method, or **SINV**, is a procedure which extracts the eigenvalues and eigenvectors in a specified range of interest. The general method of solution is first to shift the eigenvalue to the starting eigenvalue λ_0

$$\lambda = \lambda_0 + \Lambda \tag{11-13}$$

then the ***dynamic matrix*** , ***D***, is defined as:

$$\boldsymbol{D} = \left[\boldsymbol{K} - \lambda_0 \boldsymbol{M} \right] \tag{11-14}$$

and the following recurrance relation is used to iterate to the eigenvalue closest to λ_0:

$$\boldsymbol{w}_n = \boldsymbol{D}^{-1} \boldsymbol{M} \boldsymbol{u}_{n-1} \tag{11-15}$$

$$\boldsymbol{u}_n = \frac{1}{C_n} \boldsymbol{w}_n \tag{11-16}$$

where :

$$C_n = \max \left| \boldsymbol{w}_n \right| \tag{11-17}$$

and ***u*** is the trial vector which converges to the eigenvector. Because matrix decompositions are performed for each mode, the system matrices should be either well-banded or small to improve efficiency. The method does allow singularities in both the stiffness and mass matrix. However, the sum of these matrices must not be singular. It is often convenient to use the Inverse Power method when only a few eigenvalues are required because the input data for a static solution will require little additional data for normal modes analysis.

You must be aware of two disadvantages to this method. The first is the relatively high cost required to obtain a large number of modes. The second is the requirement that you provide an estimated frequency range and the number of desired roots in the range. This requires

some prior knowledge of the solution. If these values are overestimated, solution time will be greater than necessary. If they are underestimated, some of the desired modes may not be computed.

A ***Sturm Sequence Check*** is performed to insure that all eigenvalues in the requested range have been found. This is performed in the following manner. Assume that eigenvalues are desired in the open interval α to β. The following two factorizations are performed:

$$\boldsymbol{K}-\alpha\boldsymbol{M} = \boldsymbol{U}_{\alpha}^{T}\boldsymbol{D}_{\alpha}\boldsymbol{U}_{\alpha};\ \boldsymbol{K}-\beta\boldsymbol{M} = \boldsymbol{U}_{\beta}^{T}\boldsymbol{D}_{\beta}\boldsymbol{U}_{\beta} \tag{11-18}$$

The number of eigenvalues, $n_{\alpha\beta}$, in the interval is then given by:

$$n_{\alpha\beta} = n_{\beta} - n_{\alpha} \tag{11-19}$$

where n_{α} and n_{β} are the number of negative diagonal terms of $\boldsymbol{D}_{\alpha}$ and $\boldsymbol{D}_{\beta}$, respectively. If this computation indicates that not all eigenvalues have been found, then a new starting point is computed and any missing eigenvalues are extracted.

11.1.2.5 The Subspace Iteration Method

While the Lanczos Method is very robust, when either the stiffness or mass matrices are very dense, the efficiency of the method is reduced. The Subspace Iteration Method has been implemented to address this issue. It is especially critical when performing Fluid-Structure Interaction analyses which result in very dense mass matrices.

The Algorithm. The steps in the Subspace Iteration algorithm are:

1. Specify initial trial vectors $\boldsymbol{\Phi}$
2. Solve for improved vectors $\boldsymbol{\Psi}$ using:

$$\boldsymbol{K}\boldsymbol{\Psi} = \boldsymbol{M}\boldsymbol{\Phi} \tag{11-20}$$

3. Use Gram-Schmidt procedure to orthogonalize $\boldsymbol{\Psi}$ with respect to previously converged eigenvectors $\boldsymbol{\Phi}_a$:

$$\boldsymbol{\Psi} = \boldsymbol{\Psi} - \boldsymbol{\Phi}_a\boldsymbol{\Phi}_a^{T}\boldsymbol{M}\boldsymbol{\Psi} \tag{11-21}$$

4. Project the mass and stiffness matrices onto the subspace spanned by $\boldsymbol{\Psi}$:

$$\overline{\boldsymbol{M}} = \boldsymbol{\Psi}^{T}\boldsymbol{M}\boldsymbol{\Psi},\ \overline{\boldsymbol{K}} = \boldsymbol{\Psi}^{T}\boldsymbol{K}\boldsymbol{\Psi} \tag{11-22}$$

5. Solve the reduced eigenproblem:

$$[\,\overline{\boldsymbol{K}} - \lambda\overline{\boldsymbol{M}}\,]\,\overline{\boldsymbol{\Phi}} = \boldsymbol{0} \tag{11-23}$$

6. Compute the new approximate eigenvectors:

$$\boldsymbol{\Phi} = \boldsymbol{\Psi}\overline{\boldsymbol{\Phi}} \tag{11-24}$$

7. Perform convergence tests and move converged eigenvectors from $\boldsymbol{\Psi}$ to $\boldsymbol{\Phi}_a$. If all eigenvectors have converged, perform a shift and go to Step 1 to start a new block, otherwise, go to

Step 2 and continue iterating.

Subspace Dimension. The number of eigenvectors in Φ is called the ***Dimension*** of the subspace. For computational efficiency, this dimension should be small. In **UAI/*NASTRAN***, the default, called **MAXDIM**, is 60. This allows about 30 eigenvalues to be extracted from each block. The mass matrix is then shifted, as described below, and the eigenvalues extracted from a new block. This process continues until all requested eigenvalues and eigenvectors have been obtained.

Extracting Zero Eigenvalues. The handling of zero eigenvalues, such as those which represent rigid body modes and the constant pressure mode of a fluid, requires special treatment. Without such treatment, all initial trial vectors would converge to the zero values and flexible modes would be overlooked.

The constant pressure mode of a fluid is extracted first using a subspace of dimension one. All subsequent modes are then orthogonalized against this mode.

If your model contains multiple isolated compartments, then additional zero-frequency constant pressure modes will exist. These will be treated properly if you specify the number of fluid compartments or tanks using the **NTANKS** field of the **FSIDATA** Bulk Data entry. See Chapter 23 for additional details.

Convergence. The convergence of the subspace iteration is measured simply by comparing each of the eigenvalues for successive iterations:

$$\varepsilon = \left| \frac{\lambda_i - \lambda_{i-1}}{\lambda_i} \right| < 10^{-8} \qquad (11\text{-}26)$$

where i is the current iteration. The smaller eigenvalues converge quickly to the default value of 10^{-8}. The higher eigenvalues take longer. You may control the convergence criteria with the **E** field of the **EIGR** Bulk Data entry.

Diagnosing Problems. It is possible that there will be specific models for which the subspace iteration method will not converge properly. If you encounter such a problem there are several approaches that you may take. First, you may include the **DIAG 16** option in your Executive Control packet. This will activate a special output called the Subspace Iteration Summary Table. This table provides a detailed status of the extraction process. Also note that this table is automatically printed when convergence difficulties occur.

One of the most probable difficulties that may be noted is that the approximate eigenvectors quickly converge to the lower eigenvalues such that some of the vectors become linearly dependent. This leads to a singular reduced mass matrix. You may correct this problem in one of three ways:

- Relax the convergence criteria, **E**.
- When performing Fluid-Structure Interaction analyses, use a larger value of **FSIEPS** on Bulk Data entry **FSIDATA**.

11.1.3 Reduction Methods

The CPU time required to solve eigenvalue problems is proportional to the cube of the size of the matrices involved. For models encountered in *industrial-strength* analyses, an eigenextrac-

tion on the full *g-size* system of equations, even using the Lanczos or Inverse power methods, may be prohibitively expensive. As you saw in Chapter 9, **UAI/*NASTRAN*** provides you with several reduction techniques which may be used to approximate the solution to the problem by solving a much smaller set of equations. These methods are discussed in the sections below.

11.1.3.1 Static Condensation

The oldest and most frequently used method for reducing the size of a normal modes analysis is the ***static condensation*** or Guyan reduction. Recalling from Chapter 6, this method first partitions the *f-set* degrees of freedom into the *a-set*, or retained degrees of freedom, and the *o-set*, or omitted degrees of freedom for both the stiffness and mass matrices. As you will see in Section 11.3, this may be done in several different ways.

11.1.3.2 Dynamic Reduction

The newer method for improving the efficiency of normal modes analyses is the Dynamic Reduction, also described in Chapter 6. Dynamic reduction is a technique by which the equations of motion are represented by generalized coordinates. When you request a dynamic reduction, you specify either a frequency range or the number of eignevectors that you wish to use for generalized coordinates. Remember that you may include *a-set* degrees of freedom with your generalized coordinates. You must retain some *a-set* degrees of freedom if you are performing substructuring analyses.

11.1.3.3 Rigid Body Modes

When extracting normal modes from an unconstrained model, or a model which includes mechanisms, all of the eigenmethods extract a number of near-zero frequency modes equal to the number of rigid body degrees of freedom. You may use Rigid Body supports, which are defined using **SUPORT** Bulk Data entries, to set the corresponding eigenvalues to zero, and to have the rigid body mode shapes recomputed with respect to the supported degrees of freedom. This is, however, not a requirement.

11.1.3.4 Normal Modes for Modal Reduction

While of importance in its own right, Normal Modes analysis is often performed to compute eigenvectors which will be used to create a reduced dynamic model. The reduced model is formed using generalized degrees of freedom, some of which are the eigenvectors. The cost and quality of the resultant modal analysis (e.g., Modal Frequency Response) is highly dependent upon the number and quality of the normal modes used in the reduced model. **UAI/*NASTRAN*** provides a number of tools to assist the analyst in choosing a good set of normal modes. A good set of modes is small enough to be cost effective, but robust enough to capture the important dynamic behavior of the structure.

One measure of how well a set of normal modes will represent the dynamics of the system is to compute how much of the rigid body mass can be represented by the flexible modes. This is termed the ***modal effective mass*** of the system. The computation of the modal effective mass for the extracted normal modes may be selected as an output of any modal analysis using the Case Control Command **MEFFMASS**.

If one has defined a set of six rigid body displacement vectors, $\boldsymbol{D}_m$, about some point, then the rigid body mass of the structure is defined as:

$$\text{Rigid Body Mass} \equiv \boldsymbol{D}_{mg}^T \boldsymbol{M}_{gg} \boldsymbol{D}_{mg} \tag{11-27}$$

If we partition the rigid body vectors and the mass matrix to the physical analysis, or *a-set*, degrees of freedom, the same expression still holds, but there may be fewer rigid body modes. For purposes of these computations, it is equivalent to use null vectors for those unrepresented *a-set* rigid body modes.

$$\text{Rigid Body Mass} \equiv \boldsymbol{D}_{m_a}^T \boldsymbol{M}_{aa} \boldsymbol{D}_{m_a} \tag{11-28}$$

The rigid body modes may then be approximated as a linear combination of the normal modes, $\boldsymbol{\Phi}$,

$$\boldsymbol{D}_{m_a} \equiv \boldsymbol{\Phi}_a \boldsymbol{\varepsilon} \tag{11-29}$$

and solved for the participation, $\boldsymbol{\varepsilon}$, of each mode in the rigid body vector by considering the identity:

$$\boldsymbol{D}_{m_a} - \boldsymbol{\Phi}_a \boldsymbol{\varepsilon} \equiv 0 \tag{11-30}$$

which yields:

$$\boldsymbol{\varepsilon} = \frac{\boldsymbol{\Phi}_a^T \boldsymbol{M}_{aa} \boldsymbol{D}_{ma}}{\boldsymbol{\Phi}_a^T \boldsymbol{M}_{aa} \boldsymbol{\Phi}_a} \tag{11-31}$$

Given the total rigid body mass from (11-28), and the modal participation in each rigid body vector, the modal effective mass may be computed as:

$$\left[\boldsymbol{\Phi}_a \varepsilon \right]^T \boldsymbol{M}_{aa} \left[\boldsymbol{\Phi}_a \varepsilon \right] \equiv \boldsymbol{\varepsilon}^T \boldsymbol{\Phi}_a^T \boldsymbol{M}_{aa} \boldsymbol{\Phi}_a \boldsymbol{\varepsilon} \tag{11-32}$$

where the rigid body modes that cannot be represented in the *a-set* are assigned zero participation factors.

11.1.4 Energy Measures

Three additional solution quantities are available in Normal Modes analysis that can provide important information about the behavior of the model. The first is the Element Strain Energy output. It is available for a number of **UAI/*NASTRAN*** output disciplines and is computed for all of the metric and scalar elements, *j*, as:

$$E_{s_j} = \frac{1}{2} \boldsymbol{\Phi}^T \boldsymbol{K}_{ee_j} \boldsymbol{\Phi} \tag{11-33}$$

The second and third output options are only available in Normal Modes analysis, but are closely related to the strain energy in interpretation. These are the Element Kinetic Energy and the Grid Point Kinetic Energy. The Element Kinetic Energy, which is the mass matrix counterpart to the Element Strain Energy, is computed for the *jth* element as:

$$E_{k_j} = \frac{1}{2}\omega\,\Phi^T M_{ee_j}\,\omega\,\Phi = \frac{\lambda}{2}\Phi^T M_{ee_j}\Phi \qquad (11\text{-}34)$$

Notice that the choice of scaling of the kinetic energy results in the same total kinetic energy in a mode as total strain energy. This is natural, since the system is conservative and, under harmonic oscillation, energy is exchanged among these two forms.

The distribution of kinetic and strain energy among the elements, however, is different. Element Strain Energy is concentrated in elements with high stiffness in regions of high amplitude, while Element Kinetic Energy is concentrated in high mass elements in regions of high amplitude. The element strain energy, therefore, is useful in identifying the elements whose stiffnesses control the modal stiffness while the element kinetic energy is useful in identifying the elements whose masses control the modal mass. Thus, the dominant, controlling elements of your model for each normal mode may be identified from the **ESE** and **EKE** outputs.

For models using the lumped mass formulation, the Grid Point Kinetic energy can be used to examine the distribution of kinetic energy among the grid points. It is computed as:

$$E_{k_g} = \Phi_g^{mass} \otimes \left[M_{gg}\Phi_g^{mass} \right] \qquad (11\text{-}35)$$

Where Φ_g^{mass} indicates that the eigenvectors have been normalized using the **MASS** scaling option so that the total grid point kinetic energy is scaled to be unity. Note that the operator $\otimes$ indicates termwise matrix multiplication

The Grid Point Kinetic Energy output has limited meaning for a ***coupled*** mass formulation just as the Grid Point Strain Energy has limited meaning for metric elements. Since this mass formulation produces a coupling of mass across grid points, the sharing of kinetic energy among grid points can occur. In general, this obscures the meaning of the computation as a means of identifying important model parameters to control modal behavior.

11.2 INPUT DATA REQUIREMENTS

Each of the **UAI/*NASTRAN*** Input Data Packets contains commands and data which are specific to the Normal Modes analysis discipline. These are discussed in this section.

11.2.1 Executive Control Commands

Other than the selection of the Normal Modes Rigid Format no special Executive Control commands are required.

11.2.2 Case Control Commands

There are a number of Case Control commands which are used for Normal Modes analysis. A description of how you use these is found in the sections below.

11.2.2.1 Selecting the Extraction Method

You must select an eigenvalue extraction method with the Case Control command:

```
METHOD = sid
```

where *sid* is the identification number of an **EIGR** Bulk Data entry.

11.2.2.2 Automatic Static Condensation

The **AUTOOMIT** option is automatically invoked when you use the **GIV** eigenextraction method. This is done to remove mass singularities that are present due to the finite element formulations. If you wish to use any of the other **AUTOOMIT** options, then you use the command:

```
AUTOOMIT [ ( [ PRINT ] [ EPS = x ] [ PUNCH [ = { OMIT } ] ] ) ] = { YES }
         [ ( [ NOPRINT ] [ MASS = y ]          { ASET }       )   { NO  }
         [ ( [ NOZERO ] [ KEEP = z ]                          )   { OSET }
```

Table 11-2. SELECTING NORMAL MODES CONTROL DATA

TO SELECT:	CASE CONTROL IS:	BULK DATA ENTRIES USED ARE:
Eigenvalue Method	**METHOD**	**EIGR**
Output	**OMODES** **MEFFMASS** **EKE** **GPKE**	
Consistent Mass		**PARAM , COUPMASS , 1**
Static Condensation		**ASET** **OMIT**
Automatic Static Condensation	**AUTOOMIT**	
Dynamic Reduction	**DYNRED**	**DYNRED**
Rigid Body Supports		**SUPORT**

The **EPS**, **MASS** and **KEEP** options allow you to qualify the degrees of freedom in your model for reduction. The value *x* specifies a value of stiffness-to-mass ratio which defines those degrees of freedom that will be removed from the *a-set* using the rule:

$$\text{If } \frac{\boldsymbol{K}_{ii}}{\boldsymbol{M}_{ii}} > x, \text{ then place } u_i \text{ in the } o\text{-}set$$

Similarly, the **MASS** option allows you to reduce degrees of freedom whose mass in inconsequential to the dynamic charactersitcis of your model. The following rule is used:

$$\text{If } \boldsymbol{M}_{ii} < y, \text{ then place } u_i \text{ in the } o\text{-}set$$

Finally, and most useful, is the **KEEP** option. You use this option to select a specified percentage of the degrees of freedom in the model which represent the major dynamic contributions. The value *z* represents a decimal percentage, i.e. it has values ranging from **0.0** to **100.0**.

11.2.2.3 Dynamic Reduction

The Dynamic Reduction feature is selected by placing the command:

```
DYNRED = sid
```

in your Case Control Packet. The *sid* references a **DYNRED** Bulk Data entry.

11.2.2.4 Modal Effective Mass

You request the Modal Effective Mass calculation with the command:

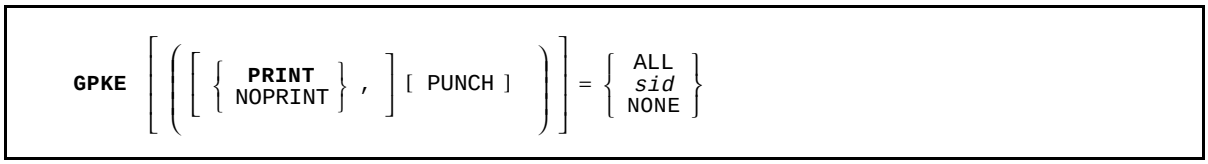
```
MEFFMASS [ ( [ { PRINT } , ] [ PUNCH , ] [ GRID = gid , ] [ THRESH = ε , ]
           [ { NOPRINT }   ]
                                  [ { MASS   } ] ) ] [ = { NONE    } ]
                                  [ { WEIGHT } ]       { SUMMARY }
                                                       { FULL    }
                                                       { DETAIL  }
```

11.2.2.5 Energy Measures

Each of the three available energy measures is selected with a special Case Control command. Element Strain Energy and Kinetic Energy are requested with the command:

```
ESE  [ ( [ { PRINT } , ] [ PUNCH ,] [ THRESH = x ] ) ] = { ALL }
EKE  [     { NOPRINT }                                   { sid }
                                                         { NONE }
```

where you may select **ALL** elements or a subset of elements defined by *sid*. In both cases, you may set a **THRESH**hold value below which values will not be printed. The Grid Point Kinetic Energy is requested using:



where you may again select **ALL** elements or a subset of elements defined by *sid*.

11

11.2.3 Bulk Data Entries

In addition to the geometric modeling data, normal modes analysis has several additional requirements. Data must be available to generate a mass matrix. As discussed earlier, this is done either by defining the density on at least one of the material property Bulk Data entries or by defining concentrated or scalar masses, or by providing non-structural mass values on the element property Bulk Data entries. If these data are defined in weight units, then you ***must*** use the **PARAM** Bulk Data entry defining **WTMASS**, which is used to convert to mass units. It is also recommended that a **SUPORT** Bulk Data entry be used if there are free body modes.

A summary of the Case Control commands and Bulk Data entries used for Normal Modes analysis is given in Table 11-2. Required commands and Bulk Data entries are indicated by the shaded entries.

11.2.3.1 Eigenextraction Data

You specify eigenextraction data using the **EIGR** Bulk Data entry. The form of this entry varies slightly depending on the method that you select. Each form of data is described below.

Lanczos. The Lanczos method has input which is the same as that of the Givens methods:

EIGR	SID	LANCZOS	FL	FU		NVEC			-cont-
-cont-	NORM	GID	DOF						

However, there is a slightly different interpretation of the fields. If the system being solved has rigid body modes, then the **FL** field should be set to **0.0** in order to compute them. You should also leave the **NVEC** field blank in such cases. Finally, when you do not specify an upper limit on the frequency, **FU**, and you do not specify the number of eigenvectors to compute, **NVEC**, you will obtain only one eigenvector.

Givens and Modified Givens. The input data for the Givens methods is the simplest. The **EIGR** data entry is:

EIGR	SID	GIV MGIV	FL	FU		NVEC			-cont-
-cont-	NORM	GID	DOF						

The **SID** is an identification number used to reference this data from the Case Control command **METHOD**. You then specify a range of frequencies within which you want eigenvectors computed. Note that ***all*** eigenvalues are always computed. You may then specify the number of eigenvectors that will be printed using **NVEC**. Finally, you may select from several **NORM**alization options for the resulting eigenvectors. You may select **MAX**, in which case eigenvectors are normalized to their maximum component displacement, **MASS**, in which case the eigenvectors are normalized to a unit value of generalized mass, or **POINT**, which requests that the eigenvector be normalized with respect to a specific **DOF** of a GRID point given by **GID**.

Inverse Power. This method has the same basic data requirements as above, but an additional field, **NEST**, has been added:

EIGR	SID	SINV	FL	FU	NEST	NVEC			-cont-
-cont-	NORM	GID	DOF						

The **NEST** field, which specifies the number of estimated roots in the range **FL** to **FU**, is required. All other fields are as for the Givens methods.

Subspace Iteration. The iterative nature of this method requires more input data and allows more options than the other methods. The Bulk Data entry is:

EIGR	SID	SUBS	FL	FU		NVEC	MAXDIM		-cont-
-cont-	NORM	GID	DOF						

The **MAXDIM** field is described in Section 11.1.2.5.

11.2.3.2 Reduction Data

If you are performing static condensation, then you may proceed in either of two ways. You may either define explicitly all of the degrees of freedom which you wish to retain in the *a-set* by using **ASET** Bulk Data entries, or you may explicitly define all of the degrees of freedom that you wish to place in the *o-set* with **OMIT** Bulk Data entries. Neither the **ASET** nor **OMIT** data are referenced by a Case Control command — their presence in the data stream causes the reduction to be performed.

When requesting a Dynamic Reduction, you must enter a **DYNRED** Bulk Data entry. The simplest form of this entry is:

DYNRED	SID	FMAX							

where **SID** is the identification number that you selected previously with the **DYNRED** Case Control command and **FMAX** specifies the maximum frequency in which you are interested.

11.3 SOLUTION RESULTS

The following sections describe the optional solution results that you may obtain when performing Normal Modes analysis, either as a single analysis, or as a prelude to solving Transient and Frequency Response analyses by the Modal Method.

11

11.3.1 The Eigensolution

The solution results of a normal modes analysis include a summary of the eigenvalues extracted and extraction information. These results, shown in Tables 11-3, are provided automatically. As you will note, the summaries are slightly different depending on the eigenvalue method. The eigenvalue summary is self-explanatory. The extraction summary provides information relative to the calculations performed and their status. The table also contains a message describing the termination status for the eigenvalue calculations. The engineering results available from the Normal Modes extraction are summarized in Table 11-4. The eigenvectors themselves are printed in the same format as static deformations. Each eigenvector includes

Table 11-3. EIGENEXTRACTION SUMMARY

a. Full Spectrum Eigenextraction

```
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
                ( GIVENS METHOD )

 NUMBER OF EIGENVALUES EXTRACTED . . . . . .       120
 NUMBER OF EIGENVECTORS COMPUTED . . . . . .         1
 NUMBER OF EIGENVALUE CONVERGENCE FAILURES . .       0
 NUMBER OF EIGENVECTOR CONVERGENCE FAILURES. .       0
 LARGEST OFF-DIAGONAL MODAL MASS TERM. . . . .  0.00E+00
                                . . .         0
          MODE PAIR. . . . . .
                                . . .         0
 NUMBER OF OFF-DIAG0NAL MODAL MASS
      TERMS FAILING CRITERION. . . . . .       0
           . . . REASON FOR TERMINATION . . .
                 NORMAL TERMINATION
```

b. Selective Eigenextraction

```
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 METHOD )

NUMBER OF EIGENVALUES EXTRACTED . . . . . .         6
NUMBER OF FACTORIZATIONS. . . . . .         1
NUMBER OF LANCZOS RUNS  . . . . . .         1
NUMBER OF LANCZOS STEPS . . . . . .         4
NUMBER OF BLOCK SOLVES. . . . . .         7
LARGEST OFF-DIAGONAL MODAL MASS TERM. . . . .  1.49E-15
                                . . .         5
          MODE PAIR. . . . . .
                                . . .         1
NUMBER OF OFF-DIAG0NAL MODAL MASS
     TERMS FAILING CRITERION. . . . . .         0

      . . . REASON FOR TERMINATION . . .
    ALL ROOTS IN THE REQUESTED RANGE FOUND
```

Table 11-4. ACTUAL EIGENVALUE SOLUTION RESULTS

R E A L E I G E N V A L U E S

MODE NO.	EXTRACTION ORDER	EIGENVALUE	RADIAN FREQUENCY	CYCLIC FREQUENCY	GENERALIZED MASS	GENERALIZED STIFFNESS
1	120	2.753958E+07	5.247817E+03	8.352160E+02	3.652545E-05	1.005895E+03
2	119	4.583932E+08	2.141012E+04	3.407527E+03	0.000000E+00	0.000000E+00
3	118	1.041507E+09	3.227238E+04	5.136309E+03	0.000000E+00	0.000000E+00
4	80	1.950819E+09	4.416807E+04	7.029566E+03	0.000000E+00	0.000000E+00
5	117	4.816505E+09	6.940105E+04	1.104552E+04	0.000000E+00	0.000000E+00
6	116	8.008242E+09	8.948878E+04	1.424258E+04	0.000000E+00	0.000000E+00
7	115	1.530027E+10	1.236943E+05	1.968655E+04	0.000000E+00	0.000000E+00
8	114	1.742490E+10	1.320034E+05	2.100899E+04	0.000000E+00	0.000000E+00
9	113	2.516920E+10	1.586480E+05	2.524962E+04	0.000000E+00	0.000000E+00
10	79	2.553930E+10	1.598102E+05	2.543458E+04	0.000000E+00	0.000000E+00
i		λ_i	$\omega = \sqrt{\lambda_i}$	$f = \frac{\sqrt{\lambda_i}}{2\pi}$	$\varphi_i^T \boldsymbol{M} \varphi_i$	$\varphi_i^T \boldsymbol{K} \varphi_i$

Generalized Mass and Stiffness are only calculated if the associated eigenvector is requested

heading information which gives its corresponding eigenvalue.

11.3.2 Modal Effective Mass

If you select the Modal Effective Mass option, then you may request a number of different output reports. In the **SUMMARY** output, Table 11-5a, the fraction of the total rigid body mass that is represented in the modal approximation of each rigid body vector is displayed. Note that these data are fractions of the *a-set* rigid body mass matrix, not the *g-set* rigid body mass matrix. A zero fraction implies that either the rigid body vector is not represented in the *a-set* OR that the normal modes cannot represent any of the rigid body mass. A **FULL** listing will indicate the difference

At the **FULL** output level, the additional items shown in Table 11-5b are displayed. Included are the modal and *a-set* rigid body mass matrices and the fraction of the mass contained in each normal mode for each of the rigid body directions (**T1**,**T2**,**T3**,**R1**,**R2**,**R3**). Note that the translational data are given first followed by the rotational data. For the null *a-set* rigid body vectors, the a-set mass matrix will have a zero diagonal.

Finally, at the **DETAIL** output level, Table 11-5c, the ε values of (11-31) themselves are printed for each rigid body vector. These modal participation factors show the contribution of each normal mode in the approximation of the rigid body vector.

11.3.2.1 Modal Effective Mass Interpretation

Equation (11-28), is the same equation used to compute the Grid Point Weight balance except that, for the modal effective mass, the computation is done in the structural *a-set* rather than the *g-set*. The modal effective mass is then compared to the *a-set* rigid body mass rather than the *g-set* "Grid Point Weight" data. The modal effective mass fractions compare the ***diagonals*** of the rigid body mass matrix and its modal approximation. Off-diagonal mass matrix terms are available from the **FULL** print option (Table 11-5b), but their meaning is less clear.

Comparing the *a-set* rigid body mass to the *g-set* rigid body mass (obtained by requesting **PARAM,GRDPNT**) is useful in that it measures the amount of structural mass that has been *removed* from the system by boundary condition specifications: e.g., SPC and MPC. In general, these two rigid body mass measures will be different, but a good modeling check is to assure yourself that the amount of mass removed is reasonable for the selected boundary conditions.

In evaluating the set of modes for a modal analysis, you want to ensure that the selected normal modes are able to capture nearly all the rigid body mass of the system. Thus, the total modal effective mass fraction should be high for a good modal model. Be aware that, if the set of normal modes includes rigid body modes, very high percentages of modal effective mass will result. This does not, however, measure the quality of the flexible modes to represent mass. To evaluate the effectiveness of the flexible modes, the rigid body modes should be excluded from Φ_a for the purposes of the modal effective mass calculation. These modes can be excluded using the controls previously discussed on the **EIGR** Bulk Data entry.

Non-participating modes and the dominant modes can be identified by looking at the modal participation factors, themselves, or by looking at the modal effective mass fractions. Using these data, you determine a cutoff frequency for the modes used in the modal analysis. This cutoff frequency ensures good mass representation, but does not ensure a good modal model: the flexibility, damping and load approximations must also be checked.

The modal quality of these latter approximations can best be examined by comparision of direct and modal results and by using the strain energy checks of the residual flexibility vectors. Using the **MEFFMASS** tools to eliminate mass as the source of mismatched results, the

11

Table 11-5. MODAL EFFECTIVE MASS RESULTS

a. SUMMARY Output Option

```
                    M O D A L   E F F E C T I V E   M A S S   S U M M A R Y


                                TOTAL EFFECTIVE MASS FRACTION
                          REFERENCE POINT AT ORIGIN OF BASIC COORDINATE SYSTEM

     T1              T2              T3              R1              R2              R3
5.836014E-01    8.527310E-01    7.044539E-01    6.272562E-01    4.644172E-01    7.486679E-01
```

b. FULL Output Option

```
                                    MODAL EFFECTIVE MASS MATRIX
                *** *** 
                *   1.580238E-01 -1.261771E-04  2.364811E-02  5.610851E-03 -2.739170E+01 -7.502327E-01 *
                *  -1.261771E-04  2.308970E-01  2.073220E-04 -1.327601E+02 -9.388702E-01  6.201858E+02 *
                *   2.364811E-02  2.073220E-04  1.907475E-01 -1.673950E-01 -4.441074E+02  7.206182E-01 *
                *   5.610851E-03 -1.327601E+02 -1.673950E-01  1.471768E+05  3.702179E+02 -3.760451E+05 *
                *  -2.739170E+01 -9.388702E-01 -4.441074E+02  3.702179E+02  1.324823E+06 -3.633177E+03 *
                *  -7.502327E-01  6.201858E+02  7.206182E-01 -3.760451E+05 -3.633177E+03  2.083108E+06 *
                *** *** 

                                    A-SET RIGID BODY MASS MATRIX
                *** *** 
                *   2.707736E-01 -2.462711E-19 -5.548791E-19  7.058765E-16  1.779568E+02  4.498776E-02 *
                *  -1.498690E-19  2.707736E-01  2.531309E-19 -1.779568E+02 -1.263162E-15  7.814799E+02 *
                *  -9.651793E-19 -5.451157E-20  2.707736E-01 -4.498776E-02 -7.814799E+02 -5.886787E-16 *
                *   7.153351E-16 -1.779568E+02 -4.498776E-02  2.346360E+05 -1.152946E+02 -5.450884E+05 *
                *   1.779568E+02 -2.130701E-16 -7.814799E+02 -1.152946E+02  2.852657E+06 -1.353049E+01 *
                *   4.498776E-02  7.814799E+02  6.435533E-16 -5.450884E+05 -1.353049E+01  2.782419E+06 *
                *** *** 
```

MODAL EFFECTIVE MASS FRACTION

MODE NO.	FREQUENCY HZ.	T1 FRACTION	T1 SUM	T2 FRACTION	T2 SUM	T3 FRACTION	T3 SUM
1	1.149278E-05	3.945056E-02	3.945056E-02	6.577788E-24	6.577788E-24	3.614572E-01	3.614572E-01
2	1.747293E+01	7.323661E-07	3.945129E-02	5.130601E-01	5.130601E-01	8.991412E-09	3.614572E-01
3	2.824589E+01	3.146978E-04	3.976599E-02	2.237894E-02	5.354390E-01	4.814039E-04	3.619386E-01
4	2.918111E+01	3.360684E-02	7.337283E-02	2.916631E-03	5.383556E-01	4.492372E-02	4.068623E-01
...							
47	9.854612E+01	2.337513E-07	5.835921E-01	2.442804E-03	8.524774E-01	2.835582E-05	7.032925E-01
48	9.951750E+01	3.518641E-06	5.835956E-01	1.772431E-04	8.526546E-01	8.231231E-04	7.041157E-01
49	9.986485E+01	1.040807E-07	5.835957E-01	1.938671E-07	8.526548E-01	1.706537E-04	7.042863E-01
50	1.001228E+02	5.719534E-06	5.836014E-01	7.613994E-05	8.527310E-01	1.675404E-04	7.044539E-01
...							

MODAL EFFECTIVE MASS FRACTION

MODE NO.	FREQUENCY HZ.	R1 FRACTION	R1 SUM	R2 FRACTION	R2 SUM	R3 FRACTION	R3 SUM
1	1.149278E-05	4.522770E-07	4.522770E-07	6.615982E-02	6.615982E-02	1.532791E-09	1.532791E-09
2	1.747293E+01	3.533681E-02	3.533727E-02	2.215706E-08	6.615984E-02	1.875996E-01	1.875996E-01
3	2.824589E+01	1.261576E-02	4.795302E-02	4.427849E-04	6.660263E-02	1.865413E-01	3.741409E-01
4	2.918111E+01	2.645272E-05	4.797947E-02	3.983771E-02	1.064404E-01	1.242922E-02	3.865701E-01
...							
47	9.854612E+01	1.692485E-04	6.268925E-01	4.330769E-05	4.627081E-01	4.054772E-03	7.484033E-01
48	9.951750E+01	2.542573E-04	6.271468E-01	1.241290E-03	4.639494E-01	1.229044E-04	7.485262E-01
49	9.986485E+01	6.248745E-05	6.272092E-01	2.458613E-04	4.641953E-01	1.012869E-05	7.485363E-01
50	1.001228E+02	4.690928E-05	6.272562E-01	2.219682E-04	4.644172E-01	1.315671E-04	7.486679E-01

c. DETAIL Output Option

MODAL PARTICIPATION FACTORS

MODE NO.	FREQUENCY HZ.	T1	T2	T3	R1	R2	R3
1	1.149278E-05	-4.000455E-01	5.165623E-12	1.210908E+00	-1.260896E+00	-1.681518E+03	-2.527740E-01
2	1.747293E+01	1.722499E-03	1.441713E+00	1.908573E-04	-3.522108E+02	9.724617E-01	2.794595E+03
3	2.824589E+01	1.312328E-01	1.106662E+00	1.623119E-01	7.734749E+02	-5.052584E+02	1.024215E+04
4	2.918111E+01	5.661876E+00	-1.667966E+00	6.546126E+00	1.478686E+02	-2.000854E+04	-1.103765E+04
...							
47	9.854612E+01	1.250464E-02	-1.278317E+00	-1.377258E-01	3.132210E+02	5.524570E+02	-5.279414E+03
48	9.951750E+01	3.630259E-02	2.576527E-01	-5.552423E-01	-2.872639E+02	2.213137E+03	6.877680E+02
49	9.986485E+01	9.831063E-03	-1.341737E-02	3.980827E-01	-2.242363E+02	-1.550894E+03	-3.108853E+02
50	1.001228E+02	8.468013E-02	-3.089637E-01	4.583116E-01	-2.257484E+02	-1.712254E+03	-1.301916E+03

modal analysis quality can then be improved by augmenting the generalized degrees of freedom (Residual Flexibility) or by including additional modes for the purposes of correlating stiffness, damping or load.

11.3.3 Energy Measures

The results of the energy solution requests **EKE** and **ESE** are shown in Table 11-6a. Both results have the same tabular presentation. The results first present the total energy for all elements selected for output. This is followed by results for each type of finite element. For each selected element, the amount of energy, and its corresponding percentage of the total energy, is shown. The kinetic or strain energy density, defined as the energy value divided by the element volume, is then given. You may use this density to adjust the interpretation of the relative importance of large elements to the overall energy.

The results of the Grid Point Kinetic Energy request **GPKE**, shown in Table 11-6b, has the same format as a displacement vector. However, the values represent the relative amount of kinetic energy in each degree of freedom of each GRID point. Note that the values are scaled such that the total kinetic energy is 1.0 (see 11-35) and the printed values are further scaled to be a percentage of the total.

Table 11-6. ENERGY SOLUTION RESULTS

a. Element Strain and Kinetic Energy

```
MODE NUMBER =    5                        FREQUENCY =  2.993804E+01 HZ.                    EIGENVALUE =  3.538396E+04

                                  E L E M E N T   K I N E T I C   E N E R G I E S

                                                                                  *
                         ELEMENT-TYPE = BEAM              TOTAL FOR ALL ELEMENTS =    1.7661719E+04
                         EKE SET ID   = ALL
                                                                                        *
                               ELEMENT-ID         KINETIC-ENERGY           PERCENT OF TOTAL   KINETIC-ENERGY-DENSITY
                                     ...
                                     817          7.747584E+00                 .0439              1.834558E-04
                                    1175          1.360905E+00                 .0077              3.174513E-05
                                    1177          7.220745E-01                 .0041              1.549538E-05

...
MODE NUMBER =    5                        FREQUENCY =  2.993804E+01 HZ.                    EIGENVALUE =  3.538396E+04

                                  E L E M E N T   K I N E T I C   E N E R G I E S

                                                                                  *
                         ELEMENT-TYPE = QUAD4             TOTAL FOR ALL ELEMENTS =    1.7661719E+04
                         EKE SET ID   = ALL
                                                                                        *
                               ELEMENT-ID         KINETIC-ENERGY           PERCENT OF TOTAL   KINETIC-ENERGY-DENSITY
                                     ...
                                     351          1.671826E+00                 .0095              1.390025E-04
                                     363          1.456006E+00                 .0082              1.210583E-04
                                     366          2.928799E+00                 .0166              1.326005E-04
                                     367          1.813847E+00                 .0103              1.303256E-04
                                     376          2.324807E+00                 .0132              1.138313E-04
```

b. Grid Point Kinetic Energy

```
MODE NUMBER =    5                        FREQUENCY =  2.993804E+01 HZ.                    EIGENVALUE =  3.538396E+04
                                G R I D   P O I N T   K I N E T I C   E N E R G Y,   P E R C E N T

POINT ID.   TYPE          T1             T2             T3             R1             R2             R3
     294    GRID     5.331688E-02   8.361877E-04   3.666900E-03   6.454621E-05  -3.090116E-03   2.706597E-06
     305    GRID     4.445497E-02   9.298442E-04   2.424846E-03   1.468836E-05  -2.981085E-03  -1.996849E-04
     346    GRID     3.210107E-03   8.699879E-05   8.923488E-04  -1.850959E-06  -1.945476E-04  -1.771267E-05
     447    GRID     1.175723E-03   3.989143E-05   3.984725E-03    .0             .0             .0
     448    GRID     4.567923E-03   2.758324E-04   1.139252E-02    .0             .0             .0
```

11.4 METHOD SELECTION GUIDELINES

The selection of an eigenextraction method depends upon such factors as the number of eigenvalues that you need relative to the total number possible in your model, or the frequency range you wish to analyze, or some *a priori* understanding of the frequency content of your forcing functions and therefore dynamic response of your finite element model.

Additionally, selection of an eigenvalue extraction procedure should also be made in consideration with other factors which include the mathematical algorithms employed by the various procedures. In practice, certain eigenvalue extraction techniques work best with sparse, highly banded matrices, **LANCZOS** and **SINV**, and other algorithms work better for smaller, more dense matrices, **GIV** and **MGIV**. For large problems with dense mass matrices, and for Fluid-Structure Interaction analyses, the **SUBS** method is ideal.

UAI/*NASTRAN* provides this variety of eigenextraction methods to cover essentially all conditions normally encountered in structural modeling and analysis situations. An understanding of which method and related analysis procedures should be used for a particular situation is developed through experience and by following the guidelines presented below.

Some of these eigenextraction methods and related analysis options remain in **UAI/*NASTRAN*** for historical reasons and to provide compatibility with earlier versions of the system. Often, there are several "correct" approaches to solving the same problem. General guidelines for Normal Mode analysis include:

- For most problems, use the **LANCZOS** method and do not use a reduction procedure. This is often both the most accurate and least expensive approach.
- Alternately, for large models, a good choice is to use Dynamic Reduction and the **GIV** or **MGIV** eigenextraction method. For relatively small models, a static condensation approach may be effectively used instead of Dynamic Reduction.
- When using the **GIV** or **MGIV** procedures, **UAI/*NASTRAN*** will automatically enable a static condensation of massless degrees of freedom. When using the **GIV** procedure, do not disable this **AUTOOMIT** feature or you will likely encounter a numerical failure due to mass matrix singularities. The **MGIV** procedure will correctly handle models with a relatively small number of mass singularities even when the static condensation operation is suppressed.
- When using static condensation, you may select the degrees of freedom to keep (or eliminate by reduction) by using Bulk Data entries **ASETi** or **OMITi**. However, a more accurate approach is provided by the **AUTOOMIT** feature with the **KEEP** option to insure that important mass characteristics are not inadvertently omitted.
- For large fluid-structure Interaction models, use the **SUBS** method and do not use reduction procedures. For additional information, see Chapter 23.

The following table provides a concise summary of the issues related to eigenvalue extraction procedure selection.

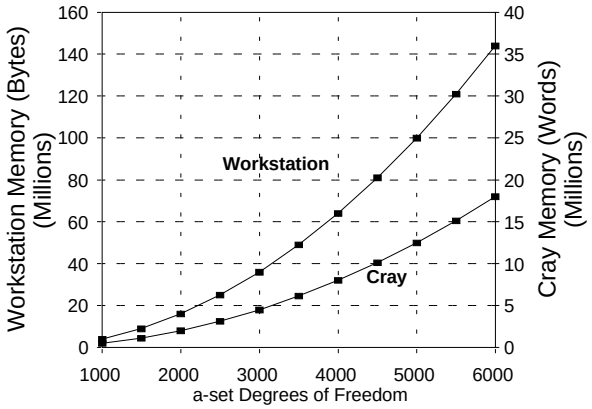
METHOD:	IS USED WHEN:	GUIDELINES:
GIVENS (GIV)	All eigenvalues of the system are required, and the Mass matrix is not singular.	The model should be reduced to a manageable size prior to using this method. Memory requirements (see Figure 11-1) and CPU time increase quickly with increasing problem size.
GIVENS (MGIV)	All eigenvalues of the system are required, and the Mass matrix may have singularities.	
LANCZOS (LANCZOS)	A few to a large number of eigenvalues are required and no reduction procedures are employed.	Usually the most efficient method of solution. Reduction procedures should not be used, because this procedure is most effficient for sparse, banded matrices.
INVERSE POWER (SINV)	Only a few eigenvalues are required in a narrow frequency range.	Generally, this method is less robust than **LANZCOS**. This procedure is primarily retained for historical purposes.
SUBSPACE ITERATION (SUBS)	A few to a large number of eigenvalues are required and no reduction procedures are employed.	This method is preferred in cases where the mass matrix is very large and dense. It is the only method that can be used to extract eigenvalues for large Fluid-Structure Interaction models.

When using either the **GIV** or **MGIV** procedures, the selection of a reduction method is very important because of its impact on the cost of your computer runs. Guidelines for selecting a method are shown in the following table.

METHOD:	IS USED WHEN:	GUIDELINES:
STATIC CONDENSATION (Guyan Reduction)	Your model has a strong, well-defined, localized mass distribution.	Provides good accuracy and a large cost savings when done properly. For the most efficient processing, either very many (>90%) or very few (<10%) of the dof should be retained in the *a-set*. Use the **AUTOOMIT(KEEP = x)** feature to insure accurate results.
DYNAMIC REDUCTION	Your model is very large with a relatively small number of important modes.	Provides accurate solution because all of the important low frequencies are retained.

Figure 11-1 shows the dramatic increase in the amount of working memory needed to perform the Givens and Modified Givens eigenextractions. In addition, as problem size grows, the CPU time required to perform the reductions necessary to arrive at a reasonably small *a-set* also increase dramatically. Current empirical data indicates that the use of Lanczos without reduction is the least expensive technique for large problem solution.

Figure 11-1. GIVENS MEMORY REQUIREMENTS



11

This page is intentionally blank.

Chapter 12

12

COMPLEX EIGENVALUE ANALYSIS

The computation of the natural modes of a structural system which is subjected to damping or which considers structural stability arising from the presence of feedback loads such as aerodynamic forces or control systems are some of the classes of problems which may be solved with the **UAI/*NASTRAN*** Complex Eigenvalue Analysis discipline. This capability is provided in two Rigid Formats. Rigid Format 7 solves the equations of motion using a direct solution of the matrix equations, the *direct* method, and Rigid Format 10 transforms the equations of motion of the structural portion of the model to modal coordinates before solving the equations, the *modal* method. This Chapter describes the procedures and options used for performing complex eigenvalue analyses.

12.1 MATHEMATICAL BACKGROUND

The complex eigenvalue analysis solves the system of second order differential equations:

$$\left[\boldsymbol{M}_{dd} p^2 + \boldsymbol{B}_{dd} p + \boldsymbol{K}_{dd} \right] \boldsymbol{u}_d = \boldsymbol{0} \tag{12-1}$$

where p is the differential operator and the matrices $\boldsymbol{K}_{dd}$, $\boldsymbol{B}_{dd}$, and $\boldsymbol{M}_{dd}$, may be complex and unsymmetric.

The degrees of freedom in $\boldsymbol{u}_d$ may include EXTRA points, those in the *e-set*, as well as structural degrees of freedom which have been reduced to the analysis set, or *a-set*. The resulting set of degrees of freedom is called the dynamic set, or *d-set*.

12

This equation represents a homogeneous system which may be solved by determining its eigenvalues. As is the case for all dynamic response solutions in **UAI/*NASTRAN***, you may employ either the direct method or the modal method in solving the equations of motion. An overview of the mathematical approach to each of these is described in this section.

12.1.1 Direct Method

In the direct method, the definitions of the $\boldsymbol{K}_{dd}$, $\boldsymbol{B}_{dd}$, and $\boldsymbol{M}_{dd}$ matrices are:

$$\boldsymbol{K}_{dd} = (1 + i\, G)\, \boldsymbol{K}^1_{dd} + \boldsymbol{K}^2_{dd} + i\, \boldsymbol{K}^4_{dd} \tag{12-2}$$

$$\boldsymbol{B}_{dd} = \boldsymbol{B}^1_{dd} + \boldsymbol{B}^2_{dd} \tag{12-3}$$

$$\boldsymbol{M}_{dd} = \boldsymbol{M}^1_{dd} + \boldsymbol{M}^2_{dd} \tag{12-4}$$

The $\boldsymbol{K}^1_{dd}$ and $\boldsymbol{M}^1_{dd}$ matrices correspond to stiffness and mass, respectively. $\boldsymbol{B}^1_{dd}$ and $\boldsymbol{K}^4_{dd}$ represent viscous and non-uniform structural damping. These are all generated from the finite element model. $\boldsymbol{K}^2_{dd}$, $\boldsymbol{B}^2_{dd}$, and $\boldsymbol{M}^2_{dd}$ are reduced from those matrices input directly by you. They may also be generated using transfer functions. *G* is the uniform structural damping coefficient.

12.1.2 Modal Method

An efficiency improvement for solving the complex eigenvalue problem is realized through the application of the modal method. In this method, the equation:

$$\left[\boldsymbol{K}_{aa} - \Lambda\, \boldsymbol{M}_{aa} \right] \boldsymbol{\Phi}_{a\xi} = \boldsymbol{0} \tag{12-5}$$

is solved to determine the natural modes of the system. Note that this is done ignoring any damping which may be present. The unknown degrees of freedom, $\boldsymbol{u}_d$ are then related to these modes by the equation:

$$\boldsymbol{u}_d = \Phi_{dh}\, \boldsymbol{q}_h \tag{12-6}$$

where Φ_{dh} is the matrix containing a prescribed number of eigenvectors of the real eigenproblem, expanded to include the EXTRA points. $\boldsymbol{q}_h$ represents the generalized coordinates in the *h-set*. This includes both the modal coordinates, ξ_i, and the EXTRA points, *e-set*. Then the complex eigenproblem may be rewritten as:

$$\left[\boldsymbol{M}_{hh} p^2 + \boldsymbol{B}_{hh} p + \boldsymbol{K}_{hh} \right] \boldsymbol{q}_h = \boldsymbol{0} \tag{12-7}$$

The definitions of the *h-set* size matrices are:

$$\boldsymbol{K}_{hh} = (1 + iG) \boldsymbol{k} + \Phi_{dh}^T \boldsymbol{K}_{dd}^2 \Phi_{dh} + i \Phi_{dh}^T \boldsymbol{K}_{dd}^4 \Phi_{dh} \tag{12-8}$$

$$\boldsymbol{B}_{hh} = \boldsymbol{b} + \Phi_{dh}^T \boldsymbol{B}_{dd}^1 \Phi_{dh} + \Phi_{dh}^T \boldsymbol{B}_{dd}^2 \Phi_{dh} \tag{12-9}$$

$$\boldsymbol{M}_{hh} = \boldsymbol{m} + \Phi_{dh}^T \boldsymbol{M}_{dd}^2 \Phi_{dh} \tag{12-10}$$

Let m_i be the terms of the diagonal matrix of modal masses, $\boldsymbol{m}$. Then the following relationships define the terms of the diagonal matrices $\boldsymbol{k}$ and $\boldsymbol{b}$ in (12-8) and (12-9):

$$k_i = (2 \pi f_i)^2 m_i \tag{12-11}$$

$$b_i = (2 \pi f_i) \; g(f_i) m_i \tag{12-12}$$

where f_i represents the frequency of each mode, and $g(f_i)$ provides a measure of structural damping as a function of frequency, or *modal damping*.

12.1.3 Eigenvalue Extraction Methods

UAI/*NASTRAN* provides three eigenextraction methods for solving the complex eigenproblem. The first, Hessenberg, is used to extract all the eigenvalues, while the other two, Inverse Power and Iterative Schur-Rayleigh-Ritz (ISSR), are used to extract a few eigenvalues and their associated eigenvectors in a specified range. A brief description of these methods is given in the following sections.

12.1.3.1 The Hessenberg Method

The first step in the Hessenberg eigenextraction method is to cast the eigenproblem in canonical form:

$$\left[\boldsymbol{A} - \lambda \boldsymbol{I} \right] \boldsymbol{\Phi} = \boldsymbol{0} \tag{12-13}$$

after which $\boldsymbol{A}$ is transformed to the upper Hessenberg form. This process depends on the existance of damping in the equations of motion (12-1). When damping is present, the transformations results in:

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \\ -\boldsymbol{M}^{-1}\boldsymbol{K} & -\boldsymbol{M}^{-1}\boldsymbol{B} \end{bmatrix} \tag{12-14}$$

and when there is no damping, the transformation is:

$$\boldsymbol{A} = -\boldsymbol{M}^{-1}\boldsymbol{K} \tag{12-15}$$

You will note that in either case the mass matric must be nonsingular. The matrix ***A*** may then be reduced to upper Hessenberg form using stabilized transformations. This techniqe is thoroughly described in [Wilkinson65].

A QR iteration is then performed to extract the smallest eigenvalue of the system [Francis61], and a shifting strategy is employed to accelerate convergence. After obtaining each eigenvalue, the ***A*** matrix is deflated. This new matrix then forms the basis for the next iteration. The process continues until the appropriate number of eigenvalues has been extracted.

12.1.3.2 The Inverse Power Method

The inverse power method is a procedure which extracts the eigenvalues and eigenvectors in a specified range of interest. The general method in **UAI/*NASTRAN*** was developed in the 1960's. The theory may be found in [NASTRAN79] and is not repeated here.

12.1.3.3 The Iterative Schur-Rayleigh-Ritz Method (ISSR)

The ***Iterative Schur-Rayleigh-Ritz Method*** (ISSR) is a procedure which extracts a specified number of roots which lie within a circle in the complex plane centered at the origin. This differs from the Inverse Power Method which extracts roots in one or more rectanular regions in the complex plane. The ISSR method computes a Schur factorization of the canonical matrix A such that:

$$\boldsymbol{A}\,\boldsymbol{Q} = \boldsymbol{Q}\,\boldsymbol{T} \tag{12-16}$$

where A has dimension n x n, Q is n x m, and T is m x m. The reduced eigenproblem:

$$\boldsymbol{T}\,\boldsymbol{y} = \lambda\,\boldsymbol{y} \tag{12-17}$$

is then solved. Transformation of y with Q recovers the eigenvectors corresponding to the eigenvalues λ_i.

This method has the advantage of solving a much smaller problem when only a few roots are required.

12.2 INPUT DATA REQUIREMENTS

This section describes the input data required for performing a complex eigenvalue analysis. Input data for complex eigenvalue analysis are organized to provide an easy transition from a static analysis model. All static analysis modeling options are available. In addition, EXTRA points are introduced for the purpose of coupling non-structural quantities to the structural model. Complex eigenvalue analysis requires additional information beyond that necessary for statics analysis, specifically:

- Mass data are required.
- A complex eigenextraction method must be supplied.
- Viscous damping, structural damping, and, in the modal Rigid Format, modal damping may be specified.
- In the modal Rigid Format, a real eigenextraction technique must be supplied.
- In the modal Rigid Format, data to specify frequencies and eigenvalue ranges may be selected.

12.2.1 Executive Control Commands

Complex Eigenvalue analysis is selected with the **SOL** Executive Control Command. There are two Rigid Formats available. For the direct method, Rigid Format 7 is used, and for the modal method, Rigid Format 10 is used. The Executive Control commands are specified as,

```
SOL 7     (For the Direct Method)
SOL 10    (For the Modal Method)
```

12.2.2 Case Control Commands

The Case Control commands, their relationship to corresponding Bulk Data entries, and the actions taken in each case are shown in Table 12-1.

As with all dynamic analyses, you may use the Guyan reduction feature to reduce the problem size. Only one boundary condition may be solved in a given execution. Also, only one set of direct input matrices may be selected with the **B2GG**, **K2GG**, and **M2GG** commands.

At least one SUBCASE must be defined for each different set of transfer functions or direct input matrices selected with any combination of **TFL**, **B2PP**, **K2PP**, and **M2PP** commands. You may select a different complex eigensolution method for each set. A second SUBCASE for each set may be used to change output requests using the **MODES** command as with normal modes analysis. The modal method employs additional Case Control commands as shown in Table 12-1.

12.2.3 Bulk Data Entries

Table 12-1 summarizes the Bulk Data entries which you may use in a Complex Eigenvalue analysis. The only entry specific to this type of analysis is **EIGC** which controls the complex eigenvalue extraction process.

Three complex eigenvalue extraction methods are available: Inverse Power, **INV**, and Iterative Schur-Rayleigh-Ritz, **ISSR**, which solve for a specific number of eigenvalues; and the Upper Hessenberg method, **HESS**, which is a general tridiagonalization method for complex eigen-

Table 12-1. SELECTING COMPLEX EIGENANALYSIS DATA

WHEN USING:	TO SELECT OR MODEL:	YOU USE THE	
		CASE CONTROL COMMAND:	BULK DATA ENTRIES:
THE DIRECT OR MODAL METHOD	Complex Eigenmethod	CMETHOD	EIGC
	Direct Input Matrices At The System Level	K2GG M2GG B2GG	DMIG
	Direct Input Matrices For All Physical Dof Including Extra Points	K2PP M2PP B2PP	DMIG
	Nonstructural effects	TFL	TF EPOINT
	Specific Modes For Output	OMODES	
	Viscous Damping Effects		CVISC,PVISC CDAMPi,PDAMP
	Uniform Structural Damping		PARAM,G
	Nonuniform Structural Damping		MATi
THE MODAL METHOD	Real Eigenmethod	METHOD	EIGR
	Modal Damping Effects	SDAMPING	TABDMP1
	Modal Coordinates		PARAM,LFREQ PARAM,HFREQ PARAM,LMODES
	Retaining Diagonal Terms Of Non-diagonal Matrices		PARAM,DIAGNLPP

value extraction which extracts all of the eigenvalues. Arguments presented in Chapter 11, Normal Modes, with regard to the suitability of these two general eigenextraction methods for your model are still valid.

EIGC	**SID**	**METHOD**	**NORM**	**GID**	**DOF**		**ND**	**EPS**	

The **EIGC** Bulk Data entry employing the Inverse Power method has the following form:

EIGC	**SID**	**METHOD**	**NORM**	**GID**	**DOF**	**E**			**-cont-**
-cont-	**PA1**	**QA1**	**PB1**	**QB1**	**W1**	**NE1**	**ND1**		**-cont-**
-cont-	**PA2**	**QA2**	**PB2**	**QB2**	**W2**	**NE2**	**ND2**		**-cont-**

Each continuation entry of the **EIGC** specifies a region in the complex plane by defining the locations of two ends of a line segment and an associated width. You may define as many as ten of these regions.

In the modal method, you may control the modal representation by specifying either the **LMODES** or the **HFREQ** and **LFREQ** parameters. **LMODES** specifies the number of lowest frequency modes to be used, whereas **LFREQ** and **HFREQ** define a range of frequencies. If you do not specify any preference for the extent of modal representation, all of the extracted real modes will be used in the modal formulation.

When you use:

PARAM	DIAGNLPP	YES							

in your Bulk Data packet, only the diagonal terms of any non-diagonal matrices are retained for the analysis. This results in a set of uncoupled equations which is computationally less expensive and provides a fairly reliable alternative. Unlike real normal modes analysis, no special provisions are included for free body modes.

12.3 SOLUTION RESULTS

The presentation of solution results for complex eigenvalue analysis begins with the eigenvalue extraction summary. When you select the Inverse Power method, a summary appears for each specified region. Samples of this summary for the two complex eigenextraction methods are shown in Table 12-2a shows the Inverse Power eigenextraction summary for a problem with two search regions, while Table 12-2b illustrates the very simple summary presented when the Upper Hessenberg method is selected.

All of the extracted complex eigenvalues are then divided into two sets: those with positive imaginary components, and those with negative imaginary components. Each set is then sorted separately by the value of the imaginary component. If these values are the same, the sort is then performed on the real part, *P*, in descending order. In the summary table, the eigenvalues with negative imaginary components are placed at the end of the list of sorted eigenvalues. A sample of the eigenvalue summary is shown in Table 12-3.

For each complex eigenvalue, the associated natural frequency, f_N, ratio of critical damping, ζ, and damped frequency, f_D, are also presented. These values are calculated from the following relationships:

$$f_N = \frac{1}{2\pi}\left[P^2 + Q^2\right]^{1/2} \tag{12-18}$$

$$\zeta = \frac{-P}{\left[P^2 + Q^2\right]^{1/2}} \tag{12-19}$$

$$f_D = f_N\left[1 - \zeta^2\right]^{1/2} \tag{12-20}$$

The results of the complex eigenvalue analysis can be obtained only in **SORT1**. This means that all point and element results are presented for each complex eigenvalue. In addition to the results available for a normal modes analysis, you may request the output of solution set eigenvectors with the **SDISP** Case Control command. These results may be obtained either as the real and imaginary parts of the complex quantity, or as magnitude and phase angle. Undeformed structural plots may also be requested.

When you select the modal method, a real eigenvalue summary, representing the modes of the undamped structure, precedes the complex eigenvalue summary as described above. The solution results and their Case Control requests are also the same as in the direct method. However, solution set eigenvectors, requested with the Case Control command **SDISP**, are identified by their modal coordinates.

Table 12-2. EIGENEXTRACTION SUMMARIES

a. Summary for Inverse Power Method

```
C O M P L E X   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

                    ( INVERSE POWER )

                      REGION NO.  1


      NUMBER OF EIGENVALUES EXTRACTED . . . . . .        1

      NUMBER OF STARTING POINTS USED. . . . . .        2

      NUMBER OF STARTING OR SHIFT POINT MOVES . . .        0

      TOTAL NUMBER OF TRIANGULAR DECOMPOSITIONS . .        2

      TOTAL NUMBER OF VECTOR ITERATIONS . . . . . .       24

              . . . REASON FOR TERMINATION . . .
                    REGIONS COMPLETED

C O M P L E X   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

                    ( INVERSE POWER )

                      REGION NO.  2


      NUMBER OF EIGENVALUES EXTRACTED . . . . . .        2

      NUMBER OF STARTING POINTS USED. . . . . .        3

      NUMBER OF STARTING OR SHIFT POINT MOVES . . .        0

      TOTAL NUMBER OF TRIANGULAR DECOMPOSITIONS . .        3

      TOTAL NUMBER OF VECTOR ITERATIONS . . . . . .       40

              . . . REASON FOR TERMINATION . . .
                    REGIONS COMPLETED
```

b. Summary for Upper Hessenberg Method

```
C O M P L E X   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

                ( UPPER HESSENBERG METHOD )


                NUMBER OF EIGENVALUES EXTRACTED . . . . . .        3

                NUMBER OF EIGENVECTORS COMPUTED . . . . . .        3
```

Table 12-3. EIGENVALUE SOLUTION RESULTS

C O M P L E X E I G E N V A L U E S U M M A R Y

MODE NO.	EXTRACTION ORDER	EIGENVALUE (REAL)	EIGENVALUE (IMAG)	FREQUENCY (CYCLES) NATURAL	FREQUENCY (CYCLES) DAMPED	DAMPING RATIO
1	3	-1.414850E+00	0.000000E+00	2.251803E-01	0.000000E+00	0.000000E+00
2	2	-5.075590E-01	8.188399E-01	1.533278E-01	1.303224E-01	5.268484E-01
3	1	5.202361E-01	3.824517E+00	6.142963E-01	6.086907E-01	-1.347854E-01
4	4	-5.075601E-01	-8.188292E-01	1.533265E-01	1.303207E-01	5.268542E-01
5	5	5.202218E-01	-3.824515E+00	6.142957E-01	6.086904E-01	-1.347818E-01

12

This page is intentionally blank.

Chapter 13

FREQUENCY AND RANDOM RESPONSE ANALYSIS

UAI/*NASTRAN* provides analysis capabilities for exciting a structural model with either sinusoidal forces or boundary motions, and with random forces or boundary motions. These capabilities are provided in two Rigid Formats and the MULTI Multidisciplinary solution sequence. Rigid Format 8 solves the equations of motion using a direct solution of the matrix equations for each frequency of excitation, the *direct* method, and Rigid Format 11 transforms the equations of motion of the structural portion of the model to modal coordinates before solving the equations, the *modal* method.

With either the direct or modal method, random response calculations are performed as a post-processing function of a frequency response analysis. In this case, the frequency response portion of the analysis is used for the purpose of computing structural transfer functions between the sources of excitation and the desired response quantities. This Chapter describes the procedures and options used for solving frequency and random response problems. Direct and modal frequency response may also be used when performing Multidisciplinary Design Optimization. Aspects of this optimization and analysis capability is described in Chapter 25.

13.1 MATHEMATICAL BACKGROUND

In frequency response analysis, the input load is defined as a series of sinusoidal loads, each operating at a unique frequency. The steady state response of the structure at each solution frequency is then assumed to be sinusoidal with the magnitude and phase angle being defined by a complex number. The general equations of motion are:

$$\left[-\omega^2 \boldsymbol{M}_{dd} + i\,\omega\,\boldsymbol{B}_{dd} + \boldsymbol{K}_{dd} \right] \boldsymbol{u}_d\, e^{i\,\omega\,t} = \boldsymbol{P}_d\, e^{i\,\omega\,t} \tag{13-1}$$

where ω, measured in radians, is the excitation frequency of the input load $\boldsymbol{P}_d$. The matrices $\boldsymbol{K}_{dd}$, $\boldsymbol{B}_{dd}$, and $\boldsymbol{M}_{dd}$, may be complex and unsymmetric. *i* is the imaginary number operator. The degrees of freedom in $\boldsymbol{u}_d$ may include EXTRA points, those in the *e-set*, as well as structural degrees of freedom which have been reduced to the analysis set, *a-set*. The resulting set of degrees of freedom is called the dynamic set, *d-set*.

As is the case for all dynamic response solutions in **UAI/*NASTRAN***, you may employ either the direct method or the modal method in solving the equations of motion. An overview of the mathematical approach to each of these is described in this section.

13

13.1.1 Frequency Response Analysis — Direct Method

In the direct method, the definitions of the $\boldsymbol{K}_{dd}$, $\boldsymbol{B}_{dd}$, and $\boldsymbol{M}_{dd}$ matrices are:

$$\boldsymbol{K}_{dd} = (1 + i\,G)\,\boldsymbol{K}^1_{dd} + \boldsymbol{K}^2_{dd} + i\,\boldsymbol{K}^4_{dd} \tag{13-2}$$

$$\boldsymbol{B}_{dd} = \boldsymbol{B}^1_{dd} + \boldsymbol{B}^2_{dd} \tag{13-3}$$

$$\boldsymbol{M}_{dd} = \boldsymbol{M}^1_{dd} + \boldsymbol{M}^2_{dd} \tag{13-4}$$

The $\boldsymbol{K}^1_{dd}$ and $\boldsymbol{M}^1_{dd}$ matrices correspond to stiffness and mass, respectively. $\boldsymbol{B}^1_{dd}$ and $\boldsymbol{K}^4_{dd}$ represent viscous and non-uniform structural damping. These are all reduced from the matrices generated from the finite element model and the symmetric matrices that you have input directly using the **K2GG**, **M2GG**, and **B2GG** Case Control commands. *G* is the uniform structural damping coefficient. $\boldsymbol{K}^2_{dd}$, $\boldsymbol{B}^2_{dd}$, and $\boldsymbol{M}^2_{dd}$ are reduced from those matrices input directly by you using the **K2PP**, **M2PP**, and **B2PP** Case Control commands. They may also be generated using transfer functions.

13.1.2 Frequency Response Analysis — Modal Method

For many frequency response problems, the modal method provides an efficient solution procedure. In this method, the natural modes of the structure, within a specified frequency range, are used as generalized degrees of freedom. This greatly reduces the problem size while retaining accuracy in the selected frequency range. Using the modal method, the equation:

$$\left[\boldsymbol{K}_{aa} - \lambda\,\boldsymbol{M}_{aa} \right] \boldsymbol{\Phi}_a = \boldsymbol{0} \tag{13-5}$$

is solved to determine the natural modes of the system. The matrices $\boldsymbol{K}_{aa}$ and $\boldsymbol{M}_{aa}$ are reduced from $\boldsymbol{K}^1_{dd}$ and $\boldsymbol{M}^1_{dd}$, and do not consider any damping effects which may be present. The unknown degrees of freedom, $\boldsymbol{u}_d$ are then related to these modes by the following relationship:

$$\boldsymbol{u}_d = \begin{Bmatrix} \boldsymbol{u}_a \\ \boldsymbol{u}_e \end{Bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_{ai} & \\ & I \end{bmatrix} \begin{Bmatrix} \boldsymbol{q}_i \\ \boldsymbol{q}_e \end{Bmatrix} = \boldsymbol{\Phi}_{dh}\boldsymbol{q}_h \tag{13-6}$$

where *i* represents the modal coordinates computed and retained for the response analysis. The manner in which you may control the extent of the modal representation is explained later. $\boldsymbol{q}_h$ represents the modal generalized coordinates in the *h-set*. This includes both the selected modal coordinates and the EXTRA points. Substituting (13-6) in (13-1), and premultiplying the relation by $\boldsymbol{\Phi}^T_{dh}$ will result in the frequency response equations of motion being transformed to the modal form:

$$\left[-\omega^2 \boldsymbol{M}_{hh} + i\,\omega\,\boldsymbol{B}_{hh} + \boldsymbol{K}_{hh} \right] \boldsymbol{q}_h\, e^{i\omega t} = \boldsymbol{P}_h\, e^{i\omega t} \tag{13-7}$$

Since $\boldsymbol{\Phi}_{dh}$ is orthogonal with respect to $\boldsymbol{M}^1_{dd}$, the resulting transformed matrix, $\boldsymbol{m}$, is a diagonal matrix, and its terms, m_i, are called the *modal* masses. The terms of the diagonal *modal stiffness*, $\boldsymbol{k}$, and *modal damping*, $\boldsymbol{b}$, matrices are then defined as:

$$k_i = (2\pi f_i)^2\, m_i \tag{13-8}$$

$$b_i = (2\pi f_i)\; g(f_i)\, m_i \tag{13-9}$$

where f_i represents the frequency of each mode, and $g(f_i)$ are the optional modal damping factors which you may specify using the **SDAMP** Case Control command.

Repeating the same transformation for (13-2) through (13-4), results in the definitions of the *h-set* size matrices:

$$\boldsymbol{K}_{hh} = (1 + i\,G)\,\boldsymbol{k} + \boldsymbol{\Phi}^T_{dh}\boldsymbol{K}^2_{dd}\boldsymbol{\Phi}_{dh} + i\,\boldsymbol{\Phi}^T_{dh}\boldsymbol{K}^4_{dd}\boldsymbol{\Phi}_{dh} \tag{13-10}$$

$$\boldsymbol{B}_{hh} = \boldsymbol{b} + \boldsymbol{\Phi}^T_{dh}\,\boldsymbol{B}^1_{dd}\boldsymbol{\Phi}_{dh} + \boldsymbol{\Phi}^T_{dh}\boldsymbol{B}^2_{dd}\boldsymbol{\Phi}_{dh} \tag{13-11}$$

$$\boldsymbol{M}_{hh} = \boldsymbol{m} + \boldsymbol{\Phi}^T_{dh}\boldsymbol{M}^2_{dd}\boldsymbol{\Phi}_{dh} \tag{13-12}$$

$$\boldsymbol{P}_h = \boldsymbol{\Phi}^T_{dh}\boldsymbol{P}_d \tag{13-13}$$

Note that while the term corresponding to $\boldsymbol{K}^1_{dd}$ is replaced by $\boldsymbol{k}$, the term corresponding to $\boldsymbol{B}^1_{dd}$ may not be a diagonal matrix. This depends on whether the viscous damping matrix is a linear combination of the stiffness and mass matrices. This is called ***Rayleigh Damping*** which results in damped, uncoupled modes. In general, the transformed direct input and damping matrices are not diagonal.

However, you may choose to retain only the diagonal terms of the transformed matrices. This can be accomplished by using the:

PARAM	DIAGNLPP	YES							

Bulk Data entry which will result in diagonal, and hence, uncoupled equations of motion.

13.1.2.1 Residual Flexibility

The quality of the modal approximation of (13-6) works well for stiffness and mass modeling, but may not be adequate to model the response of the system under point loads or in the vicinity of massless points. Under these circumstances, it may be helpful to use additional generalized degrees of freedom that represent the displacements from applied static loads that simulate the spatial distribution of the dynamic load or that simulate the displacement response to a load at the massless point. This technique for augmenting the modal approximation is termed Residual Flexibility or Residual Stiffness. It is available in **SOL MULTI** when performing Modal Frequency Response analysis and optimization.

The Residual Flexibility degrees of freedom are selected using the Case Control Command, **RESFLEX**, which refers to a Case Control Integer Set of **CASE** identification numbers of Statics cases whose displacement vectors form the additional generalized degrees of freedom. These new vectors are combined with the selected normal modes to form a new set of generalized coordinates.

Selecting a good set of additional degrees of freedom is a somewhat heuristic process with the only general guidelines being that good additional degrees of freedom can extend the accuracy of the modal approximation up to the frequency of the highest included normal mode. The user can then trade off between normal modes and residual vectors to get the largest frequency range of accuracy with the minimum number of generalized degrees of freedom. Typically, a good first set of residual vectors are the displacements due to the spatial component (e.g., the **DAREA** entries) of the dynamic loads. The Statics analysis accepts Case Control **LOAD** references to **DAREA** entries and the **RLOAD** Bulk Data entries may refer to the standard Statics load data to make this process easy for the user.

Since the choice of good vectors is difficult, the **RESFLEX** command includes options to support quantitative feedback of the quality of the added vectors. The Gram-Schmidt orthogonality check and the Strain Energy check both provide information regarding problems with the linear independence of the selected vectors. It is important to recognize, however, that the goodness of the choice can best be determined by comparison with direct results. As a matter of practice, the modal frequency analysis with or without residual flexibility vectors should be checked against a small number of direct analysis frequencies through the frequency range of interest to ensure that the modal approximation provides accurate analyses.

Residual Flexibility Augmentation. To understand the role of the Gram-Schmidt and Strain Energy checks, it is worthwhile to briefly review the mathematics of the augmentation process. This process recreates a decoupled set of mass and stiffness matrices while supporting all the damping options of a normal modal frequency response analysis. (Note that, just as for modal reduction, the viscous damping matrix may or may not be uncoupled, and direct matrix input methods may cause coupling of stiffness, damping and mass). Both the Gram-Schmidt and Strain Energy checks are methods to assist in the selection of linearly independent generalized degrees of freedom.

Revisiting (13-6) and augmenting the normal modes with the residual vectors yields:

$$\boldsymbol{u}_d = \begin{bmatrix} \boldsymbol{\Phi}_{ai} & \boldsymbol{u}_{ar} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{Bmatrix} \boldsymbol{q}_i \\ \bar{\boldsymbol{q}}_r \\ \boldsymbol{q}_e \end{Bmatrix} = \bar{\boldsymbol{\Phi}}_{dh} \bar{\boldsymbol{q}}_h \tag{13-14}$$

where $\boldsymbol{u}_{ar}$ contains the residual vectors reduced to the structural *a-set* and normalized. Notice that the *h-set* has been *augmented* by $\bar{\boldsymbol{q}}_r$. A reduced set of equations of motion:

$$\left[-\omega^2 \bar{\boldsymbol{M}}_{hh} + i\,\omega\,\bar{\boldsymbol{B}}_{hh} + \bar{\boldsymbol{K}}_{hh} \right] \bar{\boldsymbol{q}}_h\, e^{i\,\omega\,t} = \bar{\boldsymbol{P}}_h\, e^{i\,\omega\,t} \tag{13-15}$$

can be obtained in which

$$\bar{\boldsymbol{M}}_{hh} = \begin{bmatrix} \boldsymbol{\Phi}_{di}^T \boldsymbol{M}_{dd} \boldsymbol{\Phi}_{di} & \boldsymbol{\Phi}_{ai}^T \boldsymbol{M}_{aa} \boldsymbol{\Phi}_{ai} & \boldsymbol{M}_{de} \\ \boldsymbol{u}_{ar}^T \boldsymbol{M}_{aa} \boldsymbol{u}_{ai} & \boldsymbol{u}_{ar}^T \boldsymbol{M}_{aa} \boldsymbol{u}_{ar} & \boldsymbol{0} \\ \boldsymbol{M}_{ed} & \boldsymbol{0} & \boldsymbol{M}_{ee} \end{bmatrix} \tag{13-16}$$

Merging the extra points into the *a-set* as in (13-6) results in:

$$\bar{\boldsymbol{M}}_{hh} = \begin{bmatrix} \boldsymbol{\Phi}_{di}^T \boldsymbol{M}_{dd} \boldsymbol{\Phi}_{di} & \boldsymbol{\Phi}_{ai}^T \boldsymbol{M}_{aa} \boldsymbol{\Phi}_{ai} \\ \boldsymbol{u}_{ar}^T \boldsymbol{M}_{aa} \boldsymbol{u}_{ai} & \boldsymbol{u}_{ar}^T \boldsymbol{M}_{aa} \boldsymbol{u}_{ar} \end{bmatrix} \tag{13-17}$$

Similar expressions can then be developed for $\bar{\boldsymbol{K}}_{hh}$, $\bar{\boldsymbol{B}}_{hh}$ and $\bar{\boldsymbol{P}}_h$. The equations of motion, (13-15), are coupled since, in general, $\boldsymbol{u}_{ar}$ will not be orthogonal to $\boldsymbol{\Phi}_{ai}$. To uncouple the equations, a new eigenanalysis is performed that solves:

$$\left[\bar{\boldsymbol{K}}_{hh} - \bar{\lambda}\, \bar{\boldsymbol{M}}_{hh} \right] \bar{\boldsymbol{\Phi}}_{hh} = \boldsymbol{0} \tag{13-18}$$

All the eigenvalues and eigenvectors of this system are extracted using a Modified Givens method. The resultant eigenvectors are used to uncouple the equations of motion; for example

$$\boldsymbol{M}_{hh} = \bar{\boldsymbol{\Phi}}_{hh}^T \boldsymbol{\Phi}_{dh}^T \boldsymbol{M}_{dd} \boldsymbol{\Phi}_{dh} \bar{\boldsymbol{\Phi}}_{hh}^T = \boldsymbol{\Phi}_{dh_r}^T \boldsymbol{M}_{dd} \boldsymbol{\Phi}_{dh_r} \tag{13-19}$$

Rather than performing this expensive operation directly, however, advantage is taken of the relationship between $\boldsymbol{\Phi}_{dh}$ and $\bar{\boldsymbol{\Phi}}_{hh}$. $\bar{\boldsymbol{\Phi}}_{hh}$ has the structure:

$$\bar{\boldsymbol{\Phi}}_{hh} = \begin{bmatrix} \boldsymbol{I}_{ii} & \boldsymbol{\Phi}_{ir} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Phi}_{rr} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{ee} \end{bmatrix} \tag{13-20}$$

The new uncoupled equations can be assembled efficiently by using the original $\boldsymbol{M}_{hh}$, $\boldsymbol{K}_{hh}$, $\boldsymbol{B}_{hh}$ and $\boldsymbol{P}_h$ from (13-10) through (13-13). For example, the uncoupled, augmented mass matrix is:

$$M_{hh_r} = \begin{bmatrix} M_{ii} & \Phi_{ai}^T M_{aa}\left[\Phi_{ai}\Phi_{ir} + u_{ar}\Phi_{rr}\right] & M_{de} \\ \left[\Phi_{ai}^T M_{aa}\left[\Phi_{ai}\Phi_{ir} + u_{ar}\Phi_{rr}\right]\right]^T & \left[\Phi_{ai}\Phi_{ir} + u_{ar}\Phi_{rr}\right]^T M_{aa}\left[\Phi_{ai}\Phi_{ir} + u_{ar}\Phi_{rr}\right] & 0 \\ M_{ed} & 0 & M_{ee} \end{bmatrix} \quad (13\text{-}21)$$

where only the terms in the second column need to be computed; the others have already been assembled prior to **RESFLEX** processing.

The **ORTHOGONALITY** and **STRAIN ENERGY** checks come into play in providing feedback on the success of generation of the uncoupled equations of motion. The **ORTHOGONALITY** check examines the magnitude of the values in the (1,2) partition of the M_{hh_r} matrix:

$$M_{ir} = \Phi_{ai}^T M_{aa}\left[\Phi_{ai}\Phi_{ir} + u_{ar}\Phi_{rr}\right] \equiv 0 \quad (13\text{-}22)$$

13

If the residual vectors are linearly independent and do not yield any eigenvalues, $\bar{\lambda}$, that are less than the eigenvalues of the retained normal modes, the M_{hh_r} matrix should be diagonal and (13-22) should be satisfied. The ε_o value, entered on the **RESFLEX** command, is the largest absolute value of any term in M_{ir} that will be accepted in a successfully uncoupled mass matrix.

In a typical run, where the **RESFLEX** option **GRAMSCHM=AUTO** is selected, if the orthogonality criterion fails on the first pass, a Gram-Schmidt orthogonalization procedure is performed on Φ_{ai} and u_{ar} to identify the set of linearly dependent vectors. Columns of u_{ar} that are linearly dependent on Φ_{ai} or on other columns of u_{ar} are then removed from the set of residual vectors with a warning to the user. The whole uncoupled system is regenerated using the reduced set of vectors. If, on the second pass, the orthogonality criterion passes, the augmented modal analysis proceeds without warning. If it still fails, a warning is issued about the poor quality of the added degrees of freedom, but the analysis still proceeds. Downstream singularities or problems in the modal analyses ***MAY*** result.

The only way the orthogonality criterion can fail the second time is if one or more of the chosen residual vectors introduces an eigenvalue, $\bar{\lambda}$, that is less than that of the largest eigenvalue associated with the retained normal modes. This is an unusual event, since the residual vectors typically represent high frequency responses. If ***ALL*** of the low frequency normal modes are retained in the modal reduction, the problem residual vector would have been identified as redundant during the Gram-Schmidt process. Only if some low frequency modes are omitted can this problem occur. You should use the low frequency mode in that case and omit the redundant residual vector.

Under user control, the Gram-Schmidt process can be skipped or forced to occur prior to the first pass. If skipped, the redundancy warning that normally appears after the second uncoupling procedure is issued after the first failure of the orthogonality check. If forced to occur, the redundant residual vectors are identified prior to the first attempt to uncouple the equations and orthogonality failure again causes warnings on the first failure. Under no circumstances will the orthogonality check cause the execution to stop. The Gram-Schmidt process, however, may omit ***ALL*** the residual vectors in its attempts to ensure linear independence.

Strain Energy Checks. The above procedures ensure that linearly dependent residual vectors are not included in the additional set of degrees of freedom. The Strain Energy checks provide some quantitative feedback on the relative quality of a residual vector to represent displacement fields that cannot be well approximated by the normal modes. The idea is that, if all the normal modes were retained, a linear combination would be able to describe any displacement field. Therefore, comparing the strain energy in the residual vector to the strain energy in the modal approximation to the residual vector, provides a scalar measure of the relative ability of the normal modes to represent the residual vector's displacement field. If the normal modes can already represent most of the strain energy, the residual vector is not providing enough new dynamics and another vector should be considered.

The strain energy in the residual vector is computed as:

$$E = \frac{1}{2} \boldsymbol{u}_{ar}^T \boldsymbol{K}_{aa} \boldsymbol{u}_{ar} \tag{13-23}$$

and the modal approximation to the residual vector takes the form:

$$\hat{\boldsymbol{u}}_{ar} = \zeta_i \boldsymbol{\Phi}_{ai} \tag{13-24}$$

where the combinatorial factors, ζ_i, are computed as:

$$\zeta_i = \frac{\boldsymbol{\Phi}_{ai}^T \boldsymbol{M}_{aa} \boldsymbol{u}_{ar}}{\boldsymbol{\Phi}_{ai}^T \boldsymbol{M}_{aa} \boldsymbol{\Phi}_{ai}} \tag{13-25}$$

Computing the modal approximation to the strain energy follows by direct substitution:

$$\hat{E} = \frac{1}{2} \hat{\boldsymbol{u}}_{ar}^T \boldsymbol{K}_{aa} \hat{\boldsymbol{u}}_{ar} \tag{13-26}$$

The fraction of the total strain energy is simply the ratio of these two scalars. More detail is obtained, however, by examining the contribution of each mode to the strain energy. That is computed from:

$$e_i = \frac{1}{2} \zeta_i^2 \lambda_i m_i \tag{13-27}$$

which allows the identification of the modes that are most (large fraction) or least (small fraction) redundant with the residual vector. A good residual vector will have a low total fraction of modal strain energy to actual strain energy. If a high fraction results, the modal participation factors, ζ_i, and the modal strain energy fractions, e_i, can help to identify the near redundancies and indicate better choices of residual vectors.

13.1.2.2 Mode Acceleration

When you select the modal method, you typically exclude a range of higher frequency modes from the problem. This process, called *mode truncation*, tends to be less significant on the accuracy of the resulting accelerations than on the displacements. This is due to the omission of elastic forces, which may have considerable contributions from the higher modes.

You may use the mode acceleration option to alleviate this problem. In this procedure it is assumed that the solution accelerations and velocities are more accurate than the displacements. Therefore, (13-1) is rearranged, using the accelerations and velocities as known quantities:

$$\boldsymbol{K}_{dd}^{1}\bar{\boldsymbol{u}}_d = \boldsymbol{P}_d e^{i\omega t} - \left[-\omega^2 \boldsymbol{M}_{dd} + i\omega \boldsymbol{B}_{dd} + \boldsymbol{K}_{dd}^{2} \right] \boldsymbol{u}_d e^{i\omega t} \tag{13-28}$$

Note that the effects of structural damping are omitted. $\bar{\boldsymbol{u}}_d$ represents the improved displacements. The right hand side of (13-28) may be considered as an equivalent set of loads of the form:

$$\bar{\boldsymbol{P}}_d e^{i\omega t} = \boldsymbol{P}_d e^{i\omega t} - [-\omega^2 \boldsymbol{M}_{dd} + i\omega \boldsymbol{B}_{dd} + \boldsymbol{K}_{dd}^{2}] \boldsymbol{u}_d e^{i\omega t} \tag{13-29}$$

Then a static solution is performed to solve for the displacements:

$$\boldsymbol{K}_{dd}^{1}\bar{\boldsymbol{u}}_d = \bar{\boldsymbol{P}}_d \tag{13-30}$$

13

The improved displacements are then used to recover the element data.

13.1.3 Random Response Analysis

You may perform a random response analysis using the results you obtain from a frequency response analysis. This is usually done by subjecting a set of points in the model to any number of ***sources of excitation***, and measuring the random response of the desired points. The sources of excitation may be applied forces or enforced motions. You must use additional input data which defines the frequency-dependent power spectral density factors for the sources of excitation, and the statistical correlation between them, if any.

The application of frequency response techniques to the analysis of random excitations requires that the system be linear and that the excitation be stationary with respect to time. The general governing equations for calculating the power spectral densities and autocorrelation functions may then be conveniently approximated in the frequency domain.

Consider that $H_{aj}(\omega)$ is the frequency response of any physical quantity, u_a, due to an excitation source, Q_j. Then:

$$u_a(\omega) = H_{aj}(\omega)\, Q_j(\omega) \tag{13-31}$$

where $u_a(\omega)$ and $Q_j(\omega)$ are the Fourier transforms of u_a and Q_j. According to the transfer function theorem, the power spectral density of the response, $S_a(\omega)$, is related to the power spectral density of the excitation source, $S_j(\omega)$, by:

$$S_{aj}(\omega) = |H_{aj}(\omega)|^2 S_j(\omega) \tag{13-32}$$

The above relationship is significant because it allows the statistical properties, such as autocorrelation function, of the response of a system to random excitations to be evaluated using results of a frequency response analysis. Another useful result is that if sources Q_i are statistically uncorrelated, that is, if there is no cross-correlation between any pair of sources, then the

power spectral density of the total response is equal to the sum of the power spectral densities of the responses due to individual sources. Thus:

$$S_a(\omega) = \sum_j S_{aj}(\omega) = \sum_j |H_{aj}(\omega)|^2 S_j(\omega) \tag{13-33}$$

If the sources are statistically correlated, the degree of correlation can be expressed by cross-spectral density factors, S_{jk}, and the power spectral density of the response may be evaluated from:

$$S_a = \sum_j \sum_k H_{aj} H_{ak}^* S_{jk} \tag{13-34}$$

where H_{ak}^* is the complex conjugate of H_{ak}.

The autocorrelation function is evaluated for each value of time lag, τ, as a summation of the contributions from N excitation frequencies:

$$R_a(t) = \frac{1}{2\pi\tau} \sum_{i=1}^{N-1} \frac{S_a(\omega_i+1) - S_a(\omega_i)}{(\omega_{i+1} - \omega_i)\tau} [\cos(\omega_{i+1}\tau) - \cos(\omega_i\tau)] + S_a(\omega_{i+1}) \sin(\omega_{i+1}\tau) - S_a(\omega_i) \sin(\omega_i\tau) \tag{13-35}$$

The sources of excitation need not be forces at individual points. A loading condition, which is an ensemble of applied forces that are completely correlated, may also be treated as an excitation source. An example is a plane pressure wave from a specified direction. Furthermore, the response may be any physical quantity including internal element forces and stresses, as well as GRID point displacements, velocities, and accelerations.

13.1.4 Transfer Functions

Several options, outlined below, are provided for modeling nonstructural dynamic effects including transfer functions and direct input matrices. For convenience, and to avoid conflicts with static analyses, additional degrees of freedom, called EXTRA points, may be introduced on **EPOINT** Bulk Data entries for the dynamics Rigid Formats. The EXTRA points are placed in the *e-set*.

One example of the use of EXTRA points is for structural control system modeling. In order to facilitate the treatment of control systems, **UAI/*NASTRAN*** includes an input data format for the specification of a transfer function in the form:

$$(B_o + B_1 p + B_2 p^2) u_d + \sum_i (A_o(i) + A_1(i) p + A_2(i) p^2) u_i = 0 \tag{13-36}$$

which is interpreted as a differential equation with terms in the rows of $\boldsymbol{M}$, $\boldsymbol{B}$, and $\boldsymbol{K}$ corresponding to u_d. Note that if the transfer function is to act as a constraint equation, then the

row corresponding to u_d must be null. u_d and u_i can either be an EXTRA point or any structural point defined using **GRID** or **SPOINT** Bulk Data entries.

Structural loads proportional to displacements at EXTRA points are represented by adding terms in the stiffness matrix at the intersections of rows corresponding to the structural points and columns corresponding to the EXTRA points. Direct input matrix capability is provided for this purpose. The superposition of all such terms, including those generated by transfer functions, are called the direct input stiffness matrix, $\boldsymbol{K}^2_{pp}$, mass matrix, $\boldsymbol{M}^2_{pp}$, and damping matrix, $\boldsymbol{B}^2_{pp}$. Terms in the direct input matrices may refer to EXTRA points, structural points, or both. Since dynamic loads may also be applied to EXTRA points, it is possible to completely simulate the linearized subsystems by means of EXTRA points.

13.1.5 Viscoelastic Frequency Response

Many materials such as rubber and metals at high temperatures exhibit strong dependency of their material properties on frequency. Under pure sinsoidal exitation these materials show a lag between stress and strain. The behavior of such materials can be modeled by using the linear viscoelastic theory. The shear modulus for linear viscoelastic materials, G, can be written as :

$$G(f) = G'(f) + i\,G''(f) \tag{13-37}$$

where, $G'(f)$ is the shear storage modulus, $G''(f)$ is the shear loss modulus, and, f is the frequency.

The Youngs modulus $E(f)$ is obtained from the following relation:

$$E(f) = 2\,(1+\nu)\,G(f) \tag{13-38}$$

UAI/*NASTRAN* supports viscoelastic behavior in the direct frequency analysis. The viscoelastic effects are included in the total dynamic stiffness matrix in the manner described below.

The dynamic stiffness matrix is :

$$\boldsymbol{K}_{dd} = \boldsymbol{K}^e_{dd} + \boldsymbol{K}^v_{dd} \tag{13-39}$$

where, $\boldsymbol{K}^e_{dd}$ is the stiffness contribution from elastic elements and $\boldsymbol{K}^v_{dd}$ is the stiffness contribution from viscoelastic elements. $\boldsymbol{K}^v_{dd}$ can be expressed as :

$$\boldsymbol{K}^v_{dd} = \left[\frac{G'(f) + i\,G''(f)}{G_{ref}}\right]\boldsymbol{K}^{1v}_{dd} \tag{13-40}$$

where, $G_{ref}$is the reference value of $G(f)$. The matrix $\boldsymbol{K}^e_{dd}$ is given by:

$$\boldsymbol{K}^e_{dd} = (1+ig)\,\boldsymbol{K}^{1e}_{dd} = (1+ig)\left[\boldsymbol{K}^1_{dd} - \boldsymbol{K}^{1v}_{dd}\right] \tag{13-41}$$

In order to compute $\boldsymbol{K}^{1v}_{dd}$, which incorporates the viscoelastic effects into $\boldsymbol{K}_{dd}$, the $\boldsymbol{K}^4_{dd}$ must only include contributions from viscoelastic elements. This is accomplished by setting $g_e = 0$ for all elements which are not viscoelastic. (See Section 8.7)

Hence,

$$\boldsymbol{K}_{dd}^{4v} = \boldsymbol{K}_{dd}^{4} = g_{e_{ref}} \boldsymbol{K}_{dd}^{1v} \tag{13-42}$$

Substituting in (13-40) and (13-41) yields the following expressions:

$$\boldsymbol{K}_{dd}^{e} = (1 + ig)\left[\boldsymbol{K}_{dd}^{1} - \frac{1}{g_{e_{ref}}} \boldsymbol{K}_{dd}^{4} \right] \tag{13-43}$$

$$\boldsymbol{K}_{dd}^{v} = \frac{1}{g_{e_{ref}}}\left[\frac{G'(f) + i\, G''(f)}{G_{ref}} \right] \boldsymbol{K}_{dd}^{4} \tag{13-44}$$

and:

$$\boldsymbol{K}_{dd} = (1 + i\, g)\, \boldsymbol{K}_{dd}^{1} + (TR(f) + i\, TI(f))\, \boldsymbol{K}_{dd}^{4} \tag{13-45}$$

where,

$$TR(f) = \frac{1}{g_{e_{ref}}}\left[\frac{G'(f)}{G_{ref}} - 1 \right] \tag{13-46}$$

and,

$$TI(f) = \frac{1}{g_{e_{ref}}}\left[\frac{G''(f)}{G_{ref}} - g \right] \tag{13-47}$$

The viscoelastic analysis is performed by using the Case Control command:

```
SDAMPING = tid
```

where *tid* is the identification number of **TABLED1** Bulk Data entries which define the frequency dependence of both $G'(f)$ and $G''(f)$. Note that the actual identification number of the table giving $G''(f)$ must have the identification number *tid+1*. Additional information is found in Chapter 8. An alternate method of input requires that the coefficients *TR* and *TI* be entered instead of $G'(f)$ and $G''(f)$. If you do this, you must also include the Bulk Data entry:

PARAM	**VISCTAB**	**MSC**							

13.2 FREQUENCY DOMAIN EXCITATIONS

UAI/*NASTRAN* provides a large selection of options for specifying dynamic frequency or random excitations. The various capabilities are described in this section.

13.2.1 Frequency Response Loads

Dynamic loads are selected with a **DLOAD** Case Control command, which references **DLOAD**, **RLOAD1** or **RLOAD2** Bulk Data entries. You may combine two or more **RLOADi** entries by indirectly referencing them on a **DLOAD** Bulk Data entry.

The **RLOAD1** entry defines a frequency-dependent load of the form:

$$P(f) = A[C(f) + iD(f)]e^{i(\theta - 2\pi f\tau)} \qquad (13\text{-}48)$$

Similarly, the **RLOAD2** entry defines a frequency-dependent load of the form:

$$P(f) = AB(f)e^{i(C(f) + \theta - 2\pi f\tau)} \qquad (13\text{-}49)$$

13

In each case, *A* is defined using any combination of **DAREA** entries and static load entries. θ is defined on a **DPHASE** entry, and τ is defined on a **DELAY** entry. *B(f)*, *C(f)*, and *D(f)* are frequency-dependent functions which are defined in tabular form on any of the four available **TABLEDi** entries. These entries are explained to you in the **UAI/*NASTRAN Reference Manual***.

The static load coefficients, *A*, may be different for each loaded degree of freedom. Therefore, each load is applied to its corresponding degrees of freedom in the global coordinate system, and assembled into the load vector. You may use SUBCASEs to analyze a number of load cases in a single execution.

13.2.2 Enforced Motion

You may use the same functional forms defined in the previous section to define an enforced motion instead of an applied force. You must enter one of the values **1**, **2**, or **3** in the **DYNEX** field of the **RLOADi** Bulk Data entries to explicitly request that the function be treated as an enforced displacement, velocity, or acceleration, respectively.

☞ The degrees of freedom on which the motion is enforced ***must be placed in the s-set***.

13.2.3 Random Excitations

The sources of excitation in a random response analysis are considered statistically either uncorrelated, or cross-correlated, or auto-correlated. A set of sources which are auto-correlated, sometimes referred to as completely correlated, are defined in a SUBCASE. The degree of cross-correlation among sets of excitations in different SUBCASEs are expressed in terms of power spectral density factors of the form:

$$S_{jk}(f) = (X + iY)G(f) \qquad (13\text{-}35)$$

The subscripts *j* and *k* define the SUBCASE numbers of the cross-correlated excitation sources. If the sources are uncorrelated, only the terms corresponding to equal *j* and *k* must be defined.

13.2.4 Frequency Response Loads for Substructures

Frequency response loads for substructuring are defined using a combination of static load vectors declared in PHASE 1 substructuring runs and dynamic load data defined in a PHASE 2 solution execution. The PHASE 1 static load data perform the role of the **DAREA** data. These data are expanded in the frequency domain during the PHASE 2 execution using the referencing techniques described earlier for normal dynamics analyses. For random analysis, the random loading functions must be defined in the Case Control packet of a PHASE 3 execution.

UAI/*NASTRAN* provides additional Bulk Data entries to assist you in defining substructure loads. The **DAREAS** Bulk Data entry specifies the location of a dynamic load or enforced motion, and a scale factor. The **DELAYS** Bulk Data entry defines the time delay for frequency-dependent loads. The **DPHASES** Bulk Data entry defines the phase lead term for frequency-dependent loads. All of the values are defined in reference to Basic Substructures.

13.3 INPUT DATA REQUIREMENTS

This section describes the input data required for performing frequency and random response analyses. Input data for these analyses are organized to provide an easy transition from a static analysis model. All static analysis modeling options are available. In addition, EXTRA points are introduced for the purpose of coupling non-structural quantities to the structural model. Frequency response analyses require additional information beyond that necessary for statics analysis, specifically:

- Mass data are required.
- You must provide excitation frequencies and the sources of excitation must be defined as a function of frequency.
- Viscous damping, structural damping, and in the modal Rigid Format, modal damping, may be included.
- When you use the modal approach, you must specify a real eigenextraction method, and you may select data to specify frequencies and eigenvalue ranges.

13

13.3.1 Executive Control Commands

The Executive Control commands for direct and modal frequency and random response analyses are specified, for both DMAP Rigid Formats and the MULTI solution sequence, as follows:

```
SOL  8   (Direct Frequency and Random Response Analysis)
SOL 11   (Modal Frequency and Random Response Analysis)
   or
SOL MULTI
   CASE DFREQ
   CASE MFREQ
```

There are no additional Executive Control Commands required for frequency and random response analyses. However, if you wish to plot the output of your analysis, you must use an appropriate **ASSIGN** command to define the plot file.

13.3.2 Substructure Control Commands

Random response analysis for substructures is performed by completing a frequency response analysis in PHASE 2 and specifying the random loading functions in the Case Control packet in a PHASE 3 execution.

13.3.3 Case Control Commands

In addition to the normal modeling control information, there are a few Case Control command requirements for frequency response analyses. The Case Control commands, their relationship to corresponding Bulk Data entries, and the actions taken in each case are shown in Table 13-1.

As was the case for normal modes analysis, you may use the Guyan reduction features to reduce the problem size. Only one boundary condition may be solved in a given execution. Also, only one set of symmetric direct input matrices may be selected with the **B2GG**, **K2GG**, and **M2GG** commands.

Table 13-1. SELECTING FREQUENCY RESPONSE DATA

WHEN USING:	TO SELECT OR MODEL:	YOU USE THE	
		CASE CONTROL COMMAND:	BULK DATA ENTRIES:
THE DIRECT OR MODAL METHOD	Excitation Frequencies	**FREQUENCY**	**FREQ** **FREQ1** **FREQ2** **FREQ3**
	Dynamic Loads and Enforced Motions	**DLOAD**	**DLOAD** **RLOAD1** **RLOAD2**
	Direct Input Matrices At The System Level	**K2GG** **M2GG** **B2GG**	**DMIG**
	Direct Input Matrices For All Physical Dof Including Extra Points	**K2PP** **M2PP** **B2PP**	**DMIG**
	Nonstructural effects	**TFL**	**TF** **EPOINT**
	Specific Modes For Output	**MODES**	
	Viscous Damping Effects		**CVISC,PVISC** **CDAMPi,PDAMP**
	Uniform Structural Damping		**PARAM,G**
	Nonuniform Structural Damping		**MATi** **(GE** Field**)**
	Random Response	**RANDOM**	**RANDPS** **RANDT1** **TABRND1**
	Output Frequencies	**OFREQ**	
THE DIRECT METHOD	Viscoelastic Response	**SDAMPING**	**TABLED1** **PARAM,VISCTAB**
THE MODAL METHOD	Real Eigenmethod	**METHOD**	**EIGR**
	Modal Damping Effects	**SDAMPING**	**TABDMP1**
	Generalized Coordinates	**RESFLEX**[1]	**PARAM,LFREQ** **PARAM,HFREQ** **PARAM,LMODES**
	Mode Acceleration		**PARAM,MODACC**
	Retaining Diagonal Terms Of Non-diagonal Matrices		**PARAM,DIAGNLPP**

1. **RESFLEX** is available only when using the **SOL MULTI** solution sequence.

13

At least one SUBCASE must be defined for each different set of dynamic excitations, excitation frequencies, and direct input matrices, selected with any combination of **DLOAD**, **FREQ**, **B2PP**, **K2PP**, and **M2PP** commands, as shown below.

```
FREQ = 101
SPC  = 10
DISP = ALL
SUBCASE 1
      B2PP = OLDDAMP
      DLOAD = 10
SUBCASE 2
      B2PP = OLDDAMP
      DLOAD = 20
SUBCASE 3
      B2PP = NEWDAMP
      DLOAD = 10
SUBCASE 4
      FREQ = 104
      B2PP = NEWDAMP
      DLOAD = 20
BEGIN BULK
```


When you perform modal frequency response analyses, you must also select a real eigenextraction method with the Case Control command **METHOD**. This data is used to generate the generalized degrees of freedom which represent the model in the modal formulation.

If you wish to perform a random response analysis, you must first satisfy the Case Control requirements described above for the frequency response analysis. You then use the **RANDOM** Case Control command to select the appropriate Bulk Data entries. This command may ***only*** appear in the first SUBCASE of a *group* representing a different set of excitation frequencies or direct input matrices. In the above example, you may ***not*** enter a **RANDOM** command in **SUBCASE 2**. You may specify cross-correlations *only* among the sets of excitations defined in a group. This is done by referencing any number of **RANDPS** Bulk Data entries with the appropriate set identification number. A maximum of 20 such set identification numbers may be specified.

The following example shows how to perform a Modal Frequency Response solution using the modes computed in a modal analysis and select the solution vectors from two statics cases as additional degrees of freedom:

```
SOL MULTI
$
CASE 10 STATICS         First Static Load Case
LOAD = 1000
CASE 20 STATICS         Second Static Load Case
LOAD = 2000

CASE 100 MODES          Normal Modes Case
   METHOD = 1
$
CASE 200 MFREQ
USE MODES = 100         Requests modes from Case 100
SET 20100 = 10,20       Defines a set of statics cases 10 and 20
RESFLEX = 20100         Augments modes with statics results
...
```

13.3.3.1 X-Y Plotter Commands

You may plot the results of your frequency response and random analyses. These include plots of selected responses versus frequency, power spectral density functions of frequency, and autocorrelation functions of time lag. All of the X-Y Plotter commands must be entered as a subpacket of the Case Control packet. Chapter 30 of this manual explains the X-Y Plotting feature in full detail.

13.3.4 Bulk Data Entries

Table 13-1 summarizes the Bulk Data entries which you may use in a frequency and random response analysis. There are a number of Bulk Data entries that are used specifically for this analysis type. These, and the corresponding topics of interest, are explained in the following sections.

13.3.4.1 Excitation Frequencies

When performing a frequency response analysis, you must supply a set of excitation frequencies on any combination of the Bulk Data entries **FREQ**, **FREQ1**, **FREQ2**, and **FREQ3**. You then reference the appropriate set identification number from a **FREQ** Case Control command.

The **FREQ** Bulk Data entry defines an explicit set of frequencies of the form:

FREQ	**FREQID**	**F1**	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**-cont-**
-cont-	**F8**	**F9**	*CONTINUES WITH LIST OF VALUES*						**-cont-**

The **FREQ1** Bulk Data entry defines a starting frequency, frequency increment, and the number of increments desired.

FREQ1	**FREQID**	**FL**	**DF**	**NDF**					

The **FREQ2** Bulk Data entry specifies a lower frequency, an upper frequency, and the number of logarithmic increments desired.

FREQ2	**FREQID**	**FL**	**FU**	**NF**					

The **FREQ3** Bulk Data entry, which is only applicable in the modal method, specifies a range of either natural frequencies or mode numbers from the real eigenvalue analysis performed.

FREQ3	**FREQID**	**FL**	**FU**						
		NL	**NU**						

13.3.4.2 Frequency-Dependent Loads and Enforced Motion

In addition to the excitation frequencies, you must provide a set of frequency-dependent forcing functions or enforced motions to complete your frequency response analysis. These may be supplied using either the **RLOAD1** entry or the **RLOAD2** entry. If you wish to combine the two, you must use a **DLOAD** Bulk Data entry.

The **RLOAD1** and **RLOAD2** entries define frequency-dependent loads or enforced motions whose functional form is as shown in (13-33) and (13-34). The formats of these Bulk Data entries are:

RLOAD1	LID	ADEF	τDEF	θDEF	CTAB	DTAB	DYNEX		

RLOAD2	LID	ADEF	τDEF	θDEF	BTAB	CTAB	DYNEX		

These entries reference other Bulk Data entries such as **DAREA**, **DELAY**, **DPHASE**, and **TABLEDi** to define the appropriate components of the function at the specified degrees of freedom. **ADEF** may reference any static load set which is to be made frequency-dependent.

The **DYNEX** field of the **RLOADi** entries specifies whether the data on the entry is to be used as a forcing function, or as an enforced motion. If you enter a value of **0**, or leave this field blank, you request that the data provided be used as a forcing function. If you enter the values of **1**, **2**, or **3** in this field, you explicitly request that the function be treated as an enforced displacement, velocity, or acceleration, respectively. In these cases, the degrees of freedom to which the motion applies ***must be defined in the s-set***. You may accomplish this by referencing the degrees of freedom on a **SUPORT**, **SUPORTS**, or **SUPAX** entry, as applicable.

13

13.3.4.3 Entries Specific to the Modal Method

In the modal method, you may control the modal representation by specifying either the **LMODES** or the **HFREQ** and **LFREQ** parameters. **LMODES** specifies the number of lowest frequency modes to be used, whereas **LFREQ** and **HFREQ** define a range of frequencies. If you do not specify any preference for the extent of modal representation, all of the extracted real modes will be used in the modal formulation.

The **FREQ3** Bulk Data entry, as discussed in the previous section, specifies a range of either natural frequencies or mode numbers from the real eigenvalue analysis performed.

If you select modal damping with the **SDAMPING** Case Control command, then you must supply a **TABDMP1** Bulk Data entry which defines the frequency-dependent damping values. Note that you must specify damping values which are ***twice*** the critical damping ratio. For example, if you wish to specify a constant 2% critical damping for the all modes between 0. and 100. Hz., then the **TABDMP1** damping data entries must be 0.04.

13.3.4.4 Random Response Excitations

When you perform a random response analysis, you must define all cross-correlated and uncorrelated power spectral density factors, S_{jk} in (13-34), and S_j in (13-33), respectively. To do so, you use **RANDPS** Bulk Data entries:

RANDPS	SID	J	K	X	Y	TID			

All of the set identification numbers, **SID**, must be the same for a group of SUBCASEs representing one set of excitation frequencies and one set of direct input matrices. **SID** is referenced on the **RANDOM** Case Control command which must appear in the first SUBCASE of the group. **J** and **K** are the SUBCASE numbers of the cross-correlated sets of excitation. The complex number represented by **X** and **Y** controls the phase lag of the cross-correlation. For example, if two sets of excitation were completely correlated, there is no phase lag, and **Y** must be zero. On the other hand, if the two excitations are not in phase, **X** may be zero.

Each **RANDPS** entry references a **TABRND1** Bulk Data entry which defines the $G(f)$ in tabular form.

TABRND1	TID	XAXIS	YAXIS						-cont-
-cont-	f1	g1	f2	g2	f3	g3	f4	g4	-cont-

The data entered on this entry are interpolated as needed. You may specify the method by which this is performed along each axis in its appropriate **AXIS** field. The available options are **LINEAR** and **LOG**. The latter is most useful in entering the power spectral density factors corresponding to acceleration, as these are almost always provided in log-log format. You must note that with this option, the values you enter must be positive.

When you desire the output of autocorrelation calculations in the form described in (13-20), you must also enter the appropriate time lag constants, τ, on **RANDT1** entries. You must use the same set identification number which appears on the **RANDOM** Case Control command and the corresponding **RANDPS** entries.

13.4 SOLUTION RESULTS

The following sections describe the solution results available in Frequency Response analysis. Included are discussions of the structural responses and the Residual Flexibility output.

13.4.1 Structural Responses

The available GRID point solution requests are shown in Table 13-2. The Case Control commands **SACCE**, **SDISP**, and **SVELO** are only used when you perform dynamic response analyses using the modal method. All results are available in complex form expressed as real and imaginary components, **RECT**angular, or as magnitude and phase angle, **POLAR**. Solution results are given for the set of frequencies defined on a real or generated **SET** Case Control command, and referenced by the **OFREQUENCY** Case Control command. By default, output is generated for all excitation frequencies specified. They may be sorted by frequency, using the **SORT1** request, or by GRID point or element identification number for all frequencies, using the **SORT2** request.

When using the modal method, you obtain an eigenvalue summary and extraction data automatically. Other solution results are the same as in the direct method. When solution point output is requested, the modal degrees of freedom will be identified by their mode number, which corresponds to the eigenvalue extraction list.

Solution results for random response analyses consist of power spectral density functions of u_j quantities. u_j may be displacements, velocities, accelerations, reactions, element forces, or stresses. Autocorrelation functions for these quantities may also be output. ***All output for random response is requested using the X-Y Plotter features of*** **UAI*/NASTRAN***. The frequency response analysis results are also available in X-Y Plotter output format.

X-Y plot and print requests are made in the X-Y Plotter subpacket of the Case Control packet. The X-Y Plotter commands are described in Chapter 30 of this manual.

Table 13-2. CASE CONTROL COMMANDS FOR RESPONSE

COMMAND	FUNCTION
ACCE	Acceleration in the physical, or *a-set*.
DISP	Displacement, or response, in the physical, or *a-set*.
VELO	Velocity in the physical, or *a-set*.
SACCE	Acceleration in the solution, or *h-set*.
SDISP	Displacement, or response, in the solution, or *h-set*.
SVELO	Velocity in the solution, or *h-set*.

13.4.2 Residual Flexibility

Several levels of output for the Strain Energy Checks of the residual are available. In the **SUMMARY** output, Table 13-3a, the fraction of the total strain energy for each residual vector that is represented in the modal approximation of each vector is displayed. At the **FULL** output level, the additional items shown in Table 13-3b are displayed. Included are the modal and a-set strain energy values and the fraction of the strain energy represented in each normal mode. A running sum of the fraction is provided to assist in modal cut-off selection. Finally, at the **DETAIL** output level, Table 13-3c, the modal participation factors, ζ, defined by (13-25), are printed for each residual vector. These modal participation factors show the contribution of each normal mode in the approximation of the residual vector.

13.4.2.1 Residual Flexibility Strain Energy Interpretation

The residual vector Strain Energy computations provide a quantitative assessment of the ability of the residual vector to approximate additional displacement fields. If the Strain Energy of a residual vector is captured in the selected set of normal modes, the residual vector provides nothing new to the generalized set of coordinates. A Strain Energy fraction of 100%, would imply complete redundancy — an event that would trigger Gram-Schmidt and automatic removal under the default behavior.

The strain energy checks can be used in two ways to assist in the selection of the generalized degrees of freedom for a modal analysis. First, the set of residual vectors can be chosen such that they are poorly represented by the existing modes. This means that only a small fraction of their strain energy is in the modal approximation. A second application is to allow reduction of the set of normal modes if a particular (dominant) displacement response has a high fraction of strain energy in a smaller set of modes than the full retained set. Thus, by examining the strain energy output, the appropriate trades between the frequency range of accurate modal approximation and eigenvectors and residual vectors can be made.

To make best use of these tools, you should use the **MEFFMASS** and **RESFLEX** output options in a series of modal analyses, residual flexibility augmentations and a small subset of direct analyses until the proper set of generalized degrees of freedom have been identified. Then, you may proceed with further analyses and optimization with confidence in the behavior of the reduced model.

Table 13-3. RESIDUAL FLEXIBILITY OUTPUT RESULTS

a. SUMMARY Output Option

```
  M O D A L   S T R A I N   E N E R G Y   S U M M A R Y


              TOTAL STRAIN ENERGY FRACTION
FOR RESIDUAL VECTORS CREATED BY SUBCASES OF STATIC ANALYSIS

                      SUBCASE   11
                      9.897573E-01
```

b. FULL Output Option

```
                 MODAL STRAIN ENERGY

                    SUBCASE   11
                    2.596243E+00


              RESIDUAL VECTOR STRAIN ENERGY

                    SUBCASE   11
                    2.623111E+00



             MODAL STRAIN ENERGY FRACTION

MODE    FREQUENCY                SUBCASE   11
 NO.        HZ.              FRACTION            SUM

   1    3.084016E+01      8.506253E-01      8.506253E-01
   2    1.199501E+02      1.005197E-01      9.511450E-01
   3    2.701229E+02      2.852983E-02      9.796749E-01
   4    3.267094E+02      3.196212E-27      9.796749E-01
   5    4.870909E+02      1.008242E-02      9.897573E-01
   6    5.075099E+02      8.917952E-15      9.897573E-01
```

c. DETAIL Output Option

```
    MODAL PARTICIPATION FACTORS

MODE    FREQUENCY         SUBCASE   11
 NO.        HZ.

   1    3.084016E+01     -1.090174E-02
   2    1.199501E+02     -9.635370E-04
   3    2.701229E+02     -2.279458E-04
   4    3.267094E+02      6.308115E-17
   5    4.870909E+02      7.514772E-05
   6    5.075099E+02      6.783151E-11
```

13.5 MODELING GUIDELINES

The important factors which influence accuracy and computer resource requirements for dynamic analyses are described in this section. It is difficult to put forth a set of *cook book* rules, because it is not possible to formulate such rules to cover the broad range of dynamic analyses that may be performed with **UAI*/NASTRAN***. The following sections present a set of general guidelines for modeling and discuss the relative advantages and disadvantages of the two methods.

13.5.1 General

In modeling structural dynamics problems, the objectives of the analysis will help to determine what types of analyses are required. If the forced response of the structure is required, then it must be determined whether or not the dynamic force is periodic or transient. In the case of periodic force, the force must be broken down into its harmonic components by Fourier analysis in order to perform a forced harmonic calculation. If the force cannot be decomposed into its harmonic components or it is transient, then the response must be calculated by transient analysis, presented to you in the next Chapter.

In either case, it is recommended that a modal analysis be performed first to identify the natural frequencies and mode shapes in order to give insight into the forced-response results. If possible, the natural frequencies and mode shapes should be verified prior to continuing with a forced-response analysis. Having the natural frequencies and mode shapes is a virtual necessity for the interpretation of the forced-response results.

The following set of guidelines can be used in developing your finite element model for forced-response analyses of structures. These guidelines are ***not*** in any sense complete, and they should be viewed as complementary to the engineering judgement of a finite element analyst.

- ❒ You should remember that dynamic finite element analysis of a structure involves, in essence, two models for analysis: a global model which you input, and a reduced model which is used in the actual computation. It is therefore important to ***ensure*** that the global model contains sufficient details to characterize the stiffness and mass properties.
- ❒ The reduced model should have degrees of freedom at least ***twice*** the highest mode of interest. The dynamic degrees of freedom should be placed in areas of large mass and rigidity. They should be distributed in such a manner as to describe the anticipated mode shapes. These degrees of freedom do not have to be selected as constraint points, as long as the constraints are permanent. For plate type structures, the dynamic degrees of freedom in the *out-of-plane* direction should be emphasized. In addition, the rotational degrees of freedom are unnecessary.
- ❒ Symmetry cannot be used effectively for dynamic models as it is for static models. In most structures, there are asymmetric modes that may be missed if symmetry is assumed. The use of symmetry virtually requires the details of the mode shapes to be known in advance to prevent missing modes.

- The basic consideration in the selection of an appropriate finite element model of a dynamics problem is that only the lowest modes or a few intermediate modes are being excited by the load vector. Therefore, there is no need to represent the higher frequencies of the actual system in the finite element model, because the dynamic contribution in these frequencies is negligible.
- As a rule of thumb, if frequencies below F_l, are contained in the loading, then the finite element mesh should at most represent frequencies to $4F_l$ of the actual system.
- In cases where the dynamic displacement response is the only required response, a coarser mesh may be used. The amount of detail to include in the global model will depend whether or not dynamic stresses are of concern, and at what location.
- In the modal method, the structure considered, and the spatial distribution and frequency content of the loading, determine the number of modes to be used. The higher the modes of interest, the more detailed the model needs to be. For instance, with earthquake type loading, only the first 10 modes may be considered. For blast or shock loading, as many as $2n/3$ modes (where n is the total number of natural modes) need to be considered.

13.5.2 Comparing the Direct and Modal Methods

Both direct and modal methods for solution of frequency response problems are available in **UAI/*NASTRAN***. The direct method uses the actual GRID, SCALAR, and EXTRA point degrees of freedom to define the solution matrices. The modal method, however, transforms the structural degrees of freedom into generalized coordinates representing displacements of the normal modes or static solution vectors.

The direct method offers several advantages. Since all degrees of freedom are considered in the solution, high accuracy is obtained when a wide spectrum of frequencies is excited, as in ***impact*** problems. All options for damping, initial conditions, and nonlinear effects are provided, and no extra transformations need to be performed on the load and displacement vectors.

The modal method on the other hand, provides cost effective solutions when modal degrees of freedom are uncoupled. This is the case when coupled damping and direct matrix input data are not used, or when the diagonal terms of the transformed matrices are retained. This method is advantageous also when output for a large number of frequencies is required. The output of modal frequencies and mode shape vectors allows the identification of significant modes and facilitates the understanding of the dynamic responses.

Experience has shown that the modal method has been the practical choice for most structural analyses involving low frequency excitation and a large number of frequency solutions. The direct method is useful for special modeling requirements or when the modal formulation is otherwise inadequate. Modal response can be improved by the selection of appropriate residual flexibility vectors.

With both methods, the use of Guyan reduction, selected with the Case Control command **AUTOOMIT** or with **OMITi** and **ASETi** Bulk Data entries, is a valuable tool. This option performs condensations of the stiffness, mass, damping, direct input, and load matrices. The result is a set of smaller solution matrices for both real eigenvalue solutions and direct formulation response problems. However, for the direct method, Guyan reduction may significantly increase the bandwidth of the matrices and cause a less efficient solution.

A dynamic reduction procedure is provided for improved accuracy relative to the static reduction method. Dynamic reduction is requested using the Case Control command **DYNRED**, which in turn references a **DYNRED** Bulk Data entry. The maximum frequency range of interest is specified with this data entry. **UAI/*NASTRAN*** will automatically create **SCALAR** points which represent the displacement of generalized coordinates created by the procedure. Each generalized coordinate represents a flexible deformation of the *f-set* of the structure in the frequency range. As many generalized coordinates as are required for accuracy will be developed unless a limit is specified on the **DYNRED** Bulk Data entry.

13.6 EXAMPLE PROBLEMS

This section presents two example problems using both the direct method and the modal method of solving frequency response problems. In general, you should use the direct method if it is known in advance that the modal solution is inadequate, or in situations where a wide spectrum of frequencies are likely to be excited, as in impact problems.

The two example problems illustrate the use of both solution methods for determining the structural response to steady-state sinusoidal loads. The applied load is given in terms of complex values which reflect the amplitudes and phases at each selected frequencies. The steady-state response of the structure at each frequency is requested in **POLAR** format which corresponds to the magnitude and phases of the results.

13.6.1 The Direct Method

The first example problem illustrates the use of the direct method of determining the structural response.

Example Problem 13-1

A square plate is supported by hinges along its edges. The length of each side is 20 in, and the thickness of the plate is 2.0 in. The applied load is sinusoidally distributed over the panel and increases with respect to frequency, defined by the following function:

$$P(x, y, f) = (10.0 + 0.3f)\cos\frac{\pi x}{2a}\cos\frac{\pi y}{2b}$$

Only vertical displacements and bending rotations are allowed. You wish to determine the steady-state frequency response of the structure for frequencies between 0 and 10 Hz using the direct method of solution.

The finite element model and its engineering data are shown in Figure 13-1. The input data for this problem is available to you in file **FREQ1**. Since the plate is symmetric, only one quarter of the structure has been modeled with QUAD4 elements. This was done with the knowledge that any unsymmetric modes will be neglected. In general, you should be cautious when using symmetry for dynamic analyses.

The boundary conditions are input using **SPC1** Bulk Data entries, referenced by an **SPC** Case Control Command. The exterior edges of the model have hinged supports. The interior edges have planes of symmetry.

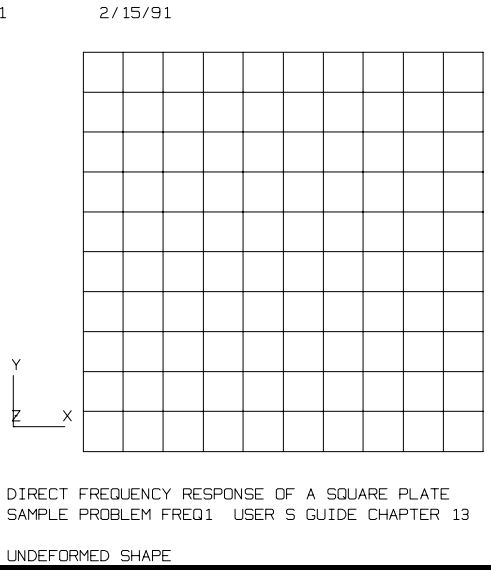
The excitation frequencies were generated using a **FREQ1** Bulk Data entry, referenced by a **FREQUENCY** Case Control command. In the specified frequency range, 110 excitation frequencies were chosen for computation.

The dynamic loads are defined on **RLOAD1** Bulk Data entries which reference other Bulk Data entries such as **DAREA** and **TABLED1**. The forcing function is in sinusoidal form, which eliminates the need for using **DELAY** and **DPHASE** Bulk Data entries. Therefore, the frequency response load defined by the **RLOAD1** reduces to the form:

$$P(x, y, f) = A(x, y)\,C(f)$$

Figure 13-1. FINITE ELEMENT MODEL

t = 2.0 in
E = 3.0 x 10^7 **psi**
ν **= 0.3**
$\mathbf{m_{ns}}$ **= 13.6 lbm/in^2**



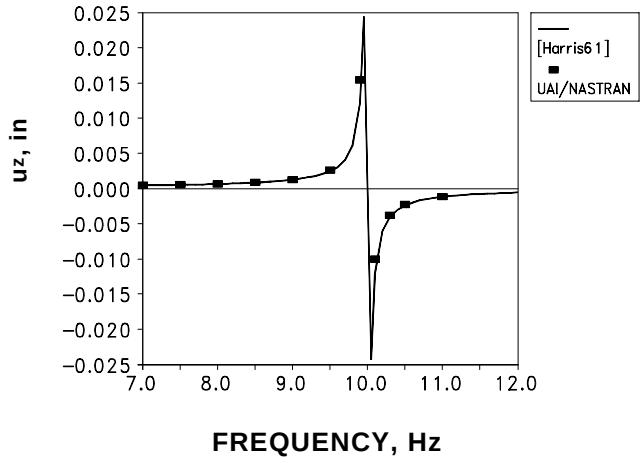
The Bulk Data entry **DAREA** is used for defining $A(x,y)$ at all GRID points, using only the sinusoidal terms of the given forcing function. The Bulk Data entry **TABLED1** is used to define $C(f)$.

The theoretical solution for this is found in [Harris61]. The excitation of the plate is orthogonal to the theoretical first mode. The responses are defined by:

$$u_z(f) = \frac{F(f)}{(2\pi)^2 \mu (f_1^2 - f^2)}$$

where $f_1 = 10Hz$ is the first natural frequency of the model, and μ is the nonstructural mass. Figure 13-2 shows the comparison between theoretical and **UAI/*NASTRAN*** results in the range 7 Hz to 11 Hz. Note that at $f = f_1$, the theoretical solution is discontinuous.

Figure 13-2. RESPONSE HISTORY



13.6.2 The Modal Method

The second example problem illustrates the use of the modal method of determining the structural response. In the modal method, the structural degrees of freedom used in the solution are the uncoupled modal displacements. The solution equations are simple and efficient. The savings in time, however, may be offset by the operations necessary to extract the modes, transform the loads to modal coordinates, and transform modal displacements to structural displacements. The savings are typically proportional to the number of excitation frequencies.

This problem is derived directly from the previous example problem. It illustrates the ease with which a direct frequency response model may be adapted for a modal analysis. In addition, it exhibits how different dynamic loads may be added together for a single analysis.

The input data for this problem is available to you in file **FREQ2**. The model in the previous example problem is used to perform a modal analysis. For this purpose, a **GIV**ens eigenvalue extraction method is chosen to extract the normal modes of the model. The modes which are extracted from the specified frequency range are used to represent the model as generalized coordinates. The applied dynamic loads are defined using both **RLOAD1** and **RLOAD2** Bulk Data entries. These are then combined with a **DLOAD** Bulk Data entry referenced on a **DLOAD** Case Control command. You will note that the formats of the **RLOADi** entries for this simple example are identical.

The results of the modal analysis are identical to those of the direct analysis. However, due to the relatively large number of excitation frequencies, a savings of over 80% of the CPU time was achieved over the direct analysis.

Chapter 14

TRANSIENT RESPONSE ANALYSIS

UAI/*NASTRAN* provides analysis capabilities for exciting a structural model with both time-dependent forces and boundary motions. This capability is provided in two Rigid Formats. Rigid Format 9 solves the equations of motion using a direct solution of the matrix equations for each time step, the *direct* method, and Rigid Format 12 transforms the equations of motion of the structural portion of the model to modal coordinates before solving the equations, the *modal* method. The Chapter describes the procedures and options used for solving transient response problems.

14.1 MATHEMATICAL BACKGROUND

Transient response analysis provides a direct simulation of the dynamic response of the system occurring due to initial conditions and time-varying excitation. The equations of motion for transient response are:

$$\boldsymbol{M}_{dd}\ddot{\boldsymbol{u}}_d(t) + \boldsymbol{B}_{dd}\dot{\boldsymbol{u}}_d(t) + \boldsymbol{K}_{dd}\boldsymbol{u}_d(t) = \boldsymbol{P}_d(t) + \boldsymbol{N}_d(t) \tag{14-1}$$

where $\boldsymbol{P}_d(t)$ is the applied load as a function of time. The nonlinear load vector, $\boldsymbol{N}_d(t)$, is calculated using functions of displacement or velocity which you define. This vector is not determined by any nonlinear material effects. The matrices $\boldsymbol{K}_{dd}$, $\boldsymbol{B}_{dd}$, and $\boldsymbol{M}_{dd}$ may be nonsymmetric.

The results of transient analyses are the displacement vectors and their first and second time derivatives, velocities and accelerations, respectively. The degrees of freedom in these vectors may include EXTRA points, *e-set*, as well as structural degrees of freedom which have been reduced to the analysis set, *a-set*. The resulting set of degrees of freedom is called the dynamic set, *d-set*.

As is the case for all dynamic response solutions in **UAI/*NASTRAN***, you may employ either the direct method or the modal method in solving the equations of motion. An overview of the mathematical approach to each of these is described in the following sections.

14

14.1.1 Direct Method

In the direct method, the definition of the $\boldsymbol{K}_{dd}$, $\boldsymbol{M}_{dd}$, and $\boldsymbol{B}_{dd}$ matrices are:

$$\boldsymbol{K}_{dd} = \boldsymbol{K}^1_{dd} + \boldsymbol{K}^2_{dd} \tag{14-2}$$

$$\boldsymbol{M}_{dd} = \boldsymbol{M}^1_{dd} + \boldsymbol{M}^2_{dd} \tag{14-3}$$

$$\boldsymbol{B}_{dd} = \boldsymbol{B}^1_{dd} + \boldsymbol{B}^2_{dd} + \frac{G}{\omega_3}\boldsymbol{K}^1_{dd} + \frac{1}{\omega_4}\boldsymbol{K}^4_{dd} \tag{14-4}$$

The $\boldsymbol{K}^1_{dd}$ and $\boldsymbol{M}^1_{dd}$ matrices correspond to stiffness and mass, respectively. $\boldsymbol{B}^1_{dd}$ and $\boldsymbol{K}^4_{dd}$ represent viscous and non-uniform structural damping. These are all reduced from the matrices generated from the finite element model and the symmetric matrices that you have input directly using the **K2GG**, **M2GG**, and **B2GG** Case Control commands. G is the uniform structural damping coefficient. $\boldsymbol{K}^2_{dd}$, $\boldsymbol{B}^2_{dd}$, and $\boldsymbol{M}^2_{dd}$ are reduced from those matrices input directly by you using the **K2PP**, **M2PP**, **B2PP**, or **TFL** Case Control commands.

Recall that the stiffness matrix with structural damping is written as:

$$\boldsymbol{K}_{dd} = (1 + i\,G)\,\boldsymbol{K}^1_{dd} + \boldsymbol{K}^2_{dd} + i\,\boldsymbol{K}^4_{dd} \tag{14-5}$$

The imaginary structural damping matrices $iG\boldsymbol{K}^1_{dd}$ and $i\boldsymbol{K}^4_{dd}$ must be converted to equivalent real damping terms by introducing ω_3 and ω_4. These represent the frequencies at which the

equivalent viscous damping matrices, $\frac{G}{\omega_3}\boldsymbol{K}_{dd}^{1}$ and $\frac{1}{\omega_4}\boldsymbol{K}_{dd}^{4}$ will produce the same magnitude of damping force.

Note that in a frequency response analysis, the structural damping terms are multiplied by the displacements, whereas in a transient analysis, they are multiplied by the velocity. This simulation of structural damping using viscous damping is only an approximation because viscous damping forces are larger at higher frequencies, those above ω_3 and ω_4, and smaller at lower frequencies, those below ω_3 and ω_4. Therefore, the quantities ω_3 and ω_4 are frequently selected to be at the center of the frequency range of interest or located at a critical, or dominant, mode. A small value of G/ω_3 is useful to insure stability of higher modes in a nonlinear transient analysis.

14.1.2 Modal Method

For many problems, an efficiency improvement for solving the transient response problem may be realized through the application of the modal method. In this method, the natural modes of the structure, within a specified frequency range, are used as generalized degrees of freedom. This greatly reduces the problem size while retaining accuracy in the selected frequency range. Using the modal method, the equation:

$$\left[\boldsymbol{K}_{aa} - \lambda\,\boldsymbol{M}_{aa}\right]\boldsymbol{\Phi}_a = \boldsymbol{0} \tag{14-6}$$

is solved to determine the natural modes of the system. The matrices $\boldsymbol{K}_{aa}$ and $\boldsymbol{M}_{aa}$ are reduced from $\boldsymbol{K}_{dd}^{1}$ and $\boldsymbol{M}_{dd}^{1}$, and do not consider any damping effects which may be present. The unknown degrees of freedom, $\boldsymbol{u}_d$, are then related to these modes by the following relationship:

$$\boldsymbol{u}_d = \begin{Bmatrix} \boldsymbol{u}_a \\ \boldsymbol{u}_e \end{Bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_{ai} & \\ & I \end{bmatrix} \begin{Bmatrix} \boldsymbol{q}_i \\ \boldsymbol{q}_e \end{Bmatrix} = \boldsymbol{\Phi}_{dh}\,\boldsymbol{q}_h \tag{14-7}$$

where *i* represents the modal coordinates computed and retained for the response analysis. The manner in which you may control the extent of the modal representation will be explained later. $\boldsymbol{q}_h$ represents the modal generalized coordinates in the *h-set*. This includes both the selected modal coordinates and the EXTRA points. Substituting (14-7) in (14-1), and premultiplying the relation by $\boldsymbol{\Phi}_{dh}^{T}$ will result in the transient respose equations of motion being transformed to the modal format:

$$\boldsymbol{M}_{hh}\,\ddot{\boldsymbol{q}}_h(t) + \boldsymbol{B}_{hh}\,\dot{\boldsymbol{q}}_h(t) + \boldsymbol{K}_{hh}\,\boldsymbol{q}_h(t) = \boldsymbol{P}_h(t) + \boldsymbol{N}_h(t) \tag{14-8}$$

Since $\boldsymbol{\Phi}_{dh}$ is orthogonal with respect to $\boldsymbol{M}_{dd}^{1}$, the resulting transformed matrix, $\boldsymbol{m}$, is a diagonal matrix, and its terms, m_i, are called the *modal* masses. The terms of the diagonal *modal stiffness*, $\boldsymbol{k}$, and *modal damping*, $\boldsymbol{b}$, matrices are then defined as:

$$k_i = (2\pi f_i)^2\, m_i \tag{14-9}$$

$$b_i = (2\pi f_i)\ g(f_i)\, m_i \tag{14-10}$$

where f_i represents the frequency of each mode, and $g(f_i)$ are the optional modal damping factors which you may specify.

Repeating the same transformation for (14-2) through (14-4), results in the definitions of the *h-set* size matrices:

$$\boldsymbol{K}_{hh} = \boldsymbol{k} + \boldsymbol{\Phi}_{dh}^T \boldsymbol{K}_{dd}^2 \boldsymbol{\Phi}_{dh} \tag{14-11}$$

$$\boldsymbol{M}_{hh} = \boldsymbol{m} + \boldsymbol{\Phi}_{dh}^T \boldsymbol{M}_{dd}^2 \boldsymbol{\Phi}_{dh} \tag{14-12}$$

$$\boldsymbol{B}_{hh} = \boldsymbol{b} + \boldsymbol{\Phi}_{dh}^T \boldsymbol{B}_{dd}^1 \boldsymbol{\Phi}_{dh} + \boldsymbol{\Phi}_{dh}^T \boldsymbol{B}_{dd}^2 \boldsymbol{\Phi}_{dh} + \frac{G}{\omega_3}\boldsymbol{k} + \frac{1}{\omega_4}\boldsymbol{\Phi}_{dh}^T \boldsymbol{K}_{dd}^4 \boldsymbol{\Phi}_{dh} \tag{14-13}$$

Note that while the term corresponding to $\boldsymbol{K}_{dd}^1$ is replaced by $\boldsymbol{k}$, the term corresponding to $\boldsymbol{B}_{dd}^1$ may not be a diagonal matrix. This depends on whether the viscous damping matrix is a linear combination of the stiffness and mass matrices. In general, the transformed direct input and damping matrices are not diagonal. However, you may choose to retain only the diagonal terms of the transformed matrices. This can be accomplished by using the:

PARAM	**DIAGNLPP**	**YES**							

Bulk Data entry, and will result in diagonal, and hence, uncoupled equations of motion. A second method, used only in the modal approach to Transient Response analysis, is a procedure wherein the equivalent viscous damping may be set independently for each modal degree-of-freedom. This method, which uses **PARAM W4MODAL**, is provided to convert the structural damping from imaginary stiffness terms to real viscous terms. See Chapter 9 for details.

14.1.2.1 Mode Acceleration

When you select the modal method, you typically exclude a range of higher frequency modes from the model. While this process, called *mode truncation*, has a minimal effect on the accuracy of the response accelerations, it may adversely affect the accuracy of the element solution results which are recovered from the displacements. This is due to the omission of stiffness and damping forces which may have considerable contributions from the higher modes.

You may use the mode acceleration option to alleviate this problem. In this procedure it is assumed that the solution accelerations and velocities are more accurate than the displacements. Therefore, (14-1) is rearranged, using the accelerations and velocities as known quantities:

$$\boldsymbol{K}_{dd}^1 \bar{\boldsymbol{u}}_d(t) = \boldsymbol{P}_d(t) + \boldsymbol{N}_d(t) - [\ \boldsymbol{M}_{dd}\ddot{\boldsymbol{u}}_d(t) + \boldsymbol{B}_{dd}\dot{\boldsymbol{u}}(t)_d + \boldsymbol{K}_{dd}^2 \boldsymbol{u}_d(t)\ ] \tag{14-14}$$

Note that the effects of structural damping are omitted. $\bar{\boldsymbol{u}}_d$ represents the improved displacements. The right hand side of (14-14) may be considered as an equivalent set of loads of the form:

$$\bar{P}_d(t) = P_d(t) + N_d(t) - [\, M_{dd}\,\ddot{u}_d(t) + B_{dd}\,\dot{u}_d(t) + K^2_{dd}\,u_d(t)\,] \tag{14-15}$$

Then a static solution is performed at each time point to solve for the improved displacements:

$$K^1_{dd}\,\bar{u}_d = \bar{P}_d \tag{14-16}$$

The improved displacements are then used to recover the element data. You may use the **OTIME** Case Control command to select an output set of time points to minimize the computational costs associated with this procedure.

14.1.3 Externally Defined Initial Conditions

The transient response solution often requires the specification of initial conditions prior to integration. Initial conditions for transient response problems may be specified for both the direct and modal methods in **UAI/*NASTRAN***.

Generally, initial condition data are input using **TIC** Bulk Data entries. Additionally, you may save Static Analysis results in **TIC** Bulk Data format using the **DISPLACEMENT** Case Control command:

```
DISP(IC=n)
```

where *n* is a set identification number that will be referenced in a subsequent analysis which uses the displacements as initial conditions.

In addition to the **TIC** Bulk Data, **UAI/*NASTRAN*** provides an even more convenient method for defining initial velocities. This is done using **TICTV** and **TICRV** Bulk Data entries.

14.1.3.1 Direct Method

In the direct method, the initial conditions which you select with the **IC** command in the Case Control packet are specified as the values of physical displacements, u, and velocities, $\dot{u}$, at time zero. Accelerations are *always* assumed to be zero at time zero. All *unspecified* initial conditions are assumed to be zero, so you must provide any desired initial conditions. Initial conditions specified for points that are eliminated by constraints or by reductions are ignored.

14.1.3.2 Modal Method

In the modal method you can also specify initial displacements and velocities either in physical coordinates or in modal coordinates. As in the direct method all unspecified initial conditions are assumed to be zero and all the initial conditions specified for points that are eliminated by constraints or by reductions are ignored.

Usually, you will find it convenient to specify initial condtions in physical coordinates rather than in modal coordinates. A least squares procedure is used in **UAI/*NASTRAN*** to convert the initial conditons from physical coordinates to modal coordinates. The relationship between the displacements in physical and modal coordinates is given by:

$$u_d(\,t=0\,) = \Phi_{dh}\,u_h(\,t=0\,) \tag{14-17}$$

where $\boldsymbol{u}_d$ is the displacement vector in the d-set, $\boldsymbol{u}_h$ is the displacement vector in the *h-set*, and $\boldsymbol{\Phi}_{dh}$ is the matrix of eigenvectors. Pre-multiplying (14-17) by $\boldsymbol{\Phi}_{dh}$ and rearranging terms results in:

$$\boldsymbol{u}_h(t=0) = [\boldsymbol{\Phi}_{dh}^T \boldsymbol{\Phi}_{dh}]^{-1} \boldsymbol{\Phi}_{dh}^T \boldsymbol{u}_d(t=0) \tag{14-18}$$

14.1.4 Internally Computed Initial Conditions

You may use the static equilibrium due to the applied loading at *t = 0* as the initial condition by including the **EQUIL** key word in the **IC** case control command. Again there is a different approach for the two methods, as described in the following sections.

14.1.4.1 Direct Method

When you use the direct method, the following equation is solved at *t = 0* for the static equilibrium:

$$\begin{bmatrix} \boldsymbol{M}_{ll} & \boldsymbol{M}_{lr} \\ \boldsymbol{M}_{rl} & \boldsymbol{M}_{rr} \end{bmatrix} \begin{Bmatrix} \ddot{\boldsymbol{u}}_l \\ \ddot{\boldsymbol{u}}_r \end{Bmatrix} + \begin{bmatrix} \boldsymbol{K}_{ll} & \boldsymbol{K}_{lr} \\ \boldsymbol{K}_{rl} & \boldsymbol{K}_{rr} \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}_l \\ \boldsymbol{u}_r \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{P}_l \\ \boldsymbol{P}_r \end{Bmatrix} \tag{14-19}$$

assuming that $\boldsymbol{u}_r = \boldsymbol{0}$ and $\ddot{\boldsymbol{u}}_l = \boldsymbol{K}_{ll}^{-1} \boldsymbol{K}_{lr} \ddot{\boldsymbol{u}}_r$.

14

14.1.4.2 Modal Method

When the modal method is used and the equations of motion are uncoupled and there are no nonlinear loads, the equations of motion may be written:

$$\begin{bmatrix} \boldsymbol{M}_o & \\ & \boldsymbol{M}_i \end{bmatrix} \begin{Bmatrix} \ddot{\boldsymbol{q}}_o \\ \ddot{\boldsymbol{q}}_i \end{Bmatrix} + \begin{bmatrix} 0 & \\ & \boldsymbol{K}_i \end{bmatrix} \begin{Bmatrix} \boldsymbol{q}_o \\ \boldsymbol{q}_i \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{P}_o \\ \boldsymbol{P}_i \end{Bmatrix} \tag{14-20}$$

and solved at *t = 0*. Where the subscript *o* denotes the rigid body mode degrees of freedom and *i* denotes the flexible mode degrees of freedom.

Now, it is assumed that $\boldsymbol{u}_r = \ddot{\boldsymbol{q}}_i = \boldsymbol{0}$ and noting that:

$$\begin{Bmatrix} \boldsymbol{u}_l \\ \boldsymbol{u}_r \end{Bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_{lo} & \boldsymbol{\Phi}_{li} \\ \boldsymbol{\Phi}_{ro} & \boldsymbol{\Phi}_{ri} \end{bmatrix} \begin{Bmatrix} \boldsymbol{q}_o \\ \boldsymbol{q}_i \end{Bmatrix} \tag{14-21}$$

The lower partition of (14-21) is now a constraint which may be written:

$$\boldsymbol{\Phi}_{ro} \boldsymbol{q}_o = -\boldsymbol{\Phi}_{ri} \boldsymbol{q}_i \tag{14-22}$$

The initial conditions may now be stated:

$$\ddot{\boldsymbol{q}}_o = \boldsymbol{M}_o^{-1} \boldsymbol{P}_o \ , \ \boldsymbol{q}_i = \boldsymbol{K}_i^{-1} \boldsymbol{P}_i \ \text{ and} \tag{14-23}$$

$$\boldsymbol{q}_o = \left[\boldsymbol{\Phi}_{ro}^T \boldsymbol{\Phi}_{ro} \right]^{-1} \boldsymbol{\Phi}_{ro}^T \boldsymbol{\Phi}_{ri} \boldsymbol{q}_i \tag{14-24}$$

Note that when you request the use of equilibrium as the initial condition, then any **TIC** Bulk Data entries are ignored.

For the uncoupled modal method, the calculation and use of static displacements as initial conditions can be also requested by using the parameter **NRB** which may be specified with the Bulk Data entry :

PARAM	**NRB**	***n***							

where ***n***, the number of rigid body modes, must be either zero or, a positive number. This alternate form of input remains for compatibility with earlier versions of **UAI/*NASTRAN***.

Also note, that in the coupled case the displacements and accelerations calculated from (14-19) are converted to modal coordinates using (14-18).

14.1.5 Time Integration and Stability

The equations of motion in transient analyses may be either coupled or uncoupled. The latter are solved analytically and therefore, are potentially far less expensive. The coupled equations, however, must be solved using the available time integration methods.

When you use the modal method, the equations are normally uncoupled because all of the matrices are diagonal. Although this is not necessarily the case when direct input matrices are present, **UAI/*NASTRAN*** allows you to retain the diagonal terms of these matrices when they are transformed to modal coordinates, thus uncoupling the equations.

In all other cases, and cases where nonlinear terms are present, the equations of motion are coupled. The coupled equations are solved by a numerical integration algorithm in **UAI/*NASTRAN***. This ***implicit*** time integration method is based on the ***Newmark*** β-***Method***, with an equivalent $\beta = \frac{1}{3}$.

It is important to select an optimal time step size in integrating the coupled equations of motion. The time step must be small enough to obtain reasonable accuracy and stability in the solution, and large enough to avoid excessive usage of resources. A stable integration method prevents any solution round-off errors from growing in successive operations. The integration method used in **UAI/*NASTRAN*** assures ***unconditional*** stability when there are no nonlinear forces. Therefore, the criteria for selecting an appropriate time step are the desired accuracy and the required resources. When nonlinear forces are present, stability is very sensitive to the time step size. In some case, the use of **DIAG 10**, discussed in Section 14.2.4, can improve the solution stability.

A reasonable estimate of the time step, Δt, may be obtained from:

$$\Delta t = \frac{T_n}{10} \tag{14-25}$$

where T_n is the period associated with the highest frequency in the range of interest.

14.1.6 Transfer Functions

Several options, outlined below, are provided for modeling nonstructural dynamic effects including transfer functions and direct input matrices. For convenience, and to avoid conflicts with static analyses, additional degrees of freedom, called EXTRA points, may be introduced on **EPOINT** Bulk Data entries for the dynamics Rigid Formats. The EXTRA points are placed in the *e-set*.

One example of the use of EXTRA points is for structural control system modeling. In order to facilitate the treatment of control systems, **UAI*/NASTRAN*** includes an input data format for the specification of a transfer function in the form:

$$\left(B_o + B_1 p + B_2 p^2\right) u_d + \sum_i \left(A_o(i) + A_1(i) p + A_2(i) p^2\right) u_i = 0 \qquad (14\text{-}26)$$

which is interpreted as a differential equation with terms in the rows of $\boldsymbol{M}$, $\boldsymbol{B}$, and $\boldsymbol{K}$ corresponding to u_d. The u_d and u_i may either be EXTRA points or structural points defined using **GRID** or **SPOINT** Bulk Data entries.

Structural loads proportional to displacements at EXTRA points are represented by adding terms in the stiffness matrix at the intersections of rows corresponding to the structural points and columns corresponding to the EXTRA points. Direct input matrix capability is provided for this purpose. The superposition of all such terms, including those generated by transfer functions, are called the direct input stiffness matrix, $\boldsymbol{K}^2$, mass matrix, $\boldsymbol{M}^2$, and damping matrix, $\boldsymbol{B}^2$. Terms in the direct input matrices may refer to EXTRA points, structural points, or both. Since dynamic loads may also be applied to EXTRA points, it is possible to completely simulate the subsystems by means of EXTRA points.

14.2 TIME DOMAIN EXCITATIONS

UAI/*NASTRAN* provides a large selection of options for specifying dynamic time domain excitations. The various capabilities are described in this section.

14.2.1 Transient Response Loads

Dynamic loads are selected with a **DLOAD** Case Control command, which references **DLOAD**, **TLOAD1** or **TLOAD2** Bulk Data entries. You may combine two or more **TLOADi** entries by indirectly referencing them on a **DLOAD** Bulk Data entry.

The **TLOAD1** entry defines a time-dependent load of the form:

$$P(t) = A\,F(t-\tau) \tag{14-27}$$

Similarly, the **TLOAD2** entry defines a time-dependent load of the form:

$$P(t) = \begin{cases} 0 & \text{when} \quad \bar{t} < 0 \quad \text{or} \quad \bar{t} > t_2 - t_1 \\ A\,\bar{t}^{B}\,e^{C\bar{t}}\cos(2\pi f\bar{t}+\theta) & \text{when} \quad 0 \le \bar{t} \le t_2 - t_1 \end{cases} \tag{14-28}$$

$$\text{where} \quad \bar{t} = t - t_1 - \tau$$

In each case, A is defined using any combination of **DAREA** entries and static load entries, and τ is defined on a **DELAY** entry. F is a time-dependent function which is defined in tabular form on any of the four available **TABLEDi** entries. These entries are explained to you in the **UAI/*NASTRAN Reference Manual***.

The static load coefficients, A, may be different for each loaded degree of freedom. Therefore, each load is applied to its corresponding degrees of freedom in the global coordinate system, and assembled into the load vector. You may use SUBCASEs to analyze a number of load cases in a single execution.

14.2.2 Enforced Motion

You may use the same functional forms defined in the previous section to define an enforced motion instead of an applied force. You must enter one of the values **1**, **2**, or **3** in the **DYNEX** field of the **TLOADi** Bulk Data entries to explicitly request that the function be treated as an enforced displacement, velocity, or acceleration, respectively.

☞ The degrees of freedom on which the motion is enforced ***must be defined in the s-set***.

14.2.3 Transient Response Loads for Substructures

Transient response loads for substructuring are defined using a combination of static load vectors declared in PHASE1 substructuring runs and dynamic load data defined in a PHASE2 solution execution. The PHASE1 static load data perform the role of the **DAREA** data. These data are expanded in the time domain during the PHASE2 execution using the referencing techniques described earlier for normal dynamics analyses.

UAI/*NASTRAN* provides additional Bulk Data entries to assist you in defining substructure loads. The **DAREAS** Bulk Data entry specifies the location of a dynamic load or enforced motion and a scale factor, and the **DELAYS** entry defines the time delay for time-dependent loads. These values are defined in reference to Basic Substructures.

14.2.4 Nonlinear Transient Loads

When you perform transient response analyses, you may define applied loads for a set of degrees of freedom as functions of the displacements at other specified degrees of freedom, hence *nonlinear* loads.

☞ It is ***required*** that the points to which the nonlinear loads are applied and the degrees of freedom on which they depend be retained in the solution set. When you use the modal method, it is further required that the nonlinear loads reference EXTRA points.

The load definitions may be of four different forms. The first form is an *arbitrary* function of the displacement at a potentially different degree of freedom:

$$N_i(u, t) = ST(\ u_j(t)\ ) \tag{14-29}$$

where N_i is the load applied to *i*, *S* is a scale factor, and $T(u_j(t))$ is a tabulated function of any permissible displacement component at *j*.

In the second form, the function is a *multiplication* of displacement values at two degrees of freedom:

$$N_i(t) = S\, u_j(t)\, u_k(t) \tag{14-30}$$

where u_j and u_k are any permissible pair of displacement components. They may be the same.

The third and fourth forms are known as *positive power function* and *negative power function*:

$$N_i(t) = \begin{cases} S\, u_j^A(t) & \text{for } u_j(t) > 0 \\ 0 & \text{for } u_j(t) \le 0 \end{cases} \tag{14-31}$$

$$N_i(t) = \begin{cases} -S\,(-u_j(t))^A & \text{for } u_j(t) > 0 \\ 0 & \text{for } u_j(t) \le 0 \end{cases} \tag{14-32}$$

where *A* is an amplification factor.

Nonlinear loads applied to a massless system without damping may ***not*** converge to a steady state solution. The use of:

```
DIAG 10
```

Executive Control command causes the term:

$$\frac{N_{n+1} + N_n + N_{n-1}}{3} \tag{14-33}$$

to be added to the nonlinear load term at each time step n+1, N_{n+1}. N_n and N_{n-1} are the values of the nonlinear load at time steps n and n-*1*. In some cases this leads to more stable solutions.

14.3 INPUT DATA REQUIREMENTS

This section describes the input data required for performing transient response analyses. Input data for these analyses are organized to provide an easy transition from a static analysis model. All static analysis modeling options are available. In addition, EXTRA points are introduced for the purpose of coupling non-structural quantities to the structural model. Transient response analyses require additional information beyond that necessary for statics analysis, specifically:

- Time step information must be provided.
- Sources of excitation must be defined as a function of time.
- Nonlinear loads may be defined.
- Initial conditions may be provided.
- Viscous damping, structural damping, and in the modal Rigid Format, modal damping may be included.
- In the modal Rigid Format, a real eigenextraction technique must be supplied.

14.3.1 Executive Control Commands

The Executive Control commands for direct and modal transient response analyses are specified as follows:

```
SOL  9    (Direct Transient Response Analysis)
SOL 12    (Modal Transient Response Analysis)
```

There are no additional Executive Control Commands required for transient response analyses. However, if you wish to plot the output of your analysis, you must use an appropriate **ASSIGN** command to define the plot file.

The automated substructuring feature is only available in **SOL 9**. However, you may extend it to the modal method by entering:

```
INCLUDE ALTERLIB( SUB09M )
```

in your Executive Control packet. This command will automatically include a standard DMAP ALTER package in your data packet which introduces the modal formulation in the framework of **SOL 9**. Note that you must then satisfy the input requirements of the modal method.

14.3.2 Case Control Commands

In addition to the normal modeling control information, there are special Case Control command requirements for transient response analyses. The Case Control commands, their relationship to corresponding Bulk Data entries, and the actions taken in each case are shown in Table 14-1.

As was the case for normal modes analysis, you may use the Guyan reduction features to reduce the problem size. Only one boundary condition may be solved in a given execution. Also, only one set of symmetric direct input matrices may be selected with the **B2GG**, **K2GG**, and **M2GG** commands.

Table 14-1. SELECTING TRANSIENT RESPONSE DATA

WHEN USING:	TO SELECT OR MODEL:	YOU USE THE	
		CASE CONTROL COMMAND:	BULK DATA ENTRIES:
THE DIRECT OR MODAL METHOD	Time Steps	TSTEP	TSTEP
	Dynamic Loads and Enforced Motions	DLOAD	DLOAD TLOAD1 TLOAD2
	Direct Input Matrices At The System Level	K2GG M2GG B2GG	DMIG
	Direct Input Matrices For All Physical Dof Including Extra Points	K2PP M2PP B2PP	DMIG
	Nonstructural effects	TFL	TF EPOINT
	Viscous Damping Effects		CVISC,PVISC CDAMPi,PDAMP
	Uniform Structural Damping		PARAM,G
	Nonuniform Structural Damping		MATi(GE Field)
	Output Times	OTIME	
DIRECT METHOD	Initial Condtions	IC	TIC TICTV TICRV
THE MODAL METHOD	Real Eigenmethod	METHOD	EIGR
	Modal Damping Effects	SDAMPING	TABDMP1
	Modal Coordinates		PARAM,LFREQ PARAM,HFREQ PARAM,LMODES
	Mode Acceleration		PARAM,MODACC
	Retaining Diagonal Terms Of Non-diagonal Matrices		PARAM,DIAGNLPP
	Initial Conditions		PARAM,NRB
		IC	TIC TICTV TICRV
	Inclusion of Rigid Body Displacements		PARAM,RBDISP
	Nonuniform Modal Structural Damping		PARAM,W4MODAL
	Normal Modes from a Previous CASE	USING MODES	

14

At least one SUBCASE must be defined for each different set of dynamic excitations, nonlinear loads, and direct input matrices, selected with any combination of **DLOAD**, **NONLINEAR**, **B2PP**, **K2PP**, and **M2PP** commands, as shown below.

```
NONLINEAR = 101
TSTEP = 32
SPC   = 10
DISP  = ALL
SUBCASE 1
      B2PP = OLDDAMP
      DLOAD = 10
SUBCASE 2
      B2PP = OLDDAMP
      DLOAD = 20
SUBCASE 3
      B2PP = NEWDAMP
      DLOAD = 10
SUBCASE 4
      NONLINEAR = 104
      B2PP = NEWDAMP
      DLOAD = 20
BEGIN BULK
```

When you perform modal transient response analyses, you must also select a real eigenextraction method with the Case Control command **METHOD**. This data is used to generate the generalized degrees of freedom which represent the model in the modal formulation.

14

14.3.2.1 X-Y Plotter Commands

In order to assist you in interpreting solution results, you may plot the results of your transient response analysis using the **UAI/*NASTRAN*** X-Y Plotter feature. The principal method for output presentation of transient results is to plot histograms of selected responses versus time. All of the X-Y Plotter commands must be entered as a subpacket of the Case Control packet. Chapter 30 of this manual explains the X-Y Plotting feature in full detail.

14.3.3 Bulk Data Entries

Table 14-1 summarizes the Bulk Data entries which you may use in a transient response analysis. There are a number of Bulk Data entries that are used specifically for this analysis type. These, and the corresponding topics of interest, are explained in the following sections.

14.3.3.1 Time-Dependent Loads and Enforced Motion

In addition to the time steps, you must provide a set of time-dependent forcing functions or enforced motions to complete your transient response analysis. These may be supplied using either the **TLOAD1** entry or the **TLOAD2** entry. If you wish to combine the two, you must use a **DLOAD** Bulk Data entry.

The **TLOAD1** and **TLOAD2** entries define time-dependent loads or enforced motions whose functional form is as shown in (14-27) and (14-28). The formats of these Bulk Data entries are:

TLOAD1	LID	ADEF	τDEF	DYNEX	FTAB				

TLOAD2	LID	ADEF	τDEF	DYNEX	t1	t2	f	θ	-cont-
-cont-	C	B							

These entries reference other Bulk Data entries such as **DAREA**, **DELAY**, and **TABLEDi** to define the appropriate components of the function at the specified degrees of freedom. **ADEF** may reference any static load set which is to be made time-dependent.

The **DYNEX** field of the **TLOADi** entries specifies whether the data on the entry is to be used as a forcing function, or as an enforced motion. If you enter a value of **0**, or leave this field blank, you request that the data provided be used as a forcing function. If you enter the values of **1**, **2**, or **3** in this field, you explicitly request that the function be treated as an enforced displacement, velocity, or acceleration, respectively. In these cases, the degrees of freedom to which the motion applies ***must be defined in the s-set***.

14.3.3.2 Nonlinear Loads

You may use the **NOLINi** Bulk Data entries to define the displacement-dependent transient loads which you have selected with the **NONLINEAR** Case Control command. The formats of these entries are:

NOLIN1	NLLID	GIDI	DOFI	S	GIDJ	DOFJ	TID		

NOLIN2	NLLID	GIDI	DOFI	S	GIDJ	DOFJ	GIDK	DOFK	

NOLIN3	NLLID	GIDI	DOFI	S	GIDJ	DOFJ	A		

NOLIN4	NLLID	GIDI	DOFI	S	GIDJ	DOFJ	A		

In all of the above entries, **S** is a scale factor, **GIDI,DOFI** defines the degree of freedom at which the load is applied, and **GIDJ,DOFJ** and **GIDK,DOFK** define the degrees of freedom whose response is used to evaluate the load.

14.3.3.3 Initial Conditions

The initial conditions may be defined in both direct and modal methods by using **TIC**, **TICTV** and **TICRV** bulk data entries. You can use **TIC** entry to specify both displacement and velocity for either a GRID point or a specific mode. The **TICTV** and **TICRV** entries are useful in specifying translational and rotational velocites for a set of GRID points. You may use any number of these entries to specify initial conditions. For a basic substructure the initial conditions are specified on a **TICS** bulk data entries when you perform a substructuring analysis. You then reference the appropriate set identification number on the **IC** case control command.

Usually, you request the calculation of static displacements and their use as initial conditions by including the **EQUIL** keyword on the **IC** Case Control command. An alternative way of requesting automatic calculation of the initial conditions when the equations of motion are uncoupled and non-linear loads are not present is to include the following:

PARAM	NRB	*n*							

in your bulk data packet. *n* is the number of rigid body modes in your analysis, and must have a zero or positive value. If you use the **IC** Case Control command with the **EQUIL** keyword, **UAI/*NASTRAN*** automatically keeps track of the number of rigid body modes in the model.

14.3.3.4 Entries Specific to the Modal Method

In the modal method, you may control the modal representation by specifying either the **LMODES** or the **HFREQ** and **LFREQ** parameters. **LMODES** specifies the number of lowest frequency modes to be used, whereas **LFREQ** and **HFREQ** define a range of frequencies. If you do not specify any preference for the extent of modal representation, all of the extracted real modes will be used in the modal formulation.

If you select modal damping with the **SDAMPING** Case Control command, then you must supply a **TABDMP1** Bulk Data entry which defines the frequency-dependent damping values. Note that you must specify damping values which are ***twice*** the critical damping ratio. For example, if you wish to specify a constant 2% critical damping for the all modes between 0. and 100. Hz., then the **TABDMP1** damping data entries must be 0.04.

14.4 SOLUTION RESULTS

The available solution results are shown in Table 14-2. These results are given for the set of times defined on a real or generated **SET** Case Control command, and referenced by the **OTIME** Case Control command. By default, output is generated for all time integration steps specified. They may be sorted by time, through a **SORT1** request, or by GRID point or element identification number for all times, through a **SORT2** request. The latter is the default condition.

When using the modal method, eigenvalue summary and extraction data are automatically printed. Other solution results are the same as in the direct method. When solution point output is requested, the modal degrees of freedom will be identified by their mode number, which corresponds to the eigenvalue extraction list.

X-Y plot and print requests are made in the X-Y Plotter subpacket of the Case Control packet. The X-Y Plotter commands are described in Chapter 30 of this manual.

14.4.1 Rigid Body Displacements

When your model is a free body, **UAI/*NASTRAN*** uses the rigid body modal displacements that are compatible with the physical displacements at time t = 0 and computes the both rigid body and flexible mode displacements at all time steps based on the results of previous time steps.

You may request that the rigid body modal displacements at t = 0 be assigned to all other times also by using:

PARAM	**RBDISP**	**NO**							

in the Bulk Data packet. The default for this parameter is **YES** which means that the rigid body mode displacements are calculated at each time step seperately.

Table 14-2. CASE CONTROL COMMANDS FOR RESPONSE

COMMAND	FUNCTION
ACCE	Acceleration in the physical, or *a-set*.
DISP	Displacement, or response, in the physical, or *a-set*.
VELO	Velocity in the physical, or *a-set*.
SACCE	Acceleration in the solution, or *h-set*.
SDISP	Displacement, or response, in the solution, or *h-set*.
SVELO	Velocity in the solution, or *h-set*.

14.5 MODELING GUIDELINES

Chapter 9 provided you with some general modeling guidelines for dynamic analysis of forced response, and some discussions comparing direct and modal methods. It is suggested that you consult those sections as needed. This section provides brief discussions on modeling of transient response problems.

Transient response analysis calculates the dynamic response of a structure due to time-dependent, but not periodic, excitations. Typical transient forces are impact, step, or ramp functions. The results of a transient response analysis for a dynamic model are the history of response versus time, usually called *histograms*, for each of the dynamic degrees of freedom. Once the transient response at these degrees of freedom has been computed, the complete response of the structure may be obtained for any time. The displacements at all other degrees of freedom are calculated for a specific time step. You must select the time step, or steps, for which the maximum displacements, and thus the maximum dynamic stresses, occur. Because the maximum displacements of all the dynamic degrees of freedom many not occur at the same time step, a certain amount of judgment must be used. After the displacements at all degrees of freedom have been calculated, the element stresses may be computed. These stresses represent the instantaneous dynamic stresses corresponding to the selected time step. Note that a full stress calculation is time consuming and computationally expensive.

An important aspect of transient analyses is the selection time step, Δt, for time integration. Some guidelines were presented in Section 14.1.4. In general, Δt is selected to correspond to the smallest period in the system.

14.5.1 Comparing the Direct and Modal Methods

The direct method yields accurate results when the integration is performed with time steps which are small compared to the period of the applied excitations. The cost of direct integration is directly proportional to the number of time steps required for the solution. Therefore, the use of direct integration method is well justified when a wide spectrum of frequencies are excited and the response for a short duration is required. Examples of these conditions are frequently found in wave propagation and shock loading problems.

The modal method, in general, is favored if only a few modes are needed to describe the response. For example, an earthquake may excite only the lowest modes of a building, and a vibrating machine may excite only the nearby frequencies of its support structure. The modal method is well suited for these types of problems. However, in the modal method, you must determine the number of modes that must be retained. To facilitate this decision, it is strongly suggested that you carefully analyze the normal modes of the structure. Computational errors in the modal method are typically the result of mode truncation effects. These effects may lead to inaccurate element solution results, such as stresses and forces, which are computed from the recovered displacements. You may correct for this effect by using the *mode acceleration* option.

14.6 EXAMPLE PROBLEM

The example problem presented in this section uses a well-known engineering problem to illustrate the steps you must take to ensure the most efficient solution for your modal transient analysis model.

Example Problem 14-1

A prismatic bar 400 in long with a rectangular box cross-section is subjected to a compound sinusoidal enforced base acceleration in a transverse direction for a time period of 20 seconds. You wish to perform a transient analysis to observe the effects of the excitation at the tip of the bar.

The structure and its engineering data are shown in Figure 14-1. The data for this problem is found in file **TRAN1**. The applied acceleration, which is of the form:

$$\ddot{u}_t = \begin{cases} 0 & \text{when} \quad t<0 \ \text{ or } \ t>20 \\ g\cos(0.6\pi t) - 0.1t + 1 & \text{when} \quad 0 \le t \le 10 \\ g\cos(0.6\pi t) & \text{when} \quad 10 \le t \le 20 \end{cases}$$

is defined on **TLOAD1** and **TLOAD2** Bulk Data entries. These entries are then combined on a **DLOAD** Bulk Data entry. The time steps are defined on the **TSTEP** Bulk Data entry.

The **GIV**ens eigenextraction method is chosen to calculate the normal modes of the structure. These modes in the frequency range of between 1.0 and 100.0 Hertz are used to represent the model in modal basis. The radian frequencies of these modes are automatically applied to the 4% structural damping with the inclusion of:

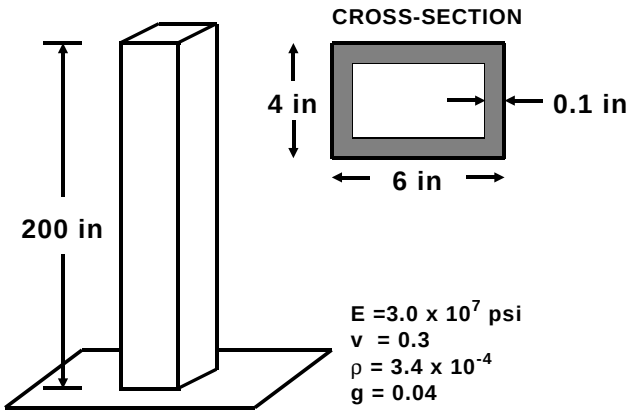
PARAM	**W4MODAL**	**YES**							

in the Bulk Data packet. Despite the presence of structural damping, the equations of motion are uncoupled using the:

PARAM	**DIAGNLPP**	**YES**							

Bulk Data entry. This entry forces the retaining of the diagonal terms of the modal matrices. This simplification also allows the initial conditions to be automatically calculated. This is possible because the applied excitations at time zero were ***not*** zero.

Figure 14-1. TRANSIENT RESPONSE MODEL



To initiate these calculations the:

PARAM	**NRB**	**0**							

is used. Note that **1** is the number of rigid body modes in the problem, represented by the degree of freedom which was not constrained.

The applied load hisory, shown in Figure 14-2, represents the loads internally calculated from the enforced accelerations. You will note the stability of the acceleration at the tip of the bar, shown in Figure 14-3. This is due to the calculation of initial conditions. If you remove any of the last two **PARAM** Bulk Data entries from the data file, you will observe the dramatic effects of the absence of the initial conditions.

Figure 14-2. APPLIED LOAD HISTORY

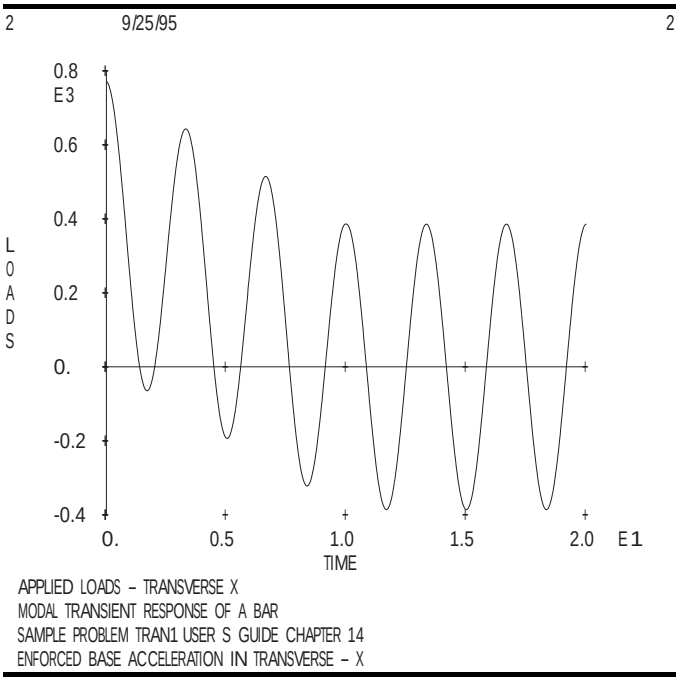
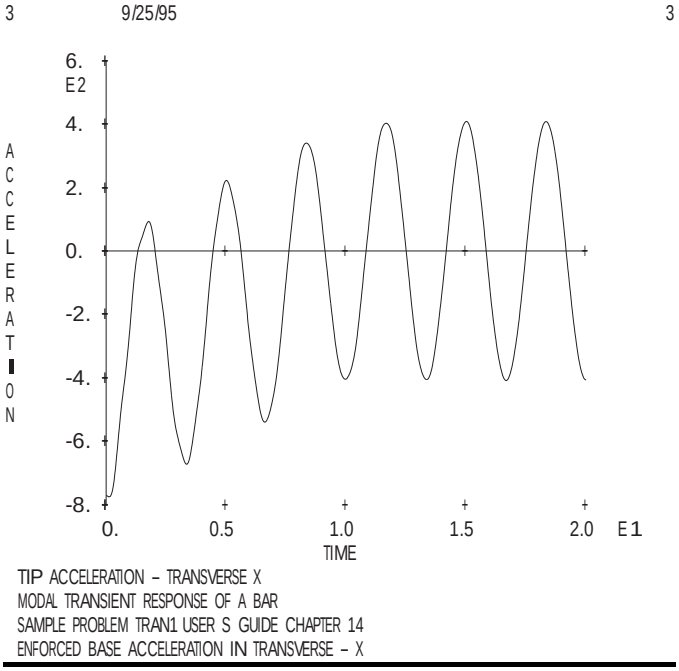


Figure 14-3. ACCELERATION HISTOGRAM



CHAPTER 15

RESPONSE SPECTRUM ANALYSIS

UAI/*NASTRAN* provides analysis capability for exciting a structural model with response spectra. This is called response spectrum analysis or shock spectrum analysis. This capability is provided in Rigid Format 17. A normal modes analysis is first performed and modal participation factors are computed for each loading direction. The modal response is computed from the modal participation factors and a table look-up of the input response spectra. The peak modal response is transformed to physical response and various combination rules are utilized to compute the combined peak structural response. This chapter describes the procedures and options used for solving response spectrum analysis problems.

15.1 GENERAL APPROACH

The response spectrum method has remained a commonly used method of analysis for a wide variety of applications including earthquake analysis and spacecraft launch dynamics. This is because it provides the designer with a rational method for specifying a stochastic loading and it is computationally efficient. Response Spectrum analysis avoids the need to define a detailed time history of the loading (the time history that produces the critical peak response in the subject design) for future earthquakes or other kinds of excitation. Instead, the design spectra is obtained by a statistical averaging of the response spectra of previously measured response time histories. As a result, you are able to determine a response envelope for anticipated load events.

While similar to the modal approach for dynamic analysis, Response Spectra analysis represents the structure by it's normal modes, however it does not allow coupled matrices. The solution itself obtains the peak modal responses, the phasing of these peak responses is then approximated with various combination rules which have been developed by different industrial groups.

In **UAI/*NASTRAN*** , response spectrum analysis solves the uncoupled modal equations for the response to an enforced motion at a single base point. Note that the enforced motion may be in a single direction, or as many as six directions. For each loading direction, the peak response of each mode is determined. For a given case condition, the total system response is determined by combining the individual modal responses in all loading directions.

A key ingredient of the Response Spectrum analysis procedure is the choice of Combination Rules used to first combine the peak responses of the individual modes to produce the peak response in the loading direction, and then to combine peak responses of different excitation directions if more than one is included. It should be noted that different combination rules can be applied for each of the two combination steps. This procedure is described in more detail in section 15.2.3.

15.2 MATHEMATICAL BACKGROUND

The importance of the response spectrum in seismic analysis and design of structures and equipment is well known to earthquake design engineers. For single degree-of-freedom problems (SDOF), the maximum response can be acquired immediately by a table look-up, and for structures modeled with multiple degrees-of-freedom, MDOF, the modal approach for dynamic analysis is used. With appropriate combination rules, the total physical response of the structure can be computed at a fraction of the cost required by a complete transient response analysis.

15.2.1 Response Spectra

For the SDOF structure shown in Figure 15-1, the equation of motion is expressed as:

$$m\ddot{u} + b\dot{u} + k u = -m\ddot{z} \tag{15-1}$$

or:

$$\ddot{u} + 2\,\zeta\omega\,\dot{u} + \omega^2 u = -\ddot{z} \tag{15-2}$$

in which ω is the radian, or circular, frequency, and ζ is the fraction of critical damping or damping ratio.

Under earthquake excitation, if the ground motion $z(t)$ is provided, the response $u(t)$ can be acquired by Duhamel's integral. For a given frequency, ω, and damping ratio, ζ, the maximum value of relative displacement is called the ***Spectral Displacement***, $S_d(\omega,\zeta)$. The value $S_v = \omega S_d$ is called the ***Spectral Pseudovelocity***, and the value $S_a = \omega S_v = \omega^2 S_d$ is called the ***Spectral Pseudoacceleration***. S_v is close to the maximum relative velocity for high frequencies, and practically equal for intermediate frequencies, but different for low frequencies. For zero damping, S_a is equal to the maximum absolute acceleration, but for damping other than zero, the two are slightly different. However, for damping levels encountered in most engineering applications, the two can be considered practically equal.

A plot of S_d versus the frequency, ω, (or natural period T) is called the ***Displacement Response Spectrum***. A set of such curves plotted for various levels of system damping are referred to as displacement response spectra. Similarly, there can be pseudovelocity response spectra and pseudoacceleration response spectra.

Figure 15-2 shows a set of pseudo-velocity response spectra for a typical earthquake. The peaks and valleys in the figure are the results of local resonances and anti-resonances of the ground motion. For design purposes, these irregularities can be smoothed out and a number of different response spectra averaged after normalizing them to a standard intensity. The result of

Figure 15-1. SIMPLE SDOF STRUCTURE

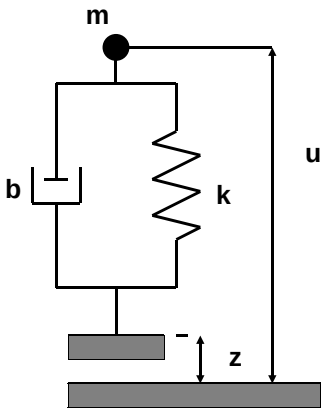


Figure 15-2. TYPICAL RESPONSE SPECTRUM

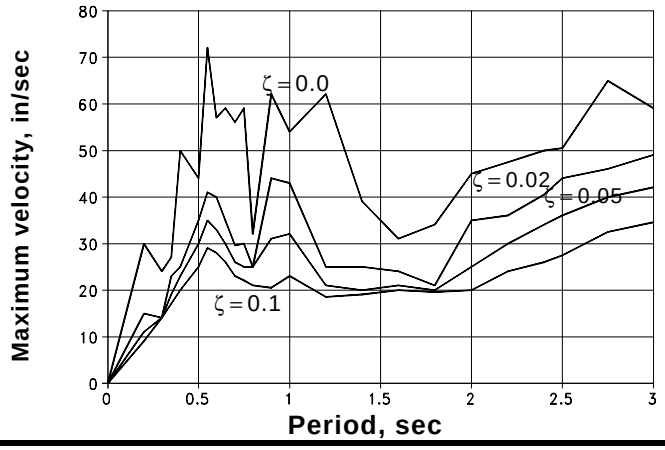
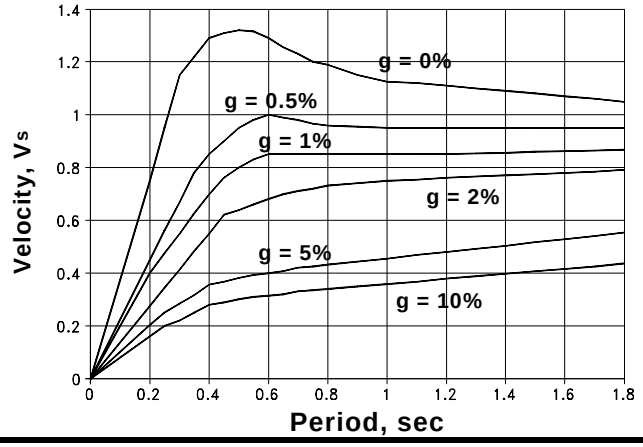


Figure 15-3. AVERAGE VELOCITY SPECTRUM



15

the above procedure will be the earthquake design spectra, which may be presented as smooth curves or even straight lines, as shown in Figure 15-3.

The relations between S_d, S_v, and S_a make it possible to plot all three sets of response spectra simultaneously on a tripartite log-log graph [Naeim89].

15.2.2 Modal Participation Factors

For an SDOF system, determining the system response is merely a table look-up process. From the value of frequency ω or period T, the maximum response comes directly off the curves in the response spectra. However, we do need to do an interpolation to get the value corresponding to the system damping.

For an MDOF structure, consider the equations of motion for enforced motion:

$$\boldsymbol{M}_{aa}\ddot{\boldsymbol{u}}_a^r + \boldsymbol{B}_{aa}\dot{\boldsymbol{u}}_a^r + \boldsymbol{K}_{aa}\boldsymbol{u}_a^r = \left[\boldsymbol{M}_{aa}\boldsymbol{K}_{aa}^{-1}\boldsymbol{K}_{as} + \boldsymbol{M}_{as}\right]\ddot{\boldsymbol{u}}_s \tag{15-3}$$

where the subscripts a and s refer to the final solution degrees of freedom, and the physically constrained degrees of freedom, respectively. When the modal approach for dynamic analysis is used, the following transformation is applied:

$$\boldsymbol{u}_a^r = \boldsymbol{\Phi}_{ah}\boldsymbol{q}_h \tag{15-4}$$

in which $\boldsymbol{\Phi}_{ah}$ represents a set of eigenvectors solved from the eigenproblem:

$$\left[\boldsymbol{K}_{aa} - \lambda\boldsymbol{M}_{aa}\right]\boldsymbol{\Phi}_{ah} = 0 \tag{15-5}$$

After applying the transformation to the equations of motion, the following equation is obtained:

$$\boldsymbol{M}_{hh}\ddot{\boldsymbol{q}}_h + \boldsymbol{B}_{hh}\dot{\boldsymbol{q}}_h + \boldsymbol{K}_{hh}\boldsymbol{q}_h = -\boldsymbol{M}_{hh}\boldsymbol{\Psi}_{sh}^T\ddot{\boldsymbol{u}}_s \tag{15-6}$$

in which, the modal participation factors, $\boldsymbol{\Psi}_{sh}$, are defined as:

$$\boldsymbol{\Psi}_{sh} = -\left[\boldsymbol{M}_{aa}\boldsymbol{K}_{aa}^{-1}\boldsymbol{K}_{as} + \boldsymbol{M}_{as}\right]^T\boldsymbol{\Phi}_{ah}\boldsymbol{M}_{hh}^{-1} \tag{15-7}$$

The subscript h represents the number of modes under consideration, and the subscript s can be from one to six, representing the DOF under excitation.

Note that (15-6) can be uncoupled since $\boldsymbol{K}_{hh}$ and $\boldsymbol{M}_{hh}$ are strictly diagonal, and though $\boldsymbol{B}_{hh}$ might have off-diagonal terms, only the diagonal terms are taken into account.

By analogy with (15-1) for SDOF, it is noted that if the displacement response, S_{jk}^d, for excitation point j and mode k is obtained by table look-up, then the corresponding modal displacement response will be $\Psi_{jk}S_{jk}^d$. Considering the different input and output possibilities, results in the following relations:

$$q_{jk} = \Psi_{jk} S_{jk} = \frac{\Psi_{jk} S^v_{jk}}{\omega_k} = \frac{\Psi_{jk} S^a_{jk}}{\omega_k^2} \tag{15-8a}$$

$$\dot{q}_{jk} = \Psi_{jk} \omega_k S^d_{jk} = \Psi_{jk} S^v_{jk} = \frac{\Psi_{jk} S^a_{jk}}{\omega_k} \tag{15-8b}$$

$$\ddot{q}_{jk} = \Psi_{jk} \omega_k^2 S_{jk} = \Psi_{jk} \omega_k S^v_{jk} = \Psi_{jk} S^a_{jk} \tag{15-8c}$$

15.2.3 Combination Rules

After acquiring the modal displacement response for excitation point j and mode k, q_{jk}, the corresponding physical peak displacements for each mode can be calculated by:

$$(u^r_a)_{jk} = (\varphi_a)_k q_{jk} \tag{15-9}$$

All other physical responses, R_{jk}, such as stress, strain, or support reactions, for the specific excitation point and mode, are obtained in the same manner.

The total structural response is computed from the combination of the peak responses in the individual modes. Several combination rules have been incorporated to produce the peak response. These rules were developed to account for the effect of relative phase or timing of the peak responses in the individual modal responses. The rules range from the highly conservative ABS rule which assumes that all modal peaks occur simultaneously to the least conservative SRSS which assumes that all peaks occur out of phase with each other. The remaining rules attempt to produce reasonably conservative results that lie between the two extreme cases.

15

The combination is performed in two steps, where the first step sums the total response for all modes for each loading, and the second step sums the total response for all loading directions. The combination rules are specified for each of the steps, and the rules are explained below.

15.2.3.1 Absolute Value Rule — ABS

The ABS rule assumes that all peaks occur simultaneously, and is therefore the most conservative rule. ABS can be applied to either step, The modes are combined using:

$$R_j = \sum_{k=1}^{h} | R_{jk} | \tag{15-10}$$

and then the points with:

$$R = \sum_{j=1}^{s} | R_j | \tag{15-11}$$

15.2.3.2 Square Root of the Sum of Squares — SRSS

The SRSS rule may be applied in either step, but is the least conservative particularly when there are closely spaced or higher frequency modes. This is due to the assumption that all peaks only occur out of phase with each other. This can be non-conservative with high-frequency modal responses which tend to respond in phase.

This also may be applied in both steps, and is a good method except when there are closely spaced modes. First combine:

$$R_j = \left[\sum_{k=1}^{h} R_{jk}^2 \right]^{1/2} \tag{15-12}$$

and then:

$$R = \left[\sum_{j=1}^{s} R_j^2 \right]^{1/2} \tag{15-13}$$

15.2.3.3 Naval Research Labs Method — NRL

NRL is a hybrid rule that only applies to the modal combination step. The NRL method assumes that closely spaced modes peak simultaneously with the largest single modal peak, and that all other modes are out of phase. The procedure is to first scan the individual responses to locate the largest response. The ABS rule is then used to combine the largest modal response with any modes which are considered to be closely spaced, based on a factor supplied by the user.

$$R_j = \sum_{k=1}^{h} \left| R_{jk} \right| + \left[\sum_{k=1}^{h} R_{jk}^2 \right]^{1/2} \tag{15-14}$$

If R_{jk} has the maximum value among that of all modes, or if $\omega_k < a\omega_{k-1}$ or if $\omega_{k+1} < a\omega_k$, then R_{jk} is placed into the first group, otherwise, it is placed in the second group. Note that a is a parameter that defines close modes, and $a \geq 1$. Note that if $a = 1$, then the NRL method is equivalent to the SRSS method except for the treatment of the maximum term. On the other hand, as $a \rightarrow \infty$, this method is equivalent to the ABS method.

15.2.3.4 Nuclear Regulatory Commission — NRC

The NRC rule, which is designed for processing closely spaced roots, is also applied only to mode combination. It includes three methods: the grouping method, the ten percent method and the double sum method.

The Grouping Method. In this method, close modes are divided into groups that include all modes having frequencies lying between the lowest frequency in the group and a frequency ten percent higher. For each group, the R_{jk} are first combined using the ABS rule. The results of all groups are then combined by SRSS.

Ten Percent Method.

With this method, the response is computed using:

$$R_j = \left[\sum_{k=1}^{h} R_{jk}^2 + 2 \sum_{k=1}^{h} \left| R_{jk} R_{jm} \right| \right]^{1/2} \quad (k \neq m) \tag{15-15}$$

The second summation is done only for close modes defined by

$$\frac{\omega_m - \omega_k}{\omega_1} \leq 0.1 \quad 1 \leq k \leq m \leq h \tag{15-16}$$

Double Sum Method. This method computes the response using:

$$R_j = \left[\sum_{k=1}^{h} \sum_{m=1}^{h} \left| R_{jk} R_{jm} \right| \varepsilon_{km} \right]^{1/2} \tag{15-17}$$

in which:

$$\varepsilon_{km} = \left[1 + \left[\frac{\omega'_k - \omega'_m}{\zeta'_k \omega_k + \zeta'_m \omega_m} \right]^2 \right]^{-1} \tag{15-18}$$

where:

$$\omega'_k = \omega_k \left(1 - \zeta_k^2 \right)^{1/2} \tag{15-19}$$

$$\zeta'_k = \zeta_k + \frac{2}{t_d \omega_k} \tag{15-20}$$

and ω_k and ζ_k are the modal frequency and damping ratio for the kth mode, respectively, and t_d is the duration of the excitation.

15.2.3.5 Complete Quadratic Combination — CQC

This is applied in step one. It takes care of close modes, and offers a significant improvement in estimating the response of certain 3-D structural systems.

$$R_j = \left[\left| \sum_{k=1}^{h} \sum_{m=1}^{h} R_{jk} p_{km} R_{jm} \right| \right]^{1/2} \tag{15-21}$$

$$p_{km} = \frac{8\,(\zeta_k\,\zeta_m)^{1/2}(\,\zeta_k + r\,\zeta_m)\,r^{3/2}}{(\,1-r^2)^2 + 4\,\zeta_k\,\zeta_m\,r\,(\,1+r^2) + 4\,(\,\zeta_k^2 + \zeta_m^2)r^2} \quad \text{and} \quad r = \frac{\omega_m}{\omega_k} \tag{15-22}$$

Note that this is the only method that takes into consideration the sign of R_{jk}.

15.2.4 Treatment of High-frequency Modes

In general, modal solution methods require that a sufficient number of modes be included such that any effects of high frequency modes are negligible. The NRC dictates that modal truncation effects may not account for more than a 10% decrease in the predicted responses. This requirement can lead to the inclusion of modes whose natural frequencies occur above the point where the spectral acceleration has returned to the "Zero Period Acceleration" level. The combination rule SRSS in particular, can produce significantly non-conservative results when applied to these 'high-frequency' modal responses.

UAI/*NASTRAN* provides a procedure to account for truncated modes. This procedure uses a psuedo-static response to inertial loads otherwise lost to mode truncation effects [NRC89] As in the usual case, responses are first determined for the modes included in the solution set. It is recommended that the user only include modes with natural frequencies below the Zero Period Acceleration (ZPA) point. In the nuclear power industry, a 33Hz cutoff has been suggested [NRC89]. The modal responses are combined according to the user specified rules described in 15.1.3. For each spectral loading condition, a static analysis is then performed to account for the inertial loading not described by the modal equations of motion. This statically applied load is computed in the following manner.

For each degree of freedom in the model, the effective mass described by the included modes is determined.

The included mass fraction d for the i^{th} degree of freedom is given by:

$$d_i = \sum_{j=1}^{h\text{–}set} \Psi_j\,\varphi_{j,i}$$

where Ψ_j is the jth mode participation factor $\varphi_{j,i}$ is the eigenvector for the j^{th} mode and i^{th} degree of freedom. The applied load at each degree of freedom is,

$$P_i = (\,1.0 - d_i)\,m_i\,\ddot{u}$$

where $\ddot{u}$ is the ZPA level acceleration. The acceleration level is obtained from the highest frequency point of the input response spectra. Note that at the ZPA, the acceleration level should be independent of the damping ratio. In the case where the input response spectra do not have the same acceleration level at the ZPA, **UAI/*NASTRAN*** will interpolate between the input spectra based on the damping ratio of the highest mode.

15.3 INPUT DATA REQUIREMENTS

This section describes the input data required for performing response spectrum analysis.

15.3.1 Executive Control Commands

The only Executive Control command for response spectrum analysis is:

```
SOL 17
```

with selects the appropriate Rigid Format. Note that neither automated substructuring nor checkpoint/restart are currently supported.

15.3.2 Case Control Commands

Two Case Control commands are used in Shock Spectrum Response analyses. The first selects the set of input response spectra, and the combination rules to be used in the analysis. Its format is:

```
SHOCK ( PCOMB = { ABS } ,
                { SRSS }

        MCOMB = { ABS                                                          }
                { NRC [ ,CLOSE = { GROUPING = x } ] [ ,HIGHFREQ ]              }
                {                { PERCENT  = y }                              }
                {                { DOUBLESUM = t }                             }
                { NRL [ CLOSE = a ]                                            }
                { SRSS                                                         }
                { CQC                                                          }
      ) = lid
```

The parameter **PCOMB** selects the shock point combination rule from sum of the absolute values, **ABS**, or square root of the sum of the squares, **SRSS**. **MCOMB** selects one of the mode combination rules. These include: the sum of the absolute values, **ABS**; Nuclear Regulatory Commission, **NRC**; Naval Research Laboratories, **NRL**; square root of the sum of the squares, **SRSS**; or the complete quadratic combination, **CQC**. When using the **NRC** method, **CLOSE** selects a closely spaced mode treatment rule.

The keyword **HIGHFREQ** determines whether the procedure for treating high-frequency modes will be performed. The default value is **NO**. Finally, *lid* specifies the identification number of **SHOCK** Bulk Data entries.

The second command, which requests the printing of modal participation factors is:

```
MPFACTOR  ( { PRINT   } [ , PUNCH ] ) = { ALL  }
          ( { NOPRINT }               )   { NONE }
```

The Case Control commands, their relationship to corresponding Bulk Data entries, and the action taken in each case are shown in Table 15-1.

15.3.2.1 Case Control Structure

The first step in the shock response analysis is to perform a normal modes analysis of the structure. The frequency data are then used along with the corresponding modal damping values to compute peak modal responses of up to six excitation conditions. You may either select to use the composite damping defined for the structure, or you may specify a modal damping-frequency relationship. In the former case, only the diagonal terms of $\boldsymbol{B}_{hh}$ will be used to get a pseudo-modal value of damping for each frequency. This is similar to the manner in which **PARAM,DIAGNLPP,YES** functions in Transient Response analyses, described in Chapter 14.

To select the shock load and combination rules, the **SHOCK** Case Control command points to the identification number of one or more **SHOCK** Bulk Data entries.

Table 15-1. SELECTING RESPONSE SPECTRUM ANALYSIS DATA

TO SELECT OR MODEL:	YOU USE THE	
	CASE CONTROL COMMAND:	BULK DATA ENTRIES:
Direct Input Matrices At The System Level	**K2GG** **M2GG** **B2GG**	**DMIG**
Real Eigenmethod	**METHOD**	**EIGR**
Viscous Damping Effects		**CVISC,PVISC** **CDAMPi,PDAMP**
Uniform Structural Damping		**PARAM,G** **PARAM,W3**
Nonuniform Structural Damping		**MATi (GE** Field**)** **PARAM,W4** **PARAM,W4MODAL**
Modal Damping Effects	**SDAMPING**	**TABDMP1**
Modal Coordinates		**PARAM,LFREQ** **PARAM,HFREQ** **PARAM,LMODES**
Modal Analysis Output	**MODES**	
GRID Point Kinetic Energy	**GPKE**	
Modal Participation Factors	**MPFACTOR**	
Shock Analysis Control	**SHOCK**	**SHOCK**

15.3.2.2 How the SORT Options Function

You may request either **SORT1** or **SORT2** for response output. The results of the two sorts are presented in the same manner in which static analysis responses are presented. The results of **SUBCASE 1**, however, do not enter the sort process. You may request the output of the Normal Modes analysis by including the appropriate Case Control commands in the first SUBCASE.

As is the case with enforced motion in dynamic analysis, extra points are not supported in response spectrum analysis. Therefore, the **K2PP**, **B2PP**, and **M2PP** options are not supported.

15.3.3 Bulk Data Entries

This capability requires a single new Bulk Data entry which is used to select the data defining the shock as a function of frequency.

SHOCK	**SID**	**GRID**	**COMP**	**DYNEX**	**DMPTYP**	**FACTOR**			**-cont-**
-cont-	**DVAL1**	**TID1**	**DVAL2**	**TID2**	***CONTINUES IN GROUPS OF 2***				**-cont-**

SID represents the shock spectra identification number which is referenced by the **SHOCK** Case Control command. This is followed by a **GRID** point and **COMP**onent degrees of freedom. The spectra for these points is then defined using **TABLED1** Bulk Data entries. The **DYNEX** field selects the type of the shock function. For shock, **DYNEX** may only have a value of **1**,**2** or **3**, which selects an enforced displacement, velocity or acceleration, respectively. The user may not select a load as in other dynamic response analyses. **DMPTYP** specifies the type of damping values, **DVALi**. The allowable values are **FRACTION**, **G**, **Q**, and **PERCENT**. The default value is **G**. **FACTOR** is the overall scale factor for all values acquired from the shock spectra specified in **TIDi**. It's default value is 1.00. The continuation entry then references a set of one or more tables, **TIDi**, which represent the shock spectra for a particular damping value, **DVALi**, as shown in Figure 15-1.

The shock application degrees of freedom are limited to six degrees of freedom of one GRID point. If there are other fixed points subjected to the same shock application, they should be connected to this point by **MPC** entries or by rigid elements.

As with the modal method for dynamic response analyses, the user must select the modes that will be used as generalized coordinates for the shock analysis. This is done by using the **PARAM** Bulk Data entries:

PARAM	**LFREQ**	*flow*							
PARAM	**HFREQ**	*fhigh*							
PARAM	**LMODES**	*nmodes*							

where you specify either the frequency range by defining the high and low frequencies to be used, or the number of modes. As described earlier in Section 15.1.4, if the **HIGHFREQ** option has been requested to account for high frequency inertia loading, then **PARAM,HFREQ** should be set at a frequency corresponding to the Zero Period Acceleration (ZPA). It should be noted that the **F2** field on the **EIGR** Bulk Data Entry can be used to produce only those structural modes below the desired cutoff frequency.

As in the case of transient analysis, parameter **G** requires the use of parameter **W3**, and if the **GE** field on **MATi** Bulk Data entries is used, then either of the parameters **W4** or **W4MODAL** is required. It should be noted that the nonuniform structural damping introduced by the **GE** field in **MATi** Bulk Data entries might cause $\boldsymbol{B}_{hh}$ matrix to have off-diagonal terms. However,

in response spectrum analysis only the diagonal terms of B_{hh} matrix are used. Thus, the default value, and in fact the only permissible value, of **PARAM DIAGNLPP** is **YES**.

Modal damping is defined by **TABDMP1** Bulk Data entries, which must be selected by Case Control command **SDAMPING**.

15.3.3.1 Bulk Data Modification

In addition to the new Bulk Data entry, two existing entries, both of which are used to define table data, have been modified. The first is:

1	2	3 ⇓	4 ⇓	5 ⇓	6	7	8	9	10
TABDMP1	TID	XAXIS	YAXIS	DMPTYP					-cont-
-cont-	f1	d1	f2	d2	*CONTINUES IN GROUPS OF 2*				-cont-

The **DMPTYP** field specifies the type of damping values, **di**. The allowable values are **FRACTION**, **G**, **Q**, and **PERCENT**. The default value is **G**. The first two new fields, **XAXIS** and **YAXIS**, are used to specify that the table entries in either or both coordinate directions are **LINEAR** or **LOG**arithmic. The default value is **LINEAR**. An analogous change has also been made for **TABLED1** entries: is:

1	2	3 ⇓	4 ⇓	5	6	7	8	9	10
TABLED1	TID	XAXIS	YAXIS						-cont-
-cont-	X1	Y1	X2	Y2	*CONTINUES IN GROUPS OF 2*				-cont-

15.4 SOLUTION RESULTS

The first SUBCASE in the Response Spectrum analysis controls the extraction of the normal modes of the model. This SUBCASE may request any of the standard solution results for this discipline. In addition, you may request the modal participation factors, shown in Table 15-2. All other solution results are the same as for standard Response Analyses, namely the displacement, velocity and acceleration vectors for each SUBCASE, and any other GRID point or element based results.

You may also print the modal damping value for each load case in your Response Spectra analysis. This is done by using the Bulk Data entry:

PARAM	**SHKDAMP**	**1**							

For Version 11.7, this entry results in a matrix print of the **OMEGA** and **DAMP** entities. The former is a vector of frequencies and the latter the damping value associated with each of these frequencies.

15

Table 15-2. MODAL PARTICIPATION FACTORS

M O D A L P A R T I C I P A T I O N F A C T O R S

MODE NO.	FREQUENCY HZ.	GRID 13 T1
1	1.328304E+00	1.226973E+00
2	3.935187E+00	-3.016631E-01
3	6.263954E+00	9.507709E-02
4	7.831073E+00	-2.056489E-02
5	1.265053E+01	1.088703E-20
6	1.561831E+01	1.822213E-04
7	1.956299E+01	-1.560755E-19
8	3.264654E+01	1.685070E-20

15.5 EXAMPLE PROBLEM

This section presents a example problem which illustrates how Response Spectra analysis can be used to analyze the response of a building subjected to an earthquake.

Example Problem 15-1

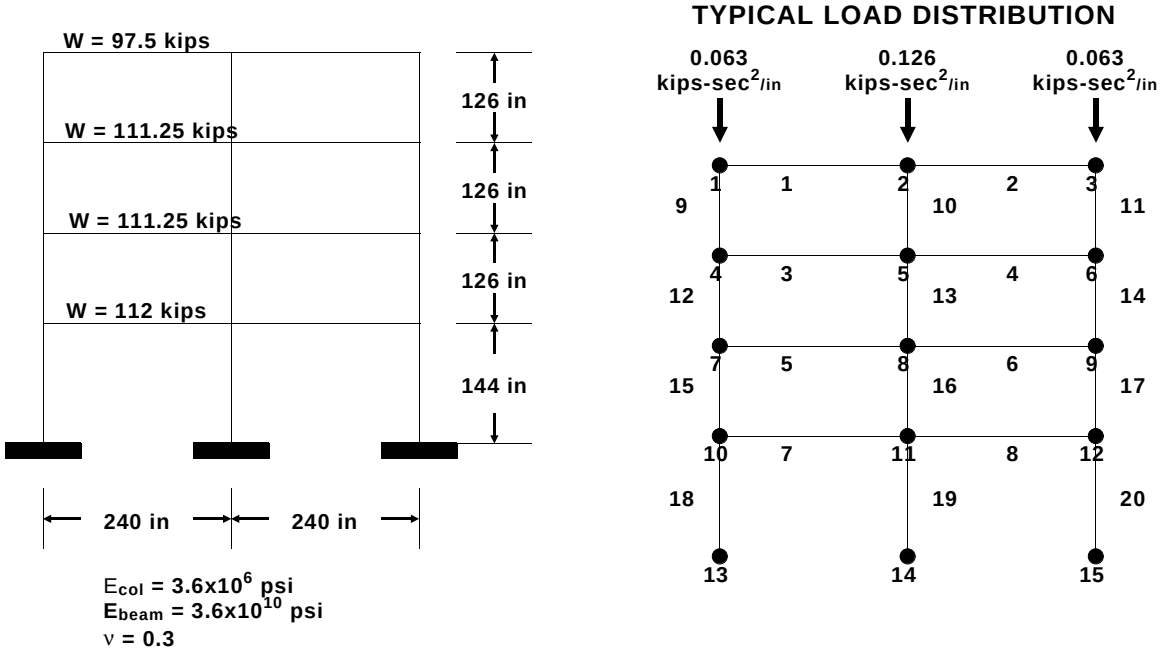
A four-story frame structure is subjected to an earthquake with a given spectral pseudovelocity. You wish to determine the modal participation factors and the displacement and acceleration responses using each of the available modal combination rules.

The finite element model and physical characteristics of a two-dimensional cut through the structure is shown in Figure 15-4. The model, found in file **SHOCK1**, is composed of 20 **BAR** elements. The weight of each floor, shown in the figure, is converted to mass and distributed, as shown, to the GRID points using **CONM2** Bulk Data entries. The columns are concrete with rectangular cross-section of 14 in x 14 in. and the beams are 12 in x 20 in. Note that this is only one cut, and that the total floor loads shown in the figure represent only one quarter of the total load.

Recalling that the shock may only be applied at a single GRID point (**13** is selected in this example) it is necessary to use MPC relationships to excite the other two GRID points, **14** and **15**, at the base of the structure. These MPCs are seen in the data file.

Case Control commands are used to define 6 SUBCASEs. The first SUBCASE controls the extraction of the normal modes of the structure. As usual, you may select any eigenextraction method and any number of desired modes.

Figure 15-4. FRAME STRUCTURE MODEL



For this example problem, mode shapes for frequencies below 33.0 Hz are selected using:

EIGR	**1**	**MGIV**	**0.0**	**33.0**					

There are eight such modes. The modal participation factors are requested for each mode and are shown in Table 15-2 in the previous section of this Chapter. The next five SUBCASEs, while selecting the same point-combination rule **PCOMB**, each request a different modal combination rule, **MCOMB**. The commands used are:

```
SUBCASE 1
   LABEL = MODAL ANALYSIS
   METHOD = 1
   MPFACTOR = ALL
SUBCASE 2
   LABEL = EXCITATION IN X DIRECTION, SRSS
   SHOCK(PCOMB=SRSS,MCOMB=SRSS) = 1
SUBCASE 3
   LABEL = EXCITATION IN X DIRECTION, ABS
   SHOCK(PCOMB=SRSS,MCOMB=ABS) = 1
SUBCASE 4
   LABEL = EXCITATION IN X DIRECTION, NRL
   SHOCK(PCOMB=SRSS,MCOMB=NRL,CLOSE=1.1) = 1
SUBCASE 5
   LABEL = EXCITATION IN X DIRECTION, NRC
   SHOCK(PCOMB=SRSS,MCOMB=NRC,CLOSE=GROUPING=0.1) = 1
SUBCASE 6
   LABEL = EXCITATION IN X DIRECTION, CQC
   SHOCK(PCOMB=SRSS,MCOMB=CQC) = 1 $
```

The input shock spectrum is shown in Figure 15-5. It is defined using three Bulk Data entries. The first entry defines the excitation point at which the shock is applied:

SHOCK	**1**	**13** ❶	**1** ❷	**2** ❸	**G**				**+S1**
+S1	**0.0**	**12** ❹	**ENDT**						

These data indicate that the load is applied to GRID **13**, ❶, in the x-direction, ❷. Field 5 indicates that the enforced motion is velocity ❸. The continuation entry indicates that for a damping value of 0.0 (note that damping is not specified for this problem) the input shock spectrum is given in a table entry defined with the identification number **12** ❹.

In this case, a **TABLED1** Bulk Data entry is used to define the design spectrum which is shown in Figure 15-4. This is done by digitizing the key points in the log-log plot:

TABLED1	**12**	**LOG**	**LOG**						**+T1**
+T1	**0.1**	**4.5**	**0.21**	**10.0**	**1.82**	**10.0**	**100**	**0.15**	**+T2**
+T2	**ENDT**								

The continuation entry gives the pairs of pseudovelocity-frequency data and interpolations are requested to be performed on a log-log scale. The final response results for each of the GRID points along the left edge of the structure are given in Table 15-3 along with the results from [Naeim89]. The responses include both displacement and acceleration and each is computed using the five available mode combination rules. Three of these rules were computed in the reference using a simple spring-mass system and hand calculations. There is a high degree of correlation, although the more detailed **UAI/*NASTRAN*** model is more flexible and includes more modes in the calculations.

Figure 15-5. DESIGN SPECTRUM

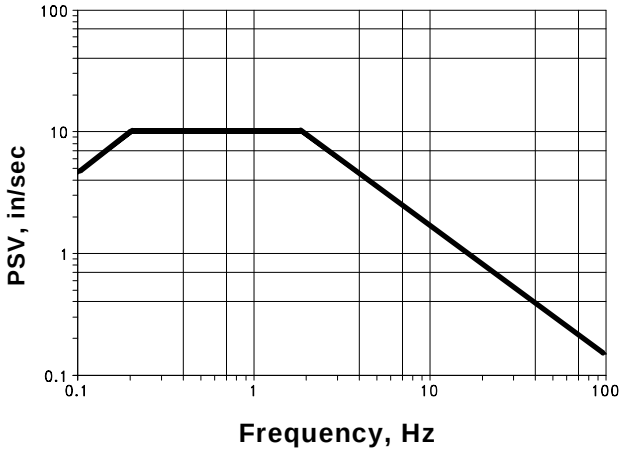


Table 15-3. SOLUTION RESPONSES

SUBCASE	COMBINATION METHOD	GRID POINT	DISPLACEMENT		ACCELERATION	
			UAI/*NASTRAN*	[NAEIM89]	UAI/*NASTRAN*	[NAEIM89]
2	SRSS	1	1.471	1.441	108.2	111.4
		4	1.326	1.310	93.2	99.3
		7	1.045	1.071	77.9	83.2
		10	0.657	0.682	58.6	68.8
3	ABS	1	1.532	1.506	148.1	168.6
		4	1.344	1.342	131.2	139.6
		7	1.094	1.112	111.4	129.9
		10	0.720	0.751	95.2	119.7
4	NRL	1	1.525		137.2	
		4	1.338		105.1	
		7	1.088		100.7	
		10	0.712		82.4	
5	NRC	1	1.471		108.2	
		4	1.326		93.2	
		7	1.045		77.9	
		10	0.657		58.6	
6	CQC	1	1.471	1.441	108.2	110.7
		4	1.326	1.310	93.2	98.9
		7	1.045	1.071	77.9	83.3
		10	0.657	0.682	58.6	70.0

This page is intentionally blank.

Chapter 16

MATERIAL NONLINEAR ANALYSIS

UAI/*NASTRAN* provides capabilities for robust analyses of structural models under the influence of a variety of nonlinear effects. Two distinct forms of nonlinear behavior may be modeled, and they also may be combined in a single simulation.

The two effects are:

- Material nonlinear behavior including: nonlinear functions of strain versus stress for one, two and three dimensional elements; nonlinear load versus deflection behavior for spring elements; and gap element open-closed and friction nonlinear effects.
- Large deflection and large rotation nonlinear effects including structural stiffening due to deflection and external load change with structural deflection.

Both effects may be simultaneously analyzed with **UAI/*NASTRAN***. The simulations are also principally limited to static analysis. The exception to this limitation is the availability of transient response load generation functions which allow loads to be generated as a function of structural motion in the time domain.

Two theoretical approaches to the solution of the static nonlinear equations of motion are provided. A Newton-Raphson technique and an Arc Length technique may be employed for most applications. For less complex geometric nonlinear problems, and for compatibility with older capability, a differential stiffness solution technique is also available.

The differential stiffness technique also is used to obtain a linear Buckling solution, and a Normal Modes solution for structures with geometric nonlinearity.

The Newton-Raphson and Arc Length solution techniques are available using the primary nonlinear analysis Rigid Format which is invoked with the Executive Control commands:

```
APPROACH NONLINEAR
SOL STATICS
```

The differential stiffness-based large deflection capability is available in Rigid Format 4, the buckling capability is available in Rigid Format 5, and normal modes with differential stiffness is available in Rigid Format 13. Buckling analysis is described in Chapter 18; all geometric nonlinear solutions are described in Chapter 17, and the material nonlinear solutions are described in this Chapter.

Many structures and structural components are fabricated using materials that have nonlinear stress-strain properties. Analyzing such components requires a nonlinear solution algorithm. **UAI/*NASTRAN*** provides you with this capability. This Chapter describes both the nonlinear solution algorithm and material models. It shows you how the special nonlinear commands and data entries are used, and it presents several example problems that demonstrate the capability.

16.1 TERMINOLOGY

This section presents the theoretical basis and practical aspects of using **UAI/*NASTRAN*** to solve a wide variety of problems where the material behaves in a nonlinear fashion. The solution techniques presented here cover two very specific conditions. The first is the case where stresses in the structure exceed the yield point and the material begins to exhibit ***plastic*** behavior, and the second is where surface contact (or lack of contact), with or without friction effects, must be simulated. This section presents some of the terminology which is used later in the chapter to describe your input requirements for defining the model and controlling the solution.

Unlike linear analysis, the simulation of nonlinear material behavior requires more than a constant value for Young's modulus and Poisson's ratio. Typically, a detailed stress-strain curve is input along with other information required to support complex behaviour including cyclic loading and unloading of the structure. Such information may include the ***yield criteria*** and material ***hardening rules*** to describe deformations beyond the yield point. **UAI/*NASTRAN*** offers a variety of constitutive laws encompassing ***nonlinear elasticity***, ***plasticity***, and a combination of these. Several yield criteria are available. Three hardening rules are supported for multidimensional problems; ***isotropic hardening***, ***kinematic hardening***, and ***combined hardening***.

Models which simulate contact phenomena require descretization of the contact surface and the use of GAP elements which connect GRID points defined on both faces of the contact material. The GAP elements are used to define an initial state (contact or not) and any required friction effects.

Static analysis for models with material nonlinear behavior involves the following concepts for obtaining an accurate solution. ***Load steps*** are defined to describe the history of loadings and unloadings. Although there is an implicit element of time involved here it is assumed that time effects are not important either between load applications (or load removal) or during the application of the load. Many problems do not require an explicit definition of load steps; there is always one load step defined by default in **UAI/*NASTRAN***.

The load change which is to occur during a load step must usually be applied in an incremental fashion in order to obtain a reliable solution. Therefore, the number of ***load increments*** must be defined, although **UAI/*NASTRAN*** provides default values based on other information provided in the problem setup.

UAI/*NASTRAN* solves for equilibrium at each load increment using an ***iterative*** process which may ***converge*** or ***diverge***. The iterative technique involves the creation of pseudo-loads, also called ***unbalanced forces***. The complete iterative technique uses the unbalanced forces and ***stiffness matrix updates*** to track a solution in a computationally efficient manner.

When the model simulation is near or at structural failure or, when the load increments are too large, the mathematical solution may ***diverge***, and equilibrium may not be achieved. **UAI/*NASTRAN*** has special solution features to recognize the onset of this condition, called ***tentative divergence***, and then to use additional algorithms to achieve a solution up to the point of structural failure.

The mathematical background for the nonlinear material solution techniques available to you in **UAI/*NASTRAN*** is described in the next section.

16.2 MATHEMATICAL BACKGROUND

This section describes the mathematical basis of the general analysis and solution procedures used in nonlinear material analysis using **UAI/*NASTRAN***. In the discussions that follow, only ***time-independent*** material nonlinearities are considered. You must remember that while the solutions to linear problems are always unique, this is often not the case for nonlinear problems. Thus a ***converged solution*** may ***not*** necessarily be the solution sought. Physical insight into the nature of the problem is required, and small incremental steps are often essential to obtain meaningful solutions.

16.2.1 General Solution Procedure

The equations of motion for nonlinear material behavior may be written as:

$$\boldsymbol{F}(\boldsymbol{u}) = \boldsymbol{P}(\boldsymbol{u}) \tag{16-1}$$

where the internal forces, $\boldsymbol{F}$, are functions of the structural deformations, $\boldsymbol{u}$. When the internal forces are linear functions of the deformations then the equations may be solved using linear static analysis as described in Chapter 10. The external loads, $\boldsymbol{P}$, are also functions of the deformations. However, for Nonlinear Material analysis, the effect of the deformations on the load is small and it is usually neglected, thus $\boldsymbol{P}(\boldsymbol{u}) = \boldsymbol{P}$. For cases in which the effect of the deformation on the external loads is large, it is necessary to have the forces follow the deformed structure. Such problems require Geometric Nonlinear analysis, as described in Chapter 17.

The solution of (16-1) cannot be obtained directly. Instead, iterative techniques are used. The general form of the recursion relation used is:

$$\boldsymbol{K}(\boldsymbol{u}_{k,0})\,\Delta\boldsymbol{u} = \boldsymbol{P} - \boldsymbol{F}(\boldsymbol{u}_{k,j-1}) \tag{16-2}$$

where K is the system stiffness matrix, and:

$$\begin{aligned} \Delta\boldsymbol{u} &= \boldsymbol{u}_{k,j} - \boldsymbol{u}_{k,j-1} \\ \boldsymbol{U}_{k,0} &= \boldsymbol{u}_{k-1,jmax} \end{aligned} \tag{16-3}$$

where j and k are iteration numbers, and $jmax$ is the maximum of j at iteration k-1.

UAI/*NASTRAN* uses a general solution procedure which is based on the concept of ***stiffness updates*** and ***unbalanced force*** iterations. A set of default solution strategies has been selected which have broad applicability for many analyses. However, because of the variability of nonlinear analyses, you are able to control all of the solution algorithm parameters using the **NLSOLVE** Bulk Data entry.

The following sections describe the basic components of the solution algorithm and how the **NLSOLVE** entry is used to control the solution process.

16.2.2 The Iteration Algorithm

UAI/*NASTRAN* features two basic nonlinear solution methods to provide maximum flexibility and reliability for the solution of a wide variety of nonlinear problems. These two basic methods are the ***Newton-Raphson method***, which includes two approaches, the ***tangential stiffness method*** and the ***secant stiffness method***, and the Arc Length method. The Arc Length method is described in Chapter 17, Geometric Nonlinear analysis, and the Newton-Raphson methods are described in this chapter.

Figure 16-1 illustrates how the Newton-Raphson methods are used to determine the value of Young's modulus from the stress-strain relationship.

Normally, the tangential stiffness method provides faster rates of convergence, and, as a result, is the default technique. However, for some problems, for example where the stress-strain curve features a negative slope as shown in Figure 16-2, the tangential stiffness method may diverge, while the secant modulus method produces a stable solution.

Two other concepts, called the ***Load Incrementation method*** and the ***Load Iteration method***, are important to the nonlinear solution procedure. These two factors and methods by which they are controlled are described in the following sections.

16.2.2.1 The **NLSOLVE** *Bulk Data Entry*

The remaining description of the nonlinear solution algorithm is best facilitated by discussing how you select the various options and control parameters. The format of the **NLSOLVE** Bulk Data entry, and a description of its fields, in shown in Table 16-1. This entry is used for both Material Nonlinear and Geometric Nonlinear analyses.

The fields used by Material Nonlinear analyses, and their default values, as indicated in the table, are described in the following Sections. Fields only used when performing Geometric Nonlinear analyses are indicated by gray shading.

Figure 16-1. STIFFNESS UPDATE METHODS

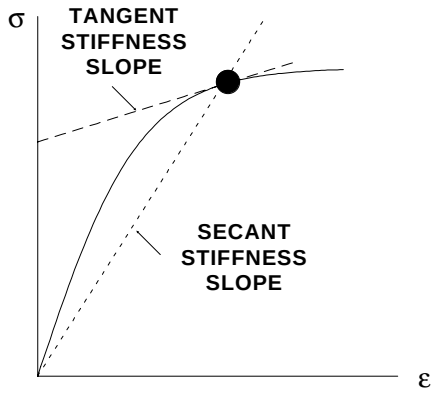
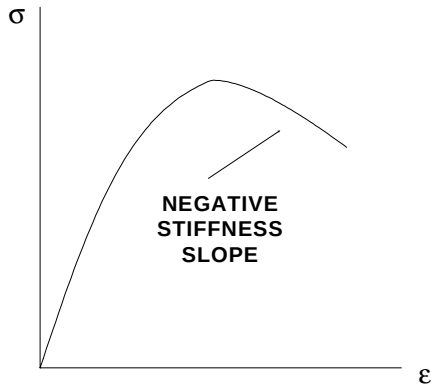


Figure 16-2. NEGATIVE σ–ε SLOPE



16

Table 16-1. NONLINEAR MATERIAL SOLUTION CONTROL

NLSOLVE	ID	SMETH	IMETH	PMETH	CONV	TENDIV	UMAX	MINK	-cont-
-cont-	MAXK	MAXP	PINC	MAXPINC	UDIV	LAMDA	EPMAX	ETAS	-cont-
-cont-	EPSE	EPSP	EPSU	ROTMAX	STNMAX	FACMAX	MINP		

TO CONTROL:	FIELDS USED:	DESCRIPTION	DEFAULT
METHOD SELECTION	SMETH	Solution method	NR
	IMETH	Iteration method	AUTO
	PMETH	External load incrementation method	AUTO
	CONV	Overall convergence criteria selector	EPU
	TENDIV	Strategy for tentative divergence	1
ITERATION	MINK	Minimum number of stiffness updates	0
	MAXK	Maximum number of stiffness updates	99
	MAXP	Maximum number of unbalanced force iterations	30
	MINP	Minimum number of unbalanced force iterations	Determined Automatically
	PINC	Number of load increments	10
	MAXPINC	Maximum number of load increments	99
CONVERGENCE	UDIV	Displacement norm threshold defining divergence	0.9999
	LAMDA	Convergence rate control parameter	0.9999
	EPMAX	Automatic load increment control parameter	Determined Automatically
	ETAS	Percentage of yield to be processed in a sub-increment	0.25
	ROTMAX	Maximum rotation allowed	
	STNMAX	Maximum strain allowed	
	EPSE	Strain energy convergence criterion	0.01
	EPSP	Load convergence criterion	1.0
	EPSU	Displacement convergence criterion	0.1
	UMAX	Maximum allowable deflection denoting divergence	10^5
	FACMAX	Maximum load factor indicating divergence	

16

16.2.2.2 Load Incrementation Method

Accurate and reliable nonlinear solution algorithms require that the nonlinear properties change by relatively small amounts between converged solutions. This means that an external load which is expected to cause severe structural nonlinear behavior must be divided into a series of successive loads, each causing a more modest change in the material strains relative to the previous loading. Thus, the external load is applied in load increments, and a converged solution is obtained at each increment. If the solution is highly nonlinear, the load increments must be relatively small, while if the structure is perfectly linear, only one load increment, the total load, must be applied.

UAI/*NASTRAN* offers you three options for selecting load increments: ***fixed***, ***automatic***, and ***semi-automatic***. With the fixed load increment method, you may specify that the external load be split into an equal number of increments, **PINC**. Selecting the automatic load increment method will cause **UAI/*NASTRAN*** to vary the load increment size automatically based on the solution characteristics and the convergence rate of the solution.

Three additional parameters allow you to control the increment size selection procedure. These include the initial load increment size, which is calculated from **PINC**, and the maximum number of load increments, **MAXPINC**, allowed in a step. Many solution characteristics, such as average displacement, the ratio of the nonlinear energy to the linear energy, and the load errors, are used to control the load increment size selection.

One characteristic you may control is the load error, **EPMAX**, above which the load increment size will be reduced. In order to obtain efficient solutions, larger load error tolerances should be used if the stiffness change is discontinuous, as in the case of GAP elements.

Finally, for the semiautomatic load increment method, the load increment size varies between those of the fixed and automatic methods. Only the rate of solution convergence is used to modify the load increment size. You may control this method with several parameters.

UAI/*NASTRAN* uses the sub-incremental technique to integrate the plasticity constitutive equations. In this technique, each load increment is subdivided into a number of sub-increments. You may also control the number of sub-increments by using **ETAS**.

16.2.2.3 Iteration Method

For any given load, the nonlinear equations must be solved using an iterative procedure. This iterative procedure uses either an unbalanced force iteration, a stiffness update, or both. Iteration procedures which use only the stiffness update approach are called Newton-Raphson methods, and procedures which feature only unbalanced force iteration are called the Modified Newton-Raphson approach or the ***initial strain method***. Both of the procedures are special cases of the general solution procedure available in **UAI/*NASTRAN***.

In general, the stiffness update method yields a faster convergence rate per load increment, but it incurs greater expense because the stiffness matrix must be reformulated and decomposed often. The unbalanced force iteration method converges more slowly, but avoids the computational costs of stiffness matrix decomposition. The most computationally efficient method is generally a combination of these two iteration procedures.

Five of the **NLSOLVE** fields are used to control the unbalanced force iteration procedure: **IMETH**, **MAXK**, **MAXP**, **MINK**, and **MINP**. **IMETH** is used to select one of three options, **FIX**, **AUTO** or **SEMI**. **MAXK** and **MAXP** are then used as described next to control the solution procedure.

When you select the fixed method of unbalanced force iteration, you may control the maximum number of unbalanced force iterations that are allowed prior to a stiffness update. A

combination of unbalanced force iterations and stiffness updates continues until convergence is achieved for the current load increment or the maximum number of stiffness updates, which you may also specify, are performed. If the maximum number of stiffness updates is reached, a message is issued and the processing for the current SUBCASE is terminated.

If you specify that the maximum number of unbalanced force iterations using **MAXP=1**, and you request a large number of stiffness updates, then the Newton-Raphson solution procedure is used. On the other hand, if you specify a large number of unbalanced force iterations so that convergence is achieved without the need for a stiffness update, then the Modified Newton-Raphson method is used.

When you select the automatic iteration method, **UAI/*NASTRAN*** evaluates the computational cost of an unbalanced force iteration and a stiffness update. It then determines the relative rates of convergence for the two methods and selects a combination of unbalanced force iterations and stiffness updates which minimize the total cost of the analysis. However, this method is subject to the limitations imposed by **MINP** and **MAXP**, i.e. the solution will perform at least **MINP**, and at the most, **MAXP**, unbalanced force iterations before performing a stiffness update.

The semiautomatic method is similar to the automatic method described above, except that at the end of each load STEP, a stiffness update is performed to prepare for stress recovery and give a more stable solution for the next load STEP. This is done to remove any remaining unbalanced forces at convergence, which will normally be present with the fixed and automatic methods.

Finally, for some severely nonlinear problems it may be necessary to perform a number of stiffness updates at the beginning of each load increment. This is done by using the parameter **MINK** which specifies a minimum number of stiffness updates to be performed. An example of the use of this parameter is given in Section 16.7.

16.2.2.4 Convergence Criteria

Convergence criteria are used to determine when the iterative procedure for nonlinear solution has converged to an acceptable result. **UAI/*NASTRAN*** provides a combination of three error measures, each of which you may select and control, to define convergence. These three measures are the load error, the strain energy error, and the displacement error. At every unbalanced force iteration, the error in each of the three measures is computed, compared with the defined tolerances, and when the values are less than the allowables, the solution is assumed to have converged. You may also specify the maximum absolute value of deflection to be allowed for nonlinear degrees of freedom. A SUBCASE will be terminated if the deformation exceeds this value.

A very general nonlinear solution algorithm, such as that installed in **UAI/*NASTRAN***, requires special attention to the recognition of a diverging solution, so that processing may terminate gracefully. This special processing is handled in **UAI/*NASTRAN*** by introducing the concept of a ***Tentatively Divergent*** solution as well as a truly ***Divergent*** solution.

The displacement norm change rate is the ratio of the norms of two successive unbalanced force iteration delta displacement vectors. The delta displacement vector for the current iteration is the difference between the total displacement at the current iteration and previous iteration. If the displacement norm change rate is greater than 1.0, then each successive load iteration is causing a greater change in deflection than the previous iteration, and the solution is said to be ***Diverging***.

The load error at unbalanced force iteration *j* is defined as:

$$\varepsilon_p^j = \frac{\left| \boldsymbol{P_i} - \boldsymbol{P_o} - \boldsymbol{F_j} \right|}{\left| \boldsymbol{P_i} - \boldsymbol{P_o} \right|} \tag{16-4}$$

where $\boldsymbol{F_j}$ are internal force vectors, the vector $\boldsymbol{P_i}$ is the external load vector at the i^{th} load increment, and $\boldsymbol{P_o}$ is the external load vector for the last load increment of the previous load step. The vertical bars indicate the norm, which for this purpose is defined as the average absolute value of the components of the vector. The convergence rate, λ, which is controlled by the **LAMDA** field of the **NLSOLVE** entry, is then defined as the ratio of load errors for two consecutive unbalanced force iterations:

$$\lambda = \frac{\varepsilon_p^j}{\varepsilon_p^{j-1}} \tag{16-5}$$

With this definition of λ, a value near 0.0 indicates rapid convergence, a value which is only slightly less than 1.0 indicates slow convergence, a value greater than or equal to 1.0 indicates that the solution may be divergent. λ may exceed 1.0 when the tangential stiffness values differ considerably from the values which would be computed using the current state of strain in the elements or when the current load increment is too large. When λ has a value greater than or equal to 1.0, the solution may be corrected to improve convergence by performing a stiffness update, reducing the size of the load increment, or both. The tentatively divergent processing algorithm in **UAI/*NASTRAN*** uses both the stiffness update and load increment reduction techniques when attempting to correct a diverging solution. If the algorithm senses that convergence cannot be achieved, then the procedure is terminated, and the analysis is considered complete with a nonlinear structural failure.

Four **NLSOLVE** Bulk Data fields are used to control tentatively divergent processing.

First, the **TENDIV** field may be used to suppress such processing. Setting this field to an integer value of **-1** will cause **UAI/*NASTRAN*** to assume divergence of the solution whenever tentative divergence is sensed. The default strategy is to perform the normal tentatively divergent processing.

The **LAMDA** field is used to set the value at which tentatively divergent processing is initiated.

Next the **UDIV** field defines the value of the displacement norm change which will cause tentatively divergent processing to begin.

The basic algorithm for tentatively divergent processing allows **TENDIV** stiffness matrix updates to the current material state to be performed. Processing continues, and if the system is still divergent, the load increment is automatically reduced by one-half. The load increment reduction will occur up to four more times, and each time the size of the load increment is reduced by one-half. If a path toward convergence is not achieved then the system is assumed to be divergent, and the current SUBCASE is terminated.

The final field that you may use to terminate the run gracefully, in the event your structure fails, is **UMAX**. If the absolute maximum displacement is greater than **UMAX**, **UAI/*NASTRAN*** executes a tentatively divergent processing with **TENDIV=0** so that the load factor closest to the structure failure will be computed. You may request the failure loads to be printed by using the **OLOAD** Case Control command.

16.3 ELEMENT LIBRARY

The elements shown in Table 16-2 may be used when you perform material nonlinear analysis using **UAI/*NASTRAN***. Both Material Nonlinear models and the techniques you use to define nonlinear elements and their material characteristics, are presented in the next section.

The GAP element is defined using two spring rates, a very stiff spring for the closed portion and a very soft spring for the open portion. In the closed position, the nonlinear effects of friction are simulated.

The ELASNL and BUSH elements use tabular load vs. deflection data to define their nonlinear characteristics. Specific details are presented in Section 16.4.3.2.

Table 16-2. ELEMENTS FOR NONLINEAR MATERIAL ANALYSIS

ELEMENT	DESCRIPTION
BAR	Nonlinear material in the BAR element allows nonlinear hinges at either end for interaction of ben.ding and axial force. In the middle sections, transverse shear and torsion are linear. Pin joints are not allowed.
BEAM	Nonlinear properties are similar to the BAR. An additional option allows the end sections, as well as the middle sections, to be nonlinear.
BUSH	Nonlinear spring behavior is available for all six degrees of freedom. This element may have nonlinear frequency dependent properties. The use of this feature is described in Chapter 13.
ELASNL	Nonlinear spring similar to the linear ELAS2 element.
GAP	Nonlinear GAP element.
HEXA	Three-dimensional nonlinear properties for the 8 node element only.
PENTA	Three-dimensional nonlinear properties for the 6 node element only.
PILE	Axial, lateral, and torsional nonlinear soil properties are available. The nonlinear structural properties are the same as that of BAR.
QUAD4 **QUADR**	Two-dimensional nonlinear properties with membrane and bending interaction. Off-plane shear is linear. There are a variable number of sample points across the thickness at the center of these elements.
ROD	Nonlinear axial material behavior is available. The torsional behavior is linear.
TETRA	Three-dimensional nonlinear properties are available for linear and quadratic TETRA elements (Five and ten node elements).
TRIA3 **TRIAR**	Two-dimensional nonlinear properties with membrane and bending interaction. Off-plane shear is linear. There are a variable number of sample points across the thickness at the center of these elements.

16.4 NONLINEAR MATERIAL MODELS

A typical nonlinear material model consists of a constitutive law, and if necessary, criteria for yield and hardening rules. Currently, three types of material nonlinearity are treated in UAI/*NASTRAN*: ***plasticity***, ***nonlinear elasticity***, and the ***combined nonlinear elasticity*** and ***plasticity***. The stress-strain responses are nonlinear in all three cases. However, in nonlinear elasticity, the unloading curve traces the loading curve to the origin when there is external unloading. On the other hand, in plasticity, the solution follows a linear stress-strain curve during unloading so that there will be permanent strains when the external loading is removed. In the combined nonlinear elastic and plastic analysis, the material is first assumed to be nonlinear elastic up to a predetermined range, after which, the material is assumed to be plastic. In the following sections, the three types of nonlinearity and the necessary input data are discussed in detail.

16.4.1 Plasticity

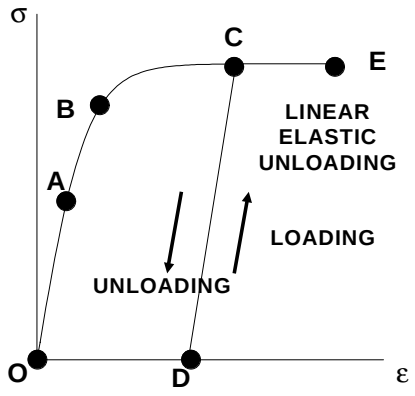
UAI/*NASTRAN* supports materials that exhibit ***elastoplastic*** behavior. Figure 16-3 shows the results of a typical tensile test. From points **O** to **A**, the stress-strain relationship is linear. Point **A** is known as the ***proportional*** limit. It is in this range that linear elastic theories are valid. As loading increases, the stresses no longer linearly increase with strain, but the material remains elastic. In other words, when the external load is removed the material returns to its original state. This condition prevails until point **B**, known as the ***Yield Point***, is reached. For materials with large differences in proportional limit and yield point it may be prudent for you to use nonlinear elastic analysis or combined nonlinear elastic and plastic analysis.

For many other materials, with little difference between proportional limit and yield point, plasticity analysis should be used. As the load increases beyond the yield point, the strain increases at a much greater rate as shown in the figure, from point **B** to **C** on the curve. However, loading must be increased to deform the specimen further. This condition is called ***Work Hardening*** or ***Strain Hardening***. If the material has an initial elastic range, negligible difference between proportional limit and yield stress, and little or no work hardening, it is termed as ***elastic-perfectly-plastic***, or simply ***Elastic-Plastic***.

For many types of steel, elastic-plastic idealization is often used. If at any stage after the yield point **B**, the external load is removed, unloading will be linear along **CD** as shown in the figure. The slope of the line **CD** is called the ***Unloading Modulus***. For most materials, the unloading modulus is equal to the elastic modulus *E*.

If the material is then loaded again, the unloading curve **CD** will be retraced with minor deviations which are usually neglected. Plastic flow does not start until the point **C** is reached. With further loading, the stress-strain response continues along **CE** as if no unloading had occurred. Point **C** can be considered as a new yield point for a strain hardening material.

Figure 16-3. ELASTO-PLASTIC MATERIAL



16

16.4.2 The Yield Criteria

In a simple tensile test, the ***Yield Stress*** is a point beyond which the material will deform plastically. However, because the stress at a point is always multi-dimensional it is necessary to relate the yield stress to generalized stress components. This relationship is defined by a ***Yield Criteria***. **UAI/*NASTRAN*** adopts the ***Work Hardening Hypothesis*** According to this hypothesis, the yield stress can be expressed as a function of the ***Effective Strain***, $\bar{\varepsilon}_p$, as shown below:

$$\bar{\sigma} = \sigma_y(\bar{\varepsilon}_p) \tag{16-6}$$

where $\bar{\sigma}$, the ***Effective Stress***, is a function of the stress vector at a point. (16-6) indicates that if the effective stress at a point is equal to the yield stress, σ_y obtained in the tensile test, the material will deform plastically at that point. The specific form of $\bar{\sigma}$ is dependent on the specific yield criterion adopted.

The following sections briefly describe the yield criteria available in **UAI/*NASTRAN***. You may consult any text book on mechanics for detailed descriptions of them.

16.4.2.1 The von Mises Yield Criterion

The von Mises criterion states that yielding occurs when, at any point in the body, the distortion energy per unit volume for a state of combined stresses becomes equal to that associated with yielding in a simple tension test. This may be expressed mathematically as:

$$\bar{\sigma} = \frac{1}{\sqrt{2}}\left[(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6(\tau_{xy}^2 + \tau_{yz}^2 + \tau_{zx}^2) \right]^{\frac{1}{2}} \tag{16-7}$$

16.4.2.2 The Tresca Criterion

The Tresca criterion states that yielding begins when the maximum shear stress at any point in the material is equal to the shear stress when the tensile test specimen yields:

$$\bar{\sigma} = \sigma_1 - \sigma_3 \tag{16-8}$$

where σ_1 and σ_3 are the maximum and minimum principal stresses respectively.

16.4.2.3 The Mohr-Coulomb Yield Criterion

The Mohr-Coulomb criterion states that the critical shear stress which causes yielding at a point in the material is related to the internal friction by the following equation:

$$\tau = c - \sigma_n \tan\varphi \tag{16-9}$$

where τ is the ***critical*** shear stress causing yielding, c is the cohesion, φ is the angle of internal friction, and σ_n is the normal stress acting on the shearing plane. This criterion is a generalization of the ***Coulomb*** friction law. The above equation can be rewritten in the form of yield criterion as:

$$\bar{\sigma}(\boldsymbol{\sigma}) = \sigma_y \tag{16-10}$$

where,

$$\bar{\sigma} = \sigma_1 - \sigma_3 \frac{1 - \sin\varphi}{1 + \sin\varphi} \tag{16-11}$$

and,

$$\sigma_y = \frac{2c\cos\varphi}{1 + \sin\varphi} \tag{16-12}$$

Note that the Mohr-Coulomb criterion reduces to that of Tresca criterion if the internal angle of friction, φ equals 0.0.

16.4.2.4 The Drucker-Prager Yield Criterion

The Drucker-Prager criterion is an approximation of the Mohr-Coulomb yield criterion. It states that yielding will occur if,

$$\sqrt{J_2} = k - \alpha I_1 \tag{16-13}$$

where,

$$I_1 = \sigma_x + \sigma_y + \sigma_z \tag{16-14}$$

$$J_2 = \frac{1}{6}\left[(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6(\tau_{xy}^2 + \tau_{yz}^2 + \tau_{zx}^2) \right] \tag{16- 15}$$

and k and α are constants. The constants k and α can be evaluated by requiring that the Drucker-Prager circle coincide with the outer corners of the Mohr-Coulomb hexagon. The above yield criterion can be written as:

$$\bar{\sigma}(\boldsymbol{\sigma}) = \sigma_y \tag{16-16}$$

where,

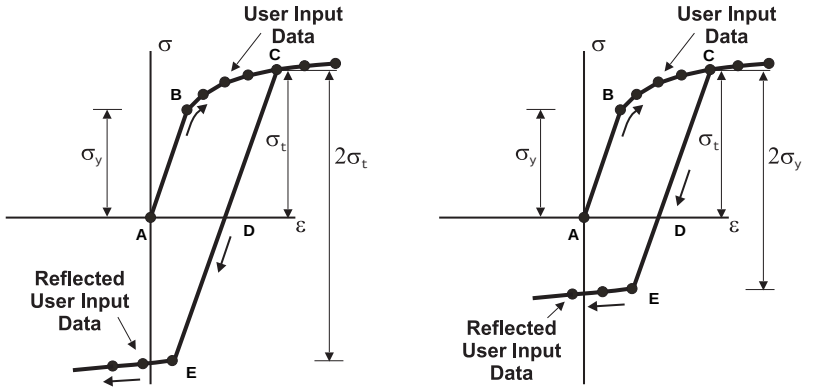
$$\bar{\sigma} = \frac{3 - \sin\varphi}{3 + \sin\varphi}\sqrt{3}\sqrt{J_2} + \frac{2\sin\varphi}{3 + \sin\varphi} I_1 \tag{16-17}$$

and,

$$\sigma_y = \frac{6c\cos\varphi}{3 + \sin\varphi} \tag{16-18}$$

The Drucker-Prager yield criterion reduces to the von Mises yield criterion if the angle of internal friction, φ equals 0.0.

Figure 16-4. ISOTROPIC AND KINEMATIC HARDENING

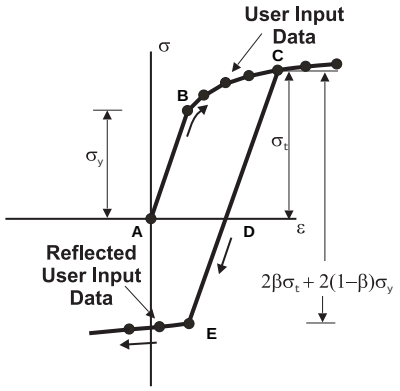


a. ISOTROPIC HARDENING **b. KINEMATIC HARDENING**

16.4.3 Hardening Rules

For many materials, the yield points in tension and compression are the same. However, if the material is first loaded in tension beyond the yield point and the load is then reversed to compression, the yield stress in the compressive regime will be considerably less than that in tension. This is known as the ***Bauschinger*** or ***Pinching*** effect. There are several simplified models to account for the Bauschinger effect. At one extreme is the model in which the hardening mechanism acts equally in tension and compression. Thus, the compressive yield stress is equal to the current tensile yield stress as shown in the figure. This model, which is shown in Figure 16-4a, has no correction for Bauschinger effects, and is called ***Isotropic Hardening*** model. At the other extreme, it is assumed that the elastic unloading and reloading, which takes place in the opposite direction, is twice the value of the first yield point, as shown by path **CDE** in Figure 16-4b. According to this model, the elastic range of the material remains unchanged due to load reversals, the compressive yield stress being reduced by the same amount as the current increment in tensile yield stress. This is called ***Kinematic Hardening***.

Figure 16-5. COMBINED HARDENING

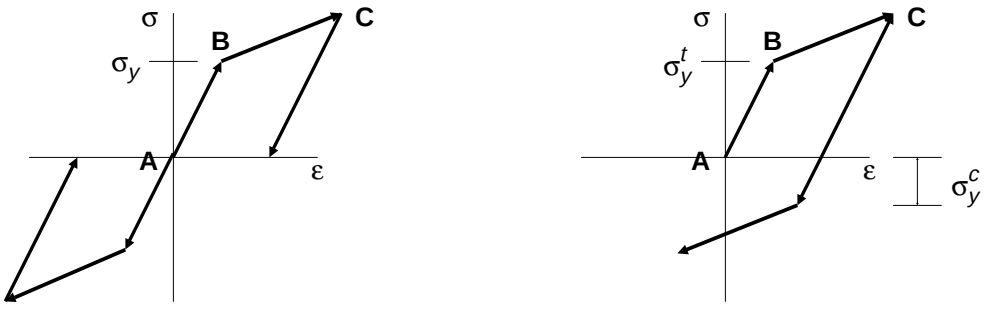


UAI/*NASTRAN* also provides a combination of the two hardening models. This called the ***Combined Hardening*** model, which states that the compressive yielding will occur when:

$$\sigma_c = -\sigma_t - 2(1-\beta)(\sigma_y - \sigma_t), \quad 0.0 \le \beta \le 1.0 \tag{16-19}$$

This is shown in Figure 16-5. Note that when $\beta = 0.0$, the combined hardening predicts kinematic hardening, and when $\beta = 1.0$, it predicts isotropic hardening. With $\beta = 0.5$, the tensile yield stress and compressive yield stress are independent of each other and always equal to

Figure 16-6. ISOTROPIC AND KINEMATIC HARDENING



a. CYCLIC HARDENING **b. NONCYCLIC HARDENING**

the initial yield stress σ_y. In **UAI/*NASTRAN***, β may be selected with the **BETA** field of the **TABLENL** Bulk Data entry.

The ***hysteresis*** loops for a few cycles of loading and unloading is completely determined by the stress-strain curve in the tensile range along with the hardening rule as shown in Figures 16-4 and 16-5. In order to obtain generality, **UAI/*NASTRAN*** offers two more cyclic loading options in addition to these hardening rules. The first one is called ***Non-standard Cyclic*** (or simply ***Noncyclic***), for which you can input both stress-strain curves in tensile and compressive ranges independently, thus resulting in a hysteresis loop shown in Figure 16-6b. Another is called ***Cyclic***, for materials with gaps, as shown in Figure 16-6a. The noncyclic and cyclic options are available only for the one-dimensional elements ELASNL, BUSH, ROD, BAR, BEAM and PILE. See Section 16.4.3.2 for additional information about the cyclic and noncyclic options.

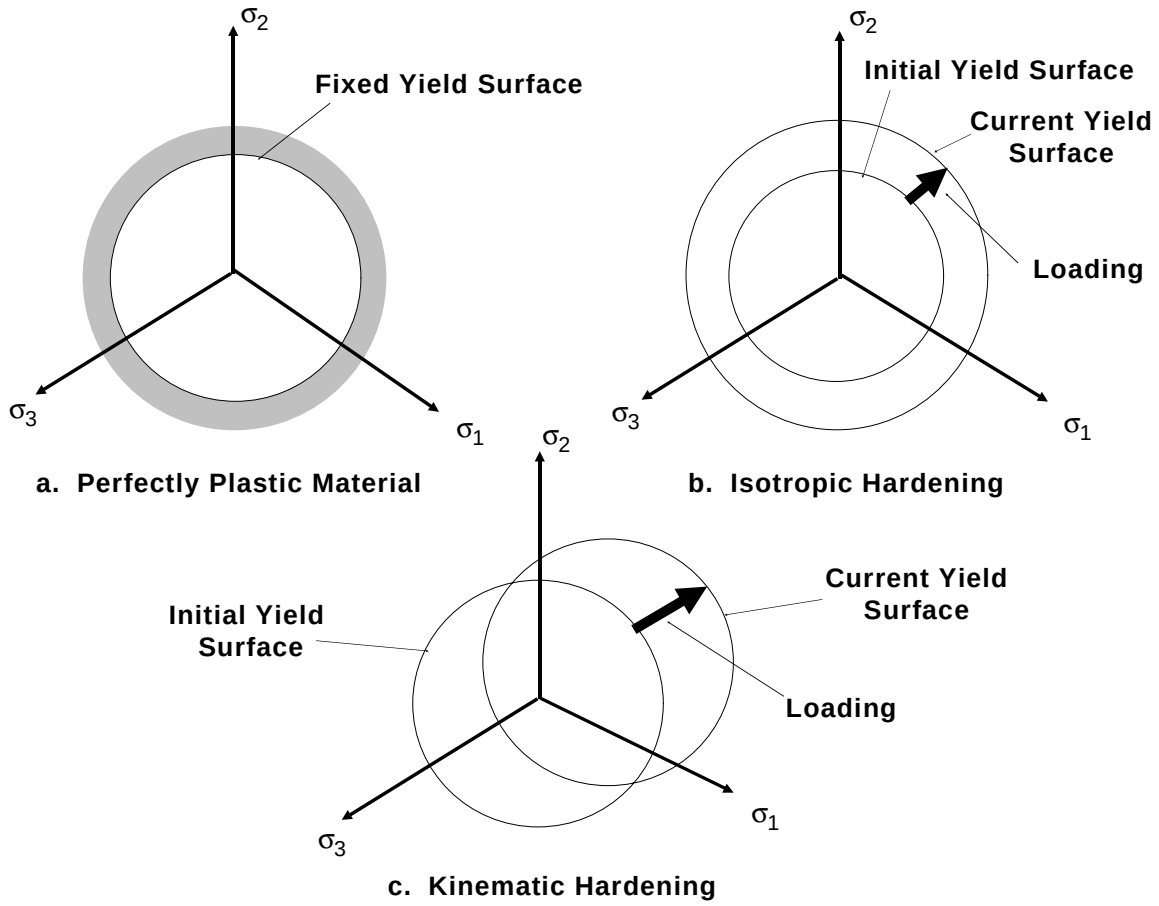
In three dimensions, the yield criteria can be represented by yield surfaces in the stress space. For a ***perfectly-plastic*** material, the yield surface remains constant at the initial yield point, as shown in Figure 16-7a, where the von Mises yield surface is used as an example. However, for a work hardening material, the yield surface must change for continued loading beyond the first yield point. How the surface changes with loading depends on the hardening rule adopted.

For isotropic hardening, the yield surface is assumed to expand uniformly outward with the stress-strain history, but it retains the same shape as the initial surface, as shown in Figure 16-7b. This hardening model does not account for the Bauschinger effect. On the other hand, the Prager-Ziegler kinematic hardening rule is adopted to account for the Bauschinger effect. In this model, the total elastic range is maintained constant by translating the initial yield surface without deforming it, as shown in Figure 16-7c.

Since the kinematic hardening model tends to over-correct for the Bauschinger effect, the combined isotropic and kinematic hardening model is considered to be a better approach to correcting for this effect. In the combined hardening model, the yield surface expands and translates. The degree of expansion or translation depends on the combined hardening parameter β. For $\beta = 0.0$, the hardening is of kinematic type, and for $\beta = 1.0$, the hardening is isotropic.

Note that the other two cyclic loading options, noncyclic and cyclic, supported by **UAI/*NASTRAN*** for one-dimensional elements, have no counterparts in the multidimensional elements.

Figure 16-7. YIELD SURFACES AND HARDENING RULES



16

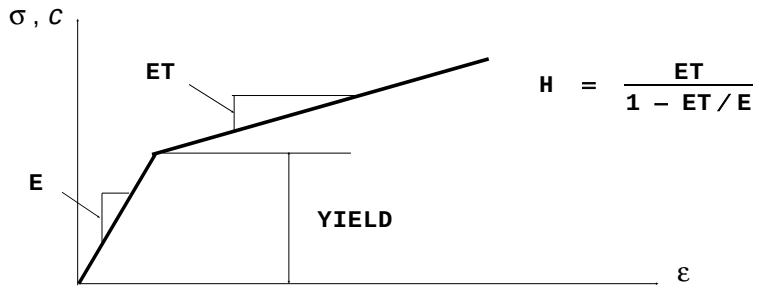
16.4.3.1 Defining Plastic Materials

The BAR, BEAM, ROD, TRIA3, TRIAR, QUAD4, QUADR, HEXA, PENTA, and TETRA elements and the structural properties of the PILE element reference material properties defined by **MAT1** Bulk Data entries. To define the nonlinear properties, the linear **MAT1** data are supplemented by a combination of **MAT1NL**, **TABLENL** or **MAT1NL1** Bulk Data entries which provide a definition of the stress-strain relationship. Consider the input:

MAT1	**MID**	**E**		**NU**	**RHO**	**ALFA**	**TREF**	**GE**	
MAT1NL	**MID**	**TNLID**	**TYPE**	**YC**					
TABLENL	**TNLID**	**EU**	**SCALE**	**HYSTYP**		Φ		**EP**	**+TI**
+TI	**X1**	**Y1**	**X2**	**Y2**	**X3**	**Y3**	**ENDT**		

Notice that there must be a linear, **MAT1**, and nonlinear, **MAT1NL**, property definition both of which have the same identification number, **MID**. The **MAT1NL** entry, in turn, references a **TABLENL** identification number, **TNLID**. The fourth field of the **MAT1NL** entry specifies that

Figure 16-8. NONLINEAR INPUT FOR BILINEAR STRESS-STRAIN



the material **TYPE** is **PLASTIC** and the fifth field, **YC**,is used to specify the yield criterion. The valid yield criteria are **VM**, **TRESCA**, **MC** and **DP**. The **EU** and **EP** fields of the **TABLENL** entry allow you to specify an alternate unloading modulus and an alternate elastic modulus to be used in the plastic range, respectively. The hardening rule is given by the **HYSTYP** field. The yield stress, or cohesion, is given by the first Y-value. Φ is the angle of internal friction which is used when the Mohr-Coulomb yield criterion, **MC**, is selected. The stress-strain curve is defined by the continuations of **TABLENL** Bulk Data entry. The slope defined by the first pair of **TABLENL** fields, **X1** and **Y1**, must be equal to the Young's modulus, **E**, defined by the **MAT1** entry. You should specify a Poisson's ratio, ν, on the **MAT1** entry. The shear modulus is then computed using Young's modulus and Poisson's ratio if it is explicitly required in the non-linear analysis.

You are cautioned about the **EU** and **EP** fields of the **TABLENL** entry. In most cases, you should use the default values for unloading modulus and elastic modulus in the plastic range. Both of these defaults are the Young's modulus, **E**. However, if you choose to specify alternate unloading modulus and/or alternate elastic modulus in plastic range, then the elastic and shear moduli used in the plastic theory are no longer the **E** and **G** from the **MAT1** Bulk Data entry. Instead, the elastic modulus is **EP**, and the shear modulus is computed by the Hooke's Law:

$$G = \frac{\mathbf{EP}}{2(1+\nu)}$$

where ν is the Poisson's ratio given on the **MAT1** entry. The initial linear stiffness matrix is still computed by using the **E** and **G** on the **MAT1** entry.

Alternately, the linear **MAT1** data may be supplemented by a **MAT1NL1** Bulk Data entry to signify that the element is a nonlinear element with a bilinear stress-strain curve. In this case, the **TABLENL** Bulk Data entry is not required. Note that **ET** is the tangential modulus of the stress-strain curve; it is not the hardening function, **H**, used in many text books. The relationship between **ET** and **H** is given in Figure 16-8.

16.4.3.2 Defining Nonlinear Spring and GAP Properties

Nonlinear data input for the ELASNL and BUSH elements and the soil properties of the PILE element are different from those described above. The material property for these types of elements is actually a spring rate. The spring rate may be defined as a constant for a linear version of the finite element; in case of ELASNL element, the input becomes identical to that of

ELAS2. If the element is nonlinear, **TABLENL** Bulk Data entries are used to define the element internal force-deflection response. The slope of this curve is the spring rate. Since this type of element does not reference a **MAT1NL** or a **MAT1NL1** entry, the unloading modulus (**EU**) field of the **TABLENL** entry is used to indicate the type of nonlinear property. A blank **EU** field or a non-zero value of **EU** indicates plastic properties. A sample input is:

CELASNL	EID	12	G1	C1	G2	C2		S	
TABLENL	12		SCALE	HYSTYP					+TI
+TI	0.001	3.+4	0.002	2.+4	ENDT				

In this example, field 3 (**EU**) of the **TABLENL** entry is left blank, which indicates that this element has plastic material properties with the default unloading modulus equal to the Young's modulus (for the example discussed, **E = 3.0E+7**). Other fields on the **TABLENL** entry are the same as described in previous examples.

The two additional cyclic loading options, ***noncyclic*** and ***cyclic***, which are supported for the one-dimensional elements only, are entered on the **HYSTYP** field of the **TABLENL** entry. If **HYSTYP=NONC**, you can input the stress-strain data independently for tension and compression ranges, the material may be loaded and unloaded, and a hysteresis curve as shown in Figure 16-9 will be followed.

If **HYSTYP=CYCLIC**, the material behavior illustrated in Figure 16-10 is used. This behavior is typical for a material that has been loaded cyclically in the past, and a ***gap effect*** is now desired. An example would be a pile in soft clay. For the **CYCLIC** option, the stress-strain curve should be defined only in the tensile range on the **TABLENL** entry. The compressive range is then created automatically.

The **SCALE** field on the **TABLENL** entry scales the Y-axis data by the factor specified. This may be useful if the data are obtained in a laboratory or field test and have not been normalized for general use, but they contain scaling data particular to the test specimen. For example, if lateral force deflection data are obtained in the field test of a pile, the force data represent the total force for that particular pile. These data must be divided by the diameter of the pile for use in **UAI/*NASTRAN***. The program assumes that the data are normalized in this manner, allowing such data to be used for other, similar piles of different diameters. The default value for **SCALE** is 1.0.

Figure 16-9. NONCYCLIC HYSTERESIS

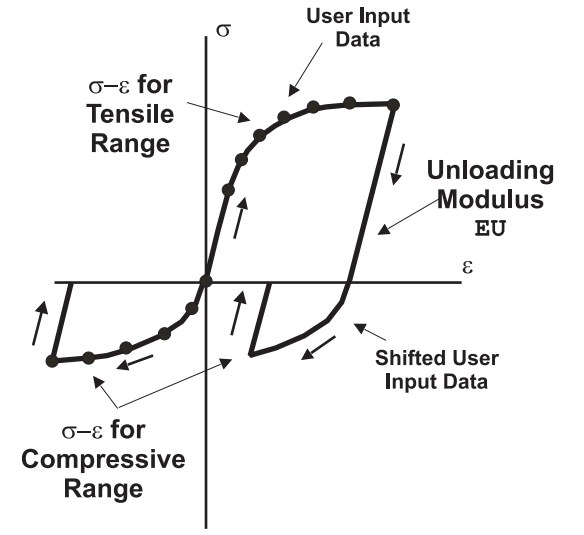
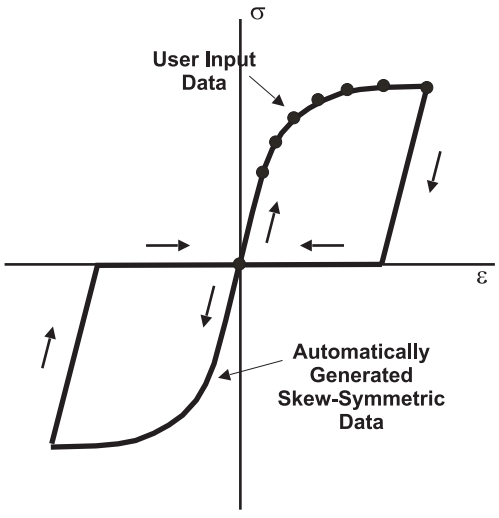


Figure 16-10. CYCLIC HYSTERESIS



The GAP element is defined using two spring rates, a very stiff spring for the closed position and a very soft spring for the open position. In the closed position, the nonlinear effects of friction are simulated. The GAP element can also be used to model a cable by reversing the two spring stiffnesses. There are no other references to the material properties by the GAP element. Also refer to Chapter 5 of this manual.

16.4.4 Nonlinear Elasticity

The stress-strain curve for a typical nonlinear response in shown in Figure 16-11. For non-linear elastic materials, the loading and unloading paths follow the same curve, so that there is no permanent deformation of the material.

The functional relationship between stress and strain for a nonlinear elastic material can be represented as:

$$\sigma = \sigma(\varepsilon) \tag{16-20}$$

The relationship can be completely established by using the data from a simple tension test. With the stress-strain curve known, the nonlinear analysis of the one-dimensional elements is rather straight forward. However, for two- and three-dimensional elements, the situation is quite different due to the additional stress components. **UAI/*NASTRAN*** adopts the model where assumptions are made to relate the multiaxial stresses to the uniaxial stress-strain curve. The use of the effective strain is made in the model. The effective strain, $\bar{\varepsilon}$, can be defined using the strain energy equation:

$$\frac{1}{2} E_s(\bar{\varepsilon})\bar{\varepsilon}^2 = \frac{1}{2}\boldsymbol{\sigma}^T\boldsymbol{\varepsilon} \tag{16-21}$$

where $E_s(\bar{\varepsilon})$ is the secant modulus shown in Figure 16-12, $\boldsymbol{\sigma}$ is the nonlinear stress vector, and $\boldsymbol{\varepsilon}$ is the strain vector. The effective stress, $\bar{\sigma}$, is the stress read from the stress-strain curve at $\bar{\varepsilon}$. In the next step, the effective stress is then used to relate the multiaxial stress-strain to the uniaxial responses. This is accomplished using the following equation:

Figure 16-11. NONLINEAR ELASTICITY

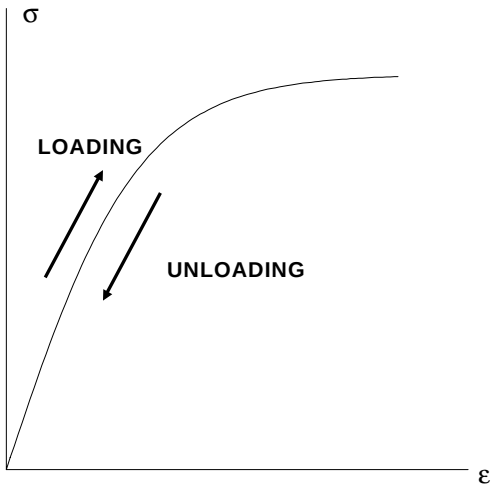
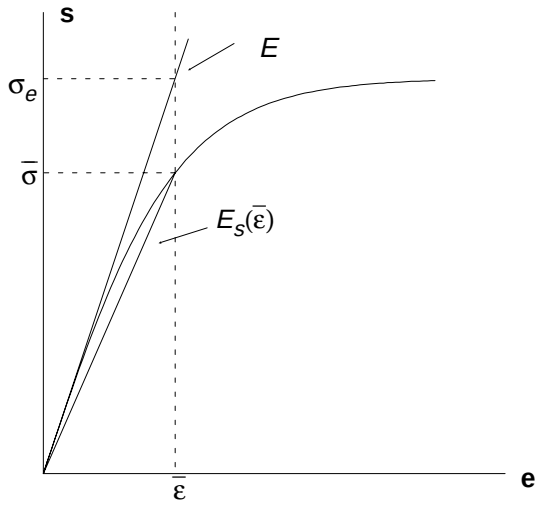


Figure 16-12. STRESS COMPUTATION



16

$$\boldsymbol{\sigma} = \frac{E_s(\bar{\varepsilon})}{E}\boldsymbol{\sigma}_e \tag{16-22}$$

where $\boldsymbol{\sigma}$ is the nonlinear stress vector, $\boldsymbol{\sigma}_e$ is the linear stress vector, and E is Young's Modulus.

The one-dimensional case for this equation is shown in Figure 16- 10. With $\bar{\varepsilon}$ and $\boldsymbol{\sigma}$, as defined above, the tangential stress-strain matrix may be computed and the nonlinear elastic analysis in multiaxial situations can be performed.

16.4.4.1 Defining Nonlinear Elastic Materials

The input data used to define nonlinear elastic materials for BAR, BEAM, ROD, TRIA3, TRIAR, QUAD4, QUADR, HEXA, PENTA, and TETRA elements is similar to that described earlier. The general form of the Bulk Data entries is:

MAT1	**MID**	**E**		**NU**	**RHO**	**ALFA**	**TREF**	**GE**	
MAT1NL	**MID**	**TNLID**	**NLELAST**						
TABLENL	**TNLID**	**EU**	**SCALE**						**+TI**
+TI	**X1**	**Y1**	**X2**	**Y2**	**X3**	**Y3**	**ENDT**		

In this case, the fourth field of the **MAT1NL** entry indicates a **TYPE** of **NLELAST** and the fifth field is not used. The **TABLENL** entry is simpler in that there is no input for the angle of internal friction, and hysteresis type.

For ELASNL, BUSH and the soil properties of the PILE element, the unloading modulus **EU** must be explicitly set to **0.0** in order to request the nonlinear elastic material.

16.4.5 Combined Nonlinear Elasticity and Plasticity

For the combined nonlinear elasticity and plasticity (**EPCOMB**), the material properties are treated as nonlinear elastic when the stress is less than the yield, and as plastic when the stress is greater than the yield stress. This is accomplished by defining a parameter η, which is computed as:

$$\eta = \frac{\bar{\sigma}}{\mathbf{YIELD}} \tag{16-23}$$

where **YIELD** is the yield stress (or cohesion) input using a **TABLENL** Bulk Data entry, and $\bar{\sigma}$ is the effective stress computed by the plastic theory. In each element, the material properties are treated as nonlinear elastic at the start of analysis. At $\eta = 1.0$, the element becomes plastic and stays plastic till the end of the analysis. It will not return to the nonlinear elastic behavior.

For the plastic range in **EPCOMB**, the elastic modulus and the shear modulus used in the plastic theory are not the initial **E** and **G** input along **MAT1** Bulk Data entry, but E_c and G_c given by:

$$E_c = \begin{cases} E_p & \text{if } \textbf{EP} \text{ is input on a } \textbf{TABLENL} \text{ entry} \\ \dfrac{\textbf{YIELD}}{\varepsilon_y} & \text{by default} \end{cases} \tag{16-24}$$

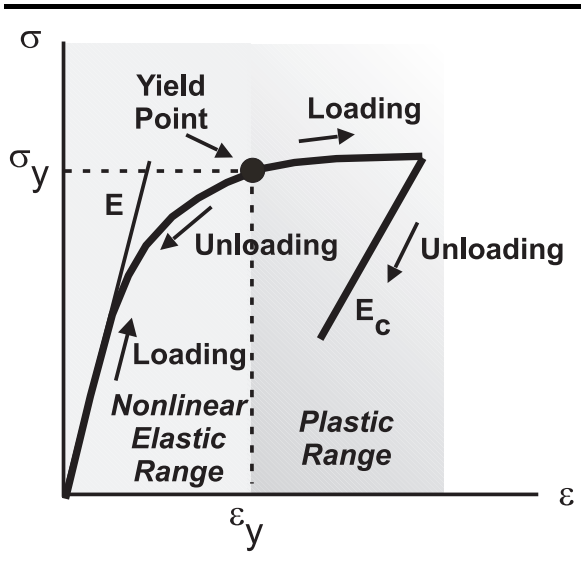
$$G_c = \frac{E_c}{2(1+\nu)} \tag{16-25}$$

where ε_y is the strain read from the stress-strain curve corresponding to **YIELD**, and ν is the Poisson's ratio input on **MAT1** entry. A schematic diagram for **EPCOMB** is illustrated in Figure 16-13.

16.4.5.1 Defining Combined Materials

When **EPCOMB** is selected, the **YIELD** field on the **TABLENL** entry must contain a value greater than **Y1** which is the first **Y** value for **X>0.0** on the **TABLENL** entry. If **YIELD**≤**Y1** then **EPCOMB** will become a plastic analysis. There are materials whose unloading modulus is not equal to the initial Young's modulus. The **EPCOMB** field with default E_c is an alternate way to input an unloading modulus which is not equal to the initial Young's modulus.

Figure 16-13. COMBINED MODEL



16.5 INPUT DATA REQUIREMENTS

This section provides information on input data required to perform Material Nonlinear analysis with **UAI/*NASTRAN***.

16.5.1 Executive Control Commands

The Executive Control commands:

```
APP NONLINEAR
SOL STATICS
```

are required to perform a nonlinear analysis. The Rigid Format for nonlinear statics analysis uses a ***Nonlinear Database***, called the **NLDB**, to store data during execution. If you are performing a single analysis execution, then you do not need define this database. However, if you intend to perform a RESTART of your analysis, then you must use an **ASSIGN** command to initialize and name the **NLDB**. The simplest form of this command is:

```
ASSIGN MYDB=SAVENL,NEW,SIZE=1000,USE=NLDB
```

This **ASSIGN**s a new database that is used for use in a subsequent RESTART as described in Section 16.5.7. For some host computers, other parameters may be required. See the ***User's Reference Manual*** for additional information.

16.5.2 Case Control Commands

The basic Case Control structure for linear analysis is also applicable to nonlinear analysis. However, there are seven special Case Control commands that are specific to nonlinear analysis. These are summarized in Table 16-3. The use of these Case Control commands and their formats are briefly described in the following sections.

16.5.2.1 The NLREDUCE Command

For computational efficiency, **UAI/*NASTRAN*** nonlinear analysis defines two complimentary sets of degrees of freedom: linear and nonlinear. The linear set includes all degrees of freedom which are connected only to linear elements, and, similarly, the nonlinear set includes all degrees of freedom which are connected to any nonlinear elements. Additionally, linear degrees of freedom which are dependent on a nonlinear degree of freedom through a rigid

Table 16-3. NONLINEAR ANALYSIS CASE CONTROL COMMANDS

COMMAND	PURPOSE
NLPRINT	To request print of solution history
NLREDUCE	To request automatic partitioning of linear and nonlinear degrees of freedom.
NLRESTART	To request nonlinear restart
NLSAVE	To select load increment data to be saved on the nonlinear analysis database.
NLSOLVE	To select nonlinear solution control data.
NLTYPE	To select the type of nonlinear analysis.
STEP	To define sequentially applied loading and unloading.

element or **MPC** equation, are placed in the nonlinear set. Unless you specify otherwise, all degrees of freedom are placed in the nonlinear set. You may request the creation of the linear set by using the **NLREDUCE** Case Control command:

```
NLREDUCE   [ = { YES } ]
               { NO  }
```

When you use this command, the linear set is automatically defined as described above. The **NLREDUCE** command is recommended because it greatly improves the efficiency of the nonlinear solution algorithm.

16.5.2.2 The NLPRINT Command

You use the **NLPRINT** command to request either **DETAIL**ed or **SUMMARY** print during the nonlinear iterations. If you do not use this command, then you will automatically get the **DETAIL**ed print. A description of the print from the solution module is presented in section 16.6. The format of this Case Control command is also explained in that section.

16.5.2.3 The NLRESTART Command

The **NLRESTART** command is used to request a nonlinear restart, which is described in detail in section 16.5.7.

16.5.2.4 The NLSAVE Command

The **NLSAVE** command controls the selection of load increment data which will be saved on the nonlinear database. The default condition is that only data at the completion of load steps will be saved. This conserves database space at the expense of reducing possibly desirable output. The format of the **NLSAVE** Case Control command is:

```
NLSAVE EVERY nth INCREMENT
```

The **NLSAVE** command may be used to override the default and force saving of all data at selected load increments. Print, punch and plot output are available only for saved data. In addition, you must use the **NLSAVE** command to save the appropriate data if you intend to perform a nonlinear increment RESTART.

16.5.2.5 The NLSOLVE Command

You use the **NLSOLVE** command to select an **NLSOLVE** Bulk Data entry which controls the nonlinear solution procedure. The format of this Case Control command is:

```
NLSOLVE = nlsid
```

This command may be placed anywhere in the Case Control packet, thus permitting different solution control parameters for different SUBCASEs and STEPs.

16.5.2.6 The NLTYPE Command

You use the **NLTYPE** command to specify the type of nonlinear analysis that you are performing. The format of the command for material nonlinear analysis is:

```
NLTYPE MAT
```

The analysis type option, **MAT**, is the default for nonlinear analysis, thus it is not necessary to use the command for this discipline. Other uses of this command are found in Chapter 17.

16.5.2.7 The STEP Command

The sequence of load applications is important in determining the final response of a nonlinear structure. The **STEP** Case Control command provides you with a mechanism for defining the SUBCASE load in terms of one or more sequentially applied loads. The format of this Case Control command is:

```
STEP stepid
```

The following examples illustrate the manner in which the **SUBCASE** and **STEP** Case Control commands are used.

1. The simplest input contains one SUBCASE and one STEP. Either or both the **SUBCASE** and **STEP** commands may be omitted from the input. In such cases, **UAI/*NASTRAN*** assigns **SUBCASE 1** and **STEP 1** to the analysis. The complete load is defined and a solution strategy is selected using the **NLSOLVE** command. One or more load increments may be used depending on the **NLSOLVE** data. If the **SUBCASE** command is used, the SUBCASE identification numbers need not begin with the integer 1. However, if the **STEP** command is used, the step id must begin with the integer **1** for each subcase.
2. With one SUBCASE and more than one step, the procedure is only slightly more complicated than previous example. Each step defines the total new external load which will be applied by the completion of the step. STEP identification numbers must be sequential integers starting at 1. Structure unloading may be modeled with steps. For example, assume a 3 step problem where at the end of step 1, the load is 50.0, at the end of step 2 the load is 75.0, and unloading occurs at step 3 and the load is 25.0. All the loads are applied along the z-axis of the basic coordinate system at GRID 67. The Case Control packet would contain:

   ```
   SUBCASE 1
      STEP 1
         LOAD 10
      STEP 2
         LOAD 20
      STEP 3
         LOAD 30
   ```

   Bulk Data entries for the applied loads may be defined in any of the normal ways used for Static Analysis.

3. Independent SUBCASEs may be executed where loads and boundary conditions change. No load history information is transmitted from one **SUBCASE** to the next. At the start of each SUBCASE the deflections, stresses and strains throughout the model are equal to zero. STEP identification numbers must begin with **1** for each **SUBCASE** and must be sequential integers incremented by **1**.

4. A Case Control command placed below the step level allows that command to vary from one step to another; if it is placed above the step level, the command remains constant for all steps in the subcase. Some of the Case Control commands may be placed either above or below the step level, others must be placed above the step level. The following is a general rule — All load requests (**LOAD**), solution technique requests (**NLSOLVE**), output requests (**NLPRINT**, **NLSTRESS**, **DISP**, **NLFORCE**, etc.) and other data saving requests (**NLSAVE**) may be placed either above or below the step level.

   However, with the exception of **SPC**, all other requests to change boundary conditions (**NLREDUCE**, **AUTOSPC**, **MPC**, etc.) ***must*** be above the step level.

16.5.3 Bulk Data

Table 16-4 presents a summary of Bulk Data entries which may be used for nonlinear structural analysis. Proper usage of these entries is described in the following sections.

16.5.4 Loads

Nonlinear static analysis supports all external mechanical loads and enforced displacement loads. Thermal and deform loads are supported only on linear elements.

The concepts of load STEP and load increment are introduced for material nonlinear analysis. ***Load steps*** are used to define a ***chronological*** order to the application of loads, and ***load increments*** are used in the solution procedure. Unlike linear analysis, the ***ultimate response*** of a structure with nonlinear material properties is ***dependent*** on the sequence of load applications to the structure. Several load steps in a subcase may be used to define such a sequence. The Case Control command **STEP** is used to define load steps. The solution for a given step may be obtained by applying the step load incrementally. The number of load increments is specified on an **NLSOLVE** Bulk Data entry which is requested for any SUBCASE or STEP.

You must remember that the load applied in each step represents the cumulative load for the subcase through that step. The step loads specified during a subcase are ***not*** automatically added to one another from one step to the next. This procedure also provides a convenient method for defining a loading and unloading sequence on a model.

16.5.5 Database Control

Much of the data required to control a nonlinear analysis, and all the output from these analyses, are stored on the NLDB, the nonlinear database. This database is required for execution, but, in many cases, it may be only a temporary file. The database is defined by an **ASSIGN** Executive Control command. The simplest form of the NLDB **ASSIGN** command is:

```
ASSIGN NLDB = TEMP
```

The database assignment for a nonlinear RESTART is described in section 16.5.7. Database allocation is fully discussed in Chapter 1 of the ***User's Reference Manual***.

16.5.6 Solution Control

The nonlinear analysis solution procedure in **UAI/*NASTRAN*** is controlled by the Case Control command **NLSOLVE**. This command is used to select an **NLSOLVE** Bulk Data entry on which you specify parameters for controlling the solution procedure. Every field of this Bulk Data entry, with the single exception of the solution identification number field, has a default value. If you use all of the default values, you need not use an **NLSOLVE** Case control com-

Table 16-4. BULK DATA FOR NONLINEAR MATERIAL ANALYSIS

BULK DATA ENTRY	FUNCTION
CBAR	Defines nonlinear or linear bar connections
CBEAM	Defines nonlinear or linear BEAM connections
CBUSH	Defines nonlinear or linear BUSH connections
CELASNL	Defines nonlinear spring connections
CGAP	Defines GAP connections
CHEXA	Defines nonlinear or linear HEXA connections
CPENTA	Defines nonlinear or linear PENTA connections
CPILE	Defines nonlinear or linear PILE connections
CQUAD4	Defines nonlinear or linear QUAD4 connections
CQUADR	Defines nonlinear or linear QUADR connections
CROD	Defines nonlinear or linear ROD connections
CTETRA	Defines nonlinear or linear TETRA connections
CTRIA3	Defines nonlinear or linear TRIA3 connections
CTRIAR	Defines nonlinear or linear TRIAR connections
MAT1NL	Defines MAT1NL data
MAT1NL1	Defines bilinear MAT1 data
NLSOLVE	Specifies solution control parameters
PBAR	Defines general BAR properties
PBAR1	Defines special BAR properties
PBEAM	Defines general BEAM properties
PBEAM1	Defines special BEAM properties
PBUSH	Defines nonlinear or linear properties for the BUSH element
PGAP	Defines GAP properties
PPILE	Defines tubular PILE properties
PPILE1	Defines general PILE properties
PROD	Defines ROD properties
PSHELL	Defines TRIA3 and QUAD4 properties
PSOIL	Defines nonlinear or linear soil properties for the PILE element
PSOLID	Defines SOLID element properties
TABLENL	Defines soil or spring force-deflection data **or** nonlinear stress-strain data

mand or Bulk Data entry. Adetailed description of the input data fields, and guidelines for preparing input data and interpreting solution output was presented in Section 16.2.2.1.

16.5.7 Nonlinear Restart

The purpose of a nonlinear RESTART is to allow you to use the material properties of a previously converged solution as a new starting point for your analysis. This is helpful when you want to change the loading sequence, the solution criteria, or to extend the analysis.

The nonlinear RESTART is different from the **UAI/*NASTRAN*** standard restart in three aspects. First, the nonlinear RESTART must start from a solution boundary by specifying a SUBCASE, STEP and LOAD increment from which execution is continued. Second, the geometry and the initial material properties of the structure modeled can not be modified. This is obvious because any modification to the geometry and the initial material properties would invalidate the previous analysis and require the nonlinear solution to start from the very beginning. In such cases, it is simpler to initiate another cold start. Third, all data required for nonlinear RESTART are saved on the nonlinear database (**NLDB**) so that checkpointing of the cold start is not needed. The following is a description of the nonlinear restart data input.

16.5.7.1 Executive Control Modifications

The nonlinear database **NLDB** must be saved in the cold start execution and used in the restart execution. The following illustrates examples of commands necessary for this operation:

In ***cold start,*** you use the command:

```
ASSIGN NLDB = filename,NEW,REALLOC
```

and then, when performing the ***restart,*** you use:

```
ASSIGN NLDB = filename,OLD
```

Please note that the cold start nonlinear database will be modified by the restart. If you expect to perform several parallel restarts from a single cold start, you must save the cold start nonlinear database on a separate file so that this file can be copied and used in alternate restarts. On the other hand, you can execute serial restarts by using the nonlinear database sequentially for each restart.

16.5.7.2 Case Control Modifications

The presence of a Case Control command **NLRESTART** indicates that the current run is a restart execution. The Case Control packet contains ***both*** subcases and steps which have been executed in the cold start, and those which are to be executed in the restart. The first subcase, step and/or increment to be executed in the restart is indicated by the options on the **NLRESTART** command. This is depicted in the following example:

```
NLRESTART SUBCASE 1, STEP 2, INCREMENT 2
SUBCASE 1
      STEP 1
          LOAD = 10
      STEP 2
          LOAD = 20
      STEP 3
          LOAD = 30
```

In the above example, the first increment of Step 1, through the first increment of Step 2, have been previously executed. The restart execution begins with the second increment of Step 2 and continues through the last increment of Step 3.

The following Case Control commands may be modified in a nonlinear restart.

- Boundary conditions such as **MPC**, **SPC**, **NLREDUCE**, and **AUTOSPC**.
- Nonlinear solution control, **NLSOLVE**.
- The **LOAD** request.
- Output requests such as **NLPRINT**, **DISP**, and **NLSTRESS**.
- The data save request, **NLSAVE**.

Depending on the option selected with the **NLRESTART** command, the nonlinear RESTART may be logically divided into three types: a CASE RESTART, a STEP RESTART, or an increment RESTART.

- The CASE RESTART begins the execution with a SUBCASE. The five types of modifications described above are legal for a CASE RESTART.
- The STEP RESTART begins the execution with a **STEP** which may be a new step or a previously executed step. Although boundary condition modifications are allowed, you have the responsibility to determine whether they are meaningful.
- For an increment RESTART you may not modify boundary conditions or loads. You need to exercise discretion when attempting other types of modification at this level.

16.5.7.3 Bulk Data Modifications

The Bulk Data packet for a nonlinear restart contains only those entries which are to be added to the cold start. The deletion Bulk Data entry (/) is not allowed in a nonlinear restart, and therefore, Bulk Data entries present in the cold start can not be deleted. This is to serve as a restriction that the geometry and the initial properties can not be modified. You may make modifications to the Bulk Data packet by introducing new entries which may be copies of the original entries with appropriate changes and new identification numbers. The following is a list of entries that can be added in a restart:

- Load entries such as **LOAD**, **FORCE**, **PLOAD**, and **SPCD**.
- **NLSOLVE** entries.
- Boundary condition entries such as **SPC**, **SPC1**, and **MPC**.

16.6 SOLUTION RESULTS

This section describes the procedures you use to obtain output results from a Material Nonlinear analysis.

16.6.1 Material Nonlinear Analysis Output

There are several differences between the output for a material nonlinear analysis and a linear static analysis. These differences are described next to help you prepare input data and interpret output. Table 16-5 summarizes the Case Control commands used by Material Nonlinear analysis. Each of them is briefly described below.

The **NLPRINT** Case Control command:

```
NLPRINT = { DETAIL }
          { SUMMARY }
          { NONE    }
```

requests a printed summary of the nonlinear solution. You may select either a **DETAIL**ed history of unbalanced force iteration data and stiffness matrix update conditions for each **SUBCASE**, **STEP** and **LOAD** increment, or a brief **SUMMARY** of the nonlinear solution control parameters and a description of the conditions of solution convergence or divergence. You may disable the command in the usual manner by specifying **NONE**. Note that if you do not use this command, a **DETAIL**ed report is printed.

The **NLFORCE** Case Control command:

```
NLFORCE [ ( [ { PRINT   } , ] [ PUNCH ] ) ] = { ALL  }
        [     { NOPRINT }   ]                 { sid  }
                                              { NONE }
```

is used to request nonlinear element forces. Similarly, the **NLSTRAIN** Case Control command generates the output of nonlinear element strains:

```
NLSTRAIN [ ( [ { PRINT   } , ] [ PUNCH ] [ { EFFECTIVE } ] ) ] = { ALL  }
         [     { NOPRINT }   ]             { VONMISES  }       { sid  }
                                           { MAXSHEAR  }       { NONE }
```

With this command, you can request the calculation of effective strain, Hencky-von Mises equivalent strains, or maximum shear strains for plate elements or octahedral shear strains for solid elements.

Nonlinear element stresses are requested with the **NLSTRESS** Case Control command:

Table 16-5. CASE CONTROL COMMANDS FOR OUTPUT

CASE CONTROL COMMAND	FUNCTION
NLPRINT	To request for summary or detailed output of nonlinear solution procedure
NLFORCE	To request for nonlinear element force results
NLSTRAIN	To request for nonlinear element strain results
NLSTRESS	To request for nonlinear element stress results

```
NLSTRESS [ ( [ { PRINT } , ] [ PUNCH ] [ { EFFECTIVE } ] ) ] = { ALL }
         [ ( [ { NOPRINT } , ]         [ { VONMISES  } ] ) ]   { sid }
                                         { MAXSHEAR  }         { NONE }
```

As with element strain, you can request the calculation of effective stress, Hencky-von Mises equivalent stress, or maximum shear stress for plate elements or octahedral shear stress for solid elements.

16.6.2 Using Other Case Control Commands

Although the element forces and stresses for the nonlinear elements must be requested using the **NLFORCE** and **NLSTRESS** Case Control commands, respectively, the **FORCE** and **STRESS** commands are still used for obtaining output for any linear elements in your model. As a result, it is normal to have both **STRESS** and **NLSTRESS** commands in the same analysis. For GRID point output, however, there is no distinction between linear and nonlinear results.

Some output that is available in linear statics analysis is not available in nonlinear analysis. In particular, Grid Point Force Balance output, which is requested with the **GPFORCE** Case Control command, and element strain energy output, which is requested with the **ESE** command, are not available in nonlinear analysis. Use of these commands will not cause an error, but the output request will be ignored, and a warning message will be issued.

Printed output for GRID point quantities such as displacements, applied loads, SPC forces, and element quantities such as forces and stresses will have additional labeling indicating the SUBCASE, STEP, and load INCREMENT numbers, and **PERCENT** of step load applied. This is the case with **SORT1** output. For many nonlinear analyses, **SORT2** output might may be easier to interpret, thus allowing you to gain a better understanding of the model behavior with applied load. **SORT2** output is labeled using SCN as a substitute for the normal SUBCASE number. The solution control number (SCN) is an eight digit composite number beginning with a two digit SUBCASE identification number, followed by a two digit STEP number, followed by a two digit INCREMENT number, an concluded with by a two digit counter for the number of stiffness updates in the current increment. For example, the output for **SUBCASE 1**, **STEP 4**, load **INCREMENT 20** will be labeled as **SUBCASE 1042000** in the **SORT2** output. You will notice that the leading zero is discarded from the SCN. The stiffness update number in the SCN is used internally by **UAI/*NASTRAN*** as the major key to data on the Nonlinear Database. As a result of the construction of SCN, the SUBCASE identification numbers, the maximum number of steps per subcase, and load increments per step are all limited to 99. Also, there may not be more than 99 stiffness updates per load increment.

16

UAI/*NASTRAN* automatically limits the printed output of GRID point and element data to the values which are computed at the end of each load STEP. This is the case if Case Control commands are placed above the STEP level in the Case Control packet. Output may be modified from one STEP to another by changing output requests at the STEP level in the same way that you control output in linear analyses, one SUBCASE to another. You may obtain additional output at the load increment level by using the **NLSAVE** Case Control command. For example, the Case Control command:

```
NLSAVE EVERY 2 INCREMENT
```

will cause output to be generated for every other load increment. The **NLSAVE** command may be used above or below the STEP level. If input above the STEP level, it applies to all STEPs until you change the request.

16.6.3 Automatic Output

Printed output which is automatically generated by the nonlinear solution procedure is controlled using the **NLPRINT** Case Control command. As you saw earlier, the minimum output is obtained using the command:

```
NLPRINT = SUMMARY
```

whereas the normal and default condition for automatic output is obtained using the command:

```
NLPRINT = DETAIL
```

The **DETAIL**ed output is automatic and does not require the use of the **NLPRINT** command. The **NLPRINT** command may be used to change the output option at either the SUBCASE or STEP level.

There are three types of automatic output. The first type consists of a summary table which defines the current nonlinear solution control data. This table is self explanatory, and an example is presented in Table 16-6a. The second type of output consists of **USER INFORMATION MESSAGE**s and **USER WARNING MESSAGE**s. The information messages are printed only when **NLPRINT** is specified as **DETAIL**, and indicate when tentative divergence has occurred, when a stiffness update is to be performed, an so forth. The warning messages are printed for both levels of **NLPRINT** request. One such message would be issued for the case when the model has failed and the solution diverges, shown in Table 16-6b.

Table 16-6. SOLUTION SUMMARY AND MESSAGES

a. Solution Summary

```
                                    N O N L I N E A R   I T E R A T I O N   S U M M A R Y

NLITER PROCESSING BEGINNING FOR SUBCASE  1, STEP  1, LOAD INCREMENT  1.

                                THE CURRENT STATUS OF NONLINEAR CONTROL DATA FOLLOWS.

                                NLSOLVE SID                                 =           1
                                THE SOLUTION METHOD                         =         TAN
                                THE ITERATION METHOD                        =        AUTO
                                THE LOAD INCREMENT METHOD                   =        AUTO
                                LOAD CONVERGENCE CRITERION USED.
                                MAXIMUM ALLOWABLE DEFLECTION PERMITTED      =  1.00000E+05

                                MAXIMUM NUMBER OF STIFFNESS UPDATES PERMITTED=          20
                                MAXIMUM NUMBER OF UNBALANCED LOAD ITERATIONS =          10
                                INITIALLY REQUESTED NUMBER OF INCREMENTS     =           1
                                MAXIMUM NUMBER OF LOAD INCREMENTS PERMITTED  =          90
                                TENTATIVELY DIVERGENT CONVERGENCE RATE       =  9.99900E-01
                                TENTATIVELY DIVERGENT DISPLACEMENT
                                  CONVERGENCE RATE                          =  9.99900E-01
                                INCREMENT SIZE REDUCING LOAD ERROR          =  2.00000E+01
                                LOAD CONVERGENCE TOLERANCE                  =  1.00000E-02
```

b. Information Messages

```
*** USER WARNING MESSAGE.          THE MAXIMUM DEFLECTION HAS EXCEEDED THE ALLOWABLE OF  1.00000E+03.

*** USER WARNING MESSAGE.          THE MAXIMUM DEFLECTION HAS EXCEEDED THE ALLOWABLE OF  1.00000E+03.

*** USER WARNING MESSAGE.          THE SOLUTION DIVERGES FOR SUBCASE  1, STEP  3, INCREMENT  5.
                                   THIS SUBCASE IS TERMINATED.  THE LOAD FACTOR IS 0.70000.
```

The third type of automatic output is illustrated in Table 16-7. This output presents a table of the unbalanced force iteration status for each load increment. It contains information which is useful for interpreting the overall results and determining the acceptability of the solution. Each column of output in this table is described in Table 16-8.

Table 16-8. NONLINEAR ITERATION SUMMARY

SUBCASE 1 STEP 2 LOAD INCREMENT 1

ITER NO.	NO.ITER LEFT	ENERGY ERROR	LOAD ERROR	DISP ERROR	CONV RATE	LOAD NORM	DISP NORM	MAX DISP	AT GRID	C	MAX DELTA DISP RATIO	AT GRID	C	TOTAL ENERGY	NONL ENERGY	LOAD FACTOR
1	-	-	0.00E+00	-	-	3.08E+00	1.44E-02	-1.502E-01	2	2	1.000E+00	2	2	2.778E+00	0.00	0.2000
1	-	-	0.00E+00	-	-	3.17E+00	1.47E-02	-1.542E-01	2	2	1.000E+00	2	2	2.930E+00	0.00	0.4000
1	-	-	0.00E+00	-	-	3.29E+00	1.53E-02	-1.603E-01	2	2	1.000E+00	2	2	3.166E+00	0.00	0.7000
1	-	-	0.00E+00	-	-	3.42E+00	1.59E-02	-1.664E-01	2	2	1.000E+00	2	2	3.411E+00	0.00	1.0000

Table 16-7. INFORMATION IN ITERATION SUMMARY

ITEM	DESCRIPTION
ITER NO	is the identification for each unbalanced force iteration for the current load increment.
NO. ITER LEFT	is the estimated number of additional unbalanced force iterations which would be required to achieve a converged solution for the current load increment. This number is computed by **UAI/*NASTRAN*** and is used in performing CPU cost estimates to determine whether or not a stiffness update should be performed to accelerate convergence.
ENERGY ERROR **LOAD ERROR** **DISP ERROR**	are measures of the accuracy of the current unbalanced force iteration solution. These numbers are checked against the values specified with the NLSOLVE Bulk Data entry to determine whether or not to accept the current solution as a converged solution. These numbers are decimal fractions, so that a value of .01 means the error is 1 percent.
CONV RATE	is the current value of the parameter **LAMDA** which is described earlier in ***Tentatively Divergent Processing.***
LOAD NORM **DISP NORM**	are the norms of the current total load vector and displacement vector (*a-set* size). The norm of a vector is also defined in ***Tentatively Divergent Processing***.
MAX DISP AT GRID,C	define the location of the degree of freedom which currently has the largest displacement quantity. This output is restricted to the *a-set.*
MAX DELTA DISP RATIO AT GRID,C	represent the ratio of the maximum absolute value of change in displacement to the current maximum displacement, and the location of this degree of freedom. Changes in the degree of freedom in this output may indicate that significant changes in material behavior may be occurring various regions of the model.
LOAD FACTOR	is the current load increment factor relative to the current **STEP** load. This value will change in a predictable manner when fixed size load increments have been selected using the **NLSOLVE** Bulk Data entry, and they may change more or less rapidly when the automatic load increment procedure has been selected. Also, if tentative divergence processing has been initiated, this column will show how the load factor is reduced in an attempt to achieve convergence.
NONL ENERGY	is the ratio of the nonlinear energy to the linear energy as if the structure were linear for the current load increment. In computing this ratio, only the incremental loads for the current load increment are used.
FLEX PARAM	is an indication of how flexible the current system stiffness is compared to its initial value. This is an approximation which is more accurate when using the arc length method than when using the Newton-Raphson method. This output occupies the same location as **NONL ENERGY**, and it is printed only when the stiffness is updated.
TOTAL ENERGY	is the total strain energy in the model at this level of applied load and displacement.

16.7 MODELING GUIDELINES

To summarize, material properties for the BAR, BEAM, ROD, TRIA3, TRIAR, QUAD4, QUADR, HEXA, PENTA, and TETRA elements, as well as the structural properties of PILE elements, are entered using **MAT1** Bulk Data entries. The nonlinearities are then specified with **MAT1NL**, **TABLENL** or **MAT1NL1** Bulk Data entries. **UAI/*NASTRAN*** allows you to specify any form of constitutive law by using the **TABLENL** entry. However, the entries in this table ***must*** correspond to uniaxial tests or it's equivalent for two- and three-dimensional problems. You must first determine that the plasticity models are most the appropriate to model the type of nonlinearity of the problem. You then choose the type of yield criteria and hardening rule to completely describe the plasticity model. The application of the total load in steps has to be worked out based on the nature of nonlinearity. **UAI/*NASTRAN*** automatically assigns the load increments during the solution phase of each load step if desired. The criteria for convergence and the solution algorithms are chosen next. You must decide on the choice of ***solution algorithms*** and ***convergence criteria.*** The solution obtained must be verified based on engineering judgement as ***many nonlinear problems do not have exact solutions.*** Frequently, a companion linear analysis is advisable to judge whether the nonlinear solution obtained is reasonable.

One of the powerful features **UAI/*NASTRAN*** provides for nonlinear analyses is the linear and nonlinear GAP elements. There are many practical situations where GAP elements play an essential role in facilitating the analysis. Examples include stresses at contact points of bearings and nonlinear interfaces in structural media such as concrete or rocks.

The GAP element is a general three-dimensional element which simulates frictional effects in the two transverse directions and stiffness in the longitudinal, or axial, direction. The GAP may be initially closed or open. The constitutive relations for the element are essentially bilinear. One slope defines the stiffness of the open GAP and the other defines the stiffness of the closed GAP. A transverse shear stiffness, which is used only when the GAP is closed, may also be defined. The GAP element may be used to model friction effects.

There may be cases in which analyses using GAP elements have difficulty in converging. Many GAP analyses can be solved using the default parameters presented in Table 16-1, such as the example found in Section 16.8.3. The default solution uses the tangential stiffness method with automatic load incrementation. These increments are generally very small. This occurs because it is assumed that when a load increment has converged, another small load increment and tangential stiffness will still be convergent. If this is not the case, you may experiment with different parameters to achieve convergence. If GAP elements are the only nonlinear elements in your model, you may try a totally different approach using the secant stiffness method coupled with a very large step size and continuous stiffness updates. This would be selected with the **NLSOLVE** Bulk Data entry:

NLSOLVE	**1**	**SEC**						**25**	**+NL1**
+NL1	**25**		**1**						

Here the whole load step uses one load increment (**PINC=1**) so that all of the GAP elements assume their final position at the start of the iteration. The secant modulus (**SMETH=SEC**) makes the solution more stable, and continuous stiffness updates (**MINK=25**) are performed until convergence is attained or **MAXK=25** stiffness updates have been performed. This method will not work if any elements are in the plastic region. In this case, the **NLSOLVE** Entry:

NLSOLVE	**1**	**TAN or SEC**						**1 to 5**	**+NL1**

In this case, the load increment size reverts to the default value, and from one to five continuous stiffness updates are performed initially at each load increment, then the normal iteration

procedure begins. This is done to allow the initial stiffness updates to remove most of the nonlinear behavior of the GAP for the load increment.

It may also be possible to improve analyses that are having convergence problems by modifying the **CGAP** Bulk Data entries. Specifically, the fields **KCLS** and **KTRAN** should not be too large when compared with other stiffnesses in the model near their location. If convergence problems occur, you may improve convergence behavior by reducing these stiffness values.

16.8 EXAMPLE PROBLEMS

This section provides you with four examples of nonlinear modeling and analysis. These include one- and three-dimensional plasticity solutions, a model demonstrating surface effects, and a nonlinear elasticity solution.

16.8.1 One-Dimensional Plastic Analysis

UAI/*NASTRAN* provides four one-dimensional nonlinear elements: the BAR, the ROD, the ELASNL, and the PILE. These elements may often be components of structures such as cooling towers, bridges, pile foundations for multistory buildings, and so on.

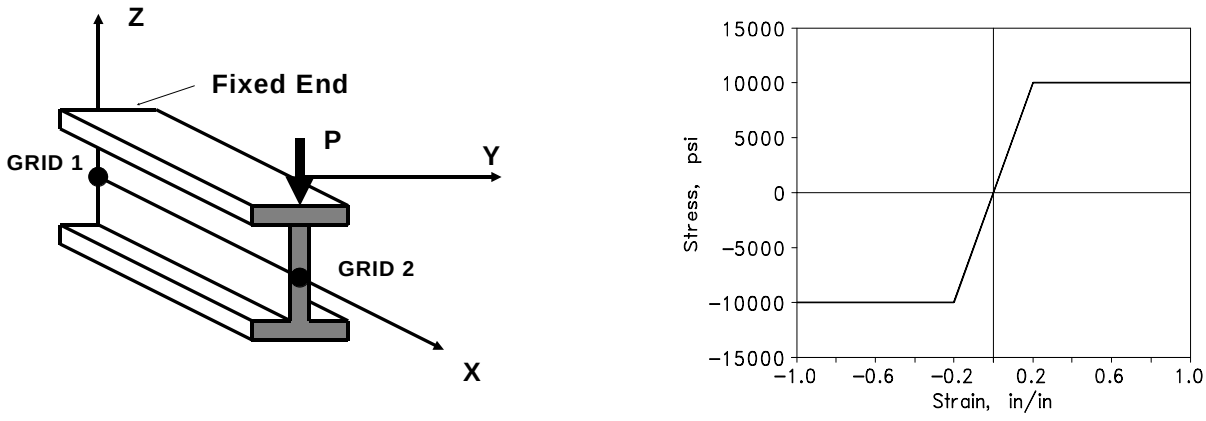
The first sample problem considers the plastic collapse of a cantilever beam. This problem demonstrates the material nonlinearity effects in a BAR element.

Example Problem 16-1
An I-Beam of length 10 in is fixed at one end and subjected to a tip load at its free end. You wish to determine the magnitude of the load which will collapse the beam.

The model consists of one BAR element with an **I** cross-section, fixed at GRID point 1, and subjected to a point load at GRID point 2, as shown in Figure 16-14a. The stress-strain curve, which is the same in tension and compression, is of the ***elastic-plastic*** type shown in Figure 16-14b. In order to perform nonlinear analysis, you must have an estimate of the load required to be applied to the structure. This estimate is often based on engineering judgement or experience with similar structures which have been analyzed before. In this case, a linear static analysis is performed first to estimate the possible solution to the problem. This is done by solving the problem for a unit load.

The results of this execution, which are shown in Table 16-9, give a maximum stress of 297.9 psi. By scaling the stress to its critical value of 10000 psi, you may estimate the load to fully stress the BAR element as 33.5 lb. The data for this baseline analysis is found in the file

Figure 16-14. I-BEAM COLLAPSE MODEL



a. I-BEAM MODEL

b. NONLINEAR STRESS-STRAIN RELATIONSHIP

16

Table 16-9. I-BEAM LINEAR SOLUTION RESULTS

```
    UNIT LOAD                                                                                                    SUBCASE 1

                              S T R E S S E S   I N   B A R   E L E M E N T S          ( C B A R )
                 STAT DIST/
ELEMENT-ID  GRID   LENGTH    SXC           SXD           SXE           SXF           S-MAX          S-MIN
         1
               1    .000   2.979146E+02 -2.979146E+02  2.383317E+02 -2.383317E+02  2.979146E+02   -2.979146E+02
               2   1.000  -4.917843E-06  4.917843E-06 -3.934275E-06  3.934275E-06  4.917843E-06   -4.917843E-06

    FULLY STRESSED DESIGN                                                                                        SUBCASE 2

                              S T R E S S E S   I N   B A R   E L E M E N T S          ( C B A R )
                 STAT DIST/
ELEMENT-ID  GRID   LENGTH    SXC           SXD           SXE           SXF           S-MAX          S-MIN        M
         1
               1    .000   1.000010E+04 -1.000010E+04  8.000080E+03 -8.000080E+03  1.000010E+04   -1.000010E+04
               2   1.000  -1.650772E-04  1.650772E-04 -1.320618E-04  1.320618E-04  1.650772E-04   -1.650772E-04
```

MNON1A. The second SUBCASE represents the load required to fully stress the element. Because the linear solution shows that the stress limit has been reached only in the flanges, you would expect the nonlinear collapse load to be larger than this value.

A nonlinear analysis, found in file **MNON1B**, is performed loading the structure with 60 lb applied over three load steps. The Case Control commands are:

```
    DISP = ALL
    OLOAD = ALL
    NLSTRESS = ALL
    NLFORCE = ALL
    NLPRINT = SUMMARY
    SPC = 1
    SUBCASE 1
          STEP 1
              LOAD = 1
          STEP 2
              LOAD = 2
          STEP 3
              LOAD = 3
```

The first **STEP** applies a load of 20 lb, the second 40 lb, and the third 60 lb. All loads are input in the normal manner. In this case, **FORCE** Bulk Data entries are used.

16

FORCE	**1**	**2**		**20**	**0.0**	**0.0**	**-1.0**		
FORCE	**2**	**2**		**40**	**0.0**	**0.0**	**-1.0**		
FORCE	**3**	**2**		**60**	**0.0**	**0.0**	**-1.0**		

When you define a nonlinear stress-strain relationship, you must define both **MAT1** Bulk Data and **MAT1NL** or **MAT1NL1** Bulk Data. The **MAT1** entry defines all of the elastic properties except the Young's Modulus, *E*. The actual value of *E* is determined by the stress-strain table that you select using your **MAT1NL** or **MAT1NL1** entries and define with **TABLENL** Bulk Data entries. For data checking purposes, the value of *E* on the **MAT1** entry must be equal to the slope of the first segment of the stress-strain relationship. The material input data for this sample problem is:

MAT1	**11**	**5.+4**		**.3**					
MAT1NL	**11**	**11**							
TABLENL	**11**								**+T1**
+T1	**SYM**		**0.2**	**10000.**	**1.**	**10000.01**	**ENDT**		

Table 16-10. COLLAPSE LOAD RESULTS

F O R C E S I N B A R E L E M E N T S (C B A R)

SUBCASE	GRID	STAT DIST/ LENGTH	- BENDING MOMENTS - PLANE 1	PLANE 2	- WEB SHEARS - PLANE 1	PLANE 2	AXIAL FORCE	TORQUE
1010100								
	1	0.000	-2.000000E+02	0.0	-2.000000E+01	0.0	0.0	0.0
	2	1.000	0.0	0.0	-2.000000E+01	0.0	0.0	0.0
1020100								
	1	0.000	-4.000000E+02	0.0	-4.000000E+01	0.0	0.0	0.0
	2	1.000	0.0	0.0	-4.000000E+01	0.0	0.0	0.0
1030300								
	1	0.000	-4.306252E+02	0.0	-4.306253E+01	0.0	0.0	0.0
	2	1.000	9.269048E-05	0.0	-4.306253E+01	0.0	0.0	0.0

S T R E S S E S I N B A R E L E M E N T S (C B A R)

SUBCASE	GRID	STAT DIST/ LENGTH	SXC	SXD	SXE	SXF	S-MAX	S-MIN
1010100								
	1	0.000	5.958292E+03	-5.958292E+03	4.766633E+03	-4.766633E+03	5.958292E+03	-5.958292E+03
	2	1.000	0.0	0.0	0.0	0.0	0.0	0.0
1020100								
	1	0.000	1.000000E+04	-1.000000E+04	9.533267E+03	-9.533267E+03	1.000000E+04	-1.000000E+04
	2	1.000	0.0	0.0	0.0	0.0	0.0	0.0
1030300								
	1	0.000	1.001487E+04	-1.001487E+04	1.001190E+04	-1.001190E+04	1.001487E+04	-1.001487E+04
	2	1.000	-2.853399E-03	2.853399E-03	-2.282719E-03	2.282719E-03	2.853399E-03	-2.853399E-03

The input data stream is then executed in **UAI/*NASTRAN***. The beam collapses by the formation of a plastic hinge at the fixed end. Since you are analyzing the failure mode of the beam, at the collapse point you expect that the stiffness matrix becomes singular and deformations become very large. This is seen in the solution results and leads to a divergence of nonlinear algorithm. At the point when failure occurs, the element force represents the actual collapse load. This output is shown in Table 16-10.

The final result compares very closely to the ***theoretical*** collapse load P_c, which is given by:

$$P_c = \frac{1}{l}\int_{Area} \sigma_y \, y \, dA = 42.25\text{lb}$$

16.8.2 3-D Plastic Analysis

The nonlinear analysis of three-dimensional structures is much more complex than those which are one-dimensional.

For three dimensional plasticity analysis, a yield criterion and a hardening rule ***must*** be supplied. The input data for different yield criteria and the hardening rules have been explained to you in earlier sections of this Chapter. **UAI/*NASTRAN*** supports the family of isoparametric solid elements for material nonlinear analyses. This example analyzes the inelastic stretching of a prismatic solid subjected to an axial load. For the sake of simplicity, the lateral strains in the solid are released, so that a state of uniform stress is obtained through out the solid. The example consists of a cantilever subjected to an axial stretching load applied in steps. Figure 16-15 shows the geometry and stress-strain relations used for this problem.

Example Problem 16-2

A cantilever beam of rectangular cross-section and length 10 in is subjected to an axial load, P, of 20000 lb, which is applied in two equal steps. You wish to determine the final axial deflection of the free end of the beam and its stress state assuming plastic properties

Figure 16-15a shows the finite element model of the cantilever beam, which is comprised of a single HEXA element, and its engineering data. The nonlinear stress-strain behavior is shown in Figure 16-15b. The input data for this problem is found in file **MNON2**.

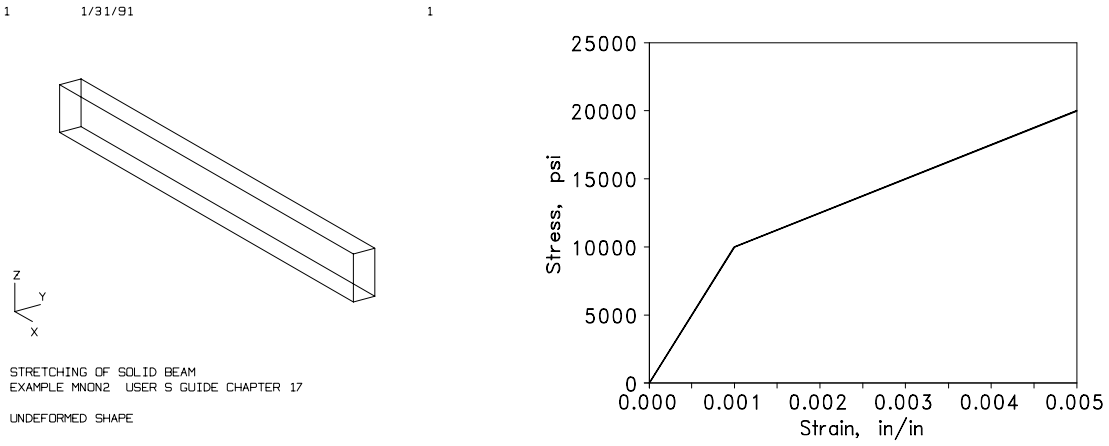
In a manner similar to that of Sample Problem 16-1, one loading condition is defined. Two **STEP**s are then used to apply the load to the structure. This time, because of the bilinear form of the stress-strain curve, a **MAT1NL1** Bulk Data entry is used. This entry specifies a **PLASTIC** material type, selects the **VONM** yield criterion, and indicates that the **ISOT**ropic hardening criterion will be used. As before, a **MAT1** Bulk Data entry is still required to define the linear value of Young's Modulus. The material property Bulk Data entries for this problem are:

MAT1	**11**	**1.+7**		**0.3**					
MAT1NL1	**11**	**PLASTIC**	**VONM**	**10000.**	**2.5+6**	**ISOT**			

In order to facilitate the exact solution, the lateral strains in the solid are released by allowing some translations at the support. **GRID 1** is fully restrained, where as **GRID 4** is allowed translation in the Y-direction. **GRID 5** is allowed to translate in the Z-direction, which implies that **GRID 8** is allowed translations in both Z and Y directions. This boundary condition results in a state of uniform stretching of the solid.

The exact solution to this problem, determined by inspection of the stress-strain relationship, yields a tip deflection of 0.005 in and a Normal-X Stress of 20000 psi. The **UAI/*NASTRAN*** results are identical within machine precision.

Figure 16-15. THREE-DIMENSIONAL PLASTIC MODEL



a. CANTILEVER BEAM MODEL

b. BILINEAR STRESS-STRAIN RELATIONSHIP

16.8.3 Modeling With Nonlinear Gap Elements

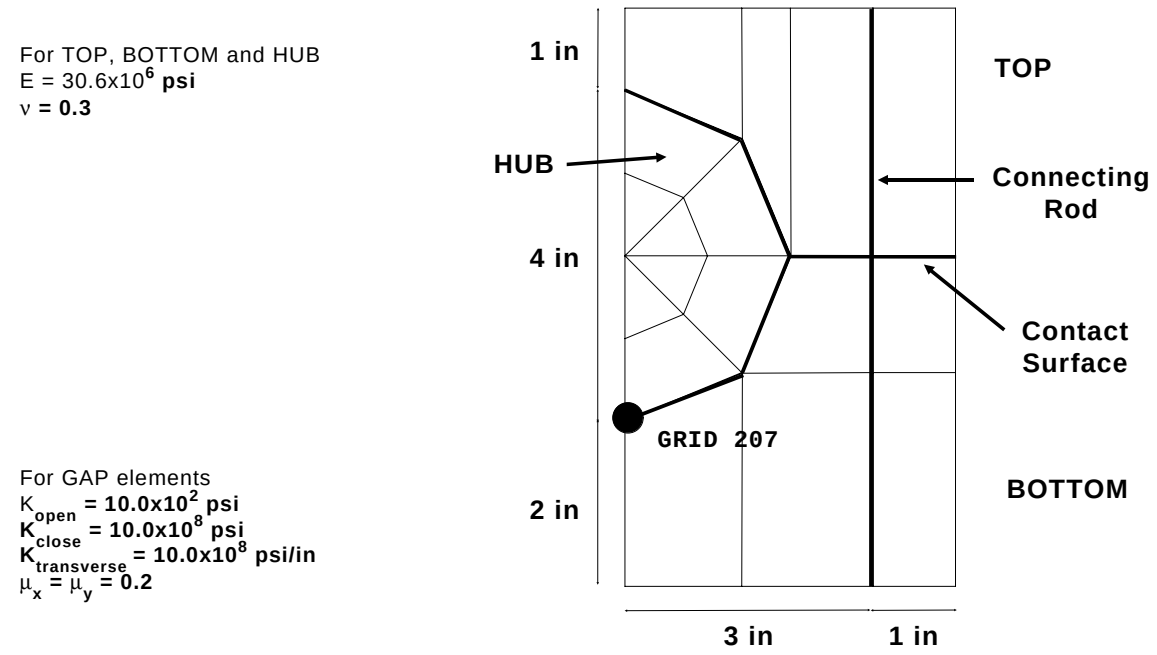
The example problem discussed in this section is the bearing block of an engine. This structure, shown in Figure 16-16, consists of three components: a top block, a bottom block, and a hub with a bearing mounted on it. The entire assembly is held together by a bolted connection as shown in the figure. The behavior of this assembly due to a temperature increase in the bolt is studied using nonlinear GAP elements of **UAI/*NASTRAN***. The increase in bolt temperature simulates a torque applied on the bolt. GAP elements are used to connect the bearing mounted on the hub to the top and bottom portions of the assembly as well as at the contact face of the top and bottom portions of the model.

The inelastic behavior in the model occurs due to the behavior at the gaps, and the rest of the domains are assumed to behave linearly elastic. Further, it is plausible to assume that the GAP elements at various points of contact of the bearing have negligible friction coefficients. The GAP elements at the interface between top and bottom portions of the model are assumed to have friction coefficients.

Example Problem 16-3

For the structure shown in Figure 16-16, assume that the temperature at the ends of the rod rises to 82.2° C from zero, due to some engine operations. The gap clearance at various contact points around the bearing is 0.01 in, and the gaps at the **TOP** and **BOTTOM** mating surface are assumed to have negligible clearance. You wish to determine the horizontal displacement at the bottom contact point of the bearing due to the imposed temperature rise in the bolt.

Figure 16-16. FINITE ELEMENT MODEL OF BEARING BLOCK



The input data for this problem is available in file **MNON3A**. There are many Multipoint Constraints in the data that are used to connect the components of the structure together. The model uses TRIA3 and QUAD4 elements for modeling the top, bottom and the hub portions of the bearing block. The connecting rod is modeled by ROD element. GAP elements are used at the contact surface between the top and bottom components, and at various points of contact between the hub, and top and bottom, which represent the bearing. Since the bearing represents smooth contact, it is assumed to be frictionless. The GAP elements are defined using **CGAP** and **PGAP** Bulk Data entries. Note that for the GAP element, there is no reference to **MATi** Bulk Data entries. Rather, you define the GAP stiffnesses directly on the **PGAP** Bulk Data entry.

Once again, it is often useful to perform a linear analysis of the structure with the GAP elements in their closed positions. The data file **MNON3B** contains the modified data to do this. The resulting deformation at GRID point 207 is 5.46×10^{-2} in. The input data file for the nonlinear analysis is then executed in **UAI/*NASTRAN*** using all of the solution algorithm defaults. The resulting deformation is 4.07×10^{-2} in which appears to be reasonable based on engineering judgement.

16.8.4 Nonlinear Elasticity

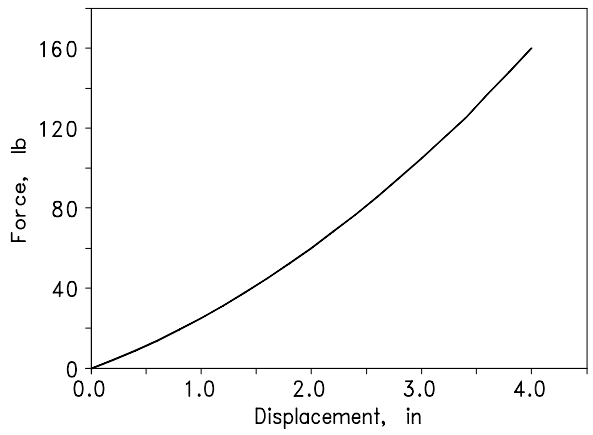
UAI/*NASTRAN* supports nonlinear elastic constitutive models as well as combined nonlinear elasticity and plasticity constitutive models for modeling material behavior. As explained earlier in this Chapter, **UAI/*NASTRAN*** adopts a model where assumptions are made to relate the multidimensional stress-strain to uniaxial stress-strain relations for nonlinear elasticity analysis. The necessary input data were also discussed. In this section, a simple example involving a one-dimensional nonlinear elastic model is presented. You should note that when using the ELASNL element, the nonlinear relationship defined using **TABLENL** Bulk Data entries is the force-displacement relation rather than the stress-strain curve that has been used in other examples.

Example Problem 16-4
A spring of length 10 in is fixed at one end and an axial load is applied to the free end. You wish to determine the axial deformation when the applied load is 160 lb.

The spring is modeled by a single ELASNL element. The force-displacement relation, which is shown in Figure 16-17, is defined using a **TABLENL** Bulk Data entry. All default values for the nonlinear solution algorithm are used. The remaining details of the solution can be found in the input data stream which is found in file **MNON4**.

The results from **UAI/*NASTRAN*** indicate an axial displacement of 4.0 in under the applied load. From the force- displacement data, you can easily see that the results match within machine-precision.

Figure 16-17. NONLINEAR ELASTICITY



Chapter 17

GEOMETRIC NONLINEAR ANALYSIS

UAI/*NASTRAN* provides capabilities for robust analyses of structural models under the influence of a variety of nonlinear effects. Two distinct forms of nonlinear behavior may be modeled, and they also may be combined in a single simulation.

The two effects are:

- Material nonlinear behavior including: nonlinear functions of strain versus stress for one, two and three dimensional elements; nonlinear load versus deflection behavior for spring elements; and gap element open-closed and friction nonlinear effects.
- Large deflection and large rotation nonlinear effects including structural stiffening due to deflection and external load change with structural deflection.

The method to deal the first effect is called Material Nonlinear analysis, and the second is called Geometric Nonlinear analysis. Both effects may occur in the same analysis in **UAI/*NASTRAN***. The simulations are also principally limited to static analysis. The exception to this limitation is the availability of transient response load generation functions which allow loads to be generated as a function of structural motion in the time domain.

Two theoretical approaches to the solution of the static geometric nonlinear equations of motion are available. Either a Newton-Raphson technique and an Arc Length technique may be employed for most applications. For less complex geometric nonlinear problems and for compatibility with older capability, a differential stiffness solution technique is also available. From the theoretical point view the first approach provides a high-order approximation to geometric nonlinear problems, while the second is only a linear approximation. In this chapter, the term ***geometric nonlinear analysis*** refers to the first approach specifically and the analysis of structures with geometric nonlinearity in general, and the second approach is explicitly called the ***differential stiffness approach***.

Rev: V20.1

The differential stiffness approach is also used to provide a linear Buckling solution and a Normal Modes solution for structures with geometric nonlinearity.

Geometric Nonlinear analysis is available using the primary nonlinear analysis Rigid Format which is invoked with the Executive Control commands:

```
APPROACH NONLINEAR
SOL STATICS
```

The differential stiffness approach is available in Rigid Format 4, the buckling capability is available in Rigid Format 5, and normal modes with differential stiffness is available in Rigid Format 13.

Buckling analysis is described in Chapter 18; the nonlinear material solutions are described in Chapter 16; and all geometric nonlinear solutions are described in this chapter.

This chapter describes how both Geometric Nonlinear analysis (Rigid Format 1, **SOL STATICS**, **APPROACH NONLINEAR**) and the Differential Stiffness approach (Rigid Formats 4 and 13) may be performed. This chapter also presents a number of examples describing how to use the various features of Geometric Nonlinear analysis. Rigid Format 5, buckling, is documented in Chapter 18 of this manual.

Only a single Case Control command is used in **SOL STATICS**, **APPROACH NONLINEAR**, to indicate whether nonlinear effects of geometry, materials, or both in combination, are to be considered. A description of this Case Control command is presented in both this chapter and the previous chapter, Chapter 16, Material Nonlinear Analysis.

17.1 TERMINOLOGY AND CAPABILITIES

This section presents a general introduction to the Geometric Nonlinear analysis capability in **UAI/*NASTRAN***, and defines some terms not usually encountered in linear analysis.

17.1.1 Nonlinear Geometry

The finite element analysis of a structure exhibits ***geometric nonlinear effects*** if the displacements cannot be accurately computed using the initial, undeformed configuration of the structure and linear stress-strain relations. Geometric nonlinear behavior is induced due to ***large displacements*** and/or ***large rotations*** of the structure.

In Material Nonlinear analysis, described in Chapter 16, the nonlinearity arises from the stress-strain relationship. The displacements and strains are infinitesimally small, and the engineering stress and strain definitions can be employed in the response description. For large displacement and small strain conditions, the structure is subjected to infinitesimally small strains, but undergoes large rigid body displacements and rotations. The stress-strain relations may be linear or nonlinear. The most ***general*** analysis case is the one in which the material is subjected to large displacements and large strains. In this case, the stress-strain relation also is usually nonlinear.

The basic requirement of a general nonlinear analysis is to find the equilibrium condition of a body subject to applied loads and boundary conditions, whether the strains are large or small. Equilibrium must be satisfied throughout the complete history of load application and consistent with the current configuration of the structure. To satisfy this requirement, it is generally necessary to employ an ***incremental loading*** formulation.

In **UAI/*NASTRAN***, the capability of analyzing structures with geometric nonlinear behavior is supported in four solution sequences: **`SOL STATICS, APPROACH NONLINEAR`**, for Geometric Nonlinear analysis; Rigid Format 4 for Static Analysis with Differential Stiffness; Rigid Format 5 for Buckling Analysis; and Rigid Format 13 for Normal Modes Analysis with Differential Stiffness. This Chapter describes the analysis and modeling procedures for Geometric Nonlinear analysis, and for static and normal modes with differential stiffness. Buckling Analysis is described in Chapter 18.

17.1.2 Displaced Element Coordinate System and Small Strains

In Geometric Nonlinear analysis, the ***displaced element coordinate system*** is defined for each element in the structural model. This system translates and rotates with each element as the structure deforms under external loads. The absolute displacements of the element, which are usually large in a Geometric Nonlinear analysis, are expressed in terms of the displaced coordinate system, and these are used for computing the element strains. Thus, the translations and rotations of an element may be arbitrarily large. However, the distortion measured with respect to the displaced element coordinate system is small such that linear strain theory can be applied. The strains computed under these assumptions are called ***small strains***.

The class of problems that can be treated by this large displacement, small strain theory includes the large deflection of a flat plate with stretching of the mid-surfaces, the snap-through of a shallow shell, and the post-buckling shape of a beam.

Displaced element coordinate systems are not used in the differential stiffness approach, Rigid Formats 4 and 13.

17.1.3 Material Points

Material points are locations in finite elements that move with the element as it undergoes geometric nonlinear behavior. The motions at material points are decomposed into two quantities, rotation and a stretch. These quantities are measured relative to the displaced element coordinate system such that absolute material point motions are the nonlinear sum of the motions of the origin and relative motions of the material points.

17.1.4 Green Strains and Stretch

In Geometric Nonlinear analysis, all strains are computed with respect to the displaced element coordinate system. By the ***Polar Decomposition Theorem*** [Chung88] the element deformation at a material point in an element can be decomposed into a rotation and a stretch of the material point. Stretch consists of the three direct stretches in the principal directions of the material point.

When the rotations of the material point with respect to the displaced element coordinate system are large and the stretches are small, the strains are computed with respect to the displaced element coordinate system. These strains are called the ***Green strains***.

When the rotations of the material point with respect to the displaced element coordinate system are small and the stretches are large, **UAI/*NASTRAN*** computes a new geometry for the element at each solution iteration to account for the large element distortion. The large stretches are measured against one load step, and the load step is divided into a number of smaller load increments. For each load increment, the stretch of a material point is assumed small so that either the linear strain theory or the incremental plasticity theory can be applied. Although the rotations of the material point with respect to the displaced coordinate may be small, the absolute rotations may still be arbitrarily large.

UAI/ *NASTRAN* does not provide a capability to analyze cases where both rotations and stretches of a material point are large, and this precludes modeling of certain classes of problems such as metal forming or most problems involving the deformation of rubber-like materials.

17.1.5 Cauchy Stresses

In linear analysis, the stresses are defined as force per unit area, where the area used to compute the stresses is that of the initial undeformed element. In Geometric Nonlinear analysis, this definition of stresses may introduce large errors because the current deformed area may be very different from the initial undeformed area in both orientation and magnitude. Thus, in Geometric Nonlinear analysis, the stresses computed from the current deformed area are called the ***Cauchy(true) stresses***.

17.1.6 The Second Piola-Kirchoff Stresses

As described above, Green strains have large rotations and small stretches with respect to the displaced element coordinate system. A coordinate system attached to the material point with Green strains may experience large rotations. However, the strains computed with respect to this material coordinate system must have both small rotations and small stretches, and the corresponding stresses computed are the Cauchy stresses. These Cauchy stresses can be rotated from the material coordinate system into the displaced element coordinated system, and these rotated stresses are called the ***Second Piola-Kirchoff stresses***. Therefore, the Second Piola-Kirchoff Stresses are the rotated Cauchy stresses and are always computed together with

Greens strains. Such a relationship between Green strains and the Second Piola-Kirchoff Stresses is called the ***conjugation of strains and stresses***.

17.1.7 Differential Stiffness

The term ***differential stiffness matrix***, also called ***geometric stiffness matrix***, ***initial stress stiffness matrix***, ***stability coefficient matrix***, or simply, ***stress stiffness matrix***, refers to a matrix that accounts for the change in potential energy associated with rotation of continuum elements under load. This matrix, designated as $\boldsymbol{K^d}$, depends on the element geometry, the displacement field, and state of stress. In **UAI/*NASTRAN***, with reference to the total Lagrangian description, the matrix, $\boldsymbol{K^d}$, represents stiffness effects that depend linearly on displacements.

The differential stiffness terms are used to completely describe the nonlinearities in the differential stiffness approach implemented in Rigid Formats 4 and 13, and are used along with other terms in Geometric Nonlinear analyses with **SOL STATICS, APPROACH NONLINEAR**.

17.1.8 Tangential Stiffness

The tangential stiffness of a structure is a function of displacement and is defined to be the rate of change of external and internal forces acting on the structure due to unit changes of the displacements:

$$\boldsymbol{K}(\boldsymbol{u}) = \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{u}_j}$$

where $\boldsymbol{f}_i$ are the forces acting on the GRID point i and $\boldsymbol{u}_j$ is the displacement at GRID point j. The tangential stiffness includes three components: the ***element stiffness***, ***differential stiffness***, and ***load stiffness***.

In the differential stiffness approach and buckling analysis, both the element stiffness and differential stiffness are computed at the initial undeformed configuration of the structure. In Geometric Nonlinear analysis, these stiffnesses are computed at the current deformed configuration.

The third component of tangential stiffness, load stiffness, is defined as the rate of change of follower forces due to unit change of displacement. **UAI/*NASTRAN*** computes the load stiffness for **PLOAD1** and **PLOAD4** applied loads.

17.1.9 Follower Forces

During analysis, the changing geometry of a structure often effects the direction and magnitude of the external load. For example, for a structure subjected to pressure, the (large) displacements of GRID points may change the direction of the pressure and the magnitude of the area on which the pressure acts. Any external loads that change their direction or magnitude as the structure is deformed are called ***follower forces***.

In linear analysis, the effects of follower forces due to displacement changes are very small and are always neglected. In Geometric Nonlinear analysis, **UAI/*NASTRAN*** computes follower forces for external loads which depend on the location of the GRID points, such as those defined by **FORCE1** and **PLOAD4** Bulk Data entries.

Follower forces are automatically processed in Geometric Nonlinear analysis, although you may manually override the automatic calculation, but they are not processed in differential stiffness approach in Rigid Formats 4 and 13.

17.1.10 Arc Length Solution Method

Geometric Nonlinear analysis provides two approaches to the solution of the nonlinear equations of motion. The primary method of solution is based on a Newton-Raphson iterative technique that includes the following features:

- Load Step definitions which allow you to define a loading and unloading history with Case Control commands
- Load Increments which are used to break the total load for a Load Step into a succession of partial loads that are applied in sequence, with solution convergence obtained at each increment
- Stiffness Updates which provide for the computation of the current tangential stiffness matrix at a particular load-displacement point in the solution
- Automatic resizing of the load increments and automatic determination of the frequency of tangential stiffness matrix updates based on solution convergence rates and the CPU cost of the respective operations
- Full user control over all of these solution options and over a variety of solution convergence criteria

Certain Geometric Nonlinear problems exhibit a situation where the external and internal loads may very rapidly and/or may have multiple values at a given structural deformation state. Such a situation is common for snap-through problems. For such cases, the standard Newton-Raphson technique may not be reliable.

For these and other cases, the Geometric Nonlinear analysis provides the ***Arc Length method***. The Arc Length method adds a load-level parameter as an additional unknown to the equations of motion, which are then solved with a Newton-Raphson strategy. This technique allows the solution to effectively track the load-deflection behavior during very large displacement changes with very small load changes.

You have complete control over the solution procedures used in Geometric Nonlinear analysis using the **NLSOLVE** Bulk Data entry. For Material Nonlinear problems, the default solution strategy is Newton-Raphson, and for Geometric Nonlinear or combined problems, the default solution strategy is Arc Length.

17.1.11 Element Library

Different finite elements have stiffness formulations consistent with the different types of nonlinear analysis solutions. These are:

- The ROD, BAR, BEAM, QUAD4, QUADR, TRIA3, TRIAR, 8-node HEXA, 6-node PENTA, and 4-node to 10-node TETRA elements have both material nonlinear and geometric nonlinear behavior. These elements are also supported in the differential stiffness solution approach, Rigid Formats 4, 5 and 13.
- The GAP, ELASNL, and BUSH only have material nonlinear behavior.
- The PILE element has both material nonlinear and geometric nonlinear behavior. However, the soil component only has material nonlinear behavior.
- The 9-to-20-node HEXA, 7-to-15-node PENTA, rigid elements, and all other elements in the **UAI/*NASTRAN*** element library only have linear formulations.

17.2 MATHEMATICAL BACKGROUND - INTRODUCTION

In linear finite element analysis, the element stiffness matrices are computed by:

$$\boldsymbol{K}^{\boldsymbol{e}} = \int_V \boldsymbol{B}^T \boldsymbol{D}\, \boldsymbol{B}\, dV \qquad (17\text{-}1)$$

where $\boldsymbol{B}$ is the strain-displacement matrix and $\boldsymbol{D}$ is the elastic constitutive matrix. For linear analysis, it is assumed that the displacements and strains are ***small*** and all integrations are performed over the original volume of each finite element. Therefore, each term of the matrix $\boldsymbol{B}$ is constant. Similarly, the assumption of linear elastic material properties means that the stress-strain matrix $\boldsymbol{D}$ is a constant.

Such assumptions are not correct for many engineering problems . For this class of problems, displacements and/or rotations may be large, such that $\sin\theta$ may not be approximated by θ, which is the normal assumption for linear finite element structural analysis. In these cases the displacements are classified as large, and additionally the strains developed during the solution may be greater than those supported by the linear region of material stress-strain property.

For accurate solution of this class of problem the ***deformed configuration*** of the structure must be used when computing the displacements and stresses. In other words, the matrix $\boldsymbol{B}$ is a function of displacement.

17.3 MATH BACKGROUND - GEOMETRIC NONLINEAR ANALYSIS

This section presents a review of the mathematical background of Geometric Nonlinear analysis as it is implemented in **UAI/*NASTRAN***.

17.3.1 Grid Point Motions

You will recall that a structure is discretized into a set of finite elements, which are inter-connected by a set of GRID points. For Geometric Nonlinear solutions, it is assumed that an infinitesimal rigid body is attached to each GRID point. The GRID point rigid body can translate and rotate, and these translations and rotations, called ***grid point displacements***, may be arbitrarily large. All element entities, such as the offset vector or the orientation vector, connected to the GRID points, translate and rotate with the GRID point rigid body. Scalar points do not have geometrical properties, thus all entities connected to scalar points do not have geometric nonlinear effects. This is also true for the material nonlinear elements ELASNL and GAP.

In **UAI/*NASTRAN***, two types of coordinate systems are related to GRID points: the basic coordinate system and the global coordinate system. The ***basic coordinate system*** is implicitly defined for the whole structural model. At each GRID point, a local coordinate system is defined and must be either directly or indirectly related to the basic coordinate system. Note that each local coordinate system is, by default, the basic coordinate system. The collection of all local coordinate systems, and the basic coordinate system, is called the ***global coordinate system***. In Geometric Nonlinear analysis, the global coordinate system is assumed fixed in space. This means that, for example, the direction of the global coordinates of a GRID point is defined by the position of the GRID point in a cylindrical coordinate system. Although the location of the GRID point may move from A to B the direction of the global coordinates of this GRID point is still defined by the original location A. All GRID point properties, such as displacements, forces, offset vectors, and orientation vectors, are expressed in terms of the fixed global coordinate system.

The most general motions of a rigid body can be represented by a translation of a base point and a rotation about the base point. Therefore, the motions of a GRID point rigid body can be decomposed into a translation of the GRID point and a rotation defined by:

$$\boldsymbol{\Theta} = \theta \cdot \boldsymbol{e}$$

where

$$\theta = |\boldsymbol{\Theta}| = \left(\boldsymbol{\Theta}^T \cdot \boldsymbol{\Theta}\right)^{1/2} \qquad (17\text{-}1)$$

and

$$\boldsymbol{e} = \frac{\boldsymbol{\Theta}}{\theta} = \begin{Bmatrix} e_x \\ e_y \\ e_z \end{Bmatrix}$$

Thus, θ is the magnitude of the rotation and $\boldsymbol{e}$ is the direction for the rotation. The displacement outputs are the three translational and three rotational components as defined above.

It is well-known that large rotations cannot be treated as vectors. As a result, 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 transformation, $\boldsymbol{R}(\theta)$. Any vector $\boldsymbol{v}$ can be rotated into a new position $\boldsymbol{v}_r$ given by the equations:

$$\boldsymbol{v}_r = \boldsymbol{R}(\Theta)\,\boldsymbol{v}$$

$$\boldsymbol{R}(\Theta) = \begin{bmatrix} \cos\theta + e_x^2(1-\cos\theta) & -e_z\sin\theta + e_x e_y(1-\cos\theta) & e_y\sin\theta + e_x e_z(1-\cos\theta) \\ e_z\sin\theta + e_x e_y(1-\cos\theta) & \cos\theta + e_y^2(1-\cos\theta) & -e_x\sin\theta + e_y e_z(1-\cos\theta) \\ -e_y\sin\theta + e_x e_z(1-\cos\theta) & e_x\sin\theta + e_y e_z(1-\cos\theta) & \cos\theta + e_z^2(1-\cos\theta) \end{bmatrix} \quad (17\text{-}2)$$

In Geometric Nonlinear analysis, the rotational change for any iteration is assumed to be small. The rotational matrix of any two consecutive iterations are related by the equations:

$$\boldsymbol{R}^{i+1} = \Delta\boldsymbol{R}\,\boldsymbol{R}^{i} \quad (17\text{-}3)$$

where

$$\Delta\boldsymbol{R} = \begin{bmatrix} 1 & -\Delta\theta_z & \Delta\theta_y \\ \Delta\theta_z & 1 & -\Delta\theta_x \\ -\Delta\theta_y & \Delta\theta_x & 1 \end{bmatrix} \quad (17\text{-}4)$$

where $\Delta\Theta = \left\{\Delta\theta_x, \Delta\theta_y, \Delta\theta_z\right\}$ is the rotational change computed from an unbalanced load iteration by assuming small rotation. This assumption is necessary because the displacement change is computed by using the tangential stiffness, which is valid only for an infinitesimal displacement change.

17.3.2 Displaced Element Coordinate System

Each element in a structural model is connected to a set of GRID points. From the location of these GRID points, an element coordinate system is defined, for example. For the QUAD4 elements, the z-axis of the coordinate system is normal to the mean plane of the element. In linear analysis, the element coordinate system is assumed to be fixed in space even though the connected GRID points move to a new position, i.e. the z-axis does not remain normal to the element mean plane after structural deformation. In Geometric Nonlinear analysis, a new type of coordinate system called the ***displaced element coordinate system*** is defined for each element. This coordinate system is computed using the same algorithm as that used for the element coordinate system, but the current GRID point locations are used instead of the initial locations. Thus, for QUAD4, the z-axis is normal to the element mean plane at every iteration throughout the analysis. In another words, the displaced element coordinate system translates and rotates with the element.

The connected GRID point displacements of the element in global coordinates are then transformed nonlinearly into those of the displaced element coordinates, which can be symbolically expressed by:

$$\boldsymbol{u}_g \Rightarrow \boldsymbol{u}_e$$

where $\boldsymbol{u}_g$ are the displacements in the global coordinate ssystem, and $\boldsymbol{u}_e$ are those in the displaced element coordinates. Since the global coordinates are fixed in space, $\boldsymbol{u}_g$ may be arbitrarily large in Geometric Nonlinear analysis. However, the definition of the displaced element coordinates has eliminated the rigid body translations and rotations in $\boldsymbol{u}_g$. Thus, $\boldsymbol{u}_e$ will be small for many problems, including the large deformation of a flat plate. In **UAI/*NASTRAN*** Geometric Nonlinear analyses, all element properties, such as strains and stresses, are computed by using the displacements in the displaced element coordinate system.

17.3.3 Strain Measurements

Options are provided for you to select from three types of strain measurements: small strain, Green strain, or large stretch. **UAI/*NASTRAN*** computes all strains in the displaced element coordinate system for each element. The terms *small*, *Green* or *large stretch* are measured against one load step. For example, small strain means that the strain at a material point in the element is small with respect to the displaced element coordinate system for one load step. The load step here is defined by the Case Control command **STEP**.

By the polar decomposition theorem, the element deformation at a material point can be decomposed into a rotation and a stretch with respect to the displaced element coordinate system. The rotation refers to the rotation of the material point, which does not include the rigid body rotation of the element, since the element rotation has been eliminated by using the displaced element coordinate system. As in small displacement theory, the stretch can be expressed by the three direct stretches in the principle directions at the material point, and the shear stretches are zero in these directions. For ***small strain***, both rotation and stretch are assumed to be small so that the linear strain theory can be applied. For ***Green strain***, the stretch is assumed small, but the rotation can be large, and for ***large stretch***, the stretch can be large, but the rotation is assumed small. In the following paragraphs, the discussion of strain measurements uses three-dimensional elements as an example. For one- and two-dimensional elements, the concept is similar, but the equations may be different.

The small strain is the engineering strain used in linear analysis:

$$\delta\varepsilon^l = \boldsymbol{B}^l \delta \overline{\boldsymbol{u}} \tag{17-4}$$

where $\delta\varepsilon^l$ is the small strain at the material point due to a variation of GRID point displacements $\delta \overline{\boldsymbol{u}}$ and $\boldsymbol{B}^l$ is the usual linear strain-displacement relationship used in linear analysis. One of the basic assumptions of small strain is that the change of geometric shape for an element will have a negligible effect on the strain. Thus, the strain is computed by using the initial element geometry, i.e. all the geometric nonlinear effects are treated by the displaced element coordinate system. Many problems, such as the large deformation of a flat plate, the snap-through of a shallow shell, and the post-buckling shape of a beam can be solved by this approach.

Green strain can be expressed by:

$$\delta\varepsilon^g = (\boldsymbol{B}^l + \boldsymbol{B}^n)\, \delta \overline{\boldsymbol{u}} \tag{17-5}$$

where $\boldsymbol{B}^n$ is the nonlinear strain-displacement relationship, which can be computed from the definition of the nonlinear strain, ε^n, given by:

$$\varepsilon^n = \begin{Bmatrix} \frac{1}{2}\left(\frac{\partial u^2}{\partial x}+\frac{\partial v^2}{\partial x}+\frac{\partial w^2}{\partial x}\right) \\ \frac{1}{2}\left(\frac{\partial u^2}{\partial y}+\frac{\partial v^2}{\partial y}+\frac{\partial w^2}{\partial y}\right) \\ \frac{1}{2}\left(\frac{\partial u^2}{\partial z}+\frac{\partial v^2}{\partial z}+\frac{\partial w^2}{\partial z}\right) \\ \frac{\partial u}{\partial x}\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\frac{\partial v}{\partial y}+\frac{\partial w}{\partial x}\frac{\partial w}{\partial y} \\ \frac{\partial u}{\partial y}\frac{\partial u}{\partial z}+\frac{\partial v}{\partial y}\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}\frac{\partial w}{\partial z} \\ \frac{\partial u}{\partial z}\frac{\partial u}{\partial x}+\frac{\partial v}{\partial z}\frac{\partial v}{\partial x}+\frac{\partial w}{\partial z}\frac{\partial w}{\partial x} \end{Bmatrix} \qquad (17\text{-}6)$$

One basic assumption of Green strain is that the stretch must be small, i.e. the strain computed by (17-5) is erroneous if the stretch is large. This limitation cannot be overcome by dividing one load step into a number of load increments so that the stretch for each load increment will be small. This is because the partial derivatives in (17-6), computed from the total displacement at the material point, will not become smaller by reducing load increment size.

On the other hand, small strain can be used to approximate Green strain if each element is divided into smaller sub-elements. When the element is divided, the stress producing rotation at a material point becomes the rigid body rotation of the sub-element such that, for each sub-element, the small strain may be applied. However, this approach may not be practical for some problems because of the extremely small element size required.

Unlike the small strain, the effect of changing geometric shape cannot be neglected in the case of large stretch. To account for this, **UAI/*NASTRAN*** divides a load step into a number of load increments and, for each increment, the solution convergence is achieved by iteration methods. At each iteration of a load increment the new locations of the connected element GRID points are computed, and these locations are used to construct a new geometry for that element. Based on the new geometry, the strains are computed using the engineering strain giving by (17-4). Thus, large stretch has the following characteristics:

- The strain is assumed small for each load increment so that linear strain theory or incremental plasticity theory can be applied.
- The geometry of the element is continually updated during the iteration to account for the effect of element distortion on computation of strain.
- Although the strain is small for one load increment, the accumulation effect of many load increments is such that the element distortion for one load step is large.
- The rotation at a material point with respect to the displaced element coordinate system must be small, however, the absolute rotation of the material point may still be large.

For large stretch, in order to guarantee that the strain is small for a load increment, its value is limited by the parameter **STNMAX**, which is input on the Bulk Data entry **NLSOLVE**. The default value is **0.03**, thus if there are ten load increments in a load step, the maximum allowable strain for the load step is **0.3**. The maximum strain for small strain should be near **0.03**

for a load step. An additional parameter available to you to limit the maximum element rotation with respect to the displacement element coordinate system per load increment is **ROTMAX**. The default value is **7.5** degrees for small strain and large stretch, and **15.0** degrees for Green strain.

17.3.4 Conjugate Stresses

For each of the three strains defined in the previous section, there is a corresponding conjugate stress. To define these these conjugate stresses, consider the virtual work done for an element in a structural model:

$$\delta w = \int_v \delta(\varepsilon^t)^T \sigma^c \, dv \tag{17-7}$$

where $\delta\varepsilon^t$ is the ***true strain*** and σ^c is the ***Cauchy (true) stress***, and the integration is performed on the current deformed volume of the element.

Then , for each type of strain, there is a conjugate stress such that the virtual work done by the strain and stress is equal to the virtual work done by the true strain and Cauchy stress as defined by (17-7).

For small strain, it has been assumed that both rotation and stretch are small with respect to the displaced element coordinate, therefore:

$$\delta\varepsilon^l \approx \delta\varepsilon^t \tag{17-8}$$

It must be emphasized that (17-8) is true only when $\delta\varepsilon^l$ is computed in the displaced element coordinate system. However, the integration of the virtual work for small strain is always done on the initial undeformed element volume v_o, and substituting (17-8) into (17-7), results in:

$$\delta w \approx \int_{v_o} \delta(\varepsilon^t)^T \sigma^c J \, dv_o = \int_{v_o} \delta(\varepsilon^t)^T \sigma^k \, dv_o \tag{17-9}$$

where $\sigma^k = J\sigma^c$ is called the ***Kirchhoff stress***, and J is the Jacobian. It has been assumed that the deformation of the element for one step is small, thus:

$$J \approx 1.0 \text{ and } \sigma^k \approx \sigma^c$$

This means that, for small strain, the strain and stress computed in the displaced coordinate system are approximately equal to the true strain and Cauchy stress.

Next consider the conjugate stress for Green strain. It can be shown that, for small stretch of a material point in the element, the Green strain and the true strain are related by:

$$\delta\varepsilon^t \approx \boldsymbol{U}_{me}\, \delta\varepsilon^g \tag{17-10}$$

where $\boldsymbol{U}_{me}$ is a rotational transformation from the displaced element coordinate system to the material coordinate system. Here, the material coordinate system is defined to be a coordinate system which rotates with the material points. This means that the Green strain is the true

strain in the material coordinate system rotated into the displaced element coordinate system. Substituting (17-10) into (17-7), results in:

$$\delta w \approx \int_{v_o} \delta(\varepsilon^g)^T \boldsymbol{U}_{me}^T \sigma^c J \, dv_o = \int_{v_o} \delta(\varepsilon^g)^T \sigma^s \, dv_o \qquad (17\text{-}11)$$

where $\sigma^s = J \boldsymbol{U}_{me}^T \sigma^c$ is the second Piola-Kirchhoff stress. Since it has been assumed that the stretch is small for the load step:

$$J \approx 1.0 \text{ and } \sigma^s \approx \boldsymbol{U}_{me}^T \sigma^c$$

This equation indicates that the strain and stress computed for the Green strain is approximately equal to the true strain and Cauchy stress in the material coordinate system of the material point.

Finally, consider the conjugate stress for large stretch. In computing strain for large stretch, the current geometry at each iteration has been used, and the integration of (17-7) is performed on the current volume. Therefore, the strain and stress computed for large stretch are the true strain and Cauchy stress, respectively.

17.3.5 Follower Forces

The changing geometry of the structure may effect the direction and magnitude for the external load. For example, for a structure subject to pressure, the displacements of the GRID points may change the direction of the pressure and the magnitude of the area to which the pressure is applied. The external load that changes its direction or magnitude with changing displacement is called a ***follower force***.

Options are provided on the Case Control command **NLTYPE** to specify whether follower forces are to be used. If they are selected, **UAI/*NASTRAN*** computes new follower forces at each iteration for all external loads which have follower forces. The computation of follower forces also depends on the strain measurement selected. For example, in computing the follower forces for **PLOAD4**, only the direction of the pressure load is changed if small strain or Green strain is selected, while both direction and magnitude are changed if large stretch is selected. Table 17-1 presents a summary of the availability of follower forces for all external loads.

17.3.6 The Arc Length Method

It is often required to obtain the solution to an unstable problem as shown in Figure 17-1. The Figure shows a plot of load factor,λ, versus displacement, ***u***, which is called the ***solution curve***. Points 1, 2, 4 and 5 are called the ***critical (limit) points***. These are points on the solution curve where the stiffness is either zero or infinity. From point 1 to point 3 the displacement snaps through at constant load level and from point 5 to point 6 the load snaps through at constant displacement. For this type of problem, the Newton-Raphson solution method will fail. Many techniques have been devised to solve this type of problem, the most successful of which is the ***Arc Length method***. Not only the Arc Length method solve problems with critical points, it is also an efficient method for solving problems without critical points.

The implementation of the Arc Length method in **UAI/*NASTRAN*** is similar to the Modified Rick's method [Rick79, Crisfield83]. In this implementation, the load factor is added to the displacements to create a new set of unknowns for the solution. This allows the load level to

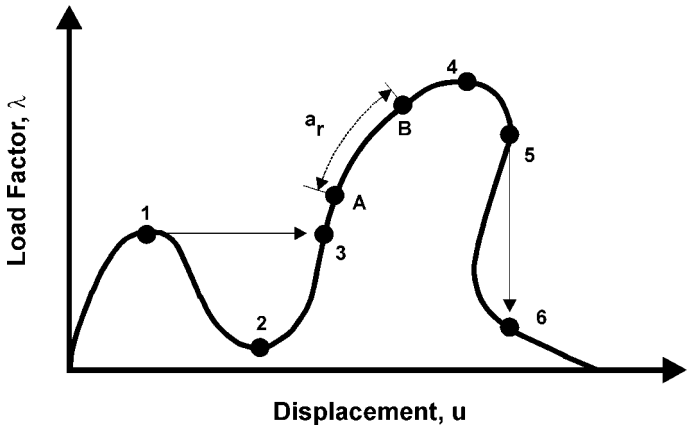
Table 17-1. AVAILABLE FOLLOWER FORCES

BULK DATA ENTRIES	FOLLOWER FORCE IMPLEMENTATION
FORCE, MOMENT	Have no follower forces because both their direction and magnitude are fixed.
FORCE1, FORCE2 MOMENT1, MOMENT2	Have follower forces because their direction may change.
GRAV	Has no follower force. The direction of gravity load is fixed in the space and its magnitude depends on the system mass matrix, which produces neglegible change due to displacements.
PLOAD1 and PLOAD4	Have follower forces because both direction and magnitude may change.
PLOAD and PLOAD2	Have follower forces, but not implemented. Use **PLOAD4** instead.

change during the course of iteration. This contrasts with the usual automatic load incrementation method, in which, although the load level is automatically incremented, the iterations are still performed at a constant load level. A fundamental weakness of the Newton-Raphson method is that the iteration at a constant level is generally unreliable for solving problems with critical points.

The load factor λ and the n-displacement unknowns form a *n+1* dimensional solution space. The equilibrium solution points constitute a curve in the solution space. This is called the solution curve, as shown Figure 17-1. The solution proceeds from any known point A on the curve to an unknown point B by constrained Newton-Raphson iteration. The constrained iteration is such that, upon convergence, the solution will have traveled an arc length, a_r, along the solution curve.

Figure 17-1. SOLUTION FOR UNSTABLE PROBLEM



This distance along the solution curve is called the ***arc length size***. The corresponding change in the load is called the ***load increment***. In the Newton-Raphson method, the load increment is defined to be the load factor λ increased by a constant amount $\Delta\lambda_o$, called the ***load increment size***. On the other hand, in the Arc Length method, $\Delta\lambda_o$ is **only** the initial load increment size, which is related to the arc length size by:

$$\begin{aligned} \boldsymbol{u}_s &= \boldsymbol{K}^{-1}\boldsymbol{p}_s \\ a_r &= \Delta\lambda_o\left(\boldsymbol{u}_s^{\boldsymbol{T}}\boldsymbol{u}_s + \gamma^2\right)^{1/2} \end{aligned} \tag{17-12}$$

where $\boldsymbol{K}$ is the current tangential stiffness matrix, $\boldsymbol{p}_s$ is the external load for the load step and γ is a load scale factor that will be defined later. From this equation, it is noted that if the arc length size is held constant from one load increment to another, then the initial load increment size computed will be dependent on the current tangential stiffness matrix — the softer the stiffness, the smaller the initial load increment size. Conversely, if the load increment size is held constant as in the Newton-Raphson method, the arc length size becomes larger with a softer tangential stiffness matrix. At the load critical point, point 1 in Figure 17-1, the Arc Length size becomes infinite, and the Newton-Raphson method fails.

As in the Newton-Raphson method, you may specify the arc length incrementation method on the **NLSOLVE** Bulk Data entry, and the Arc Length size may be selected to change **AUTO**matically from one load increment to another, **SEMI**-automatically, or it may be kept constant, **FIX**.

At each load increment before convergence, the solution is performed using the following recurrence relations:

$$\begin{aligned} \delta\boldsymbol{u} &= \boldsymbol{K}^{-1}\boldsymbol{p}_{er} \\ \boldsymbol{V}_{i-1} &= \begin{Bmatrix} \Delta\lambda_{i-1}\,\gamma \\ \Delta\boldsymbol{u}_{i-1} \end{Bmatrix} \\ \Delta\boldsymbol{V} &= \begin{Bmatrix} \Delta\lambda\,\gamma \\ \delta\boldsymbol{u} + \delta\lambda\,\boldsymbol{u}_s \end{Bmatrix} \\ \boldsymbol{V}_i &= \begin{Bmatrix} \Delta\lambda_i\,\gamma \\ \Delta\boldsymbol{u}_i \end{Bmatrix} \\ \boldsymbol{V}_i &= \boldsymbol{V}_{i-1} + \Delta\boldsymbol{V} \end{aligned} \tag{17-13}$$

17

where $\boldsymbol{p}_{er}$ is the error vector. In this set of equations, $\delta\lambda$ is an additional unknown that requires an additional equation. This equation is supplied by imposing the condition that the solution $\Delta\boldsymbol{V}$ will be searched in a direction normal to the known vector $\boldsymbol{V}_{i-1}$:

$$\boldsymbol{V}_{i-1}^{T}\cdot\Delta\boldsymbol{V} = 0 \tag{17-14}$$

Substituting (17-13) into (17-14) yields:

$$\delta\lambda = -\frac{\Delta \boldsymbol{u}_{i-1}^T \, \delta \boldsymbol{u}}{\delta\lambda_{i-1}\,\gamma^2 + \Delta \boldsymbol{u}_{i-1}^T \, \boldsymbol{u}_s} \tag{17-15}$$

Now the load scale factor γ is defined. When $\gamma = (\boldsymbol{u}_s^T \boldsymbol{u}_s)^{1/2}$, this is the normal Arc Length method called the **ARCFIX** method. When $\gamma = 0$, the solution incrementation is controlled by a set of generalized displacements, and it is thus called the **DISP**lacement control method. When $\lambda = \infty$, then equation (17-13) reduces to the Newton-Raphson method. This is called the **LOAD** control method. **ARCFIX**, **DISP** and **LOAD** are all variations of the Arc Length method. **UAI/*NASTRAN*** includes another method called the automatic **ARC** length method. The **ARC** method uses **ARCFIX** as its main solution method, and it switches to **DISP** or **LOAD** if one of these methods will give a better solution, or if there are convergence difficulties.

17.4 MATH BACKGROUND - DIFFERENTIAL STIFFNESS APPROACH

This section presents a review of the matehematical background of the Differential Stiffness approach. This approach provides a simple approximation for certain geometric nonlinear effects such as the caternary problem.

17.4.1 Overview of Approach

In **UAI/*NASTRAN*** the formulation based on ***differential stiffness*** theory assumes that geometrically nonlinear problems can be approximated by adding a differential stiffness matrix to the structural stiffness matrix, and iterating on the resulting system of equations. The original configuration of the structure is retained. The engineering stress and strain measures are used in the computations, and the change in geometry of the structure is ***only*** reflected on the stiffness of the structure.

The term differential stiffness applies to linear terms in the equations of motion of an elastic body that arise from a simultaneous consideration of large nonlinear motions and the applied loads. The theory of differential stiffness is not an exact theory and it involves inherent assumptions that are arbitrary and that may be changed depending on their practical effect. The approach in **UAI/*NASTRAN*** to the theory of differential stiffness is based on ***Lagrange's*** equations of motion for a system with a finite number of degrees of freedom.

A fundamental assumption in the differential stiffness implementation in **UAI/*NASTRAN*** is that external loads do not change in magnitude or direction as the structure deflects. However, the loads do ***travel*** with the GRID points. This must be kept in mind when interpreting results. For problems with ***strong*** geometric nonlinear effects, that is, problems where the magnitude and direction of the load change, you can use the Geometric Nonlinear analysis, as described in section 17.3.

There are some important properties of the differential stiffness matrix, $\boldsymbol{K^d}$, that you should note. $\boldsymbol{K^d}$ is ***symmetric***, it is independent of elastic properties, and depends on the element geometry, displacement field, and the state of stress. The overall structural stiffness with respect to the buckling modes becomes zero at the critical loads. In general, $\boldsymbol{K^d}$ is indefinite, and it cannot be inverted.

17.4.2 Static Analysis Approach

The static analysis of structures, whose stiffness properties are a function of both the static loads and the deformed shape, are solved using Rigid Format 4. Such problems include, for example, a hanging cable, a pendulum supported by a clock spring, or a thin cylindrical shell under point loads that do not change direction as the structure deforms.

The **UAI/*NASTRAN*** algorithm for static solution with differential stiffness involves iterations for computing the differential stiffness matrix. In its simplest form, the iteration algorithm is:

$$\left[\boldsymbol{K} + \boldsymbol{K^d}(\boldsymbol{u}_i) \right] \boldsymbol{u}_{i+1} = \boldsymbol{P} \qquad (17\text{-}2)$$

where the subscripts i and $i+1$ indicate consecutive iterations. Mathematical analysis shows that the above iteration is equivalent to retaining some of the second order terms in the calculation of differential stiffness.

Criteria are needed to determine when the solution is sufficiently converged, and when the displacement vector should be updated to the current estimated solution. These criteria are

computed automatically based on parameters that you may supply. These are described in Section 17.3.3. Convergence occurs when the difference between successive right hand sides of 17-2 is sufficiently small. A weighted criterion for this event is,

$$\varepsilon_i = \frac{\left| \boldsymbol{u}_{i+1} \bullet (\boldsymbol{P}_{i+1} - \boldsymbol{P}_i) \right|}{\left| \boldsymbol{u}_{i+1} \bullet \boldsymbol{P}_i \right|} < \varepsilon_o \tag{17-3}$$

where, ε_o, is a parameter that you may supply, and, P_i, is the right hand side of (17-2) at the i^{th} iteration.

If an exponential decay in the error is assumed, and if, $\lambda = \varepsilon_{i-1}/\varepsilon_i$ is the ratio of the error on two successive iterations, then the number of additional iterations required to achieve the desired accuracy is approximately,

$$N_f = \frac{\log \left(\dfrac{\varepsilon_i}{\varepsilon_o} \right)}{\log\ (\lambda)} \tag{17-4}$$

The number of additional iterations, N_f, may be used in conjunction with the known computer times for matrix decomposition and for one cycle of iteration, to decide whether it is more efficient to continue iterating, or to ***shift***, i.e., replace the initial elastic solution with the current approximate solution.

17.4.3 Normal Modes Approach

Normal modes analysis is used to compute the natural frequencies of an undamped structure. The inclusion of geometric nonlinearity causes the governing differential equation to be nonlinear. The response of such systems is solved by a step-by-step numerical integration procedure known as the ***average acceleration*** method.

The equations of motion for natural frequency analysis including differential stiffness can be written as:

$$\left(\left[\boldsymbol{K} + \boldsymbol{K}^{\boldsymbol{d}} \right] - \omega^2 \boldsymbol{M} \right) \boldsymbol{u} = \boldsymbol{0} \tag{17-5}$$

where $\boldsymbol{K}$ is the stiffness matrix, $\boldsymbol{K}^d$ is the differential stiffness matrix, $\boldsymbol{M}$, is the mass matrix, and, $\boldsymbol{u}$, represents virtual, ***kinematically*** admissible displacements describing the natural modes of the structure. The frequency of these modes is given by ω.

The difficulty of computing the natural frequencies of a finite element model with the inclusion of differential stiffness arises from the dependence of the differential stiffness matrix on the displacements. It is therefore necessary to ***load*** the structure, in order to compute differential stiffness of the structure. A static differential stiffness analysis must therefore be performed before proceeding to the dynamic analysis. Due to this ***preloading***, the natural frequencies of the structure are altered, and one can study the ***softening*** or ***stiffening*** effects of loads on the natural frequencies. Tensile membrane forces increase the natural frequencies.

Compressive membrane forces decrease them, and produce a root $\omega = 0$ if $\lambda = \lambda_{cr}$, where λ_{cr} refers to the load factor corresponding to the critical buckling load.

The **UAI/*NASTRAN*** solution algorithm for dynamic analysis with differential stiffness involves a two-stage process. The first stage is the iterative solution of the static case (described in the previous section). Using the ***converged*** static solution results, the dynamic solution is carried out in the second stage. As in the static differential stiffness solution, engineering stress and strain measures are used. The solution results for dynamic analysis are therefore dependent on the static preload.

17.5 INPUT DATA FOR GEOMETRIC NONLINEAR ANALYSIS

This section provides a description of the input data required to perform Geometric Nonlinear analysis with **UAI/*NASTRAN***.

17.5.1 Executive Control Commands

The Executive Control Commands for Geometric Nonlinear analysis are the same as those of the material nonlinear analysis, which are described in section 16.5.1.

17.5.2 Case Control Commands

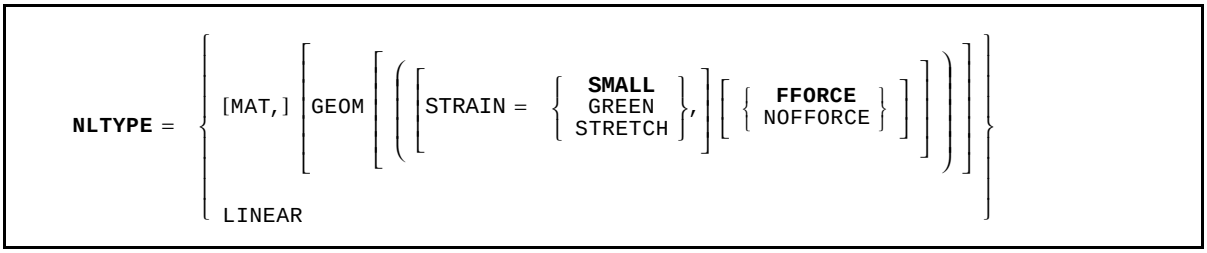
The basic Case Control structure for linear analysis is also applicable to nonlinear analysis. The seven nonlinear specific Case Control commands: **AUTOREDUCE**, **NLPRINT**, **NLRESTART**, **NLSAVE**, **NLSOLVE**, **NLTYPE**, and **STEP**, which are described for material nonlinear analysis in section 16.5.2, can also be used in Geometric Nonlinear analysis. Only usages of these seven commands that are different from the material nonlinear analysis are discussed in the following sections.

17.5.2.1 The AUTOREDUCE Command

The **AUTOREDUCE** command is used to partition the structural model into a set of linear elements and a set of nonlinear elements. While this feature, described in section 16.5.2.1, greatly improves computational efficiency when performing material nonlinear analyses, it cannot be used in Geometric Nonlinear analysis, because the entire structure behaves nonlinearly. Thus, the **AUTOREDUCE** command may not be used.

17.5.2.2 The NLTYPE Command

The **NLTYPE** command is used to select the analysis type. Its format is:



The analysis types that you may specify include: a nonlinear material analysis, **MAT**; a Geometric Nonlinear analysis, **GEOM**; a combined nonlinear analysis; and a **LINEAR** analysis. If the **NLTYPE** command is not used, then a material nonlinear analysis will be performed.

When Geometric Nonlinear analysis is selected, you may select the manner in which the **STRAIN** will be computed for each geometrically nonlinear element in the structural model. Selections are **SMALL** strain, **GREEN** strain, or large **STRETCH**. You may also specify the use of follower forces, **FFORCE**. If you select **FFORCE**, **UAI/*NASTRAN*** will compute follower forces and load stiffness as appropriate. If you define multiple SUBCASES in the Case Control packet, the **NLTYPE** command may be different from one SUBCASE to another. Examples of how to use **NLTYPE** are given in the User's Reference Manual.

UAI/ ***NASTRAN*** allows the **NLTYPE** to be placed at STEP level, i.e., the analysis type may vary from one step to another. For example:

```
SUBCASE  1
   STEP  1
      NLTYPE = GEOM
   STEP  2
      NLTYPE = GEOM,MAT
```

In this example, the first step performs a Geometric Nonlinear analysis, and the second step a combined analysis. This is perfectly legal if you know that the stresses in **STEP 1** for all elements are within the linear limits, and it may be advantageous because Geometric Nonlinear analysis is much less expensive than combined analysis. However, in many situations specifying the **NLTYPE** at the STEP level has no physical meaning even though the solution converges. Consider, for example, that a Geometric Nonlinear analysis is performed in **STEP 1**, and a material nonlinear analysis in **STEP 2**. When convergence is reached for **STEP 1**, the elements may have moved to new positions which are far from their initial positions. If this is then followed by a material nonlinear analysis which uses small displacement theory, i.e., the analysis is performed based on the initial position of the elements, then clearly the wrong solution will be obtained. Therefore, you must ensure that if the **NLTYPE** command is used at the STEP level in your problem, it has appropriate physical meaning.

17.5.2.3 Restart with the NLSAVE and NLRESTART Commands

The basic concepts of RESTART for nonlinear analyses are described in Section 16.5.7. When performing geometric, or combined, nonlinear analyses, the **NLSAVE** command is slightly different. The format of this command is:

```
NLSAVE [([MAT,][GEOM])] [ EVERY  n  INCREMENT ]
```

The specific data saved on the NLDB are different for the different analysis types specified by the **NLTYPE** command. The table below summarizes the types of restarts that you may perform based on the **NLSAVE** command used and the **NLTYPE** of the analysis.

If your NLSAVE Command is:	And if your previous run was:	Then you can restart:
None	**NLTYPE(MAT,GEOM)**	**NLTYPE(MAT,GEOM)** **NLTYPE(MAT)** **NLTYPE(GEOM)**
	NLTYPE(MAT)	**NLTYPE(MAT)**
	NLTYPE(GEOM)	**NLTYPE(GEOM)**
NLSAVE(MAT,GEOM)	**NLTYPE(MAT)**	**NLTYPE(MAT,GEOM)** **NLTYPE(MAT)** **NLTYPE(GEOM)**
	NLTYPE(GEOM)	**NLTYPE(MAT,GEOM)** **NLTYPE(MAT)** **NLTYPE(GEOM)**

For example, consider the following Case Control packet used in the initial run:

```
NLTYPE = GEOM
NLSAVE(GEOM,MAT)
SUBCASE  1
   STEP  1
   STEP  2
```

These commands specify that a Geometric Nonlinear analysis be performed in both STEPs, and data for a combined analysis are to be saved on NLDB so that the following restart can be performed:

```
NLRESTART SUBCASE 1, STEP 2
SUBCASE  1
   STEP  1
      NLTYPE = GEOM
   STEP  2
      NLTYPE = GEOM,MAT
```

This restart will perform a combined analysis in **STEP 2**. In this example, if **NLSAVE(GEOM,MAT)** was not present in the Case Control packet for the initial run, then only Geometric Nonlinear analysis is allowed in the restart. As was the case for **NLTYPE** commands placed at the STEP level, you must also make sure that performing a restart with a different analysis type has an appropriate physical meaning for your particular problem.

17.5.3 Bulk Data Entries

The bulk data entries used in Geometric Nonlinear analysis is the same as those of Material Nonlinear analysis, a summary of which is given by Table 18-4.

17.5.3.1 Solution Control with the NLSOLVE Bulk Data Entry

The format of the **NLSOLVE** Bulk Data entry, and a description of its fields, in shown in Table 17-2. This entry is used for both Material and Geometric Nonlinear analyses. The fields used by Geometric Nonlinear analyses, and their default values, are indicated in the table. Fields used only when performing Material Nonlinear analysis are indicated by gray shading. In the following sections, only those fields used by Geometric Nonlinear analysis are discussed. The other fields are described in Sections 16.2.1 and 16.2.2, because they have the same usage as those of Material Nonlinear analysis.

The Solution Method. The **SMETHD** is used in material nonlinear analysis to select the solution method **NR** or **SEC** for the Newton-Raphson method. In Geometric Nonlinear analysis only **NR** is applicable since there is no secant modulus in this type of analysis. As a result, **NR** will be used if you select **SEC**. However, in combined Geometric and Material Nonlinear analysis, both **NR** and **SEC** are valid solution methods. In addition, **UAI/*NASTRAN*** includes the Arc Length method, for which you can select the automatic **ARC** length method, the fixed Arc Length method (**ARCFIX**), the **DISP**lacement control method, or the **LOAD** control method. **ARCFIX**, **DISP**, and **LOAD** are all variations of the Arc Length method. The **ARC** method uses **ARCFIX** as its main solution method, and will switch to **DISP** or **LOAD** if the other method gives a better solution or there are convergence difficulties.

Load Incrementation Method. The **PMETHD** is used to control the load incrementation method. For the Newton-Raphson method, this may be selected from **AUTO**, **SEMI** or **FIX**. These options are described in Chapter 16.2.2.1. For the Arc-Length method, **PMETHD** is used to control the Arc Length incrementation method, which may also be **AUTO**, **SEMI** or **FIX**. The **PINC**

Table 17-2. GEOMETRIC NONLINEAR SOLUTION CONTROL

NLSOLVE	ID	SMETH	IMETH	PMETH	CONV	TENDIV	UMAX	MINK	-cont-
-cont-	MAXK	MAXP	PINC	MAXPINC	UDIV	LAMDA	EPMAX	ETAS	-cont-
-cont-	EPSE	EPSP	EPSU	ROTMAX	STNMAX	FACMAX	MINP		

TO CONTROL:	FIELDS USED:	DESCRIPTION	DEFAULT
METHOD SELECTION	**SMETH**	Solution method	**ARC**
	IMETH	Iteration method	**AUTO**
	PMETH	External load incrementation method	**AUTO**
	CONV	Overall convergence criteria selector	**EPU**
	TENDIV	Strategy for tentative divergence	**1**
ITERATION	**MAXK**	Maximum number of stiffness updates	**99**
	MAXP	Maximum number of unbalanced force iterations	**30**
	MINP	Minimum number of unbalanced force iterations	Determined Automatically
	PINC	Number of load increments	Determined Automatically
	MAXPINC	Maximum number of load increments	**99**
CONVERGENCE	**UDIV**	Displacement norm threshold defining divergence	**0.9999**
	LAMDA	Convergence rate control parameter	**0.9999**
	EPMAX	Automatic load increment control parameter	Determined Automatically
	ETAS	Percentage of yield to be processed in a sub-increment	**0.25**
	ROTMAX	Maximum rotation allowed	**7.5** (for small strain) **15.0** (for Green strain)
	STNMAX	Maximum strain allowed	**0.03**
	EPSE	Strain energy convergence criterion	**0.001** (for **ARC**) **0.01** (For **NR**)
	EPSP	Load convergence criterion	**1.0**
	EPSU	Displacement convergence criterion	**0.01** (for **ARC**) **0.1** (For **NR**)
	UMAX	Maximum allowable deflection denoting divergence	See Text
	MINK	Minimum number of stiffness updates	**0**
	FACMAX	Maximum load factor indicating divergence	**10.0**

parameter is used to control the load increment size for the Newton-Raphson method. For the Arc- Length method, the increment size required is the amount of the Arc Length to be increased, however, it is very difficult for you to specify the actual value of Arc Length size. Therefore, the Arc Length size is controlled indirectly by **PINC**. In this case, **UAI/*NASTRAN***

computes the equivalent Arc Length size from the load size determined by **PINC** and uses the computed Arc Length size for all later processing. An exception to this is the computation of the default Arc Length size. This is based on the load error instead of using the equivalent Arc Length size computed by **PINC=10**, which is the default for the Newton-Raphson method.

Solution Termination. The termination of an analysis is usually signaled by the load factor. Specifically, when load factor reaches 1.0, the solution is completed. However, in Geometric Nonlinear analysis, the displacement may increase forever and the load factor may become negative without ever reaching 1.0. In order to terminate the analysis gracefully, two parameters, **UMAX** and **FACMAX**, are provided. If the absolute maximum displacement is greater than **UMAX**, UAI/*NASTRAN* initiates the tentatively divergent solution strategy with **TENDIV = 0** so that the load factor closest to **UMAX** will be found. The default value for **UMAX** is 10^5 if there is no critical point. If critical points are found during the solution, the default values is set to $10 \times U_{\max,c}$ where $U_{\max,c}$ is the maximum absolute displacement at the first critical point. A limitation is also put on the load factor. If the absolute maximum load factor is greater than **FACMAX**, then the solution is terminated.

Element Rotation and Strain. Two parameters, **ROTMAX** and **STNMAX**, are used to control the maximum element rotation and maximum strain per load increment in all elements of the model. When the element rotation or strain is greater than these thresholds, tentatively divergence processing (i.e., **TENDIV=0**) is performed. This results in a continuation of the solution using a reduced load increment size, or a termination of the solution when either of **ROTMAX** or **STNMAX** is exceeded. Note that the **STNMAX** check is performed only if you have requested the large stretch option on your **NLTYPE** Case Control command. **ROTMAX** is checked for all strain measurements. These two parameters are very useful if you want to keep the load increment size small while using the automatic load incrementation method, **PMETHD=AUTO**, because the values of the parameters are used in computing the increment size.

17.5.4 Solution Results

Solution results for Geometric Nonlinear analysis is the same as those of the Material Nonlinear analysis which are described in Chapter 16.6.

17.5.5 Modeling Guidelines

To summarize, in order to perform a Geometric Nonlinear analysis, you must ensure that the input data stream includes: the analysis type, requested with the Case Control command **NLTYPE**, and the solution method, requested with the Bulk Data entry **NLSOLVE**. If you have done so, then every geometric nonlinear element such as BEAM and QUAD4, will behave nonlinearly in the analysis. Remember, however, that only elements which have nonlinear material properties are considered during the Material Nonlinear portion of a combined analysis.

You must use MPC constraints and rigid elements with caution in a Geometric Nonlinear analysis. MPC equations and rigid elements use a small rotation theory which may or may not be valid in your particular Geometric Nonlinear analysis. However, SPC constraints and enforced displacements such as **SPCD** are perfectly legal in a Geometric Nonlinear analysis.

Since both the Newton-Raphson method and the Arc Length method can be used to solve the geometric nonlinear problem, it may be necessary to determine which to use for a specific analysis. If you expect that there are stability problems, such as snap-through of shallow shell, in your structure, you ***must*** use the Arc Length method, because the Newton-Raphson method will fail. For structures without stability problems, experience shows that, for problems without convergence difficulty, the Newton-Raphson method converges faster. For problems with

convergence difficulties, the Arc Length method usually converges faster. For this reason, the Arc Length method has been selected as the default method for Geometric Nonlinear analyses.

Also, if you know that your problem is within the scope of small strain, you should not use the large stretch feature to solve the problem. Although the two methods will give similar results, the computation of new geometry at each iteration for large stretch adds additional expense to the iteration procedure, and the solution may require considerably more iterations to converge.

17.6 EXAMPLE PROBLEMS - GEOMETRIC NONLINEAR ANALYSIS

This section provides you with seven examples of Geometric Nonlinear problems, some of which also have material nonlinearity. These models have different geometries including a rod, beam, flat plate, cylindrical shell, elliptical shell, and pressure vessel. In addition to providing examples, these problems serve as verification of the Geometric and Material Nonlinear analysis procedure implemented in **UAI/*NASTRAN***.

17.6.1 Rod Element Snap-Through

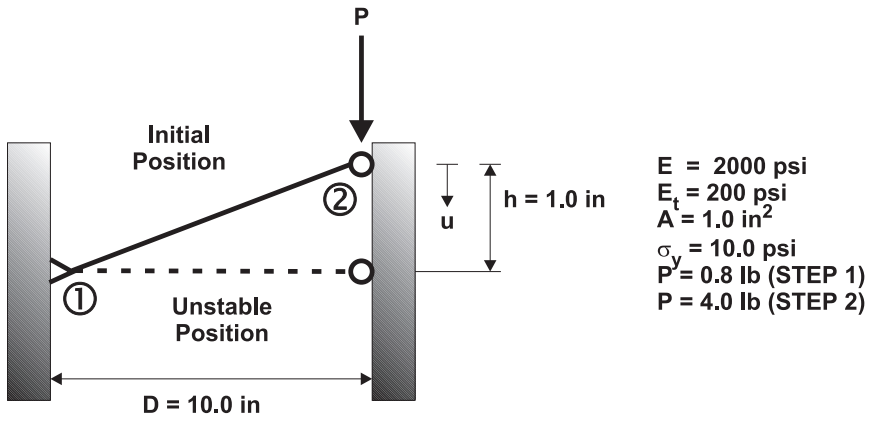
A simple structural problem is used to show how certain types of buckling conditions can not be approximated by the linear methods described in Chapter 18. This problem illustrates:

- ❒ The load critical point
- ❒ Nonlinear buckling load
- ❒ Large deformations
- ❒ Combined Material and Geometric Nonlinear Analysis

Example Problem 17-1
A slightly inclined rod is pinned at one end and free to slide vertically at the other end. A downward load, ***P***, is applied to the sliding end. The problem is to determine the nonlinear buckling load, solution curve, and internal forces due to the large rotation of the rod.

The model consists of one ROD element. The element is constrained in the x- and y- directions at GRID point 1, and constrained only in the x-direction at GRID point 2. The structure and its physical properties are shown in Figure 17-2.

Figure 17-2. SNAP-THROUGH OF INCLINED ROD



The theoretical solution is given by:

$$l_o = \sqrt{d^2 + h^2}$$

$$l = \sqrt{d^2 + (h-u)^2}$$

$$F = EA\left(1 - \frac{l_o}{l}\right)$$

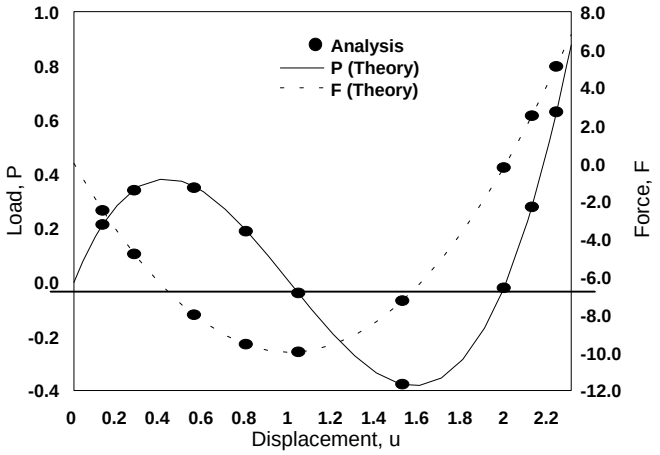
$$P = -F\frac{(h-u)}{l}$$

A load ***P*** is applied in downward in the y-direction as shown in Figure 17-2. Figure 17-3 shows the results of the **UAI*/NASTRAN*** Geometric Nonlinear analysis, in the form of the solution curve. From the plots, you can see that there is an excellent agreement between theory and analysis results.

This problem, although structurally simple, contains many features of Geometric Nonlinear analyses. At its initial position, because of the inclination of the rod, a non-zero force is required to begin displacing GRID point 2 downward. The force reaches a maximum when the differential stiffness cancels the element stiffness of the rod in the vertical direction. This point is called a ***load critical point*** and the load taken by the structure is called the ***nonlinear buckling load***. At this point, the structure is unstable. A slight increase in the force causes the displacement to change from about 0.4 to more than 2.1., as shown in Figure 17-3.

Many analysts use the buckling analysis method described in Chapter 17 to estimate the buckling load, which is called the ***linear buckling load***. For this particular problem, the linear buckling load is 1.990, while the nonlinear buckling load computed using the Geometric Nonlinear analysis procedure is 0.382. The ratio of linear and nonlinear results is more than 5.0. This illustrates the limitations of the linear buckling analysis and indicates that its results should be used with caution.

Figure 17-3. SOLUTION CURVE FOR INCLINED ROD



17

In order to show the Geometric Nonlinear and combined analysis features, two steps are used in analyzing this problem: P = 0.8 lb for the first step and P = 4.0 lb for the second step. For the first step, the stress in the rod is within the linear range and the analysis type used is **GEOM**. Since structural instabilities are involved in this step, the solution method used must be the **ARC** length method. For the second step, the stress in the rod becomes plastic and the analysis type used is **GEOM,MAT**, a combined nonlinear analysis. Since there is no structural stability in this step, either the Arc Length method or the Newton-Raphson method can be used. For this example problem, the Newton-Raphson method has used. The solution curve shown in Figure 17-3 is for **STEP 1** only. The Case Control commands for the nonlinear analysis are:

```
NLFORCE(SORT2,PRINT,PUNCH) = 2
SUBCASE  1
   STEP  1
      NLTYPE = GEOM
      LOAD = 4
      NLSOLVE = 1
   STEP  2
      NLTYPE = GEOM,MAT
      LOAD = 6
      NLSOLVE = 4
```

The Bulk Data entries required are:

MAT1	**3**	**2000.0**		**1.0-7**					
MATNL1	**3**	**PLASTIC**	**VM**	**10.0**	**200.0**				
FORCE	**4**	**2**		**0.8**	**0.0**	**-1.0**	**0.0**		
FORCE	**6**	**2**		**4.0**	**0.0**	**-1.0**	**0.0**		
NLSOLVE	**1**	**NR**	**AUTO**	**AUTO**					
NLSOLVE	**4**	**ARC**	**AUTO**	**AUTO**					

You can find the input data for this problem in the file **GNON3**.

17.6.2 Large Deflection of a Beam

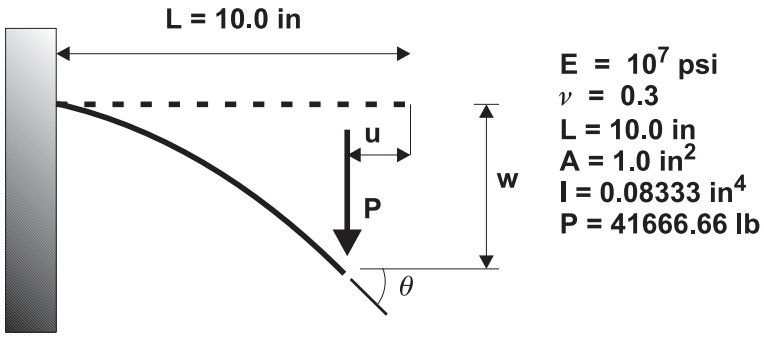
When performing Geometric Nonlinear analysis, you may select from two strain models: small strain and Green strain. These are selected by the **NLTYPE** Case Control command. The small strain model is used in cases where the rotation at a material point is small with respect to the displaced element coordinate system. On the other hand, Green strain is used when this rotation is large. It is possible to use the small strain model for large rotations by creating a fine mesh so that the relative rotation of individual elements is smaller. This problem shows the relative accuracy of these models.

Example Problem 17-2

A beam of length 10.0 in is fixed at one end and subjected to a transverse force at its free end. You wish to determine the displacements at the free end due to large displacement effects. You also wish to study the differences in the solution that occur with the small strain and Green strain models for several different mesh refinements.

The structure together with its geometry and material properties are shown in Figure 17-4. To test the effects of mesh refinement and the strain model, four separate finite element models are used: one BAR element with small strain; one BAR element with Green strain; four BAR elements with small strain; and four BAR elements with Green strain. Only the second case has been included with the test problems. It is found in file **GNON4**.

Figure 17-4. LARGE DEFORMATION OF CANTILEVER BEAM



The required Case Control packet is:

```
NLTYPE = GEOM(STRAIN=GREEN)
SUBCASE  1
   STEP  1
      LOAD = 1
      NLSOLVE = 1
```

The results of these four analyses, together with their theoretical solution reported in [Mattiasson85], are shown in the table below.

	U	W	STRAIN MODEL	NUMBER OF ELEMENTS
THEORY	3.88	7.14		
UAI/*NASTRAN*	4.04	7.19	Green	1
	4.22	8.20	Small	1
	3.87	7.16	Green	4
	3.86	7.20	Small	4

For the models with a single element, Green strain gives much better results than small strain, and the results for Green strain are very good, considering only one BAR element is used in the model. For the four element models, the results for small strain and Green strain are very similar showing that mesh refinement improves the accuracy of the small strain model.

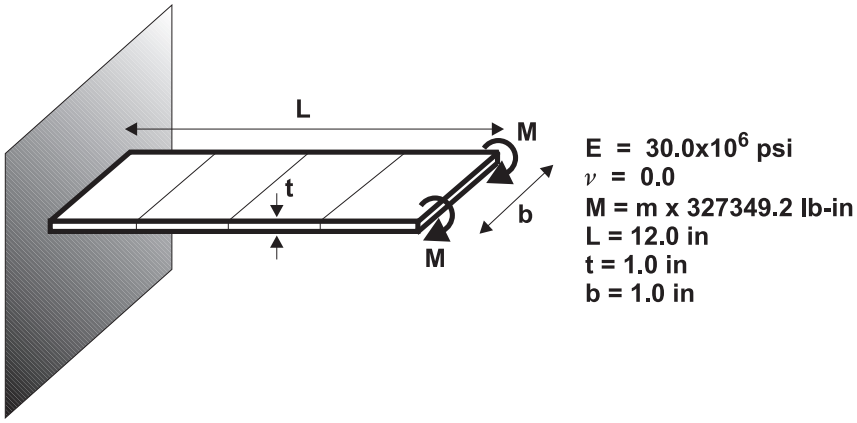
17

17.6.3 Circular Deformation of a Beam

Section 17.2.1 indicated that rotations and translations of a GRID point rigid body may be arbitrarily large. This sample problem is intended to give you a feeling for the range of applicability of Geometric Nonlinear analysis.

Sample Problem 17-3
A beam of length 12 in is fixed at one end and subjected to an end moment at its free end. You wish to track the deformed shape due to large displacements.

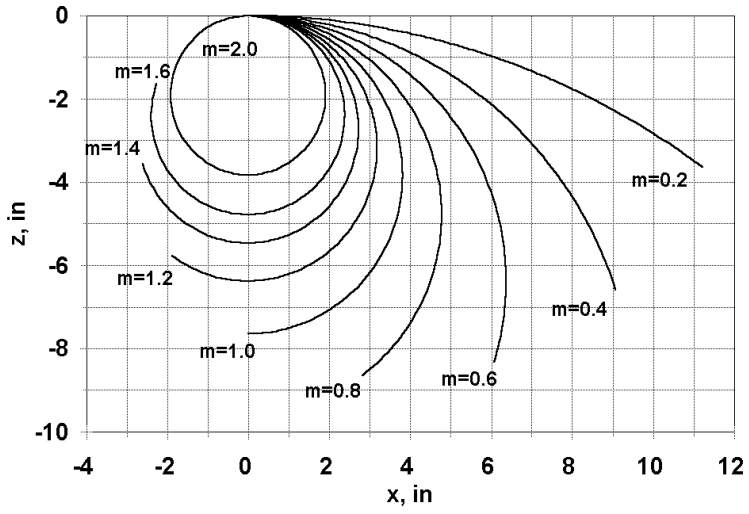
Figure 17-5. BEAM SUBJECTED TO END MOMENT



The structural model, as well as its geometry and material properties are shown in Figure 17-5. The structure is model by four QUAD4 elements. It is subjected to an end-moment of *m*x 327249.2 lb-in, where *m* is varied from 0.2 to 2.0. A Geometric Nonlinear analysis (**NLTYPE=GEOM**) is performed and the deformed shapes for different values of *m* are plotted in Figure 17-6. When *m*=2.0, the beam is bent backward into a complete circle. [Surana83] also analyzed this problem and obtained similar results. The **UAI/*NASTRAN*** results are presented in Figure 17-6.

You can find the input data for this problem in file **GNON5**.

Figure 17-6. DEFORMED SHAPE OF BEAM



17.6.4 Flat Plate with a Pressure Load

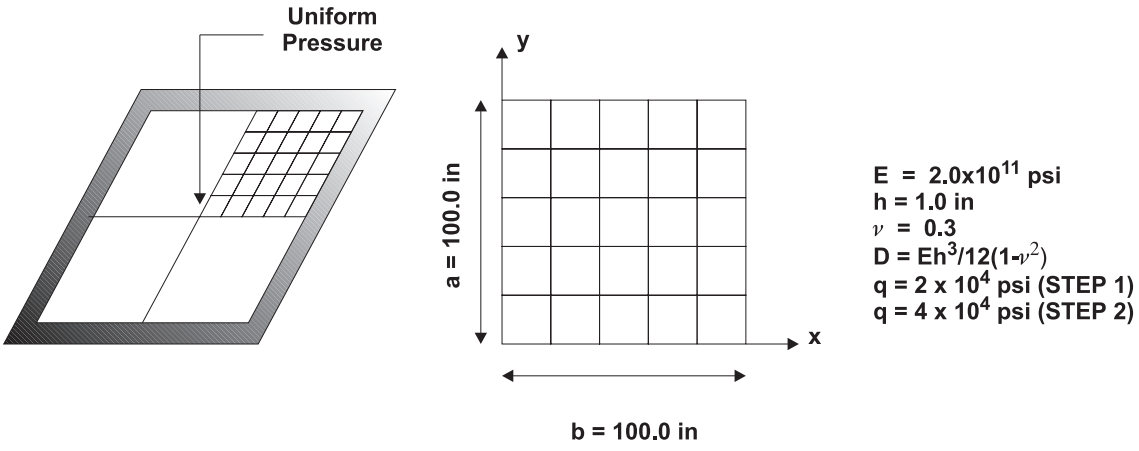
One of the basic assumptions used in performing the linear analysis of plates is that the strain at the mid-surface of the plate is negligible. This is generally true if the deflection of the plate in the direction normal to the plate is small in comparison with its thickness. When this assumption is no longer true, a Geometric Nonlinear analysis is required. Such a case is presented in this sample problem.

Sample Problem 17-4
A thin plate 200 in by 200 in is fixed along its four sides. It is subjected to uniform pressure. You wish to find the large deflection solution and compare it with the linear solution.

The structural model, geometry and material properties are shown in Figure 17-7. A quarter of the structure is modeled using a 5x5 mesh of QUAD4 elements, and symmetric boundary conditions are applied. This problem is then analyzed using **NLTYPE=GEOM** for two steps. The pressure load for these two steps and corresponding results for linear and nonlinear analysis are shown in Figure 17-8. The theoretical solution is given by [Timoshenko59]. You can see that the nonlinear solution computed by **UAI/*NASTRAN*** is in good agreement with theoretical results. Since the load for this problem is pressure, which has follower forces. The effect of follower forces is negligible.

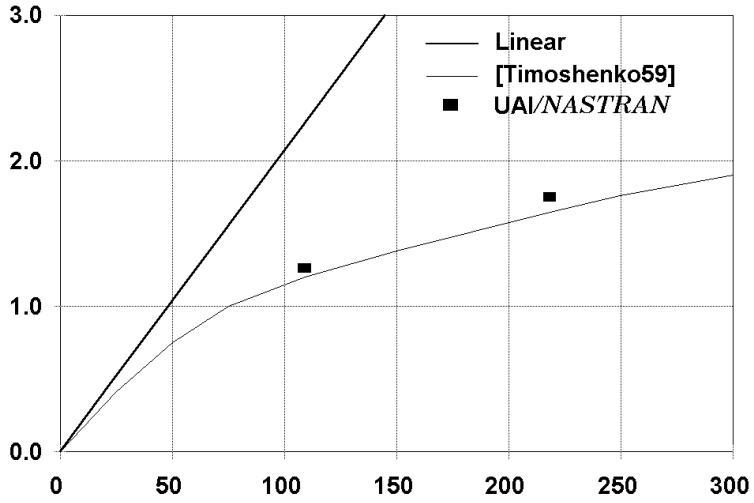
You can find the input data for this problem in the file **GNON6**.

Figure 17-7. LARGE DEFLECTION OF SQUARE PLATE



17

Figure 17-8. NONLINEAR SOLUTION FOR SQUARE PLATE



17.6.5 Shell Roof Snap-Through

A classic problem that often poses difficulties in solution is the snap-through of a shallow shell which has critical points on the solution curve. The Newton-Raphson method fails for such problems because the stiffness matrix becomes either singular or infinitely rigid at these points. The Arc Length method, however, is ideally suited to solving this problem as shown in this sample problem.

Sample Problem 17-5
A cylindrical shell roof, simply supported along its straight edges and free along its curved edges, is subjected a concentrated load at its center. You wish to determine the nonlinear buckling load and the geometric nonlinear solution curve.

The structure, geometry and material properties, boundary conditions, and the concentrated force applied are shown in Figure 17-9. A quarter of the structure is modeled by an mesh of 4x4 QUAD4 elements. This problem is analyzed with **NLTYPE=GEOM** and the Arc Length solution method. The **UAI/*NASTRAN*** solution curve is plotted with the results given by [Sabir73] in Figure 17-10. The nonlinear buckling load is 0.6 and the linear buckling load is 1.23. Note that the linear buckling load is more than twice of the nonlinear buckling load.

The complexity of this problem is made evident by the fact that there are two types of critical points on the solution curve: the ***load critical point*** and the ***displacement critical point***. When proceeding along the solution curve, there is actually a phase where the displacement under the load decreases or *snaps back*. From a numerical point-of-view, this problem is considerably more complex than the more conventional load snap-through problem. It is clear that the **UAI/*NASTRAN*** nonlinear analysis procedure works well under this extreme condition.

You can find the input data for this problem in the file **GNON7**.

Figure 17-9. CYLINDRICAL SHELL ROOF

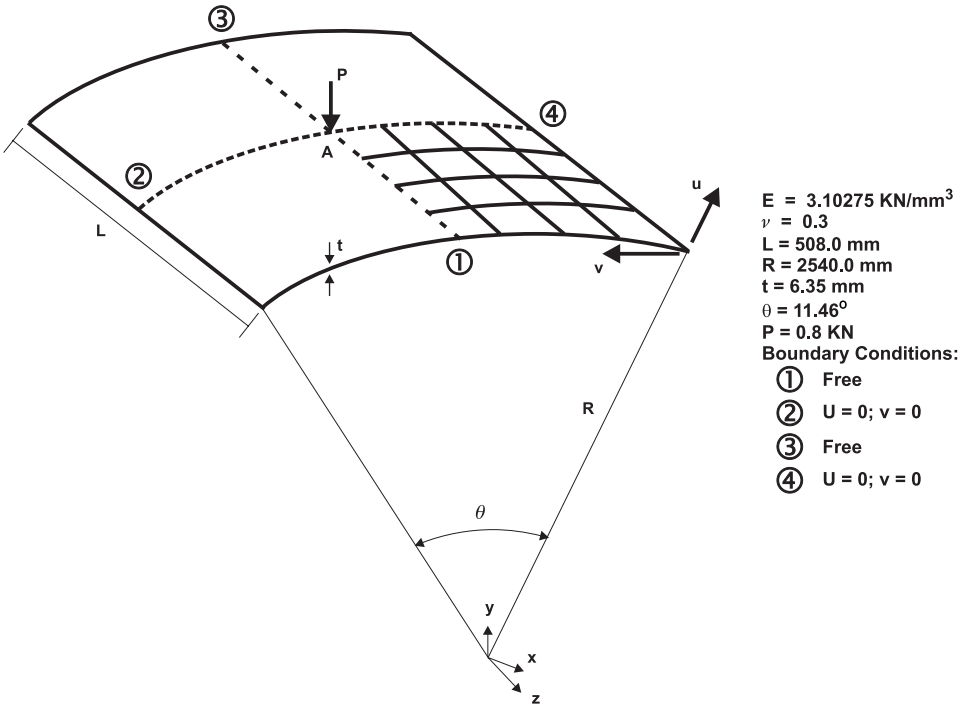
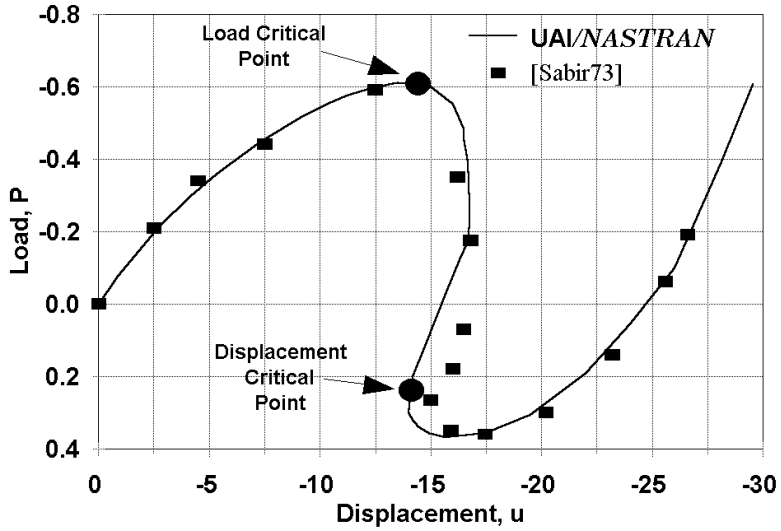


Figure 17-10. NONLINEAR SOLUTION



17

17.6.6 Elliptical Cylinder Subject to Internal Pressure

Follower forces are external loads which are unique to Geometric Nonlinear analysis. This type of load changes its direction or magnitude due to the deformation of the structure. By default, **UAI/*NASTRAN*** computes the effects of follower forces for Geometric Nonlinear analyses. While the sample problem given in Section 17.7.4 has follower forces, their effects were negligible. In this problem, however, the predominant nonlinear effect is the follower forces.

Sample Problem 17-6

A thin walled cylinder with elliptic shape is subjected to an increasing internal pressure. You wish to see the follower force effect of pressure on the deformed shape of the cylinder considering both geometric and material nonlinearities.

Figure 17-11 shows the cross-section of the elliptic cylinder which is modeled with 12 QUAD4 elements. The material property of the cylinder is elastoplastic, thus it obeys the von Mises yield criterion using the isotropic hardening rules. The nonlinear properties are specified by **MAT**, **MATNL**, and **TABLENL** Bulk Data entries. The problem is highly nonlinear, especially at the initial stage of loading. In order to facilitate convergence, six load steps are used. The steps are initially very small (0.01, 0.05, 0.1) and then they increase rapidly (0.5, 1.0, 5.0). Although a combined nonlinear analysis is performed, the effects of the material nonlinearity are minimal. The nonlinear Case Control commands are:

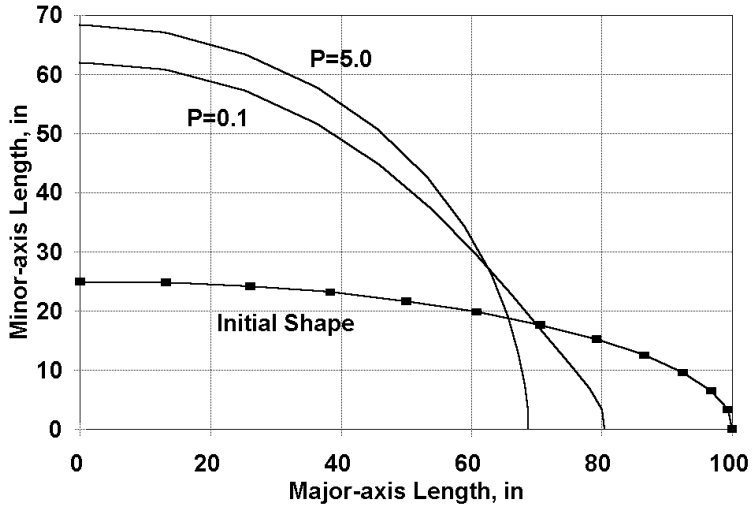
```
NLTYPE = GEOM,MAT
SUBCASE  1
   STEP  1
      LOAD = 5
      NLSOLVE = 100
   STEP  2
      LOAD = 10
      NLSOLVE = 100
   STEP  3
      LOAD = 20
      NLSOLVE = 200
   STEP  4
      LOAD = 30
      NLSOLVE = 200
   STEP  5
      LOAD = 40
      NLSOLVE = 200
   STEP  6
      LOAD = 50
      NLSOLVE = 200
```

There are two important groups of Bulk Data entries. The first group defines the six load levels. These data are given by:

PLOAD4	**5**	**101**	**0.01**				**THRU**	**112**	
PLOAD4	**10**	**101**	**0.05**				**THRU**	**112**	
PLOAD4	**20**	**101**	**0.1**				**THRU**	**112**	
PLOAD4	**30**	**101**	**0.5**				**THRU**	**112**	
PLOAD4	**40**	**101**	**1.0**				**THRU**	**112**	
PLOAD4	**50**	**101**	**5.0**				**THRU**	**112**	

The shaded fields define the actual loading.

Figure 17-11. DEFORMED SHAPE OF ELLIPTICAL CYLINDER



The second group of data define the nonlinear solution control. The first set, **100,** requests that 20 load increments be performed during each STEP (note the shaded field). The second set, **200**, reduces the number of increments to 10, reflecting the stability of the solution:

NLSOLVE	100	ARC	AUTO	AUTO					+NL1
+NL1			20						
NLSOLVE	200	ARC	AUTO	AUTO					+NL2
+NL2			10						

The analysis is then executed with **NLTYPE=GEOM,MAT** and,as shown above, the Arc Length method is used to perform the solution. The deformed shape for p = 0.1 and p = 5.0 are plotted in Figure 17-11. You can see that the effect of the follower forces makes the initial elliptical shape into a circular shape.

You can find the input data for this problem in the file **GNON8**.

17.6.7 Steel Vessel under Internal Pressure

An example of a combined nonlinear analysis was presented in Section 17.6.6, but the solution was not verified. This sample problem provides an experimental confirmation of such combined analyses, even when follower forces are included.

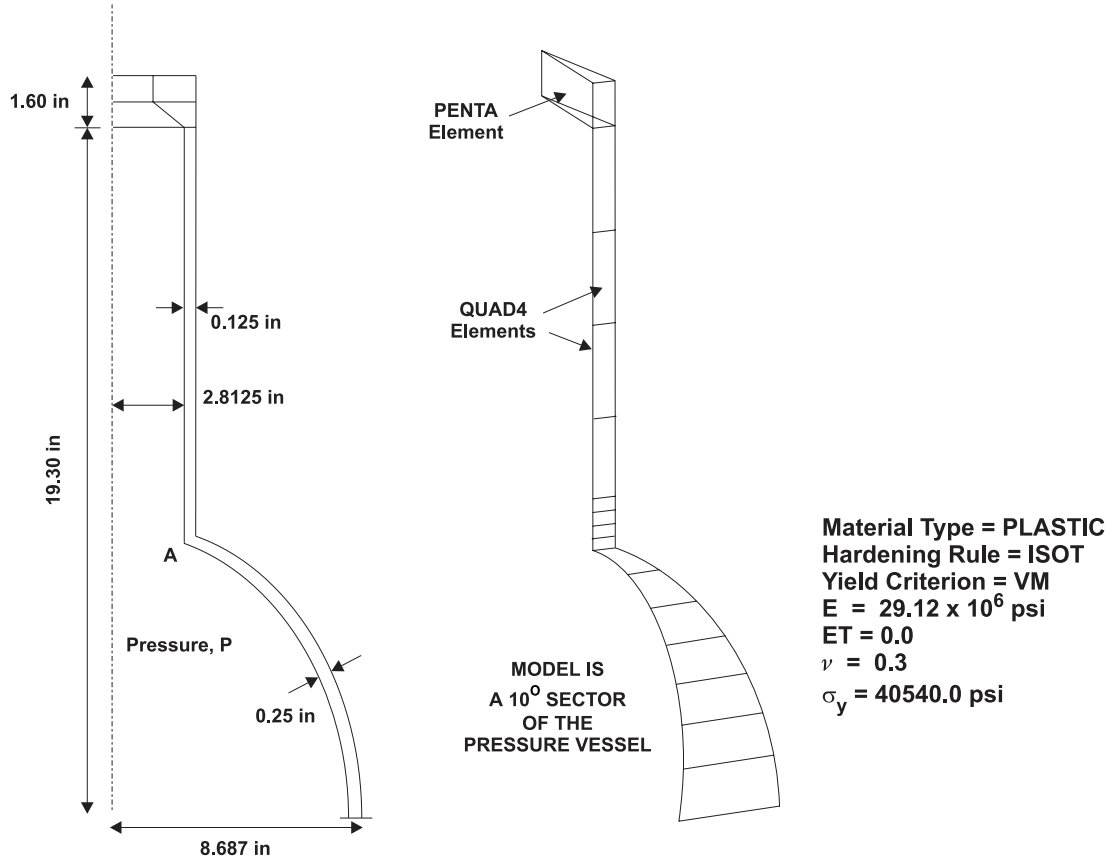
17

Sample Problem 17-7

A bottle shaped pressure vessel is subjected to internal pressure. You wish to see the comparison between the nonlinear analysis solution and the available experimental data.

The pressure vessel, its structural model, geometric and material properties, and internal pressure load are shown in Figure 17-12. This problem has geometric and material nonlinearities together with effect of follower forces. An combined analysis, **NLTYPE=GEOM,MAT** is performed with the Newton-Raphson solution method.

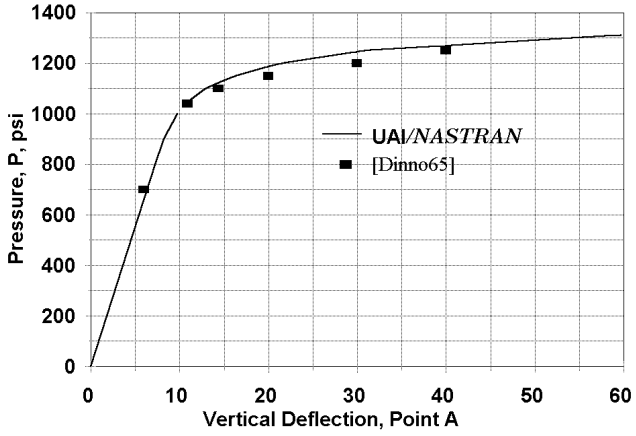
Figure 17-12. STEEL PRESSURE VESSEL



The experimental results found in [Dinno65] are plotted together with the **UAI/*NASTRAN*** nonlinear analysis results in Figure 17-13. From the plots, you can see that the agreement between the experimental and analytical results is very good.

You can find the input data for this problem in the file **GNON9**.

Figure 17-13. NONLINEAR SOLUTION



17.6.8 The Hanging Cable

The solution to the hanging cable problem is presented in Section 17.8.1. It is solved using both the Arc Length method and the Differential Stiffness approach. A comparison of the two methods are shown there. The data for this version of the example problem is found in file **GNON10**.

17.7 INPUT DATA - DIFFERENTIAL STIFFNESS APPROACH

This section describes the unique commands and data entries required to perform nonlinear geometry analysis using the differential stiffness approach in **UAI/*NASTRAN***.

17.7.1 Executive Control Commands

For static and normal modes analyses with differential stiffness, Rigid Formats 4 and 13 are requested in the Executive Control command packet, as shown below:

```
SOL 4      (Static Differential Stiffness Analysis)
SOL 13     (Normal Modes with Differential Stiffness)
```

17.7.2 Case Control Commands

For Static Analysis with differential stiffness, the Case Control packet ***must*** contain two subcases. All constraints and one static loading condition must be defined before any **SUBCASE** commands. The subcases themselves are used only for output selection, subcase 1 for the linear solution and subcase 2 for the nonlinear solution. No other special Case Control commands are required.

For Normal Modes with differential stiffness, the Case Control packet ***must*** contain three subcases. Static loads are requested above the subcase level. **LOAD**, **TEMPERATURE (LOAD)** or **DEFORM** commands may be used, or enforced displacements may be specified with **SPC** Bulk Data entries. All constraint requests must be above the subcase level. This is also true of the required **METHOD** command and the optional **DYNRED** command. The first subcase is used to request results from the linear statics analysis. The third subcase is used to request output from the modal solution.

In addition, for Rigid Format 13, a **DSCOEFFICIENT** Case Control command ***must*** be used to define scale factors for applied loads and stiffness matrices. This command is specified as,

```
DSCOEFFICIENT     = { n       }
                    { DEFAULT }
```

where *n* refers to the set identification number of the accompanying **DSFACT** Bulk Data entry. The **DSCOEF** Case Control command ***must*** appear in the second subcase of the input Case Control Packet.

17.7.3 Bulk Data Entries

Differential stiffness effects are available for the most frequently used structural elements. These are the BAR, BEAM and ROD; TRIA3, TRIA6, TRIAR, QUAD4, QUAD8, QUADR and SHEAR; and HEXA, PENTA and TETRA. Other elements such as scalar springs may be used in the model, but they will only produce linear stiffness terms.

If the **DSCOEFFICIENT** Case Control command has been used with a set identification number, then a **DSFACT** Bulk Data entry must be input. This entry defines the scale factors for applied loads and stiffness matrices. The format of this entry is:

DSFACT	SID	B1	B2	B3	B4	B5	B6	B7	-cont-
-cont-	B8	B9	*CONTINUES WITH LIST OF VALUES*						

The nonlinear solution is performed by iterating on the combined applied load vector and the incremental load vector due to differential stiffness effects. If convergence is not quickly obtained, the differential stiffness matrix is updated and load iterations are started again. The maximum number of ***load iterations*** for a given differential stiffness may be defined by the Bulk Data entry:

PARAM	MAXPITER	*maxpit*							

where the default value for *maxpit* is **4**. **UAI/*NASTRAN*** automatically determines when to initiate a new differential stiffness generation based on the rate of convergence in the load iterations and the computer time required to achieve convergence.

The maximum number of ***stiffness updates*** may be specified with the entry:

PARAM	MAXKITER	*maxkit*							

The default value for *maxkit* is **4**. The solution convergence criterion may also be specified with the entry:

PARAM	EPSIO	*eps*							

where the default value for *eps* is $\mathbf{10^{-5}}$. The iteration has converged when equation 17-3 is satisfied.

17.7.4 Solution Results

For Static Analysis with Differential Stiffness, Rigid Format 4, the **UAI/*NASTRAN*** print and plot output is requested using the same Case Control commands as described for Static Analysis, Rigid Format 1, except that output for the linear solution is requested in subcase 1 and output for the nonlinear solution is requested in subcase 2. For normal modes analysis with differential stiffness, an additional subcase, 3, is necessary for requesting output from normal modes analysis.

Rigid Format 4 also produces automatic output that describes the status of the solution at each step of the iteration. These status descriptions are shown below:

- The iteration has been completed but convergence has not yet been obtained.
- The iteration procedure has converged to the specified **EPSIO** value.
- The iteration is diverging from the **EPSIO** value
- There is not enough computer time remaining to continue iteration.
- The maximum number of iterations

An example of the iteration history, taken from Sample Problem 17-1, is shown in Table 17-2.

Table 17-3. ITERATION HISTORY

```
*** USER INFORMATION MESSAGE, DIFFERENTIAL STIFFNESS ITERATION NO.    1 HAS BEEN COMPLETED.
    CONVERGENCE WAS NOT OBTAINED ON THIS ITERATION.
    ANOTHER LOAD ITERATION WILL BE PERFORMED.


*** USER INFORMATION MESSAGE, DIFFERENTIAL STIFFNESS ITERATION NO.    2 HAS BEEN COMPLETED.
    CONVERGENCE WAS NOT OBTAINED ON THIS ITERATION.
    ESTIMATE FOR NUMBER OF LOAD ITERATIONS TO CONVERGE IS        3
    ANOTHER LOAD ITERATION WILL BE PERFORMED.


*** USER INFORMATION MESSAGE, DIFFERENTIAL STIFFNESS ITERATION NO.    3 HAS BEEN COMPLETED.
    CONVERGENCE WAS NOT OBTAINED ON THIS ITERATION.
    ESTIMATE FOR NUMBER OF LOAD ITERATIONS TO CONVERGE IS        1
    ANOTHER LOAD ITERATION WILL BE PERFORMED.


*** USER INFORMATION MESSAGE, DIFFERENTIAL STIFFNESS ITERATION NO.    4 HAS BEEN COMPLETED.
    CONVERGENCE WAS NOT OBTAINED ON THIS ITERATION.
    ESTIMATE FOR NUMBER OF LOAD ITERATIONS TO CONVERGE IS        4
    ANOTHER LOAD ITERATION WILL BE PERFORMED.


*** USER INFORMATION MESSAGE, DIFFERENTIAL STIFFNESS ITERATION NO.    5 HAS BEEN COMPLETED.
    CONVERGENCE WAS NOT OBTAINED ON THIS ITERATION.
    ESTIMATE FOR NUMBER OF LOAD ITERATIONS TO CONVERGE IS        3
    ANOTHER LOAD ITERATION WILL BE PERFORMED.


*** USER INFORMATION MESSAGE, DIFFERENTIAL STIFFNESS ITERATION NO.    6 HAS BEEN COMPLETED.
    THE ITERATION PROCEDURE IS DIVERGING WITH AN ERROR RATIO OF EPSI =  8.94328E-04.
    THE ALLOWABLE ERROR RATIO IS  1.00000E-05.
    A DIFFERENTIAL STIFFNESS UPDATE WILL BE PERFORMED.


*** USER INFORMATION MESSAGE, DIFFERENTIAL STIFFNESS ITERATION NO.    7 HAS BEEN COMPLETED.
    CONVERGENCE WAS NOT OBTAINED ON THIS ITERATION.
    ANOTHER LOAD ITERATION WILL BE PERFORMED.


*** USER INFORMATION MESSAGE, DIFFERENTIAL STIFFNESS ITERATION NO.    8 HAS BEEN COMPLETED.
    THE ITERATION PROCEDURE HAS CONVERGED TO THE ERROR RATIO OF EPSI =  6.98699E-06.
    THE ALLOWABLE ERROR RATIO WAS  1.00000E-05.
```

17.7.5 Modeling Guidelines

This section provides you with modeling guidelines for Statics and Normal Modes differential stiffness analysis. It is assumed that you are familiar with Linear Statics and Real Eigenvalue analyses. It is important to use the appropriate eigenextraction method in your analysis.

A key question arises as to when to perform a Geometric Nonlinear finite element analysis. In lieu of a set of well defined guidelines which are difficult to put forth, one often relies on engineering judgment to accomplish this. In general, this type of analysis is justified by considering the geometry of the structure and the type of loading. Slender flexural members (high span to depth ratio) with in- plane (or membrane) type loads are very likely to yield erroneous results if a Linear Statics analysis is performed. Frequently, these errors reflect on the predicted displacements (as will be seen in the statics differential stiffness analysis example). Conversely, it may be found that at a certain load, the deflections increase more rapidly than predicted by a linear solution. The finite element analyst must therefore be experienced enough to recognize these situations and perform the appropriate type of analysis.

17.7.5.1 Static Analysis Modeling

There are many static problems where ignoring the geometric nonlinear aspects of structural response often leads to inaccurate solutions. As remarked earlier, for the accurate determina-

tion of displacements and the prediction of load-displacement responses of slender structures with membrane type forces, a Geometric Nonlinear analysis must be performed. The applications of such analyses are of significant importance in aerospace engineering, cooling towers, cable-stayed bridges and other slender structures. The modeling example discussed in this section demonstrates the importance of including the geometric nonlinearity in the analysis.

17.7.5.2 Normal Modes Modeling

Although there are many similarities between Rigid Format 13, normal modes with differential stiffness, and Rigid Format 3, linear normal modes analysis, there are also important ***differences*** in the requirements of the Case Control and Bulk Data packets. As mentioned earlier, the Case Control packet ***must*** contain three subcases for dynamic modeling with differential stiffness. The first subcase is used only to request solution results from the Linear Statics analysis. The second subcase is used to control the statics differential stiffness solution. A **DSCOEFFICIENT** Case Control command ***must*** appear in the second subcase, either to select a **DSFACT** Bulk Data packet, or to explicitly select the default value of unity. The third subcase is used to control output requests from the modal solution. All other requirements previously defined for Normal Modes also apply to Normal Modes with Differential Stiffness.

The guidelines discussed earlier for consideration of geometric nonlinearity in finite element modeling are applicable to both statics and dynamic modeling. You should include the geometrical effects in the computation of natural frequencies if the geometry of the structure is slender. As mentioned earlier in this chapter, tensile membrane forces increase the natural frequencies and the compressive membrane forces decrease them. Therefore, the natural frequencies you determine from the normal modes analysis with differential stiffness are ***dependent*** on the static preload. This is inevitable, as the differential stiffness of a structure is dependent on the static displacements.

You should be aware of an important case in normal modes analysis with differential stiffness. If the static preload you prescribe in the analysis happens to be the ***buckling*** load, the solution for natural frequencies produce a result of $\omega = 0$ as $\lambda = \lambda_{cr}$. For instance, if the static load you prescribed on the structure is the first buckling load, then the frequency for the first mode of free vibration should be computed as zero. This outcome however depends on the method of eigenextraction you choose. The modeling example described below discusses this situation in some detail.

17.8 EXAMPLE PROBLEMS - DIFFERENTIAL STIFFNESS APPROACH

This section provides two examples of Differential Stiffness analyses. The first is a classic problem of a cable hanging under its own weight. The second problem illustrates the effects of differential stiffness on the normal modes of a beam structure.

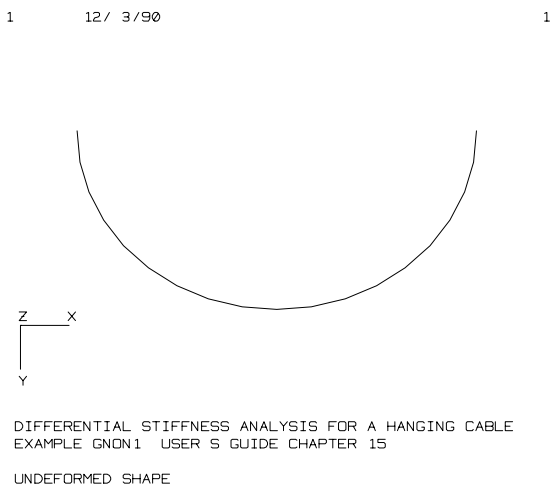
17.8.1 Static Analysis of a Cable

The first example is the modeling and analysis of a hanging cable sagging under its own weight. This is a classical Geometric Nonlinear statics problem in which the deformation of the cable assumes the shape of a catenary.

Sample Problem 17-9
A semicircular cable is suspended between two fixed supports. The radius of the circle is 10 ft and it is sagging under its own weight. You wish to determine the deformed shape of the cable.

The finite element model is comprised of 19 GRID points connected with 18 BAR elements. The GRID points are spaced at 10° intervals along the initial semicircular shape. An undeformed plot, created with the ***uaiplot*** utility, is shown in Figure 17-14. The input data for this problem is found in file **GNON1**. The bending stiffness of the elements is nominally small in order to provide a non-singular linear solution. The axial stiffness of the BAR elements was made sufficiently large to limit extensional displacements.

Figure 17-14. HANGING CABLE MODEL



As mentioned earlier, the Case Control packet must contain two subcases, one for linear statics solution, and the other for the differential stiffness solution. The static loads are defined before the **SUBCASE** commands, as shown below for this problem:

```
LOAD = 100
SPC  = 100
SUBCASE 1
   LABEL = LINEAR SOLUTION
SUBCASE 2
   LABEL = NONLINEAR SOLUTION
BEGIN BULK
```

Since the cable deforms under its own weight, the load is applied using a **GRAV** Bulk Data entry. The remaining data for this problem follow the format for Linear Statics analysis. The sample problem is then executed in **UAI/*NASTRAN***.

The theoretical solution for this problem is developed in [Spiegel58] and will not be repeated here. The equilibrium position of the cable is defined by a catenary curve. It is important to compare the Linear Statics solution with the differential stiffness solution. **UAI/*NASTRAN*** provides the results of both solutions in the output. Figure 17-15 shows the deformed shape of the cable computed by the Linear Statics solution, and Figure 17-16 the differential stiffness solution.

The comparison of these figures clearly demonstrates that it is pointless to perform a Linear Statics finite element analysis for this problem. Only the Geometric Nonlinear solution has meaningful results with tolerable errors. The reason for this is simple. Due to its geometry, the cable undergoes large rigid body displacements that do not produce any stresses.

Figure 17-15. LINEAR SOLUTION

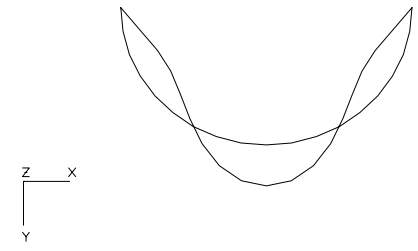
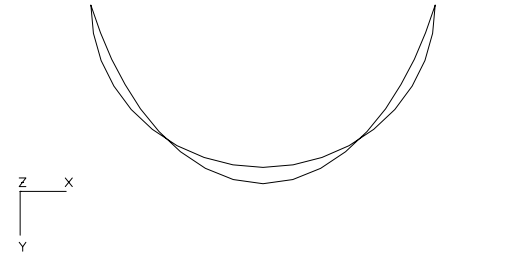


Figure 17-16. NONLINEAR SOLUTION



This problem can be also solved using Geometric Nonlinear analysis with **SOL STATICS, APPROACH NONLINEAR** as described in section 17.5. It is interesting to compare the results from the Geometric Nonlinear analysis and those from the differential stiffness approach, which are given by the following table:

Vertical Displacement at GRID Point 19		
Theory	**Geometric Nonlinear Analysis**	**Differential Stiffness Analysis**
0.9338	**0.9386**	**1.2168**

You can see that Geometric Nonlinear analysis gives better results. This is because these results are far more accurate than those computed using the linear approximations of the differential stiffness approach.

17.8.2 Normal Modes of a Beam

The modeling example discussed in this section demonstrates the effect of geometric non-linearity on the natural frequencies of a beam structure. It compares the effects of both tensile and compressive preloads to the natural frequencies assuming linear behavior.

Sample Problem 17-10
An aluminum beam with a span of 100 ft. is subjected to an axial compressive load equal of 400 lb. The beam

The finite element model and its engineering data are shown in Figure 17-17. This model is found in file **GNON2A**. Unlike Static Differential Stiffness analyses, the Normal Modes solution requires three subcases in the Case Control packet:

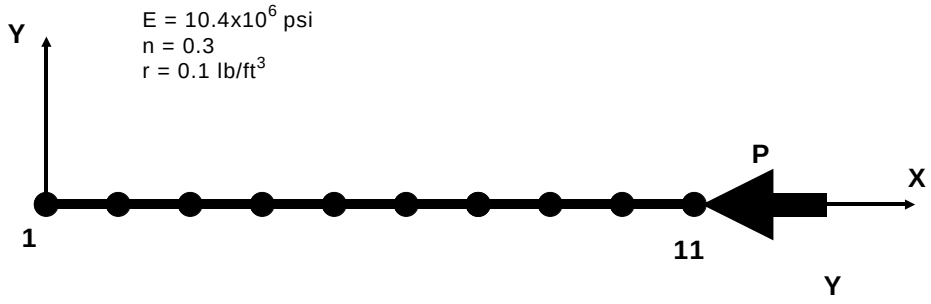
```
SUBCASE 100
   LABEL = STATICS SOLUTION WITH COMPRESSIVE LOAD
   LOAD  = 100
SUBCASE 200
   LABEL = SECOND ORDER STATICS SOLUTION
   DSCOEFFICIENT = DEFAULT
SUBCASE 300
   LABEL  = NORMAL MODES WITH DIFFERENTIAL STIFFNESS
   METHOD = 101
BEGIN BULK
```

The **DSCOEFFICIENT** Case Control command selects the default scale factors for the applied load and stiffness matrix. The Givens method was used for eigenvalue extraction by using the **EIGR** Bulk Data entry. The eigenvalue extraction method is selected in the third subcase using the **METHOD** Case Control command. While the Givens method extracts all eigenvalues, you must specify the frequency range for which the eigenvectors will be computed. In this case, the range is specified as **0.0** to **10.0** Hz. The corresponding **EIGR** Bulk Data entry is:

EIGR	**101**	**GIV**	**0.0**	**10.**					

The left end of the span is fixed in u_x and u_y ,the right end is fixed in u_y, and all points are fixed in u_z, θ_x, and θ_y. The applied load is 400 lb which is approximately one half of the Euler buckling load. As you will see at the end of this section, loads must be less than the critical buckling load for meaningful solutions.

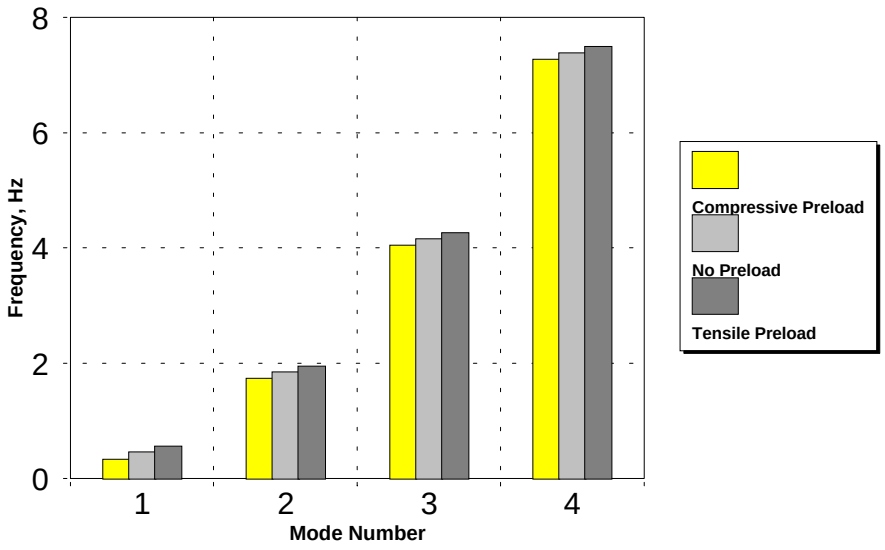
Figure 17-17. BEAM WITH AXIAL LOAD



In a second case, found in file **GNON2B**, the load was changed from compression to tension, and the natural frequencies were again computed. Finally, a third case, file **GNON2C**, was executed in which the normal modes of the structure without differential stiffness effects were computed. Figure 17-18 shows **UAI/*NASTRAN*** results of frequencies for all three cases.

These results clearly demonstrate the influence of differential stiffness on the natural frequencies of the structure. As expected, the stiffening effects of the compressive loads decreased the natural frequencies while the tensile loads increased them. These frequencies bound the natural frequencies of structure without differential stiffness effects. You will also note that the influence of differential stiffness on the natural frequencies becomes negligible for higher modes of vibration of the structure.

Figure 17-18. COMPARISON OF NATURAL FREQUENCIES



17

The theoretical natural frequency for the first mode considering differential stiffness effects due to a compressive membrane force is calculated by:

$$f = \left[\frac{1}{4\rho A l^2} \left(\frac{\pi^2 EI}{l^2} - F \right) \right]^{\frac{1}{2}} \text{ Hz} \qquad (17\text{-}15)$$

This results in a theoretical frequency is 0.337 Hz which is the result obtained with **UAI/*NASTRAN***.

You will note that when the applied load equals the critical buckling load the frequency is 0.0 Hz. The eigenextraction methods are numerical techniques that may yield results that have appreciable errors for this case. Furthermore, for loads exceeding the critical load, the solution is undefined. This may often result in large negative eigenvalues that do not represent physically meaningful results. Such results indicate that you have formulated your problem incorrectly.

17.9 REFERENCES

[Chung88]: Chung, T. J., "Continuum Mechanics", Prentice Hall, Englewood Cliffs, N.J., 1788.

[Crisfield81]: Crisfield, M. A., "A Fast Incremental/Iterative Solution Procedure that Handles Snap-through", Computers and Structures, Vol. 13, 55-62, 1981.

[Dinno65]: Dinno and Gill, "An Experimental Investigation into the Plastic behavior of Flush Nozzles in Spherical Pressure Vessels", Int. J. Mesh. Sci., Vol. 7, 817, 1965.

[Mattiasson85]: Mattiasson, K., Bongtsson, A. and Samuelsson, A., "On the Accuracy and Efficiency of Numerical Algorithms for Geometrically Nonlinear Structural Analysis", Finite Element Methods for Nonlinear Problems, Proc. of the Europe-US Symposium, Trondheim, Norway, Aug12- 16, 1985, edited by Bergan, P. G. Bathe, K. J. and Wunderlich, W.

[Ricks79]: Ricks, E., "An Incremental Approach to the Solution of Snapping and Buckling Problems", Int. J. Solids and Structures, Vol 15, 524-551, 1979.

[Sabir73]: Sabir, A. B. and Lock, A. C., "The application of finite elements to the large deflection geometrically nonlinear behaviour of cylindrical shells", Variational Methods in Eng., Southampton University Press, 7/66-7/74, 1973.

[Spiegel58]: Spiegel, M.R., *Applied Differential Equations*, Prentice-Hall, 1958.

[Surana83]: Surana K. S., "Geometrically Nonlinear Formulation for the Curved Shell Elements", Int. J. for Numerical Methods in Eng., Vol. 17, 581-618, 1983.

[Timoshenko59]: Timoshenko. S. and Woinowsky-Krieger, S.,"Theory of Plates and Shells", McGraw-Hill, Fig. 208, 244, 1959.

17.10 BIBLIOGRAPHY

[Badawi92]: Badawi, M. and Cusens A. R., "Symmetry of the Stiffness Matrices for Geometrically Nonlinear Analysis", Communications in Applied Numerical Methods, Vol. 8, 135-140, 1992.

[Batoz79]: Batoz, J-L, Dhatt, G., "Incremental Displacement Algorithms for Nonlinear Problems", Int. J. for Numerical Methods in Eng., Vol. 14, 1262-1266, 1979.

[Bergan78]: Bergan, D. G., Horrigmoe, G., Krakeland, B. and Sereide, T. H., "Solution Techniques for Nonlinear Finite Element Problems", Int. J. for Numerical Methods in Eng., Vol. 12, 1677-1696, 1978.

[Crisfield83]: Crisfield, M. A., "An Arc-Length Method Including Line Searches and Accelerations", Int. J. for Numerical Methods in Eng., Vol. 17, 1269-1289, 1983.

[Farard93]: Farard, M. and Massicotte, B., "Geometrical Interpretation of the Arc-Length Method", Computers and Sturctures, Vol. 46, No. 4, 603-615, 1993.

[Haisler77]: Haisler, W. E. and Stricklin, J. A., "Displacement Incrementation in Nonlinear Structural Analysis by the Self-Correcting Method", Int. J. for Numerical Method in Eng., Vol 11, 3-10, 1977.

[Hibbitt79]: Hibbitt, H. D., "Some Follower Forces and Load Stiffness", Int. J. for Numerical Methods in Eng., Vol. 14, 937-941, 1979.

[Kanok81]: Kanok-Nukulchai, W., Taylar, R. L. and Hughes, T. J. R., "A Large Deformation Formulation for Shell Analysis by the Finite Element Method", Computers and Structures, Vol. 13, 17-27, 1981.

[Mang80]: Mang, H. A., "Symmetricablity of Pressure Stiffness for Shells with Load Free Edges", Int. J. for Numerical Methods in Eng., Vol. 15, 981-990, 1980.

[Mang81]: Mang, H. A. and Gallagher, R. H., "Finite Element Analysis of Thin Shells of General Form for Displacement Dependent Loads", Nonlinear Finite Analysis of Plates and Shells, The winter annual meeting of ASME, Washington, D. C., Nov. 15-20, 1981, edited by Hughes, T. J. R., Pifko, A. and Jay, A.

[Nagtegaal81]: Nagtegaal, J. C. and DeJong, J. E., "Some Computational Aspects of Elastic-Plastic Large Strain Analysis", Int. J. for Numerical Methods in Eng., Vol. 17, 15-41, 1981.

[Powell81]: Powell, G. and Simons, J., "Improved Iteration Strategy for Nonliear Strucures", Int. J. for Numerical Methods in Eng., Vol. 17, 1455-1467, 1981.

[Rheinboldt83]: Rheinboldt, W. C. and Riks, E., "Solution Techniques for Nonlinear Finite Equations", State-of-the-art Surveys on Finite Element Technology, 1983, chapter 7, Sponsored by ASME, Edited by Norr, A. K. and Pilkey, W. D.

[Shah86]: Shah, J. J., Busby, H. R. and Kinzel, G. L., "Geometry-Based Stiffness Parameters for Predicting Approach of Limit Points in Finite Element Studies", Trans. of ASME, Vol. 108, 462-468, 1986.

[Wood78] Wood, R. D. and Schrefler, B.,"Geometrically Nonlinear Analysis - A Correlation of Finite Element Notations", Int. J. for Numerical Methods in Eng., Vol. 12, 635-642, 1979.

This page is intentionally blank.

Chapter 18

BUCKLING ANALYSIS

UAI/*NASTRAN* provides the capability for the analysis of certain types of geometric nonlinear problems using the ***differential stiffness*** approach that is described in Chapter 17. This chapter describes how this approach applies to Buckling analyses. It describes how such analyses are performed and their solution results interpreted. Finally, two tutorial examples are presented.

Buckling Analysis using the differential stiffness approach is performed using Rigid Format 5.

18.1 TERMINOLOGY

Buckling can be defined as the sudden large deformation of a structure due to a slight increase of an existing load under which the structure had exhibited little, if any, deformation before the load was increased. For many structures, the ***critical buckling load*** which is the maximum load for which the structure is in stable equilibrium, occurs at stress levels much less than the proportional limit for the material. For these types of structures, the ***material nonlinearities*** can be ignored, and ***only*** the ***geometric nonlinearities*** need to be considered in computing the buckling load.

Since there are different ways by which the new equilibrium configuration may be reached, buckling can be classified by the use of proper adjectives. The ***classical or bifurcation buckling***, which has been most extensively studied, is characterized by the fact that, as the load passes through its critical stage, the structure passes from its unbuckled equilibrium configuration to a buckled equilibrium configuration with an infinitesimally small increment in load. Buckling of long columns, thin plates loaded by in-plane forces, and buckling of rings are classical examples of this kind of buckling.

For some structures, the loss of stiffness after buckling is so great that the buckled equilibrium configuration can only be maintained by returning to a lower load level. This type of buckling is known as ***finite-disturbance buckling***. Examples of this type are buckling of thin cylindrical shells under axial compression, and buckling of complete, spherical shells under uniform external pressure.

A third type of buckling known, as ***snapthrough buckling*** or ***oil-canning***, is characterized by a visible and sudden jump from one equilibrium configuration to another equilibrium configuration with large displacements. These are called ***nonadjacent*** equilibrium states. Examples of this type are snapping of a low pinned arch under lateral loads, and snapping of clamped shallow spherical caps under uniform lateral pressure.

The **UAI*/NASTRAN*** approach to Buckling Analysis is based on the differential stiffness theory explained in Chapter 17. An important assumption of this theory is that the external loads do not change in magnitude or direction as the structure deflects, but instead, they travel with the GRID points. As a result, the buckling of ***nonconservative*** systems, such as the ***follower-force*** problem, may not be solved using this approach. In addition, only the bifurcation buckling and finite-disturbing buckling problems can be solved using Rigid Format 5.

Snap-through buckling analysis requires a complete geometric nonlinear analysis using the procedures described in Chapter 17.

18.2 MATHEMATICAL BACKGROUND

The linearized buckling problem is solved in two steps. First, the linear structural response is computed from:

$$\boldsymbol{K}_{ll}\boldsymbol{u}_{l} = \boldsymbol{P}_{l} \tag{18-1}$$

Once the current deformations have been determined, the differential stiffness matrices for each of the elements, $\boldsymbol{k}_{e}^{d}$, are computed. The standard assembly procedure is then used to assemble the complete g-size differential stiffness matrix:

$$\boldsymbol{K}_{gg}^{d} = \mathbf{A}\ \boldsymbol{k}_{e}^{d} \tag{18-2}$$

This is followed by performing all of the partitioning operations to $\boldsymbol{K}_{gg}^{d}$ to arrive at the solution size matrix, $\boldsymbol{K}_{ll}^{d}$.

$$\left[\boldsymbol{K}_{ll} + \lambda \boldsymbol{K}_{ll}^{d}\right]\boldsymbol{u}_{l} = \mathbf{0} \tag{18-3}$$

where $\boldsymbol{K}_{ll}$ is the linear stiffness matrix, $\boldsymbol{K}_{ll}^{d}$ is the differential stiffness matrix, λ are eigenvalues, and $\boldsymbol{u}$ are vectors of normalized buckling shapes associated with the eigenvalues. The eigenvalues represent the load factors that will cause buckling.

Buckling Analysis requires a static structural model, one static loading condition, and an eigenvalue analysis to determine the buckling factors associated with that loading condition. Other static load cases and determination of their buckling factors must be performed in separate executions.

There are some important properties of the differential stiffness matrix, $\boldsymbol{K}^{\boldsymbol{d}}$, that you should note. $\boldsymbol{K}^{\boldsymbol{d}}$ is ***symmetric***, and is independent of elastic properties, and it depends only on the element geometry, displacement field, and the state of stress. The overall structural stiffness with respect to the buckling modes becomes zero at the critical loads. In general, $\boldsymbol{K}^{\boldsymbol{d}}$ is indefinite, and it cannot be inverted.

18.3 INPUT DATA REQUIREMENTS

This section describes the input data requirements for Buckling Analysis using **UAI/*NASTRAN***.

18.3.1 Executive Control Commands

In **UAI/*NASTRAN***, Buckling Analysis is requested by using Rigid Format 5. This is selected with the Executive Control command:

```
SOL 5 (Buckling Analysis)
```

18.3.2 Case Control Commands

The Case Control packet for buckling analysis ***must*** contain two subcases. One static loading condition ***must*** be defined as the first subcase, and a **METHOD** command, which selects an **EIGB** Bulk Data entry, ***must*** appear in the second subcase. Only one set of boundary constraints may be defined, and they must be defined prior to the **SUBCASE** commands. Output requests are made in the usual manner.

18.3.3 Bulk Data Entries

Buckling behavior may be computed only for those elements that support differential stiffness. The only special Bulk Data entry for Buckling Analysis is the **EIGB** entry:

EIGB	SID	METHOD	FL	FU		ND		E	-cont-
-cont-	NORM	GID	DOF						

The **SID** field is the identification number that is referenced by the **METHOD** Case Control command. The second field specifies the eigenextraction **METHOD**. You may select from two methods, **SINV**, the inverse power method with Sturm Sequence checks, or **LANCZOS**. You use the fields **FL** and **FU** to specify the range of load factors in which you are interested. When using the **LANCZOS** method, you may specify the parameter **ND** which defines the number of lowest, i.e. critical, positive eigenvalues to be extracted. Finally, you may specify the normalization method for the buckling mode shapes. Generally, these are only normalized to the maximum value, in which case you do not need the continuation entry.

18.4 SOLUTION RESULTS

Print and plot output from Buckling Analysis is requested in the usual manner using Case Control commands ***except*** that output for the linear statics solution is requested in the first subcase, and output for the buckling solution is requested in the second subcase. An example of the Case Control commands required to do this is shown below:

```
SET 10 = 1,32
SUBCASE 1
   LABEL = STATICS SOLUTION
   LOAD = 100
   OLOAD = 10
SUBCASE 2
   LABEL  = BUCKLING SOLUTION
   METHOD = 100
   SPCFORCE = 10
BEGIN BULK
```

The buckling output consists of requested data, such as displacements or stresses, printed for each eigenvalue or buckling factor computed. The load factors are presented in tabular form as shown in Table 18-1. The eigenvectors are the normalized buckling shapes. They are measured relative to a specific degree of freedom or the maximum value depending on the option you specify with your **EIGB** Bulk Data entry. These deformations are ***not*** actual post-buckling deflections.

Table 18-1. BUCKLING LOAD FACTOR SUMMARY

B U C K L I N G L O A D F A C T O R S

MODE NO.	EXTRACTION ORDER	LOAD FACTOR
1	3	7.766561E+00
2	1	7.086994E+01
3	2	2.019339E+02

18.5 MODELING GUIDELINES

This section addresses the analysis and modeling aspects of buckling using **UAI/*NASTRAN***. The most commonly used element types are supported for Buckling Analysis capability. Chapter 17 presents a list of elements available for solution with differential stiffness analysis.

Buckling Analysis in **UAI/*NASTRAN*** can be applied to the classical bifurcation buckling problems as well as finite-disturbance buckling problems. With this capability, you can model many practical structures such as framed multi story buildings, bridges, automobile chasis and body shell, rocket shells, and so on.

There are certain facts you should be aware of in the finite element modeling of buckling problems. The most important is that, ***buckling of a structure does not imply buckling of each element***.

Many axisymmetric structures such as shells of revolution, produce nonaxisymmetric buckling modes even when the loading is axisymmetric The results of Buckling Analysis using **UAI/*NASTRAN*** are presented as buckling load factors, and the corresponding eigenvectors represent buckling modes. For classical bifurcation buckling problems, the ***first*** buckling mode corresponds to the lowest buckling factor, followed sequentially by the higher order buckling load factors and buckling modes. However, this is ***not*** always true for finite-disturbance buckling problems. In the analysis of shells of revolution for example, the lowest buckling mode computed may have more than one ***half sine wave***. In the classical column buckling sense, this would correspond to the second or subsequent buckling modes. You must plot the mode shapes of the buckled structure corresponding the lowest buckling load factor, to get the lowest buckling mode. As you will see in the following analysis example, the lowest mode of buckling computed, corresponds to a full sine wave pattern of buckled shape.

18.6 PERFORMING BUCKLING ANALYSES

This section presents two example problems which illustrate the methods used to perform Buckling Analyses and interpret the solution results.

18.6.1 Euler Buckling

The first example problem solves a very simple Euler buckling problem by simulating a beam with an assemblage of plate elements.

Example Problem 18-1

A rectangular plate 50 in long and 5 in wide is clamped along one short edge and a compressive load of 2 lb is applied to the free edge. You wish to determine all buckling load factors and shapes up to 25,000 lb.

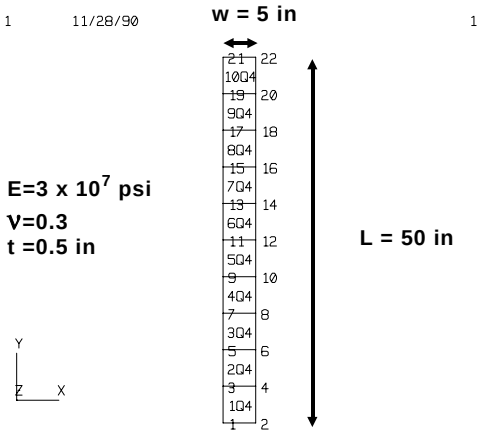
The finite element model and its physical characteristics are shown in Figure 18-1. This model, found in the file **BUCK1**, is comprised of 10 QUAD4 elements. The applied load is given a unit value since the resulting factor will determine the critical buckling load. From beam theory, see [Popov68], the exact loads may be determined from:

$$P_{cr} = n^2 \frac{\pi^2 EI}{4L^2} \quad n = 1, 3, \ldots$$

As indicated previously, you must define two subcases. The first subcase specifies the static loading, and the second the eigenvalue extraction data. For this problem, the following **EIGB** Bulk Data entry is used:

EIGB	**300**	**LANZCOS**	**0.0**	**25000.**					

Figure 18-1. EULER BUCKLING MODEL

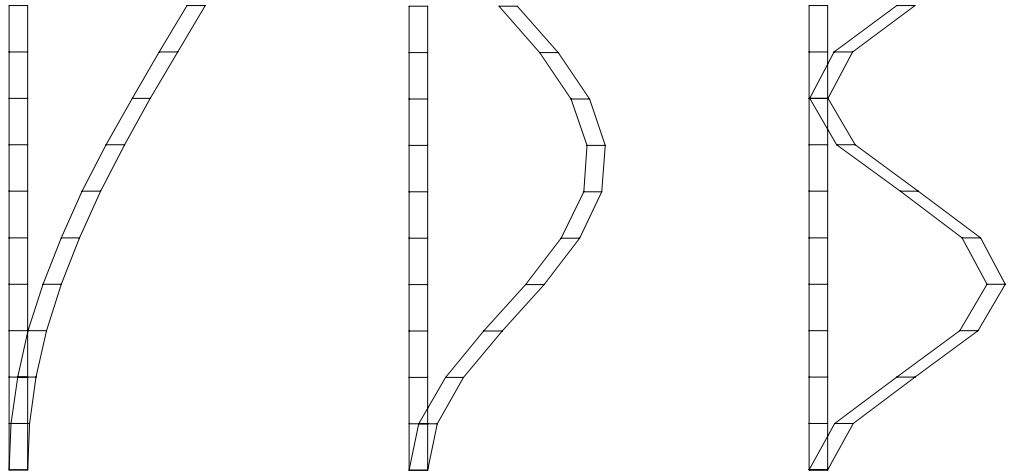


Note that you select the critical load range on the **EIGB** entry. The first three buckling mode shapes are shown in Figure 18-2. The table below compares the **UAI/*NASTRAN*** load factors with the theoretical beam results.

MODE	THEORY	UAI/*NASTRAN*
1	771.0	775.9
2	6940.0	7028.8
3	19280.0	19787.3

The agreement is very good with less than a 5% difference even for the third mode.

Figure 18-2. EULER BUCKLING MODE SHAPES



18.6.2 Cylindrical Shell Buckling

The second example problem performs the buckling analysis of a cylindrical shell subjected to axial compressive load. The cylindrical shell selected is short with large radius so that axisymmetric buckling modes will result due to the applied axial compressive load.

Example Problem 18-2

You wish to find the buckling loads and mode shapes for the first four modes of buckling of a short cylindrical shell of internal radius 80 in, height 50 in and wall thickness of 2.5 in, as shown in Figure 18-3. Assume that the top and bottom edges are constrained with respect to radial movements.

The finite element model and its physical engineering data are shown in Figure 18-3. Because the structure is axisymmetric, only a 15° section is modeled, and symmetric boundary conditions are applied. The input data for this problem is available in file **BUCK2**. Quadrilateral shell elements, QUAD4, are used in the model.

Again, you must specify two subcases in the Case Control packet for buckling problems. The static solution loads and output requests are in the first subcase, and the buckling analysis eigenextraction selection and output requests appear in the second subcase.

The top and bottom edges are constrained in u_r. All points are constrained in u_θ, θ_r, and θ_z. A compressive load of 2400 lb is applied to the whole cylinder along both the top and bottom edges. The load for the modeled segment, *P*, is thus 100 lb.

The extraction of buckling modes is controlled with the **EIGB** Bulk Data entry:

EIGB	**300**	**LANCZOS**	**0.0**	**250.0**					

The region of interest was specified as **0.0** to **250.0**. The example problem data file is then executed in **UAI/*NASTRAN***.

A theoretical solution for axisymmetric buckling of a thin shell is presented in [Timoshenko51], the number of half-waves which occur when the shell buckles at minimum load is given as:

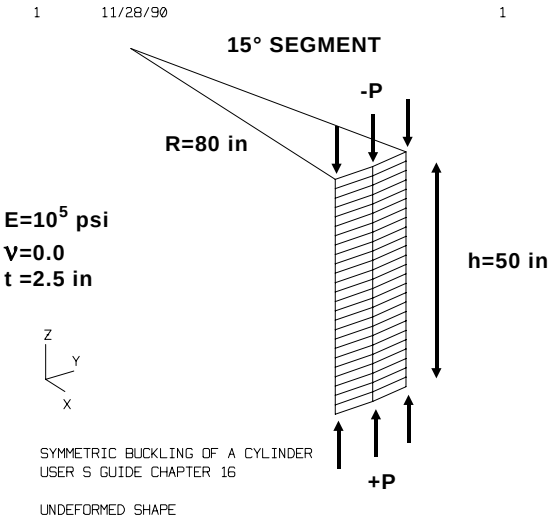
$$m = \frac{h}{\pi}\left[\frac{12(1-\nu^2)}{R^2 t^2}\right]^{\frac{1}{4}}$$

where *m* is the closest integer to the right-hand side values. The corresponding critical stress is:

$$\sigma_{cr} = \frac{E t^2 m^2 \pi^2}{12 h^2 (1-\nu^2)} + \frac{E h^2}{R^2 m^2 \pi^2}$$

For the given problem, the number of half-waves, *m*, at the minimum buckling load of the structure from the

Figure 18-3. CYLINDRICAL SHELL MODEL



above equation, is, 2. The corresponding critical stress, σ_{cr}, is 224.1 Ksi. The results of finite element Buckling Analysis from **UAI/*NASTRAN*** are output as buckling load factors, *F*, which are multipliers of the axial load imposed on the structure. To compute the theoretical load factor for mode *i*, the relationship:

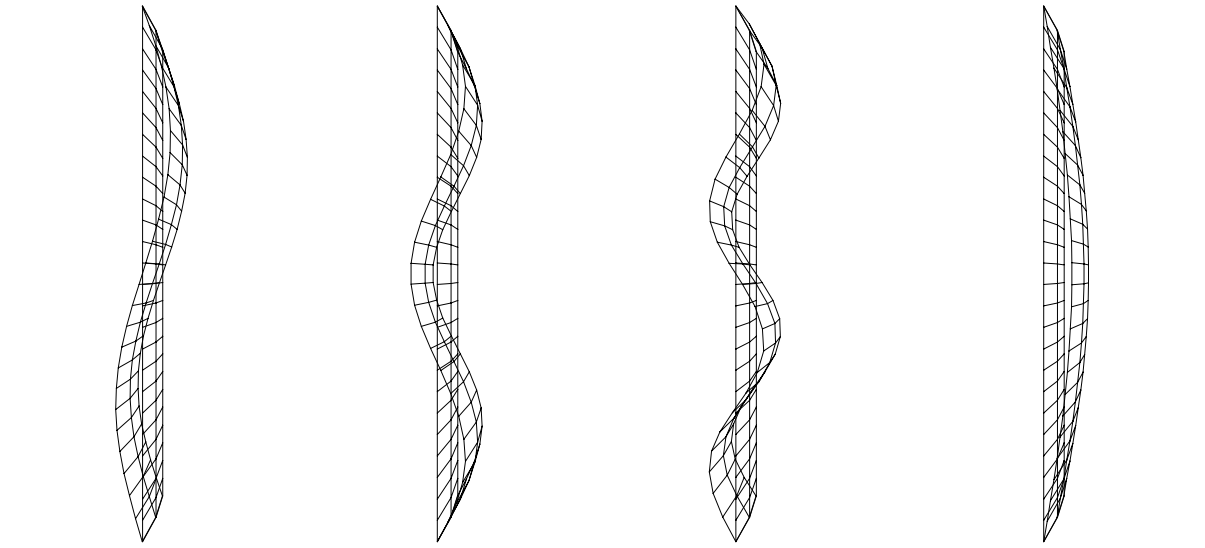
$$F_i = \left(\frac{360.0}{15.0}\right)\frac{(\sigma_{cr})_i A}{P} = \left(\frac{360.0}{15.0}\right)\frac{(\sigma_{cr})_i \pi (80.0^2 - 77.5^2)}{P}$$

can be used. The fraction represents the multiplier needed to compute the critical load for the entire cylinder given the results for the 15° segment. A comparison of the computed **UAI/*NASTRAN*** load factors and the theoretical factors are presented in the table below.

NUMBER OF HALF WAVES *m*	LOAD FACTOR, *F*	
	THEORY	UAI/*NASTRAN*
1	214.6	217.9
2	93.4	94.4
3	118.0	119.6
4	182.3	185.3

Note that the lowest buckling mode occurs for two half waves. It can be seen that the results from the **UAI/*NASTRAN*** solution are within 3% of the theoretical solution for all modes. The buckling mode shapes corresponding to these loads are shown in Figure 18-4.

Figure 18-4. CYLINDRICAL SHELL BUCKLING MODES



Chapter 19

SUBSTRUCTURING ANALYSIS

The Substructuring Analysis feature of **UAI/*NASTRAN*** provides you with a sophisticated technique for modeling large and complex structures. Substructuring allows you to analyze complex structures from models of smaller, simpler components. For example, the finite element idealization of the design model (e.g. CAD drawings) of a complete automobile, aircraft or spacecraft may be built from the relatively separate and independent finite element models of different parts such as an automobile frame, a hood, an aircraft engine or wing, or a spacecraft solar array or antenna.

Thus, substructuring provides a convenient environment for different, geographically separated engineering organizations to participate in the development of complex system models without rigorous pre-planning and coordination for the complex model. Structural parts which are duplicated in the structure need only be modeled once, thus simplifying the time-consuming tasks of model building, checkout, correction, and tuning. The single finite element model of such a part is assigned a substructure name and may then be replicated for use any number of times throughout the model. These copied parts are given their own unique substructure names, and they may be translated, rotated and reflected for use elsewhere in the model.

Substructuring is controlled by English language commands which are called ***Substructuring Commands***. Commands include those used to create and duplicate substructures, to combine two or more substructures, and to perform either static or modal reductions on substructures. Ultimately, you construct a system model and solve the resulting equations of motion for static response, dynamic response or eigenvalue extraction. Finally, you may perform detailed data recovery for any or all of the original finite element model parts.

Rev: V20.1

19

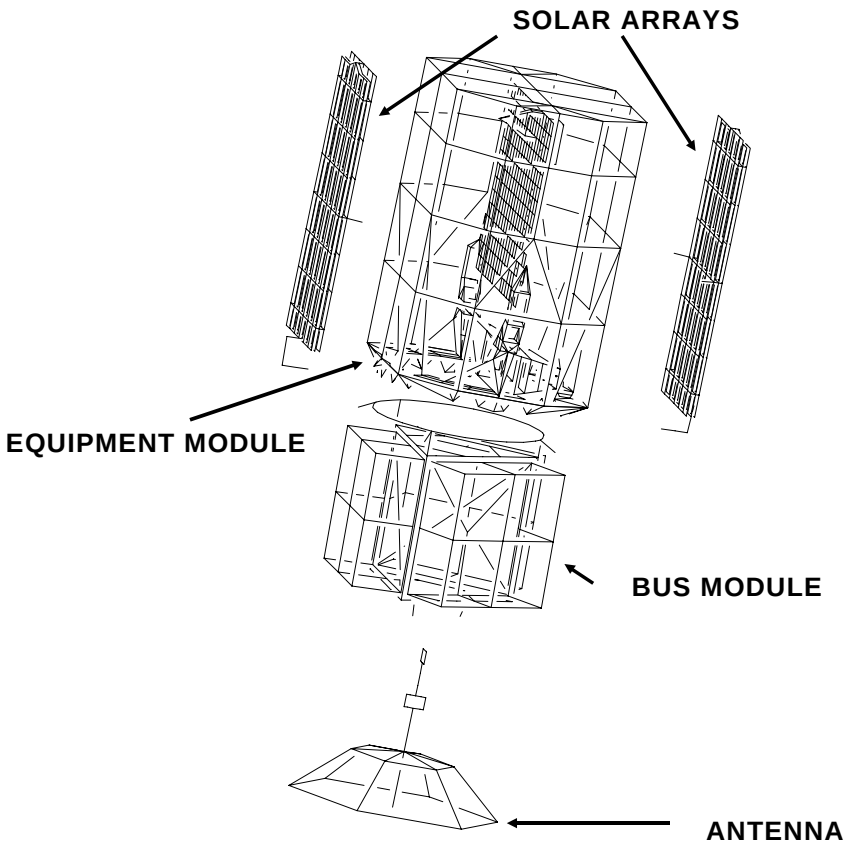
Advantages to using substructuring techniques include: ease of modeling and graphical display of component substructures compared with manipulating very large system models; the ability to *debug* the finite element model while working on one part at a time; the elimination of the need to model portions of the structure which are repetitive; the ability to assemble a complex model after the components have been developed by multiple engineering organizations; and, finally, the partitioning of the computer processing task into several smaller, easier to control steps which require smaller blocks of computer processing time than large monolithic models.

The database technology which is used by the Substructuring system allows you to manage your disk space requirements in an efficient manner. Based on your computing system resources, you may make selections to trade disk space for CPU time for certain operations. You may also elect to store portions of your data on magnetic tape when not needed.

Substructuring is available in **UAI/*NASTRAN*** for statics, normal modes, direct and modal frequency response, and direct and modal transient response structural analysis. The procedure is implemented using the standard Rigid Formats. Special DMAP ALTERS are then automatically generated based on your substructuring commands.

A typical substructuring application is illustrated in Figure 19-1. This application shows a spacecraft consisting of a *bus module*, an *equipment module*, a *high gain antenna* and two *solar arrays* configured for a launch condition. In this case, only one of the solar arrays is actually modeled and represented with a **UAI/*NASTRAN*** input file. The other solar array is created with a Substructuring command and then translated and reflected across the spacecraft body at the time of model assembly — this is how structures are synthesized from their smaller components.

Figure 19-1. TYPICAL SUBSTRUCTURING APPLICATION



19.1 COMPARING SUBSTRUCTURING AND SUPERELEMENTS

There are two basic approaches to solving large models in a partitioned manner. One, developed in the early 1970's is called Superelements, and the other, with roots traceable to the early 1960's is Substructuring. Both of these methods had the same basic goal: to solve large problems that were beyond the computing capabilities of the host computer. However, the philosophical basis of the approaches is very different.

19.1.1 Superelements — Simple Partitioning

In the original superelement approach, you defined your entire model and then specified how it should be partitioned in some number of smaller pieces that are more computationally tractable. As a result, you must create the complex model and use unique numbering systems for all of its GRID points and elements. Close coordination and careful planning are needed to create the complex model in such a way as to allow its disassembly.

19.1.2 Substructuring — Structural Synthesis

Unlike superelements, the origins of substructuring are based as much on the management of large, multi-organizational design and analysis tasks as on computing efficiency. The U.S. space program required that many contractors work on single launch and payload systems. It was then necessary to couple the analytical models developed by each of them into a single system model. The substructuring method was done manually — each organization providing stiffness, mass and load data to the contractor responsible for system integration. In 1973, NASA funded the development of Automated Multi-Stage Substructuring in NASTRAN to make these procedures more efficient. The axioms for the design of this capability included:

- Independent development of models
- Elimination of restrictions on numbering schemes and coordinate systems
- Unlimited multi-level combinations of substructures

As a result, substructuring more closely resembles the process model of large-scale finite element analysis. In addition to its role in large-scale analysis, used properly, it can be a powerful management tool to facilitate the development of analytical models by multiple subcontractors, even if such organizations are geographically separated.

19

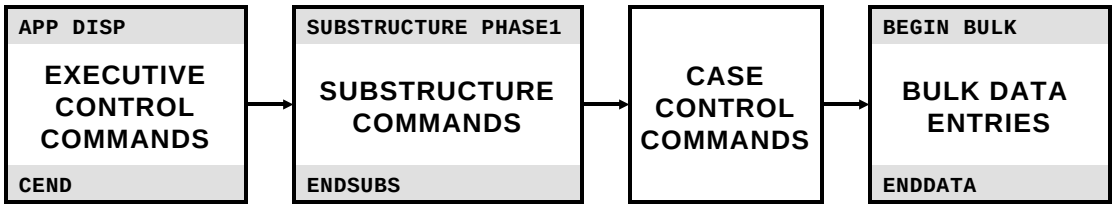
19.2 BASIC CONCEPTS AND TERMINOLOGY

This Section describes the basic concepts, control mechanisms, and terminology used when performing Substructuring Analyses.

19.2.1 Analysis Control

You request and control Substructuring Analyses by placing Substructure Commands in your input data stream in the Substructure Command packet, which immediately following the Executive Control Packet as shown in Figure 19-2.

Figure 19-2. SUBSTRUCTURE CONTROL PACKET LOCATION



You use Substructure Commands to control all substructure operations. Using these commands, you assign names to your substructures. These names may contain as many as eight characters. Proper use of these commands is described later in this chapter, and detailed instructions on each command are found in Chapter 3 of the ***User's Reference Manual***. Table 19-2 provides a summary of the available commands.

UAI/*NASTRAN* processes the Substructure Control packet and develops a DMAP ALTER which is based on the commands used and on the Rigid Format selected in the Executive Control packet. There are no special Rigid Formats for Substructuring. Table 19-1 defines which Rigid Format is used for each analysis.

The automated substructuring system stores data on a separate ***eBase*** database called the ***Substructure Operating File***, or SOF. The SOF may be exported in either binary or character form. This allows you to move data between different computers. SOF databases may be merged using techniques described later, and specific model data may be imported to the SOF from external sources or exported from the SOF for external use with another computer program.

To effectively employ Substructuring, you must have an overall understanding of the terminology, the basic approach, the database used, and the control functions provided. These topics are discussed in the sections which follow. Suggestions, recommendations, and cautions to be observed when using substructuring are also given.

19.2.2 Basic Substructures

Substructuring analysis begins with the creation of finite element models of parts, or components, of a complete system model. These components are called ***Basic Substructures***. Each model part must be created and stored in a separate **UAI/*NASTRAN*** input data file from the others. These input files are processed by **UAI/*NASTRAN***, converting the finite element model

data into matrix form. These matrices, and other required table descriptions of these data, are then written to the SOF. This collection of data defines the basic substructure. Basic substructures may then be combined to build more complex substructures which, in turn, can be progressively combined with other substructures to eventually arrive at a final structural model.

19.2.3 Substructure Naming Conventions

To simplify data entry, all specific references to GRID points and loadings for Substructuring Analyses are made by referring to the basic substructure name. For this reason, you may not use a component substructure more than once while building the solution structure. That is, every component substructure name must be unique. If you wish to use the same component substructure more than once, then you must use the equivalence operation to replicate the substructure and assign a new name to it.

19.2.4 Image Substructures

Image Substructures are created when a complex substructure is replicated using the **EQUIVALENCE** command. Those substructures which would logically precede the equivalenced substructure are automatically created and are called image substructures. This is explained in more detail, along with an example, in a later Section. Any substructure which is not an image substructure is called a ***Primary Substructure***.

19.2.5 Substructure Tree Diagram

A diagram which shows the logical relationship between all substructures, including substructure names, which operations are to be performed on which substructure, and in what order, is called a ***Substructure Tree Diagram***.

The Substructure tree diagram is useful for configuration management of the analysis, scheduling the computational steps, and understanding the data flow through the system. Figure 19-3 illustrates a system model to be constructed from three separate basic substructures. Figure 19-4 presents a Substructure tree diagram which defines all the operations and substructures created during the process of building the final model named **I**.

Basic substructures are created at the PHASE 1 level. Substructures **A**, **B**, and **E** are shown in solid boxes to indicate they are physically different models. The shaded boxes are image substructures which are created during an equivalence operation. These image substructures do not require a PHASE 1 input file and they are never executed in **UAI/*NASTRAN***. The equivalence operation defines a new substructure which is a duplicate of an existing substructure *and automatically creates all equivalent lower level component substructures.* Note that the term *lower level* refers to the less complex of the component substructures which are used to create a *higher level,* or more complex, substructure. Since the substructure created by an equivalence operation requires very little space on the SOF, this feature reduces disk space requirements as well as data preparation time.

Image substructures are completely independent substructures from their parents. Naturally, however, they are physically identical in size, have the same GRID point and element labeling, and have the same load set definitions as their parents. Note that in Figure 19-4, the names of the image substructures are identical to the equivalent parent substructure, with the exception of either a prefix character or a suffix character. The new names are created automatically by **UAI/*NASTRAN*** using the alphabetic letter specified by the **PREFIX** or **SUFFIX** subcommand of the **EQUIVALENCE** command. You use these new names to reference the appropriate compo-

19

Figure 19-3. SUBSTRUCTURES FOR TREE DIAGRAM

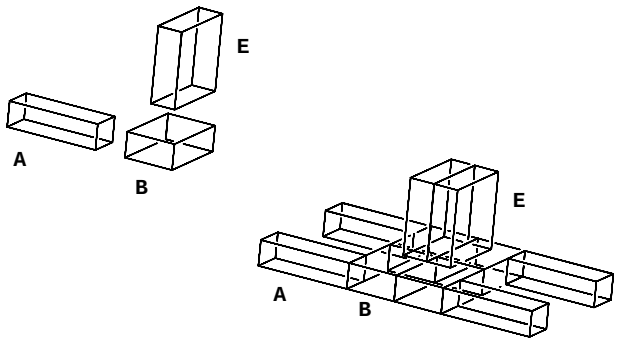
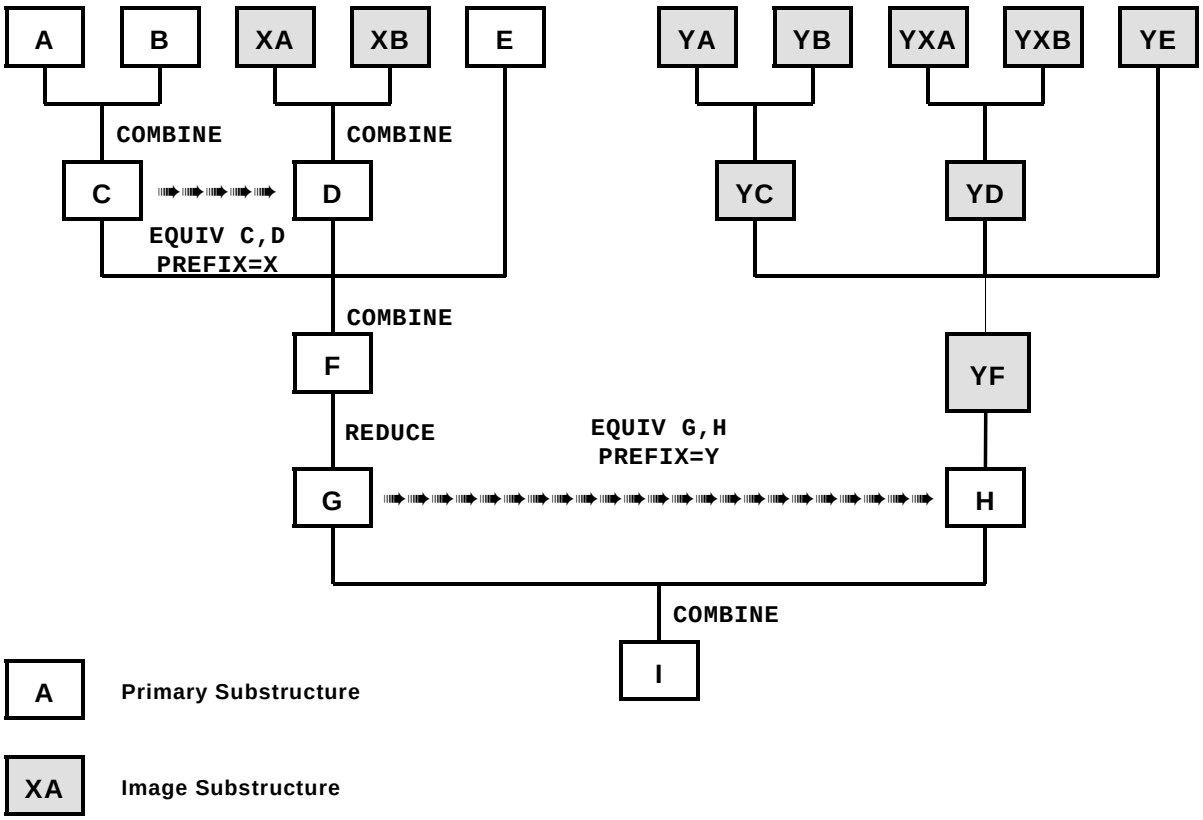


Figure 19-4. SUBSTRUCTURE TREE DIAGRAM



nent substructure as if it had been created independently with a Bulk Data packet and executed in a PHASE 1 job.

All substructures shown in Figure 19-4, with either solid or shaded boxes, are separate and distinct substructures. They may have different applied loads, boundary conditions, and responses. For example, though only **A**, **B** and **E** represent actual PHASE 1 executions, PHASE 3 data recovery may be made for **A**, **B**, **E**, **XA**, **XB**, **YA**, **YB**, **YXA**, **YXB**, and **YE**, each of which generally would have different results.

Although you need only generate a model for A and not for XA, YA or YXA, the A data must include all possible distinct loads which will be applied to any of its image substructures. These loads are assigned separate load identification numbers in separate SUBCASEs during PHASE 1 processing for **A**. Likewise, any degrees of freedom which will represent distinct boundary conditions for image substructures of **A** must also be retained as boundary degrees of freedom when processing **A** or its descendents. When you later perform a solution, all of the loads and boundary conditions will then be available for use.

19.2.6 Substructure Phases

Each **UAI/*NASTRAN*** execution of the automated substructuring system is classified by a substructure phase. There are three phases of substructure analysis:

- PHASE 1 - ***Substructure Generation***. Processes finite element data to create a basic substructure
- PHASE 2 - ***Substructure Assembly and Solution***. Performs operations such as **COMBINE**, **REDUCE** and **SOLVE**.
- PHASE 3 - ***Substructure Data Recovery.*** Performs complete *g-set* GRID point displacement and element data recovery for any basic substructure

19.2.6.1 PHASE 1

A PHASE 1 **UAI/*NASTRAN*** execution is performed for each basic substructure that you create. The PHASE 1 data represent a complete finite element model analogous to any normal structural model. Generally, permanent boundary conditions are specified and load sets are defined. It is possible, however, for certain modeling characteristics such as loads and boundary conditions to be added or modified at a later time in the analysis cycle.

Depending on the Rigid Format, all appropriate matrix partitioning and reduction is performed to reach the equations of motion at the *a-set* level. Therefore, at the completion of a PHASE 1 execution for substructure *i*, the matrices $\boldsymbol{K}^{i}_{aa}$, $\boldsymbol{M}^{i}_{aa}$, $\boldsymbol{B}^{i}_{aa}$, $\boldsymbol{K}^{4i}_{aa}$, $\boldsymbol{P}^{i}_{a}$ are created. These are then placed on the SOF database for later use in other substructuring operations.

The *a-set* degrees of freedom are, as usual, those which are retained during the reduction processes. In substructuring, these are called ***boundary dof***. If you plan to combine substructures, all points that you wish to have connected must be contained in the *a-set*. This is extremely important when using the dynamic reduction (**DYNRED**), **AUTOOMIT**, or **AUTOSPC** features. You must explicitly select the physical *a-set* degrees of freedom required for any later **COMBINE** operations so that they will be retained. This is done using the Case Control command:

```
BOUNDARY = sid
```

The set identification number, *sid*, references **BDYS** and **BDYS1** Bulk Data entries which define the boundary, or retained, degrees of freedom.

Basic substructures may be developed with independent numbering conventions for GRID points, elements or coordinate systems of other substructures. For example, all basic substructures may contain a GRID point with identification number 1. The most important consideration for modeling a basic substructure is the proper location in space of GRID points which will ultimately connect with similar GRID points of another basic substructure. The relative orientation in space of connecting GRID points must be maintained between all basic substructures. However, this does not mean that the GRID point identification numbers or numbering sequence must be the same between substructures. It is not necessary that the same basic coordinate system be used when developing basic substructure models. Any differences between basic coordinate system orientation can be resolved when substructures are combined. Furthermore, there are no restrictions on the use of local coordinate systems and their identification numbers when constructing basic substructure models. All differences in coordinate systems can also be resolved when substructures are combined.

19.2.6.2 PHASE 2

PHASE 2 executions **COMBINE**, **REDUCE**, and **SOLVE** structures. These operations may be performed in one or more executions, each of which accesses the same SOF database. Any number of **COMBINE** or **REDUCE** operations may be performed in one execution; however, only one solution, or **SOLVE** operation, may be performed in a single execution. Initial data recovery for substructures must be performed in PHASE 2. However, the PHASE 2 operation recovers only the *a-set* displacements for basic substructures. Therefore, a PHASE 3 recover must be performed if *g-set* displacements and element stresses and forces are desired for any basic substructure. For ease-of-use, you may also perform all PHASE 2 operations except **SOLVE** and **RECOVER** in a PHASE 1 execution.

The PHASE 2 executions are most often used to create substructures from assemblages of basic substructures or other complex substructures previously created. Each combine operation allows you to create a new substructure from as many as seven component substructures. Each of the component substructures may be translated, rotated, or symmetrically reflected prior to combination. The equivalence operation allows you to create new substructures which are identical to other existing ones.

19.2.6.3 PHASE 3

PHASE 3 executions are used to recover element stresses and forces and all *g-set* motions of a PHASE 1 model. A separate PHASE 3 execution is made for each basic or image substructure for which data recovery is desired. It is also possible to recover the results for all equivalent basic and image substructures in the same PHASE 3 execution. For example, in Figure 19-4, substructures **A**, **XA**, **YA**, and **YXA** may all be processed in the same PHASE 3 job.

19.2.7 The Substructuring Disciplines

The substructuring approach may be used in all of the analytical disciplines shown in Table 19-1.

Table 19-1 RIGID FORMATS FOR SUBSTRUCTURING ANALYSIS

ANALYSIS DISCIPLINE	RIGID FORMAT COMMAND
LINEAR STATICS	SOL 1
LINEAR STATICS (FREEBODY)	SOL 2
NORMAL MODES	SOL 3
DIRECT FREQUENCY RESPONSE	SOL 8
MODAL FREQUENCY RESPONSE	SOL 11
DIRECT TRANSIENT RESPONSE	SOL 9
MODAL TRANSIENT RESPONSE	SOL 12

19

19.3 THE SUBSTRUCTURING OPERATIONS

The operation and control functions of Substructuring are best illustrated by example. Before beginning a major substructuring effort, the strategy to be used should be outlined. This includes the definition of the basic substructures and the manner in which these will be reduced and combined to yield the complete solution model. This outline can be used to create a Substructuring Tree Diagram. The tree is useful for configuration management of the analysis, scheduling the computational steps, and understanding the data flow through the system.

The substructuring system provides a great deal of flexibility with its command language. A brief overview of the major functions is presented below. All of the commands are summarized in Table 19-2 and described in detail in Chapter 3 of the ***User's Reference Manual***.

Table 19-2 SUMMARY OF SUBSTRUCTURE COMMANDS

COMMAND TYPE	COMMAND	DESCRIPTION
CONTROL COMMANDS	ENDSUBS	End of substructure control
	OPTIONS	Defines matrix operations
	RUN	Controls execution mode
	SUBSTRUCTURE	Selects execution phase
SUBSTRUCTURE OPERATIONS	COMBINE	Combines substructures
	EQUIVALENCE	Creates an equivalent substructure and image substructures
	REDUCE	Performs a Guyan reduction
	MREDUCE	Performs a modal reduction
	CREDUCE	Performs a complex modal reduction
	MRECOVER	Recovers eigenvectors from a modal reduction
	SOLVE	Executes the Rigid Format solution process
	RECOVER	Recovers the Rigid Format solution results
	BRECOVER	Recovers solution results for basic substructures
	PLOT	Requests substructure undeformed plots
	MASSBAL	Requests substructure weight data
SOF OPERATIONS	CHECK	Verifies the contents of a **DUMP**ed sof
	DELETE	Removes selected data items from the SOF
	DESTROY	Removes whole substructure trees
	EDIT	Removes groups of items from the SOF
	SOFOUT/SOFIN	SOF interface to external files
	SOFPRINT	Prints SOF data items

19.3.1 The EQUIV Operation

The **EQUIVALENCE** command allows you to replicate any substructure as many time as necessary to synthesize your structure. When you perform an EQUIV operation you may use either the **PREFIX** subcommand or the **SUFFIX** subcommand. These subcommands define an additional character which is used as a prefix or suffix for all of the image substructures created during the operation. If no additional character is specified, then a default **PREFIX** of **E** is used.

You should take care when performing successive equivalence operations to monitor the growth of image substructure names so as not to exceed the eight character limit. If the limit is exceeded, the right-most character will be truncated. Therefore, it is possible for you to inadvertently create duplicate substructure names as more prefixes are added. If you do so, **UAI/*NASTRAN*** will issue an error message, and the operation will not be performed.

19.3.2 The COMBINE Operation

The **COMBINE** command, with its subcommands, is used to assemble from two to seven substructures into a higher level substructure. For example, consider the operation shown in Figure 19-5. The relationship of the degrees of freedom of the combined substructure, C, to its components, A and B, is:

$$\boldsymbol{u}_A^{old} = \boldsymbol{H}_{AC}\boldsymbol{u}_C^{new}$$

$$\boldsymbol{u}_B^{old} = \boldsymbol{H}_{BC}\boldsymbol{u}_C^{new}$$

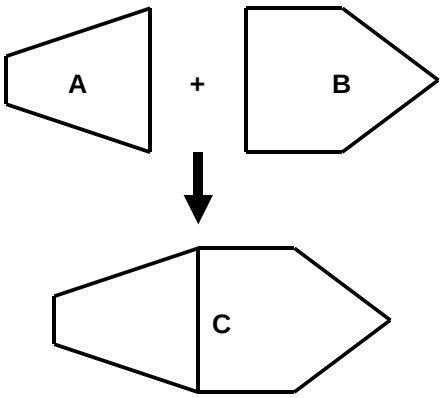
$$\boldsymbol{M}_C = \boldsymbol{H}_{AC}^T\boldsymbol{M}_A\boldsymbol{H}_{AC} + \boldsymbol{H}_{BC}^T\boldsymbol{M}_B\boldsymbol{H}_{BC}$$

$$\boldsymbol{K}_C = \boldsymbol{H}_{AC}^T\boldsymbol{K}_A\boldsymbol{H}_{AC} + \boldsymbol{H}_{BC}^T\boldsymbol{K}_B\boldsymbol{H}_{BC}d$$

$$\boldsymbol{P}_C = \boldsymbol{H}_{AC}^T\boldsymbol{P}_A + \boldsymbol{H}_{BC}^T\boldsymbol{P}_B$$

The $\boldsymbol{H}$ matrix is simply a mapping of the combined degrees of freedom to each of the component substructures. The system matrices are then computed from these components. You may also translate, rotate, or reflect substructures prior to their combination. For example, the right wing of an aircraft is first modeled and an equivalence operation is performed to define a duplicate wing. Then, when performing the combine, a **SYMTRANSFORM** operation is applied so that the second wing appears as the actual left wing, a mirror image of the right wing. A **TRANSFORM** operation is then used to properly position it on the left side of the aircraft. These operations simply result in terms of the H matrix which repre-

Figure 19-5. THE COMBINE OPERATION



sent either direction cosines, in the case of the **TRANSFORM** operation, or sign reversals in the case of the **SYMTRANSFORM** operation. Caution is advised in that the symmetry transformation, **SYMTRANSFORM**, is always applied to the component in its own basic coordinate system before the usual translation and rotation **TRANSFORM** for final positioning.

19.3.3 The Reduction Operations

Several reduction operations are available to you when performing substructuring. The **REDUCE** operation is used to perform the Guyan reduction of a substructure. In fact, you may perform this type of reduction in either of two ways. The first way, which is the most efficient, is to define the *a-set* in your PHASE 1 execution. This is, of course, done in the normal manner using **ASETi** or **OMITi** Bulk Data entries. [Note that there are special provisions for using the **AUTOSPC**, **AUTOOMIT** and Dynamic Reduction features during PHASE 1. These are described in a later section.] The second way is to perform the reduction in PHASE 2. This is the only way to perform a multilevel reduction. Using the **REDUCE** command requires that you specify all of the a-set degrees of freedom explicitly. This is done with the **BDYC**, **BDYS**, and **BDYS1** Bulk Data entries. The selected degrees of freedom are called ***Boundary*** degrees of freedom even though it is not necessary for them to be on the physical boundary of a substructure.

When performing dynamic response analyses, you may perform a modal reduction by using the **MREDUCE** and **CREDUCE** commands. This is the modal synthesis approach wherein the structure is represented by a set of its normal mode shapes and additional physical degrees of freedom. The modal reduction feature may be used when the model is represented by real symmetric mass and stiffness matrices. The complex modal reduction function provides a general modal reduction capability when damped modes are desired or when complex or unsymmetric matrices are present. Constraints may also be applied to the structure only for the purpose of calculating the modes. **BDYC**, **BDYS**, or **BDYS1** Bulk Data are used to define these degrees of freedom and are requested by the subcommand **FIXED**.

Note that for both the **REDUCE** and **MREDUCE** substructure commands, the mass and stiffness matrices, $\boldsymbol{K}$ and $\boldsymbol{M}$, the damping matrices, $\boldsymbol{B}$ and $\boldsymbol{K}^4$, and the load vectors, $\boldsymbol{P}$, are transformed to the reduced set of components. The reduced substructures may be processed with any of the other substructure operations. However, substructures generated with the complex modal reduction may not be processed with any commands requiring real arithmetic, namely **REDUCE**, **MREDUCE**, or **SOLVE** with Rigid Formats 1, 2, 3, 9, or 12.

19.3.4 The SOLVE and RECOVER Operations

Although you may use as many equivalence, combination, and reduction operations as desired in PHASE 1 or PHASE 2, only one solve operation is allowed in any single PHASE 2 execution, and the solve operation is not allowed at all in PHASE 1. The **SOLVE** command requests that a specific substructure solution be performed. Rigid Formats 1, 2, 3, 8, 9, 11, and 12 may be used for substructuring solutions. The **RECOVER** command is used to recover the solution data for successively lower level substructures. Only the displacements, forces of constraint, and applied loads can be selectively output for any component substructure with this command.

You may perform more than one solution of a given substructure, in separate PHASE 2 executions, by using the subcommand **SUBSCRIPT** of the **RECOVER** command which follows the **SOLVE**. For example:

FIRST SOLUTION	SECOND SOLUTION

...
SUBSTRUCTURE PHASE2
SOLVE BIGMODEL
RECOVER BIGMODEL
  SUBSCRIPT = 1
...
	...
	SUBSTRUCTURE PHASE2
	SOLVE BIGMODEL
	RECOVER BIGMODEL
	  SUBSCRIPT = 2
	...

The **RECOVER** command is then used in a PHASE 3 execution to obtain all of response output normally provided by **UAI/*NASTRAN*** for each desired basic substructure, and any other substructures which have been equivalenced to it. You may also merge the results of several PHASE 2 solutions when recovering the solution results for a substructure in PHASE 3.

The command **MRECOVER** is used to recover mode shapes for substructures that have been reduced by the modal method.

19.3.5 The OPTION Command

All of the substructuring PHASEs may be executed using only some of the system matrices for the specified operations. The format of this command is:

```
OPTION = opt_list
```

where *opt_list* is a list of one or more of the symbols **K**, **M**, **P**, **PA**, **B**, and **K4**. These symbols correspond to the named system matrix, with the exception of **PA** which signifies appended load vectors.

You may modify matrices or loads by rerunning the Substructuring Analysis using only the desired options. However, if you do this, you must delete any existing old data from the SOF using the **EDIT** or **DELETE** commands.

The load append option, **PA**, is used when you wish to add new loading conditions without regenerating loads that were previously created. To generate the new loads, you rerun all of the PHASE 1 models which have new loads using the:

```
OPTION = PA
```

command. You then resubmit your PHASE 2 jobs also using the **PA** option. Each of the PHASE 2 operations will result in the new load vectors being appended to the existing loads. As a result, all of the loads will be available for your solution.

19.3.6 Other Operations

The **PLOT** command requests undeformed Structural Plots in a PHASE 1 or PHASE 2 execution. Deformed Structural Plots and can be obtained from a PHASE 3 execution. Finally, X-Y Plots may be obtained either during PHASE 2 or PHASE 3.

The **MASSBAL** command provides Grid Point Weight Generator summaries at various stages of a substructure analysis. Grid Point Weight data may be requested for any substructure during PHASE 1 or PHASE 2. The subcommand **RNAME** contains the name of a reference basic sub-

structure, which must be a component of the named substructure. The subcommand **RGRID** contains the number of a reference GRID point which must be in the reference substructure. The rigid body mass properties of the substructure are calculated with respect to that reference GRID point. The **RNAME** and **RGRID** subcommands are optional. If they are not used, the origin of the basic coordinate system of the substructure will be used as the reference point.

19.4 THE SOF DATABASE

The data required for each basic substructure and for all subsequent combinations and reductions of substructures are stored on an ***eBase*** database called the SOF. The SOF database is an archival database which must reside on disk during a **UAI/*NASTRAN*** execution, but the database may also be stored on tape between runs to provide backup or for subsequent input to other computers.

The SOF allows you to perform a Substructuring Analysis in separate steps without using the checkpoint/restart feature of **UAI/*NASTRAN***. A PHASE 1 execution is required to create each basic substructure and to place its data on the SOF. This must be done prior to any combination operations using that substructure. Substructures may then be assembled in stages from these basic substructures and added later to other component substructures already on the SOF. The same SOF may contain data for more than one solution structure, or, in fact, for more than one analysis.

Like all ***eBase*** Databases, while the SOF is a single logical database, it may actually reside on many physical files as described in Chapter 2 of this manual. You must maintain the physical files on the computer from execution to execution. You should backup large SOFs to tape for protection between executions. The preferred way to do this is by using your computer system utilities. Alternately, you may use the **SOFOUT** and **SOFIN** Substructure commands.

The physical SOF file must be defined in each execution by using the **ASSIGN** Executive Control command:

```
ASSIGN  MYSOF = name_prefix [ , PASSWORD = pass ] , { NEW } , USE = SOF
                                                    { OLD }
```

The details of the parameters on this command are found in Chapter 2 of the ***User's Reference Manual***. Briefly, *name-prefix* is the prefix of the name used for several files that represent the database. The actual names are created by **UAI/*NASTRAN***. Again, more about this is found in Section 2.5. *pass* is an optional security code that allows you to protect your data. The two parameters **NEW** and **OLD** are used to indicate that the file has, or has not, been previously created.

All data stored on the SOF are referenced by their substructure names. For each substructure, various types of SOF data entities may be stored. These entities are called ITEMS, and they are accessed by their item name. Thus, the substructure name and item name are all that is required to access any block of data on the SOF. **UAI/*NASTRAN*** automatically keeps track of the data, stores the data as it is created, and retrieves the data when required. It is your responsibility to maintain the file. It must be accessible when needed. You should remove data containing errors and data that are no longer needed for subsequent analyses from the SOF. You may do this with the **DELETE**, **EDIT** or **DESTROY** commands. Also, data may be selectively stored on a backup tape or file for later retrieval if additional space is needed for subsequent operations. You should not use other programs, such as ***eShell***, to delete or modify data on your SOF. These programs do not include knowledge about the substructuring tree and will not correctly modify critical information if data are deleted.

19.4.1 The SOF Data Contents

The SOF contains all of the information created during a substructuring analysis. These data may be characterized as model definition data, substructure transformation data, and solution data.

19.4.1.1 The SOF Database Table of Contents

A directory is maintained that summarizes the data that has been saved on the SOF database. This directory is called the ***Table of Contents*** or TOC of the SOF. A TOC may be requested at any point of the analysis by using command:

```
SOFPRINT TOC
```

A typical table of contents is shown in Table 19-3. This analysis tree represents the sample problem presented in Section 19-8. The TOC contains an entry for each substructure that has been created during the analysis and its data items. These items have the four letter code names printed at the top of the TOC. There is also special information that defines the substructuring tree given at the start of the TOC. These data are shown in Table 19-4.

19.4.2 The SOF Data Entities — "Items"

The actual data entities, called ***items***, stored on the SOF fall into one of three categories: model definition data, substructure transformation data, and solution results. Although normally, the specific contents of these items are not used, you may find it necessary to verify certain data if errors are encountered. In the first category, there are sixteen items for each substructure that define the model. These, as well as the items in the other categories, are summarized in Table 19-5. The second category of items arises from the various substructure reduce and combine operations which are expressed in terms of various transformation matrices. Finally, after the last model has been created, a solution is performed and various solution data are saved on the SOF. The specific items depend on the analysis discipline that you select. If you have performed multiple solutions using the subscript feature, a summay of this information is also printed.

Table 19-3. SOF TABLE OF CONTENTS

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

SUBSTRUCTURE NO.	NAME	TYPE	SS	PS	LL	CS	HL	EQSS	BGSS	CSTM	LODS	PLTS	KMTX	MMTX	PVEC	POVE	UPRT	HORG	UVEC	QVEC	SOLN	PAPP	POAP	LOAP	LMTX	GIMS	PHIS	LAMS	K4MX	BMTX	PHIL	HLFT
1	RIGHT	B	4	0	0	2	3	2	2	1	1		4		2			3	2	2	1											
2	LEFT	B	5	1	0	1	3	2	0	0	1		0		0			3														
3	UPPER	C	6	0	2	6	7	2	2		1		4		2			3	2		1											
4	LRIGHT	IB	2	1	0	5	6	2	0	0	1		0		0			0														
5	LLEFT	IB	0	1	0	4	6	2	0	0	1		0		0			0														
6	LOWER	C	0	3	5	3	7	2	0		1		0		0			3														
7	CIRCLE	C	0	0	6	0	0	2	2		1		5		3				2		1											

SIZE OF ITEM IS GIVEN IN POWERS OF TEN (0 INDICATES DATA ARE STORED IN PRIMARY)

Table 19-4. POINTERS DEFINING THE SUBSTRUCTURE TREE

POINTER	DESCRIPTION
LL	One-way pointer from a substructure created by a COMBINE or REDUCE operation to one of its simpler components. If there is more than one component, as there is with COMBINE, these components are linked together with their CS pointers.
CS	Circular list of pointers between COMBINEd substructures.
HL	One-way pointer from a substructure to a higher level, or more complex, substructure of which it is a component.
SS	Chain of pointers starting from a primary substructure to all secondary substructures which are EQUIVed to it.
PS	One-way pointer from a secondary substructure to the primary substructure from which it was created by an EQUIV operation.

Table 19-5. SOF DATABASE ENTITIES — "ITEMS"

DATA TYPE	SOF ITEM	DESCRIPTION
MODEL DEFINITION	**EQSS**	External to internal GRID point map.
	BGSS	Basic GRID point coordinates.
	CSTM	Coordinate system transformations.
	LODS	Load set summary.
	LOAP	Appended load set summary.
	KMTX	Substructure stiffness matrix.
	MMTX	Substructure mass matrix.
	PAPP	Appended load vectors.
	PLTS	Plot request information.
	PVEC	Load vectors.
	PVGV	Load vector, *g-set*.
	PVSV	Load vector, *s-set*.
	PVOV	Load vector, *o-set*.
	YVSV	Enforced displacement vector, *o-set*.
	K4MX	Substructure structural damping matrix.
	BMTX	Substructure viscous damping matrix.
SUBSTRUCTURE TRANSFORMATION	**LMTX**	Decomposition result from REDUCE operation.
	POAP	Appended load vectors at OMITted points.
	POVE	Load vectors on OMITted points.
	UPRT	Partitioning vector for REDUCE operations.
	HORG	Substructure transformation matrices.
	GIMS	Modal reduction transformation.
	PHIL	Complex modal reduction transformation.
	HLFT	Complex modal reduction transformation.
SOLUTION DATA	**UVEC**	Solution displacements or eigenvectors.
	QVEC	Reaction force vectors.
	LAMS	Modal reduction eigenvalue summary.
	PHIS	Modal reduction transformation.
	SOLN	Solution summary data.

19

19.5 INPUT DATA REQUIREMENTS

This Section reviews the input data requirements for performing Substructuring Analyses with **UAI/*NASTRAN***. There are two distinct steps in performing a substructuring analysis. In the first step, you create your basic substructures. You then use these basic substructures as building blocks to synthesize successively larger, more complex structures. In the second step, you solve the structure for one of the analytical disciplines and recover data for the basic substructures. These two steps are described in detail in the following sections.

19.5.1 Substructure Control Commands — Building Your Structure

You build your substructuring model in several steps that generally span both PHASE 1 and PHASE 2 executions. During PHASE 1, you create basic substructures and, optionally, define loading conditions for those analytical disciplines which require them. Then, during PHASE 2, you create more complex substructures. The techniques for building your structure in this manner are described in this section.

19.5.1.1 PHASE 1 — Static Analysis

You must create each basic substructure of your model in a single PHASE 1 **UAI/*NASTRAN*** execution. The input data stream for each contains all of the geometric and engineering data necessary to define the structure and, optionally, its loads. The Substructure Control packet that you use to create a new substructure, for static analysis, is of the form:

```
SUBSTRUCTURE PHASE1
   BASIC = A
ENDSUBS
SPC = 100
SUBCASE 100
   LOAD = 100
SUBCASE 200
   LOAD = 200
```

Once these substructures have been created and placed on the SOF, you may use them as *building blocks* for other more complex substructures.

Note that the loads that you define will be used later during the PHASE 2 solution operation. Also note that you may only specify a single boundary condition in PHASE 1 because only one set of matrices is placed on the SOF for each basic substructure. This is in no way a limitation, because you may define as many boundary conditions as necessary during the PHASE 2 solution. SUPORT points should not be used during the PHASE 1 execution. If rigid body supports are required, then they should be applied during PHASE 2 with the **SUPORTS** Bulk Data entry.

You must be careful to retain all of the boundary points for the basic substructure during PHASE 1. This incudes points which will be connected to other substructures, or those to which boundary conditions will be applied. The explicit reduction specified by **ASET** or **OMIT** Bulk Data entries is the most straight-forward method of doing this. However, it may also be inconvenient. More often, you might choose to use the **AUTOSPC**, **AUTOOMIT** or Dynamic Reduction features. When using these features, the special Case Control command:

```
BOUNDARY = sid
```

may be used to identify the boundary points.

The set identification number, *sid*, references **BDYS** or **BDYS1** Bulk Data entries which define the degrees of freedom on the boundary. The format of these entries is:

BDYS	SID	GID1	DOF1	GID2	DOF2	GID3	DOF3		

BDYS1	SID	DOF	*GRID ID LIST*	-cont-

where **SID** is the set identification number, **DOFi** the DOF code specifying the degrees of freedom in the boundary set, and **GIDi** the GRID points whose *dof* are placed in the boundary set. All boundary *dof* will be ignored during the **AUTOSPC** and **AUTOOMIT** operations. In the case of Dynamic Reduction, these *dof* are placed in the *a-set*.

You may also perform a substructuring **REDUCE** operation on your basic substructure. This is done with the commands:

```
REDUCE name
   NAME newname
   BOUNDARY = bsid
   RSAVE
```

where *name* is the name of the substructure being reduced and *newname* is the name of the resulting reduced substructure. The *bsid* set specifies the *dof* that will be in the *a-set* of the reduced substructure. This identification number references the Bulk Data entry:

BDYC	BSID	*name*	SID						

This entry, which has a more general form described later, combines data defined by **BDYS** and **BDYS1** entries which have the identification number **SID**.

The difference between the usual Static Condensation and the **REDUCE** command is subtle. When you use the former, the reduction transformation $\boldsymbol{G}_o$ is not saved. Therefore, it must be recomputed later when you perform a recovery of the solution data for the basic substructure. On the other hand, when you use **REDUCE**, the transformation is saved on the SOF. As a result, you must consider the trade-off between available disk space and computing time when selecting the reduction method. Furthermore, if you **CHKPNT** the PHASE 1 job and do a **RESTART** in PHASE 3, you will also eliminate the need to recompute the transformation. However, the **CHKPNT** file itself requires significant disk space.

☞ The selection of Static Condensation or substructure **REDUCE** in PHASE 1 depends on the relative importance of disk space and CPU time at your facility.

19.5.1.2 PHASE 1 — Normal Modes Analysis

For Normal Modes analysis, the PHASE 1 input stream is similar to that of Static Analysis except that there are no SUBCASE definitions. A typical input stream is:

```
...
SUBSTRUCTURE PHASE1
   BASIC = A
ENDSUBS
SPC = 100
MPC = 200
BEGIN BULK
...
```

You may also perform a PHASE 1 reduction using either **ASET** and **OMIT** Bulk Data entries or the Substructure **REDUCE** command.

19.5.1.3 PHASE 1 — Transient Response Analysis

When performing Transient Response analyses, the PHASE 1 input stream is the same as that of a Static analysis. Only the static loads are defined. The time varying characteristics of the load are defined during the PHASE 2 solution as you will see later in this Chapter. A typical PHASE 1 input stream is:

```
...
SUBSTRUCTURE PHASE1
   BASIC = A
ENDSUBS
SPC = 100
MPC = 200
LOAD = 100
BEGIN BULK
...
```

You may also perform reductions in PHASE 1 as described for Static Analyses. In addition, you may perform real or complex modal reductions. The first of these is requested with the command:

```
MREDUCE name
   NAME = newname
   BOUNDARY = bsid
   FIXED = fsid
   METHOD = methid
   RANGE = f1,f2
   NMAX = nmodes
   RESIDUALEFFECTS
   RSAVE
```

You specify the *name* of the basic substructure and the *newname* that will be assigned to the reduction. You then must specify a boundary set, *bsid*, which defines physical degrees of freedom to be retained along with the generalized coordinates. The set identification number, *bsid*, references a **BDYC** Bulk Data entry which, in turn, references **BDYS** or **BDYS1** Bulk Data entries as shown previously. You must retain all physical degrees of freedom that will be used to connect with other substructures later. These typically represent the physical boundaries of the substructure, although this need not be the case.

You then specify a **METHOD** which references an **EIGR** Bulk Data entry. This entry defines the eigenextraction data, including either the frequency range for which eigenvectors will be computed, or the number of eigenvectors to extract. These eigenvectors will then be used as the modal coordinates which represent the reduced structure. You may use either the subcommand **RANGE** or **NMAX** to modify the frequency range or number of vectors. These values are only used if they include fewer vectors than the number actually extracted.

By including residual effects in the reduction you can greatly reduce modal truncation error. The **RESIDUALEFFECTS** subcommand results in the computation of six rigid body motions, three translations and three rotations, which represent inertia relief deflection shapes. These shapes are then added to the other generalized coordinates.

Finally, you may choose to save the lower triangular factor of the decomposed matrix used to perform the reduction. This matrix is the most costly to compute, and it requires the most disk space. You must determine the tradeoff of these resourses on your host computer. The **RSAVE** subcommand is used to save these data.

19.5.1.4 PHASE 1 — Frequency Response Analysis

As described in more detail in Chapter 13 of this manual, you may define multiple SUBCASEs when performing Frequency Response Analyses. As a result, the PHASE 1 input stream is nearly the same as it is for Static Analysis. There are two major differences. First, you may perform either of the modal reductions **MREDUCE** or **CREDUCE** on the basic substructure, and second, you may only specify static loads in each SUBCASE.

19.5.1.5 PHASE 2 — Building the Model

Once you have created basic substructures, you may build a more complex model using the basic substructures as building blocks. The operations used to accomplish this are described in this section.

The most commonly used PHASE 2 operations for building a solution structure are **EQUIV** and **COMBINE**. The syntax of the **EQUIV** command is:

```
EQUIV namep,names

  { PREFIX } = letter
  { SUFFIX }
```

where *namep* is the name of an existing substructure, called the *primary substructure*, and *names* is the name of the new *secondary substructure* which will be created as an identical copy of the primary. The data for the secondary substructure is not instantiated, that is, it is not physically copied. If the primary substructure is a pseudostructure itself, the result of **COMBINE** operations, then all of its component substructures are also equivalenced as shown in Figure 19-4. The **PREFIX** or **SUFFIX** subcommand is used to prepend, or append, a single *letter* to the names of all these component substructures so that their names will be unique within the substructuring tree.

The **EQUIV** command allows you to reuse a single substructure several times when creating a complex model. As you will see in the following discussion, each secondary substructure may be independently manipulated in space during the synthesis of a solution structure.

If you perform many levels of **EQUIV**alence, each successive prefix or suffice reduces the uniqueness of the original names that you have assigned to your substructures. If your names are too long, or they have to many common letters in them, they may become nonunique. For example, if you used the names **WINGLEFT** and **WINGRITE**, then after four **EQUIV** operations, all components of these substructures would have the same names!!

The creation of a solution structure is usually performed by combining several basic substructures. This is done with the **COMBINE** operation which is specified with the command:

COMBINE [({ MANUAL / AUTO / MATCH })] *sub_name_list*

NAME = *newname*
TOLERANCE = *tol*
CONNECT = *n*
COORD = *choice*
OUTPUT = *op1,...,optn*
COMPONENT = *cname*
TRANSFORM = *tid*
SYMTRAN = [X][Y][Z]

The **COMBINE** operation is used to connect any number of substructures specified in the *sub_name_list*. The operation may be performed either **MANUAL**ly or **AUTO**matically. A special option allows GRID points with **MATCH**ing identification numbers to be combined. When the **MANUAL** option is used, you must specify the explicit degrees of freedom that will be connected using **CONCT** or **CONCT1** Bulk Data entries. The format of these entries is:

CONCT	SID	DOF	SNAME1	SNAME2					-cont-
-cont-	GID11	GID21	GID12	GID22	*CONTINUES IN GROUPS OF 2*				-cont-

CONCT1	SID	SNAME1	SNAME2	SNAME3	SNAME4	SNAME5	SNAME6	SNAME7	-cont-
-cont-	DOF1	GID11	GID12	GID13	GID14	GID15	GID16	GID17	-cont-
-cont-	DOF2	GID21	GID22	GID23	GID24	GID25	GID26	GID27	-cont-

The **CONCT** entry allows you to specify pairwise connections between two substructures, **SNAME1** and **SNAME2** at specific components, **DOF**. The **CONCT1** entry allows you to define as many as seven substructure connections, **SNAME1** through **SNAME7**, at one time. You select these entries with the **CONNECT** subcommand of **COMBINE**.

When using the **AUTO**matic option, you must specify a **TOLERANCE**. This floating point value specifies a distance used for determining which GRID points are to be connected. Points within a sphere of radius *tol* are considered to be coincident. This feature is especially useful for combining substructures created by different groups where, for example, a different number of significant figures has been input for coordinates. Note that after the **COMBINE** has been performed, the coordinates used for the resulting substructure are taken as the average coordinates of all points that were connected.

Normally, you should use the **AUTO COMBINE** option. The **MANUAL** approach may be useful if you have other pre-processor programs which have provided connection data in a readily available form.

There may be cases in which you wish to override connections between substructure degrees of freedom to represent modeling situations such as hinged joints. You may do this by using the Bulk Data entries:

RELES	SID	SNAME	GID1	DOF1	GID2	DOF2	GID3	DOF3	-cont-
-cont-	GID4	DOF4	*CONTINUES IN GROUPS OF 2*						-cont-

RELES1	SID	SNAME	DOF	*GRID ID LIST*	-cont-

These entries specify that specific degrees of freedom, **DOF**, for GRID points, **GID**, in substructure **SNAME**, ***not be connected*** during the **COMBINE** operation. Note that when the degrees of freedom between two points are to be **RELES**ed, only one of them must be specified on the Bulk Data entry. Note that the **RELES** data are also selected with the **CONNECT** subcommand.

One of the most powerful features of the substucturing capability allows you to manipulate the substructures in space when performing the **COMBINE**. This is done with the series of commands:

```
COMPONENT = cname
   TRANSFORM = tid
   SYMTRAN = [X][Y][Z]
```

A **COMPONENT**, *cname*, is defined as one of the substructures which appears in the *sub_name_list* of the **COMBINE** command. For each **COMPONENT** that you select, you may perform a coordinate transformation that translates and rotates the model in space prior to its combination. The **TRANSFORM** subcommand references a **TRANS** Bulk Data entry whose format is:

TRANS	TID		A1	A2	A3	B1	B2	B3	-cont-
-cont-	C1	C2	C3						

This entry defines a rectangular coordinate system (identical to the **CORD2R** Bulk Data entry) by three points, **A**, **B**, and **C**, which are expressed in the ***Basic Coordinate system of the resultant combined substructure***. You may also use **CORD2R** Bulk Data entries. This feature allows you to use nested coordinate systems (i.e. those which reference points in other coordinate systems), a capability not allowed with **TRANS** entries. You may also select a symmetry transformation, **SYMTRAN**, about one or all of the coodinate directions of the component substructure. This transformation is a mirror reflection of the geometry of the component.

☞ ***If you select one or three planes of symmetry, the coordinate system of the component is reversed, that is, it becomes left-handed***. This is the case for resulting displacements, as well.

In order to connect two points during a **COMBINE** operation they must have the same output coordinate system. The common output system will, in fact, be the coordinate system of the **COMBINE**d point in all final output presentation. If the two points have different coordinate systems, a fatal error message is issued and your job terminates. This problem can be overcome in two ways. The **GTRAN** Bulk Data entry can be used to manually specify an output coordinate system for individual GRID points. The format of this entry is:

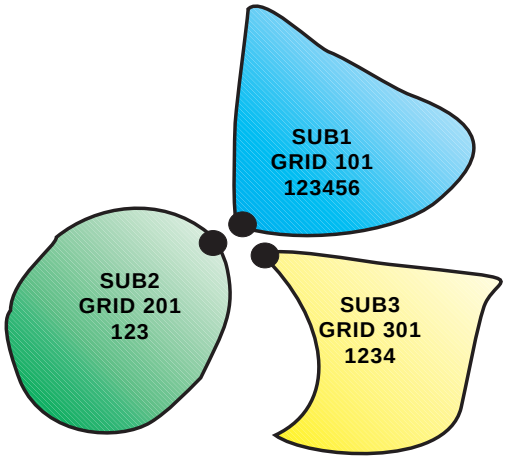
GTRAN	SID	SNAME	GID	TID					

You select these data using the **TRANSFORM** *sid* value. The GRID point **GID** in substructure **SNAME** will have the coordinate system specified by **TID**. If you leave this field blank, or set it

to zero, then the GRID point displacements will be transformed to the ***Basic Coordinate System of the COMBINEd substructure***. If you set **TID** to **SID**, then the point will remain fixed to the substructure without transformation. If you specify another **TID** value, it references a **TRANS** Bulk Data entry ***which defines a specific coordinate system***.

It is also possible to let **UAI/*NASTRAN*** automatically generate **GTRAN** data entries for GRID points with mismatched coordinate systems. You request this by using the **COORD** subcommand. This subcommand allows you to transform the coordinates systems to either the Basic system, or to a specific coordinate system that you select. This subcommand is global in nature and applies to all components in the **COMBINE** operation.

Understanding Substructure COMBINE Results. During the **COMBINE** operation, a new substructure is created. The degrees of freedom of this substructure are defined by ***Internal Points***. In the simplest case, an internal point is defined for each unique GRID point in the resulting substructure. Thus, several points which are connected result in a single internal point. This assignment of internal points is more complex when there is a mismatch of degrees of freedom at two points which are to be connected. This occurs when, for example, you have



used the **RELES** feature described above. When this happens, several internal points may result. Consider the following sketch.

The three identified points are to be connected. Notice that each has a different set of degrees of freedom. After the **COMBINE** operation, three internal points are created, as summarized in the following table:

INTERNAL POINT	DOF	SUB1	SUB2	SUB3
1	123	101	201	301
2	4	101		301
3	56		201	

The **COMBINE** Connectivity Map, shown in Table 19-7, provides a description of the internal points in the combined substructure and their origin. It is possible, using these internal point numbers, to introduce new data during the solution step. However, as you will see in following sections, there are simpler ways to do this. This connectivity map, along with other optional output, is selected with the **OUTPUT** subcommand of **COMBINE**.

19.5.1.6 PHASE 2 — Multilevel Reduction

You may also perform additional reductions during PHASE 2 with the command:

```
REDUCE name
   NAME = newname
   BOUNDARY = bsid
   RSAVE
```

where *name* is the name of the substructure to be reduced, and *newname* is the resulting substructure. The *bsid* identifies a boundary set that defines the degrees of freedom that will be ***retained*** after the reduction. You specify this set with the Bulk Data entry:

BDYC	SID	SNAME1	SID1	SNAME2	SID2	SNAME3	SID3		-cont-
-cont-		SNAME4	SID4	*CONTINUES IN GROUPS OF 2*					-cont-

Where each of the identification numbers **SIDi** reference **BDYS** and **BDYS1** Bulk Data entries that were described in the previous section.

The use of the PHASE 2 **REDUCE** should be limited to operating on substructures that were created from previous **COMBINE** operations. For basic substructures, it is recommended that the reductions be performed during PHASE 1 in the usual manner, i.e. with **ASET** or **OMIT** Bulk Data, or with the **AUTOOMIT** feature.

To understand this, consider the substructure analysis tree that is shown in Figure 19-4. Here, the first PHASE 2 operation in the tree combines **A** and **B** to form **C**. An equivalence operation is then used to create a secondary substructure named **D**. You will note that the image substructures **XA** and **XB** are created automatically by this operation using the **PREFIX** specified in the **EQUIVALENCE** command. The command packet for this sequence could be:

```
SUBSTRUCTURE PHASE2
   COMBINE A,B
      NAME = C
   EQUIV C,D
      PREFIX = X
ENDSUBS
```

The substructures **C**, **D**, and **E** are then combined to form **F**. This new substructure is then reduced using any of the methods allowed for the given solution type. In the case of statics, a Guyan reduction may be used. This new substructure, **G**, is then replicated with another **EQUIV** command to create **H**. A final **COMBINE** then creates the solution structure **I**:

```
SUBSTRUCTURE PHASE2
   COMBINE C,D,E
      NAME = F
   REDUCE F
      NAME = G
      BOUNDARY = 101
   EQUIV G,H
      PREFIX = Y
   COMBINE G,H
      NAME = I
      COMPONENT H
      TRANS = 999
   SOFPRINT(TOC)
ENDSUBS
```

Note that a boundary set, **101**, was specified for the **REDUCE** operation. This defined the degrees of freedom to be retained in the substructure *a-set*. The equivalence operation generated a large subtree of image substructures. The new image substructures **YXA** and **YXB** illustrate how substructure names grow as more prefixes or suffixes are added during **EQUIV** operations. The final **COMBINE** of **G** and **H** illustrates the use of the **TRANS** subcommand for substructure **H**. This allows **H** to be translated or rotated prior to combination.

19.5.2 Substructure Control Commands — Solving Your Model

The solution of your model also proceeds in two steps. In the first, you perform a PHASE 2 solution of a structure and recover the results. You may then print or save these results on the database. The PHASE 2 solution only computes nodal response quantities. Therefore, you perform the second step which is PHASE 3 data recovery. The PHASE 3 recovery allows you to obtain all element-based solution results such as stresses, for your model or any of its component substructures. The following sections describe the methods for performing these steps.

19.5.2.1 PHASE 2 Solve — Statics

After you have built your final model, you solve, recover, and **PRINT**, or **SAVE**, the solution results in a PHASE 2 execution. Consider the following example:

```
SUBSTRUCTURE PHASE2
   SOLVE I
   RECOVER I
      PRINT A
         DISP = ALL
         SPCF = ALL
      PRINT C
         DISP = ALL
         SPCF = ALL
      PRINT H
         DISP = ALL
         SPCF = ALL
ENDSUBS
SUBCASE 100
   LOAD = 100
 SUBCASE 200
   LOAD = 200
```

In this example, the solution is performed for each SUBCASE specified in the Case Control packet, and data selected are printed for each of the named substructures. The results are only computed at the boundary, or *a-set*, points of the selected substructures. To recover results for all points, you must execute PHASE 3 for each substructure. The **PRINT** command is used to both print the solution results and to save them on the database. You may use the **SAVE** subcommand instead to save, but not print, the results.

The definition of SUBCASEs in the PHASE 2 solution is critical for performing the PHASE 3 data recovery. Each PHASE 2 SUBCASE results in a single displacement vector. When using these vectors, they must be referenced in the order that they were computed.

The complete **SOLVE** command is:

```
SOLVE name
   OVERWRITE
   MODEL = n
```

The **OVERWRITE** and **MODEL** subcommands are described in subsequent sections of the Chapter.

19.5.2.2 PHASE 2 Solve — Statics — Defining Boundary Conditions

You may apply new boundary conditions to your model during the PHASE 2 solution. These include single-point constraints and multipoint constraints. Additionally, you may use the AUTOSPC feature to remove singularities automatically.

Single-point constraints are defined using the Bulk Data entries:

SPCS	**SID**	**SNAME**	**GID1**	**DOF1**	**GID2**	**DOF2**	**GID3**	**DOF3**	**-cont-**
-cont-	**GID4**	**DOF4**	***CONTINUES IN GROUP OF 2***						**-cont-**

SPCS1	**SID**	**SNAME**	**DOF**	***GRID ID LIST***	**-cont-**

These data are referenced in the usual manner with an **SPC** Case Control command. All data are selected using the substructure name, **SNAME**, GRID point identification number, **GIDi**, and DOF Code, **DOFi**. Enforced displacements may also be specified using:

SPCSD	**LID**	**SNAME**	**GID1**	**DOF1**	$\delta 1$	**GID2**	**DOF2**	$\delta 2$	**-cont-**

You may apply multipoint constraints within, or between, substructures. These constraints are of the form:

$$u_d = -\frac{1}{B}\sum_i \sum_j A_{ij} u_j$$

where u_d represents a dependent degree of freedom and u_j a set of independent degrees of freedom. The coefficient matrix, $\boldsymbol{A}$, defines the linear relationship among these degrees of freedom. You define such relationships with the Bulk Data entry:

MPCS	**SID**	**SNAMED**	**GIDD**	**DOFD**	**B**				**-cont-**
-cont-		**SNAME1**	**GID11**	**DOF11**	**A11**	**GID12**	**DOF12**	**A12**	**-cont-**
-cont-		**SNAME2**	**GID21**	**DOF21**	**A21**	**GID22**	**DOF22**	**A22**	**-cont-**

These data are referenced in the usual manner with the **MPC** Case Control command.

19.5.2.3 PHASE 2 Solve — Statics — Defining Loading Conditions

There are a number of ways in which you may specify PHASE 2 loading conditions. The first method is to combine loads which have been defined for the basic components of the solution structure during PHASE 1 processing. This is done by using the **LOADC** Bulk Data entry:

LOADC	**LID**	**S**	**SNAME1**	**LID1**	**S1**	**SNAME2**	**LID2**	**S2**	**-cont-**
-cont-			**SNAME3**	**LID3**	**S3**	***CONTIUNUES IN GROUPS OF 3***			**-cont-**

This load, *L*, is defined as:

$$L = S \sum_i S_i \, LID_i$$

where **S** is the overall coefficient, **LIDi** are load set identification numbers defined in the PHASE 1 job for substructures **SNAMEi**, and **Si** are coefficients applied to each **LIDi**. Note that the same substructure may be referenced more than once on the **LOADC** entry. SUBCASEs are defined in the normal manner.

The second method is to apply the loads directly in PHASE 2. This is done by referencing the points within any or all basic substructures and applying any GRID-based loads desired. This is facilitated by the **GRIDS** Bulk Data entry discussed in Section 19.5.2.4.

The third method is to automatically combine all PHASE 1 loads which have the same identification numbers. This is done using the Case Control command:

```
LOADCOMBINE = lid
```

where *lid* is a load set identification number that appears in one or more basic components of the solution structure. The effect is the same as if you had specified a **LOADC** Bulk Data entry to combine all loads with a scale factor of 1.0. You may also combine new PHASE 2 loads with these PHASE 1 loads if you give them the same identification number:

```
...
SUBCASE 101
   LOADCOMBINE = 100
   LOAD = 100
...
```

The fourth, and final, method is to automatically combine all PHASE 1 loads having the same identification number. This is done with the Case Control command:

```
LOADCOMBINE = ALL
```

In this case, a separate SUBCASE is generated for every unique PHASE 1 load set across all of the component substructures of the solution structure. The SUBCASE identification numbers are assigned consecutive values beginning with **1**. Detailed examples of these methods are shown later in this Section.

19.5.2.4 PHASE 2 — Adding New Data

During the PHASE 2 operations, unique internal identification numbers are assigned to each of the GRID points in a newly created substructure. When the **SOLVE** operation is performed, the actual model may be represented either by external GRID point identification numbers, or by internal identification numbers. The usual approach is to use the external numbering, but there are two cases where this is not possible:

- If two different internal points represent the same external identification number, then external numbering cannot be used.

- If the same internal point is a boundary point where two different substructures are being combined, and the external identification numbers are different for each point. This condition is normally allowed, and the smaller external point identification number is used.

To help insure unique external point identification numbers, each substructuring command that adds new basic substructure GRID points includes a special subcommand called **GRIDOFFSET**. This subcommand allows you to control the values of the external GRID point identification numbers created. The commands supporting **GRIDOFFSET** include **BASIC**, **EQUIV**, **MREDUCE**, and **CREDUCE**.

If external identification numbers cannot be used, then the internal numbers are used instead. The **MODEL** subcommand of **SOLVE** allows you to control this behavior. You can explicitly request the use of external or internal GRID point identification numbers, and you can specify the action to be taken if the request cannot be performed. These options are described in Chapter 3 of the User's Reference Manual.

When internal identification numbers are used, adding PHASE 2 data is very inconvenient because you must reference these internal numbers. In order to simplify the referencing of new data you may use the **GRIDS** Bulk Data entry:

GRIDS	**NEWGID**	**SNAME**	**GID**						

where **NEWGID** is a new GRID point identification number that you are assigning to **GID** in substructure **SNAME**. If you inadvertently select a value for **NEWGID** which is the same as an internal point number, the **NEWGID** value takes precedence. Once you have defined these new GRID points, you may add any data to your PHASE 2 input stream for use during the **SOLVE** operation. This may include elements, constraints, loads, and so forth.

There are many useful engineering applications of this feature. For example, in aircraft models it may be necessary to add new actuators or wing stores to the same base model to study the impact of these minor redesigns. Such elements may be added in PHASE 2 without modification of the base structure. Case Control requests can then be used to request output selections for the actual model being **SOLVE**d. The GRID point identification numbers used in the output will be either the external numbers, the internal numbers, or the **GRIDS** numbers described above. These output requests may include all GRID and element solution results and plot data. Element-based results are generated only for those elements added during the **SOLVE** operation. A PHASE 3 **RECOVER** is still required to obtain element solution results for the PHASE 1 model. These Case Control output requests are distinct from any output requests used in the **RECOVER** command. The requests apply to the full solution model with no references to the component substructures that were used to construct the final model. This is different from the normal **RECOVER** command which partitions the output for each separate component. Because of this, external GRID point identification numbers are always used in the **RECOVER** output presentation. It also differs in that only GRID point output requests are allowed in the normal **RECOVER**.

19.5.2.5 PHASE 2 — Performing More than One Solution

There may be cases when you wish to perform multiple solutions of the same substructure, saving all results on the database. This is useful in a number of cases such as adding new loading conditions, adding new boundary conditions, or changing a portion of the model. To do this, you assign a ***subscript*** to the data when it is recovered.

For example:

FIRST SOLUTION	SECOND SOLUTION

...
SUBSTRUCTURE PHASE2
SOLVE BIGMODEL
RECOVER BIGMODEL
  SUBSCRIPT = 1
...
	...
	SUBSTRUCTURE PHASE2
	SOLVE BIGMODEL
	RECOVER BIGMODEL
	  SUBSCRIPT = 2
	...

In this example, all results for the first solution is given a subscript of 1, and that for the second solution a subscript of 2. Both solutions are saved on the database and can be accessed at a later time. The **SUBSCRIPT** subcommand is not required. However, if you perform an unsubscripted solution, you cannot later perform one with a subscript. Therefore, you should consider in advance whether multiple solutions will be performed. In the worst case, you simply must delete the unsubscripted solution before continuing your analysis.

Consistency checks are made during the solution to prevent you from destroying previous solution results. The entire substructure tree is analyzed prior to the solution to make certain that there are no previous solution data with the same identification. The solution is considered to be duplicated if there is a solution with the same subscript value or if any unsubscripted solution exists.

There may be cases in which you wish to re-solve a substructure without saving previous solution data. To do this, you may manually delete all solution data (with the same identifier) from the entire substructure tree. Since this can be inconvenient for large trees, **SOLVE** has a special **OVERWRITE** subcommand which automatically deletes the required items.

19.5.2.6 PHASE 2 Recovery

The PHASE 2 **RECOVER** command is used to save and, optionally, print the results of the **SOLVE** operation. A **SOLVE** command should normally be followed by a **RECOVER** command so that you may specify which solution data are to be recovered and stored on the SOF. Even if the **RECOVER** command is not used, the solution for the actual substructure that was **SOLVE**d is automatically saved. However, this is usually not sufficient for performing successful PHASE 3 recovery. Additional PHASE 2 executions are required to recover the data. The basic form of the **RECOVER** command is:

```
RECOVER sub_name
   SUBSCRIPT = n
   { PRINT } comp_name
   { SAVE  }
      Output requests for all basic components of comp_name
      BASIC bas_name
         Output requests for bas_name
         ...
   ...
```

For any given solution substructure, you may print the results for any of its component substructures. For the example in Figure 19-4, if you **SOLVE I**, then you could **PRINT** or **SAVE** any of the nine substuctures contained in its tree.

During **RECOVER** processing of the **PRINT** or **SAVE** subcommands, all data required to obtain the requested results is automatically computed and saved. For example, if you initially re-

quest **PRINT** for substructure **D**, then **RECOVER** first computes and saves the results for **I**, **G**, **F**, and **D**, in that order. These solution results are then immediately available for any future **PRINT** or **SAVE** requests performed during the **RECOVER**, or in a subsequent PHASE 2 execution. Before you can perform a PHASE 3 **RECOVER** operation on any of the basic components of the solution, you must make sure to **RECOVER** the solutions for that component, i.e. a PHASE 2 **RECOVER PRINT** or **SAVE** operation must have been performed for that basic component.

The output from the **PRINT** subcommand is partitioned by the basic components that comprise the substucture. For example, if you specify **PRINT F**, then a separate set of output appears for basic components **A, B, XA, XB, and E**. During the PHASE 2 **RECOVER**, you can request — depending on the analytical discipline — **DISP**, **SPCF**, **OLOAD**, **VELO**, and **ACCE**. You can also use the **BASIC** subcommand to further restrict the output for each basic component.

Limiting the amount of output can also be accomplished with additional subcommands. You can control explicitly the SUBCASEs, modes, frequencies, or time steps that are processed. You can also control the output sort order. Complete details of these options are found in Chapter 3 of the ***User's Reference Manual***.

19.5.2.7 PHASE 3 Recovery — Statics

One you have performed your PHASE 2 solution, you may recover data for any of the basic substructure components of the solution structure which have been **SAVE**d during the solution. (Remember that the PHASE 2 **PRINT** request also saves the solution). The PHASE 3 input stream is the PHASE 1 data for the basic substructure with modifications made in the Substructure Control packet and the Case Control packet. The Substructure Command packet includes the commands:

```
SUBSTRUCTURE PHASE3
   RECOVER name

      CASEIDS = { PHASE2     }
                { ORDINAL    }
                { POSITIONAL }

      OFFSET = off_val
      SUBSCRIPT = sub_val
      OUTPUT = OP1,...OPn
ENDSUBS
```

where *name* is the name of the basic substucture to be recovered. The **CASEIDS** and **OFFSET** subcommands define the relationship of the PHASE 3 subcases to those used during the PHASE 2 solution. The subcommand **SUBSCRIPT** is used only when you have performed multiple PHASE 2 solutions.

The basic rules for constructing your input data stream for PHASE 3 are:

- The Bulk Data packet must be the same as it was for the PHASE 1 execution.
- The constraint definitions must be the same as it was for the PHASE 1 execution.
- All SUBCASE definitions must be based on the PHASE 2 solution cases.
- Static load definitions are ignored in PHASE 3.

There are three ways of relating the PHASE 3 SUBCASE definitions to the PHASE 2 solution. These methods, which are controlled with the **CASEIDS** subcommand, are:

The PHASE2 Option. When using this option, the SUBCASE identification numbers used in PHASE 3 must match those from PHASE 2. You may recover a subset of the PHASE 2 SUBCASEs by simply specifying in PHASE 3 Case Control those that you want. To futher simplify the input, if you wish to use the complete set of solutions from PHASE 2, then it is not necessary to enter any SUBCASEs in PHASE 3 Case Control. Examples of this are given in section 19.5.3. When using solution merging (discussed later) you may have a situation where the PHASE 2 SUBCASE identification numbers are not unique. The **OFFSET** subcommand is used in such cases to offset the identification numbers by a given value for each set of PHASE 2 solutions. You may, or may not, specify **TITLE, SUBTITLE and LABEL** information in PHASE 3. These titles will be taken from PHASE 2 unless you explicitly redefine them.

The Ordinal Option. This option is similar to the PHASE 2 option except that the SUBCASE identification numbers are assign values from 1 to n. The rules for partial output and titling are as described above. Since this method assures unique SUBCASE identifiers, it may be more convenient when merging many solutions. Also, it is not necessary to use the **OFFSET** subcommand with this option. However, the meanings associated with the PHASE 2 SUBCASE identifiers are lost.

The Positional Option. This option is primarily provided for compatibility with previous versions of **UAI/*NASTRAN***. Here, the SUBCASE identification numbers are completely arbitrary. The first SUBCASE is used for the first solution, the second SUBCASE for the second solution, and so on. The ability to obtain partial output, to generate SUBCASEs, and to reuse titling information are not available.

Merging Multiple Solutions. It is also possible to merge multiple solutions during a PHASE 3 **RECOVER** operation. You may perform such a merger in two ways:

- Results from multiple subscripted solutions for the same substructure may be merged using the **SUBSCRIPT** subcommand. If all of the subscripted solutions are desired, then a single **RECOVER** command may be used. Otherwise, you must use a new **RECOVER** command for each solution. For example:

ALL SOLUTIONS	SELECTED SOLUTIONS

...
SUBSTRUCTURE PHASE3
RECOVER NAME
  CASEIDS = ORDINAL
  SUBSCRIPT = ALL
...
	...
	SUBSTRUCTURE PHASE3
	RECOVER NAME
	  SUBSCRIPT = 1
	  OFFSET = 200
	RECOVER NAME
	  SUBSCRIPT = 5
	  OFFSET = 200
	...

  Also remember that for all cases of solution merging, if you use the **CASEIDS=PHASE2** option, then the SUBCASE identification numbers must be unique. When using multiple **RECOVER** commands this may be controlled with the **OFFSET** subcommand. If you select the **ALL** option, you must use the **CASEIDS=ORDINAL** subcommand to ensure uniqueness. If, however, all SUBCASE identification numbers are, in fact, unique then this subcommand is not required.

- When your model includes basic substructures that have been **EQUIV**ed, solutions for these substructures may also be merged, but you must use a second technique. Suppose that you have a model of an airplane. Only one wing is modeled, say the left, and then an **EQUIV** is used to create the right wing as a mirror image of the left. It is often desirable, for post-processing, to obtain the solution results for both wings during a PHASE 3 recover execution. This could be done in the following manner:

```
SUBSTRUCTURE PHASE3
   RECOVER LWING
      OFFSET = 100
   RECOVER RWING
      OFFSET = 200
```

  Notice that in this case there will always be duplicate SUBCASE identification numbers because both sets of results are from the same solution. As before, either the **OFFSET** or **CASEIDS=ORDINAL** subcommands can be used to resolve the duplicates.

  This type of solution merging, where different substructures are being combined, will only work if the substructures are all **EQUIV**ed to one another. The method cannot be used if the substructures were created in individual PHASE 1 executions.

It is also possible to mix any combination of **SUBSCRIPT** and substructure merging in the same PHASE 3 **RECOVER** execution.

19.5.3 A Complete Example

The following sections present a complete example of substructure use.

19.5.3.1 The PHASE 1 Load Definition

The following table shows the PHASE 1 input for three basic substructures:

PHASE 1 DATA FOR MODEL1	PHASE 1 DATA FOR MODEL2	PHASE 1 DATA FOR MODEL3

...
SUBSTRUCTURE PHASE1
  BASIC = MODEL1
ENDSUBS
SPC = 100
BEGIN BULK
  ...
ENDDATA
	...
	SUBSTRUCTURE PHASE1
	  BASIC = MODEL2
	ENDSUBS
	SPC = 200
	SUBCASE 1
	  LOAD = 10
	SUBCASE 2
	  LOAD = 20
	BEGIN BULK
	  ...
	ENDDATA
		...
		SUBSTRUCTURE PHASE1
		  BASIC = MODEL3
		ENDSUBS
		SPC = 200
		SUBCASE 1
		  LOAD = 10
		BEGIN BULK
		  ...
		ENDDATA

For this example, there are three basic substructures, named **MODEL1**, **MODEL2**, and **MODEL3**. For **MODEL1**, there are no loading conditions defined in PHASE 1 as indicated by the lack of SUBCASE definitions. For **MODEL2**, there are two SUBCASEs which select loads with identification numbers **10** and **20**, and for **MODEL3**, there is one SUBCASE with identification numbers **10**.

☞ Remember that you may only specify a single boundary condition in PHASE 1 because only one set of system matrices may be saved on the SOF.

☞ When referencing PHASE 1 load conditions, it is important to remember that the PHASE 1 SUBCASE numbers are unimportant. The load condition identification number, **10** and **20** in the example, are those which are referenced in the PHASE 2 **SOLVE**.

19.5.3.2 The PHASE 2 Loading Conditions

All actual loading conditions applied to a solution structure are defined during the PHASE 2 solution. You can use loads that were defined on any or all of the PHASE 1 substructures, or you may enter the loads directly in PHASE 2. As illustrated earlier, PHASE 1 loads are applied with the **LOADC** Bulk Data entry. This entry allows you to combine the PHASE 1 loads for any of the basic substructures that are components of the solution structure. When defining loads directly in PHASE 2, you may use the **GRIDS** Bulk Data entry to identify the actual points to be loaded. The following example illustrates both of these conditions.

```
SUBSTRUCTURE PHASE2
   COMBINE MODEL1,MODEL2,MODEL3
      NAME = BIGMODEL
   SOLVE BIGMODEL
   RECOVER BIGMODEL
   SAVE BIGMODEL
ENDSUBS
SUBCASE 1
   LABEL = COMBINATION OF PHASE 1 LOADS SETS 10 AND 20 FROM MODEL2
   LOAD = 1
SUBCASE 2
   LABEL = COMBINATION OF PHASE 1 LOAD 10 FROM MODEL 2 AND 10 FROM MODEL 3
   LOAD = 2
SUBCASE 3
   LABEL = PHASE 2 LOADS ENTERED DIRECTLY
   LOAD = 3
SUBCASE 4
   LABEL = COMBINATION OF PHASE 1 AND PHASE 2 LOADS
   LOAD =4
BEGIN BULK
LOADC,1,1.0,MODEL2,10,1.0,MODEL2,20 ❶
LOADC,2,1.0,MODEL2,10,1.0,MODEL3,10 ❷
GRIDS,999,MODEL2,1 ❸
FORCE,3,999,,1.+3,1.0,0.0,0.0 ❹
LOAD,4,1.0,1.0,1,1.0,2 ❺
ENDDATA
```

In this example, there are four SUBCASEs defined. The first SUBCASE combines loads **10** and **20** that were defined in the PHASE 1 job for **MODEL2**, and the second SUBCASE combines load **10** of **MODEL2** with load **10** of **MODEL3**. The **LOADC** Bulk Data entries ❶ and ❷ define these combinations and assign them identification numbers of **1** and **2**, respectively. The third SUBCASE, number **3**, references load set **3** defined by a **FORCE** Bulk Data entry, ❹. This entry, in turn, references GRID Point 1 of substructure **MODEL2** as specified on the **GRIDS** entry, ❸. The fourth SUBCASE specifies a load set, **4**, which is a combination of the loads used in SUBCASEs **1** and **2**. This is defined by the **LOAD** Bulk Data entry, ❺.

It is often the case that a number of basic substructures will have loads applied which are part of a single loading condition for the final solution structure. If you know that this will be the case, then you may plan in advance to use the same load identification number for each

PHASE 1 substructure. In the example, suppose this is the case for load set 10 which is defined for both **MODEL2** and **MODEL3**. Then **SUBCASE 2** could be specified as:

```
...
SUBCASE 2
   LOADCOMBINE = 10
...
```

This requests that the loads from all basic substructures with the identification number of **10** be combined for this SUBCASE. This is equivalent to a **LOADC** Bulk Data entry which names every load with an identification number of **10** and uses coefficients of unity. Thus, for this example, it is equivalent to:

```
LOADC,2,1.0,MODEL2,10,1.0,MODEL3,10
```

The use of **LOADCOMBINE** can greatly reduce input requirements for defining load conditions. However, its use does mean that you must assign the same identification number to all appropriate PHASE 1 loading conditions. For even more convenience, if all of your loading conditions have been specified in this manner, that is every basic substructure uses the same load set identification number for loads that should be combined, then you can use the special form:

```
LOADCOMBINE = ALL
```

In this case, a separate SUBCASE is generated for every unique **PHASE 1** load set across all of the component substructures of the solution structure. The SUBCASE identification numbers are assigned consecutive values beginning with **1**. For the example problem, two SUBCASEs would result. These would be equivalent to:

```
SUBCASE 1
   LABEL = LOAD 10
SUBCASE 2
   LABEL = LOAD 20
```

When using the **ALL** feature of **LOADCOMBINE**, you should not include any SUBCASE definitions in your input data stream. They will be generated automatically.

You must assign the same identification number to the load in each PHASE 1 substructure if you plan to use the **LOADCOMBINE** command to create a total loading condition across multiple substructures.

19.5.3.3 The PHASE 3 Loading Conditions

You generally use a PHASE 3 execution to perform recovery of solution results for a basic substructure which is a component of a solution structure that has been solved in PHASE 2. The SUBCASEs that are used in PHASE 3 are identical to those used in the PHASE 2 solution. Thus, to recover solution results for **MODEL1** of our example, the following input data could be used:

```
...
SUBSTRUCTURE PHASE3
   RECOVER MODEL1
ENDSUBS
SPC = 100
DISP = ALL
STRESS = ALL
SUBCASE 1
   LABEL = COMBINATION OF PHASE 1 LOADS SETS 10 AND 20 FROM MODEL2
SUBCASE 2
   LABEL = COMBINATION OF PHASE 1 LOAD 10 FROM MODEL 2 AND 10 FROM MODEL 3
SUBCASE 3
   LABEL = PHASE 2 LOADS ENTERED DIRECTLY
SUBCASE 4
   LABEL = COMBINATION OF PHASE 1 AND PHASE 2 LOADS
BEGIN BULK
...
```

Note that there are no **LOAD** commands and that there are four SUBCASEs, corresponding to the SUBCASEs of the PHASE 2 execution. Any **LOAD** commands in PHASE 3 are ignored. The boundary condition used in the PHASE 1 job must be used in PHASE 3. Note that the **SUBCASE** identification numbers match those used in PHASE 2. This also allows you to recover only selected SUBCASEs if you wish. For example:

```
...
SUBSTRUCTURE PHASE3
   RECOVER MODEL1
ENDSUBS
SPC = 100
DISP = ALL
STRESS = ALL
SUBCASE 3
   LABEL = PHASE 2 LOADS ENTERED DIRECTLY
BEGIN BULK
...
```

is perfectly valid and will result in a solution for **SUBCASE 3** as defined in PHASE 2. If you wish to recover a solution for all PHASE 2 SUBCASEs, then it is not necessary to enter any SUBCASE data at all. You may simply use:

```
...
SUBSTRUCTURE PHASE3
   RECOVER MODEL1
ENDSUBS
SPC = 100
DISP = ALL
STRESS = ALL
BEGIN BULK
...
```

If no SUBCASEs are defined in the PHASE 3 **RECOVER**, then the SUBCASEs will be generated automatically to match those used in the PHASE 2 solution. Additionally, any titling information used in PHASE2 will be used in PHASE 3 unless you explicitly override them with new **TITLE**, **SUBTITLE**, or **LABEL** commands.

19.5.4 AUTOMATIC ALTERS FOR SUBSTRUCTURING

The **UAI/*NASTRAN*** Substructuring capability translates the Substructure Control commands into DMAP ALTERs which are automatically inserted into the Rigid Formats during substruc-

turing analyses. As a result, you must follow special procedures if you wish to include your own ALTERs during a substructuring run. This section describes how to do this.

19.5.4.1 The Location of Substructuring ALTERs

The ALTERs generated when performing substructuring analyses depend both on the analysis discipline and the PHASE of the analysis. You can determine the location of the ALTERs by making a test run and using the Executive Control command:

```
DIAG 23,43
```

This will print the current ALTER locations and then terminate your **UAI/*NASTRAN*** execution.

19.5.4.2 The Procedure For Adding Your Own ALTERs

You may use your own ALTERs with substructuring if they do not conflict with the statement numbers listed above. If you have conflicts, you may merge your ALTERs with those generated by the substructuring system using the following procedure.

1. For the **UAI/*NASTRAN*** substructuring execution to be ALTERed, you must modify the input data stream as follows:
   a. To request a file where the substructure ALTERs will be written, add to the Executive Control Packet the command:

      ```
      ASSIGN ...,USE=PUNCH
      ```

   b. Requests that the ALTERs be printed and PUNCHed and job execution terminated after the preface by adding the command:

      ```
      DIAG 23,24,43
      ```

2. Submit the modified run which will create the PUNCH file containing the substructuring ALTERs.
3. Using an editor, modify the **UAI/*NASTRAN*** input file submitted in Step 2 as follows
   a. Remove the command:

      ```
      ASSIGN ...,USE=PUNCH
      ```

   b. Modify the Executive Control command:

      APP DISP *to* **APP DISP,SUBS,1**

   c. Merge the contents of the PUNCH file from the previous execution into the Executive Control Packet. Remove the **DIAG** statement inserted in the previous run.
   d. **It is crucial that you make no changes to the Substructure Control Packet because the DMAP retrieved only applies to the specific Substructure Control packet used when you punch the ALTERs.**
   e. Modify the ALTER subpacket as desired, merging your ALTERs with those generated by the substructuring system. Add an **ENDALTER** statement at the bottom of the ALTER sequence. Now there will not be a conflict between automatic ALTERs and user ALTERs. This is because the automatic ALTER feature is suppressed with the

      ```
      APP DISP,SUBS,1
      ```

command, and all ALTERs are present in the Executive Control Packet. Following the normal rules for DMAP ALTERs, your ALTERs must be inserted in the proper sequence with those which were previously punched.

4. Submit this run, now modified for the second time, for execution.

19.5.5 Using APP DMAP with SOF Data

You may write your own DMAP sequence to use data from the SOF database. To do this, you must, as always, include the Executive control command, **APP DMAP** in your input data stream, and you must **ASSIGN** the SOF database:

```
ASSIGN SOF='my_sof_file',OLD,USE=SOF
...
APP DMAP
...
dmap program commands
...
CEND
```

In addition to all of the normal DMAP modules, the only substructuring modules that may be used in the DMAP sequence are **SOFI**, **SOFO**, and **SOFUT**. Since the SOF is an ***eBase*** database, you may also use the modules **DBIN** and **DBOUT** to import and export data.

19.5.6 Case Control Commands

For substructuring analysis, the Case Control packet controls loading conditions, constraint set selection, output requests, and method of analysis just as in any non-substructuring analysis. However, there are very important relationships among the Case Control commands for each of the three phases of substructuring. Compatibility between the phases must be maintained for load sets, constraint sets, and SUBCASE definitions.

In PHASE 1, there are two requirements. First, you may specify only one set of constraints. You may change the constraint sets later during the PHASE 2 solution. Second, you must define one SUBCASE for each loading condition which is to be saved on the SOF. This loading condition may consist of any combination of external static loads, thermal loads, element deformation loads, or enforced displacements. These loads and those from other basic substructures may then be combined during the PHASE 2 solution step.

The PHASE 2 Case Control packet is exactly like the Case Control used in a non-substructuring analysis. When you are not performing a **SOLVE** operation, only the **TITLE**, **BEGIN BULK**, and structural plotting commands are used. When you are performing a **SOLVE**, the SUBCASE definitions, load and constraint set selections, and other commands are used in the usual fashion to control the solution process.

When performing PHASE 3 solution recovery, several requirements must be met.

☞ All **MPC**, **SPC**, **ASET**, and **OMIT** constraint sets must be identical to those used in PHASE 1 for the substructure.

☞ The SUBCASE definitions must be based on the PHASE 2 solution SUBCASE definitions. Refer to section 19.5.2.7 for a detailed discussion of the SUBCASE setup procedure.

19.5.7 Bulk Data Entries

There are nearly twenty special Bulk Data entries that you must use when performing Substructuring Analyses. These entries, summarized in Table 19-6, form three functional groups. Those used for reductions, those used for the combine operation, and those used when you perform the solution. Most of these entries are similar to other data used for non-substructuring analyses, differing only in the fact that substructure names are also required as part of the input data. Examples of some of these Bulk Data entries are given in Section 19.7. See the ***User's Reference Manual*** for detailed descriptions of them all.

Table 19-6. SUBSTRUCTURING ANALYSIS BULK DATA

TO PERFORM:	USE BULK DATA:	TO:
SUBSTRUCTURE REDUCTION	**BDYC**	Define REDUCE boundary sets.
	BDYS, BDYS1	Define dof in the boundary set.
SUBSTRUCTURE COMBINATION	**CONCT, CONCT1**	Define dof for manual COMBINE.
	GTRAN	Define an output coordinate system.
	RELES, RELES1	Specify *dof* not to be connected.
	CORD2R (Formerly **TRANS**)	Define coordinate transformations.
SUBSTRUCTURE SOLUTION	**DAREAS**	Specify dynamic loading.
	DELAYS	Specify dynamic load time delays.
	DPHASES	Specify dynamic phase lead terms.
	LOADC	Combine static loading conditions.
	MPCS	Specify multipoint constraints.
	SPCS, SPCS1	Specify single-point constraints.
	SPCSD	Define enforced displacements.
	TICS	Define transient initial conditions.

19

19.6 SOLUTION RESULTS

The actual solution results for Substructuring Analyses are the same as those for nonsubstructuring analyses. However, the substructuring operations also have special solution results which are discussed in this Section.

19.6.1 PHASE 1

The creation of a basic substructure in PHASE 1 does not result in any significant output because only the fundmental database entities for the substructure are computed and saved. If you also perform PHASE 2 operations in your PHASE 1 execution, then you will get some of the results described in the next Section.

19.6.2 PHASE 2

During PHASE 2, the final solution structure is synthesized by combining groups of substructures and, optionally, performing different types of reductions. These steps are followed by the actual solve operation. This section briefly describes the results of these operations.

19.6.2.1 The COMBINE Operation

You may combine from two to seven substructures to form a new substructure which may represent either a final solution structure, or simply another intermediate substructure used in synthesizing your solution structure. A **COMBINE** operation is performed in two steps. The first step performs the geometric combination of the substructures and the second step performs the appropriate matrix operations that define the transformations that will be used in subsequent operations or for data recovery.

The most useful result of a **COMBINE** operation is the table of substructure connectivities and the final substructure map. These detail the actual degrees of freedom that have been connected and provide a list of all the *internal points* in the substructure resulting from the **COMBINE**. You will find this useful for validating the operation and checking that no spurious mechanisms, such as hinges or pin points, have been introduced in the model. This map, shown in Table 19-7, is requested with the **OUTPUT** subcommand (Option **10**) of the **COMBINE** command.

Table 19-7. TYPICAL COMBINE CONNECTIVITY MAP

SUMMARY OF SUBSTRUCTURE CONNECTIVITIES

INTERNAL POINT NO	INTERNAL DOF NO	DEGREES OF FREEDOM	********** S U B S T R U C T U R E N A M E S **********	
17	81	12345	LEFT	RIGHT
			106	706
18	86	12345	LEFT	RIGHT
			105	705
19	91	12345	LEFT	RIGHT
			104	704
20	96	12345	LEFT	RIGHT
			103	703
21	101	12345	LEFT	RIGHT
			102	702

19.6.2.2 The REDUCE Operations

The three reduction operations, **REDUCE**, **MREDUCE** and **CREDUCE** also have an **OUTPUT** subcommand that may be used to check the results of the operation. The results vary depending upon the reduction used.

When performing the static condensation, or Guyan reduction, you may obtain a summary of the *boundary* degrees of freedom which can be used to verify that the correct *a-set* has been retained. Modal reductions produce summaries of the your input requests and tables that indicate the frequencies computed for the generalized coordinates that are created in these operations. You may also request a list of the physical boundary points.

19.6.2.3 The SOLVE Operation

When you perform a solution in PHASE 2, only the results for the *a-set* degrees of freedom of the final solution structure and its components are available. To recover the results of a *g-set* data for a substructure, a PHASE 3 execution must be made.

The interpretation of forces of constraint resulting from PHASE 2 is different than in a non-substructuring analysis. Additional terms are included in the calculations. These differ depending on the solution being performed. The equations used for the computation of these forces, Q, are shown below:

$\boldsymbol{Q} = \boldsymbol{Ku} - \boldsymbol{P}$	Rigid Formats 1 and 2
$\boldsymbol{Q} = \boldsymbol{Ku} - \omega^2 \boldsymbol{Mu}$	Rigid Format 3
$\boldsymbol{Q} = \boldsymbol{Ku} + p\boldsymbol{Bu} + p^2\boldsymbol{Mu}$	Rigid Formats 7, and 10
$\boldsymbol{Q} = \boldsymbol{Ku} + \boldsymbol{B}\dot{\boldsymbol{u}} + \boldsymbol{M}\ddot{\boldsymbol{u}} - \boldsymbol{P}$	Rigid Formats 8, 9, 11, and 12

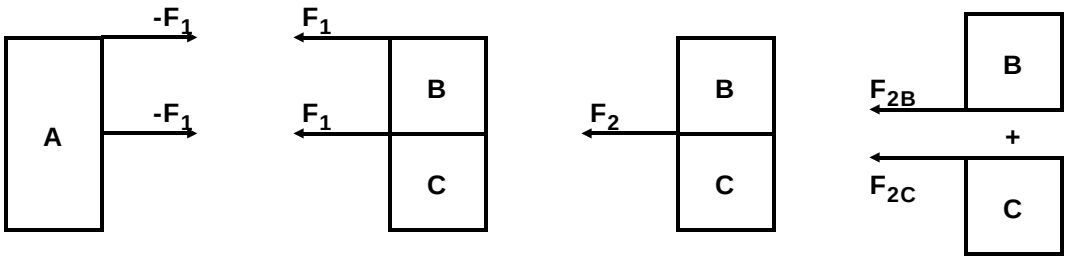
In these equations, $\boldsymbol{Q}$ are the forces of constraint, $\boldsymbol{P}$ are the applied loads, $\boldsymbol{u}$ is the displacement vector, $\boldsymbol{K}$ is the stiffness, $\boldsymbol{B}$ is the damping, $\boldsymbol{M}$ is the mass, ω are eigenvalues from a real modes analysis, and p are complex eigenvalues from a complex modal reduction. The force vectors $\boldsymbol{Q}$ contain all the terms due to:

- Inertia forces
- Damping forces
- Single-point constraints
- Multi-point constraints
- Forces transferred from other connected substructures
- Residual forces due to numerical round-off

The equations presented above for calculation of forces of constraint provide especially useful information such as the forces of substructure interconnection as shown in Figure 19-6a. Forces $\mathbf{F_1}$ and $\mathbf{F_2}$, recovered as forces of constraint for substructure **A** and for substructure **BC**, represent the forces of interconnection. Force $\mathbf{F_2}$ represents the sum of two component forces, one from each component substructure **B** and **C**, acting at their common GRID point. The separate contributions to $\mathbf{F_2}$ from each **B** and **C** may be determined by using the **RECOVER** command for the component substructures **B** and **C** individually, as shown in Figure 19-6b.

19

Figure 19-6. SUBSTRUCTURE CONSTRAINT FORCES



a. INTERCONNECTION FORCES b. COMPONENT FORCES

19.6.3 PHASE 3

The results of a PHASE 3 execution are identical to the results that you would obtain from performing a nonsubstructuring analysis on the same model.

19.7 MODELING GUIDELINES

This Section provides you with suggestions and recommendations useful when performing Substructuring Analyses. It also describes possible pitfalls and how you may avoid them.

19.7.1 Pilot Models

It is strongly recommended that before you perform a large substructuring analysis, that you simulate your structure with a simplified *pilot* or prototype model. This approach allows all data, including operational commands and control of the SOF, to be tested using small matrices at low cost.

19.7.2 Reductions

Generally, the most economical analyses result from the use of relatively small basic substructures or by performing significant reductions in PHASE 1. When you use Guyan reduction in static analyses, either a very large or very small percentage of the degrees of freedom should be eliminated. When you use modal reduction, the **FIXED** set should be used to help approximate the expected solution mode shapes. If inertia relief shapes are requested, six shapes are created. However, if the model is not fully three-dimensional, some of these shapes may be null. This must be corrected prior to subsequent operations. Certain mode shapes which introduce singularities, such as rigid body shapes at zero frequency, are automatically excluded from assignment to the reduced substructure. The rigid body shapes are not needed because the boundary points, by definition, must provide the rigid body description of the structure.

19.7.3 Load Append

If you need to define new loading conditions, you may use the ***Load Append*** feature. This avoids performing redundant PHASE 2 computations.

19.7.4 Singularities

Certain GRID point degrees of freedom which often result in stiffness singularities, such as the rotations normal to a plate, may be constrained in PHASE 1. However, if these GRID points are later transformed to a new output coordinate system during a combine operation, the singularity may be re-introduced into the problem. The substructuring system transforms GRID point degrees of freedom in groups of three translations and three rotations. Thus, if one or more translational or rotational degrees of freedom exist for a GRID point, and a general transformation is applied, three translational or rotational degrees of freedom will exist for the resulting structure at that GRID point. As a result, the stiffness matrix will be singular, and this must be considered in subsequent operations. For example, in future reduction operations some of these degrees of freedom must be kept in the boundary set so that the interior point stiffness matrix is non-singular. You must eventually remove the extraneous singularities using **AUTOSPC**, **SPCS** or **MPCS** Bulk Data entries when you solve your problem.

19.7.5 Solution Items

The SOF has numerous protection features. At no time are any SOF items destroyed by **UAI/*NASTRAN*** unless you specifically request such action. In addition, **UAI/*NASTRAN*** does not allow more than one substructure item to exist for each substructure at any one time. This means that it is sometimes necessary for you to manually edit the SOF to remove specific data items. For example, to perform a new solution for a substructure that had previously been

solved, the old solution data items must be removed from the SOF to make space available for the new data. That is, the **SOLN** and **UVEC** items, the solution summary and displacement vectors respectively, created in the earlier solve operation, should be deleted if a new solution with new loads or frequency range is desired for the same substructure. These same items also must be removed from each lower level substructure for which the new solution data will be recovered. The **EDIT** or **DELETE** commands may be used for this purpose. **SOLN** and **UVEC** items are also created by **MRECOVER** and must be deleted prior to a solve and recover for the same structure. By using the equivalence operation to create an identical structure, a new solution may be obtained for the same structure without deleting the older solution data items.

19.7.6 Structural Design Considerations

You should combine substructures which may change due to design iterations as late in the sequence of combine operations as possible. This will minimize the cost of creating a new solution structure. Also, if the design iteration changes are minor and their impact on other substructures in the model can be neglected, then recover operations only from the lowest level of substructure affected by the changes need be performed. Frequently, these design changes can be evaluated using only the PHASE 3 recovery calculations. Of course, care must be taken to maintain compatibility with the boundary GRID points and connections that were previously defined.

19.7.7 The Superelement Convertor

UAI/*NASTRAN* has the ability to accept superelement models and convert them into a sequence of substructuring data files. Separate files are created for each Basic substructure and for the Phase II **COMBINE** and **SOLVE** operations. To select this feature, you include the Executive Control command:

```
SECONVERT [ = [ { YES } ] [ , OUTPUT = { BULK } ] ]
          [   { NO  } ]   [            { SPLIT } ]
```

in your input stream along with a complete superelement data file. The **OUTPUT** option results that all resulting Bulk Data be placed in a single **BULK** file, or **SPLIT** among separate files.

19.8 EXAMPLE PROBLEMS

This section presents two example problems which illustrate most of the important features of Substructuring Analysis.

19.8.1 Circular Plate Model

The first example problem performs a very simple statics analysis which illustrates how a structure is partitioned into substructures, how substructures may be combined with both TRANSFORM and SYMTRANSFORM options, and how loads are selected during the PHASE 2 solution.

Example Problem 19-1
A circular plate of radius 11 in is fixed along its circumference and subjected to an unsymmetric load of 1 psi over a 15° sector of the plate. You wish to determine the maximum normal displacement using substructuring methods.

The physical model of the plate and its engineering data are shown in Figure 19-7. The finite element model of basic substructure **RIGHT** is constructed for the upper right quadrant of the plate as shown in Figure 19-8. The model is composed of QUAD4 and TRIA3 elements. The PHASE 1 model data is found in file **SUB1A**. One SUBCASE is defined which applies the uniform load to the first 15° sector of the model. The loads are applied using **PLOAD2** Bulk Data entries in the normal manner. To reduce the computational requirements, a Guyan reduction is performed during PHASE 1. The retained degrees of freedom are those along the boundaries that will later be joined, as shown in Figure 19-8. The remaining points on the outer edge of the plate are fixed for all degrees of freedom.

You are now ready to create your solution structure. It is very important that you determine the sequence of operations that you intend to perform in advance. For this example problem, the general outline of operations is described by the substructure tree shown in Figure 19-9. The sample data for this job is found in file **SUB1B**.

Figure 19-7. CIRCULAR PLATE

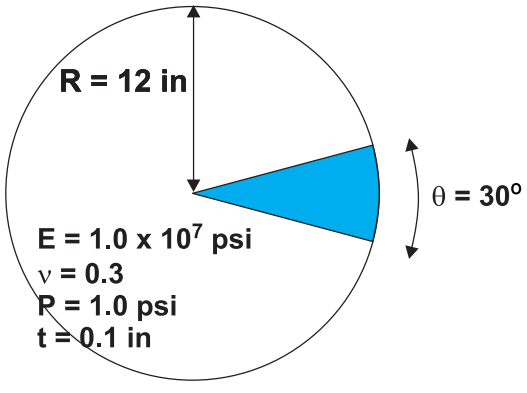
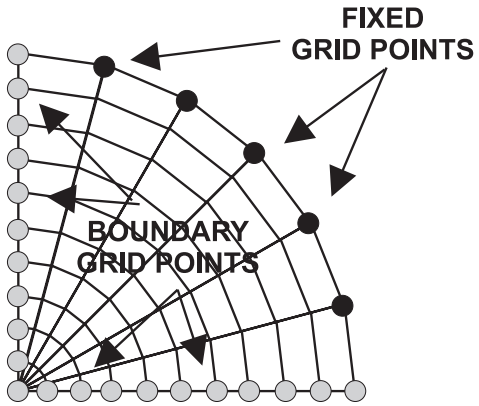
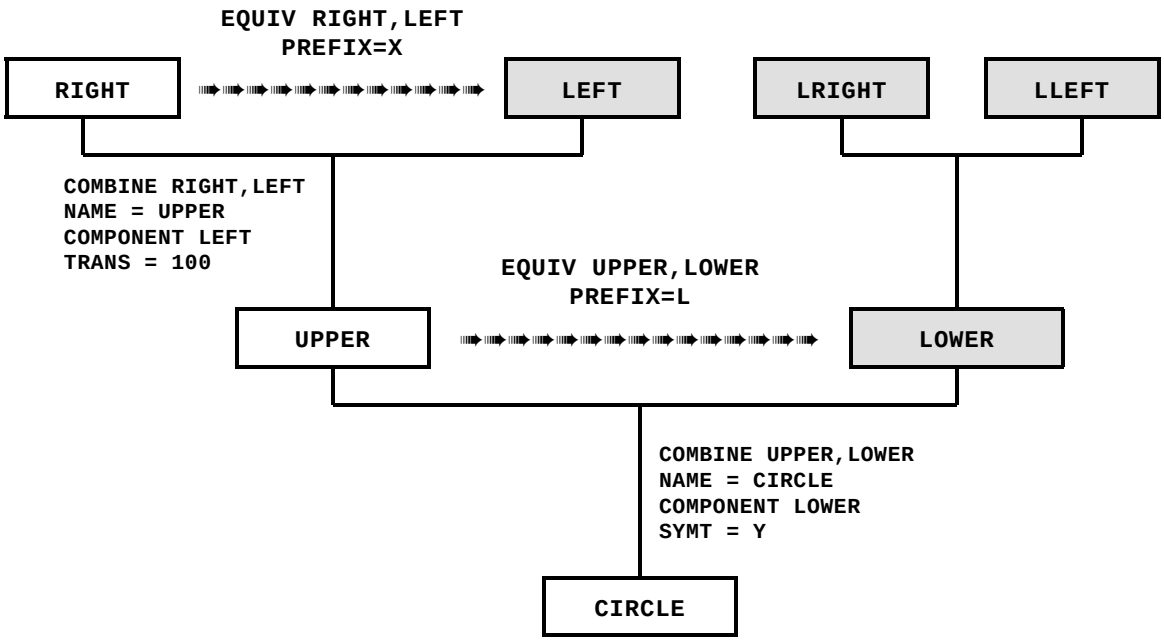


Figure 19-8. FINITE ELEMENT MODEL



19

Figure 19-9. SUBSTRUCTURE TREE DIAGRAM



The first PHASE 2 operation is to create substructure **LEFT** by replicating substructure **RIGHT**. You do this with an **EQUIV** command. Now, you wish to **COMBINE RIGHT** and **LEFT** to create the **UPPER** half of the plate. The Substructuring commands to do this are:

```
COMBINE RIGHT,LEFT
      NAME = UPPER
      OUTPUT = 12
      TOLER = 0.0001
      COMPONENT = LEFT
          TRANS = 100
```

The **TRANS** subcommand is used, in this case, to rotate substructure **LEFT** 90° before performing the **COMBINE** operation. The identification number, **100**, references a **TRANS** Bulk Data entry that defines any rotations and translations that you wish to perform. In this case, this entry is:

TRANS	**100**		**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**1.0**	
	0.0	**1.0**	**0.0**						

Figure 19-10 shows how you determine the transformation. The basic coordinate system selected for the complete model is indicated by X_B,Y_B. The PHASE 1 substructure **RIGHT** was defined in the same system, although this is not a requirement. Substructure **LEFT**, after the 90° rotation is in the indicated position. The **TRANS** Bulk Data entry is then used to define a rectangular coordinate system that relates the system of the substructure to the basic coordinate system you wish to use. The three points, **A**, **B** and **C** in Figure 19-10 define the substructure coordinate system in the usual manner. The coordinates of these points in the basic are entered on the data entry. As a result, point **C** on the **TRANS** entry is specified as **(0.,1.,0.)**. This indicates that the substructure X-axis will be rotated to coincide with the basic Y-axis.

Figure 19-10. FIRST COMBINE OPERATION

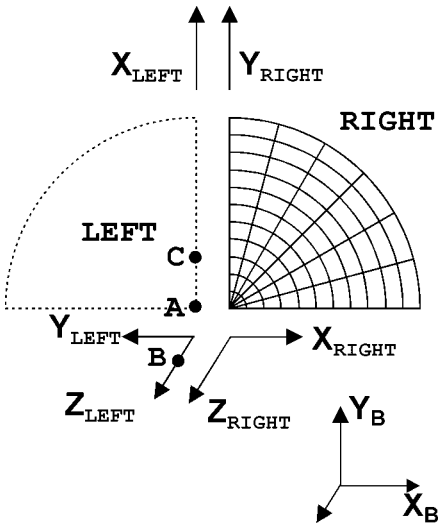
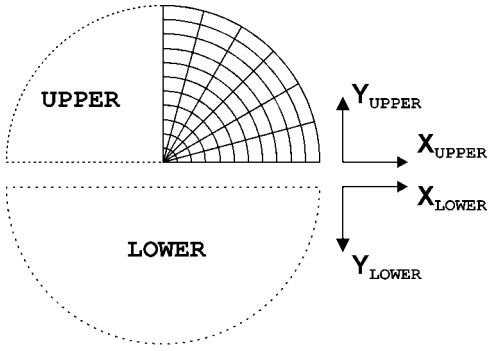


Figure 19-11. SECOND COMBINE OPERATION



NOTE THAT THE COORDINATE SYSTEM FOR LOWER IS LEFT-HANDED AFTER THE SYMT

The ***Multilevel*** flexibility of Substructuring Analysis now allows you to **EQUIV** the upper half of the circle to a new substructure called **LOWER**. Note that the components of **LOWER** appear in the tree as image substructures. Now, to create the full circle, you perform another **COMBINE** operation:

```
COMBINE UPPER,LOWER
      NAME = CIRCLE
      OUTPUT = 12
      TOLER = 0.0001
      COMPONENT = LOWER
          SYMT = Y
```

This time, you use a symmetric transformation on the **LOWER** substructure, reflecting it about the X-Axis. The **SYMT** subcommand may be interpreted as specifying the coordinates whose signs will be changed in the component substructure. The resulting coordinate systems are shown in Figure 19-11. The coordinate system for **LOWER** is now left-handed.

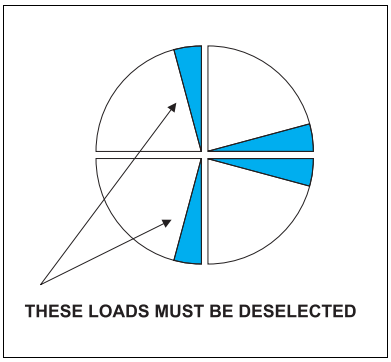
Now that you have created the entire circle, you perform the static solution by simply entering the command:

```
SOLVE CIRCLE
```

The recovery of data for individual substructures which are components of your solution structure may then be requested. For this example problem, recovery is performed only for the basic substructure **RIGHT**:

```
RECOVER CIRCLE
PRINT RIGHT
```

The application of loads during the solution is accomplished by selecting any combination of all loads defined for your basic substructures. You must remember that the loads remain fixed to the substructure when the **EQUIV** operation is used to replicate parts of your model. The adjoining figure illustrates the loads that are available for the combined substructure **CIRCLE**. Since you only wish to load the 30° segment which spans **RIGHT** and **LRIGHT**, you must select these loads thus deselecting the loads that you do not wish to apply. This is done using the **LOADC** Bulk Data entry:



LOADC	**10**	**1.0**	**RIGHT**	**1**	**1.0**	**LRIGHT**	**1**	**1.0**	

This entry defines a linear combination of loads drawn from one or more of the basic substructures in your model. Additionally, you must insure that the boundary conditions for the solution are correct. Because the outer circumference of the circle must be fixed, you must define **SPC**s for the points on the outside of the interface boundaries. This is done with the Bulk Data entries:

SPCS	**10**	**RIGHT**	**111**	**123456**	**711**	**123456**			
SPCS	**10**	**LEFT**	**111**	**123456**	**711**	**123456**			
SPCS	**10**	**LRIGHT**	**111**	**123456**	**711**	**123456**			
SPCS	**10**	**LLEFT**	**111**	**123456**	**711**	**123456**			

The solution for this load condition is then performed. The PHASE 2 solution results are only the GRID point quantities for all degrees of freedom in the *a-set* of each substructure. To recover the *g-set* values and element quantities, you must execute PHASE 3 for each substructure. The PHASE 3 data file for substructure **RIGHT** is found in file **SUB1C**.

19.8.2 Airplane Model

The second example problem performs the analysis of an entire aircraft. It illustrates more advanced features including multiple solutions for different configurations of the structure.

Example Problem 19-2

A full vehicle aircraft model is assembled and solved for static response using a variety of scenarios to illustrate particular features of substructuring.

Scenario 1: Performs a complete analysis using the most basic, straight-forward substructuring approach.
Scenario 2: Illustrates recovering multiple solutions for a single basic substructure and how this feature can be used to manage the analysis of different configurations of the vehicle.
Scenario 3: Illustrates the use of reduction methods during Phase 1.
Scenario 4: Illustrates the exporting and importing of component SOF databases for allowing detailed analysis of your workshare component using a substructure component from another contractor.

The physical model of the complete airplane is shown in Figure 19-12. The example problems described in the remainder of this section are all based on the components shown in Figure 19-13.

Figure 19-12. FULL AIRPLANE MODEL

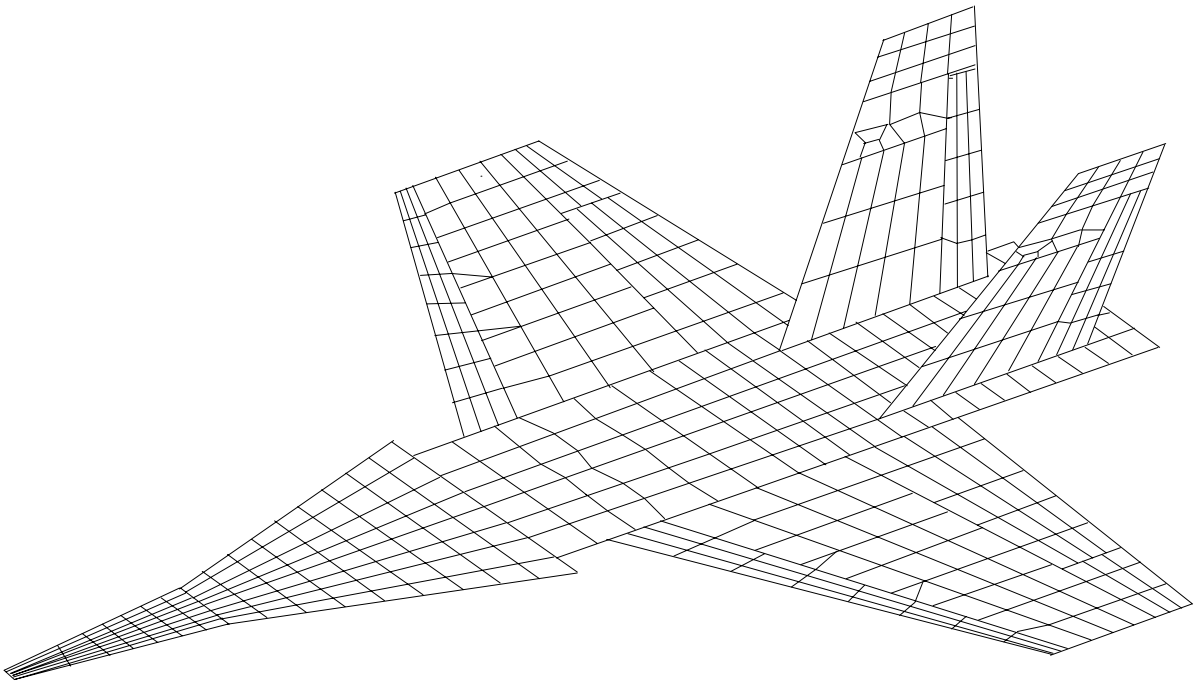


Figure 19-13. AIRPLANE MODEL SUBSTRUCTURES

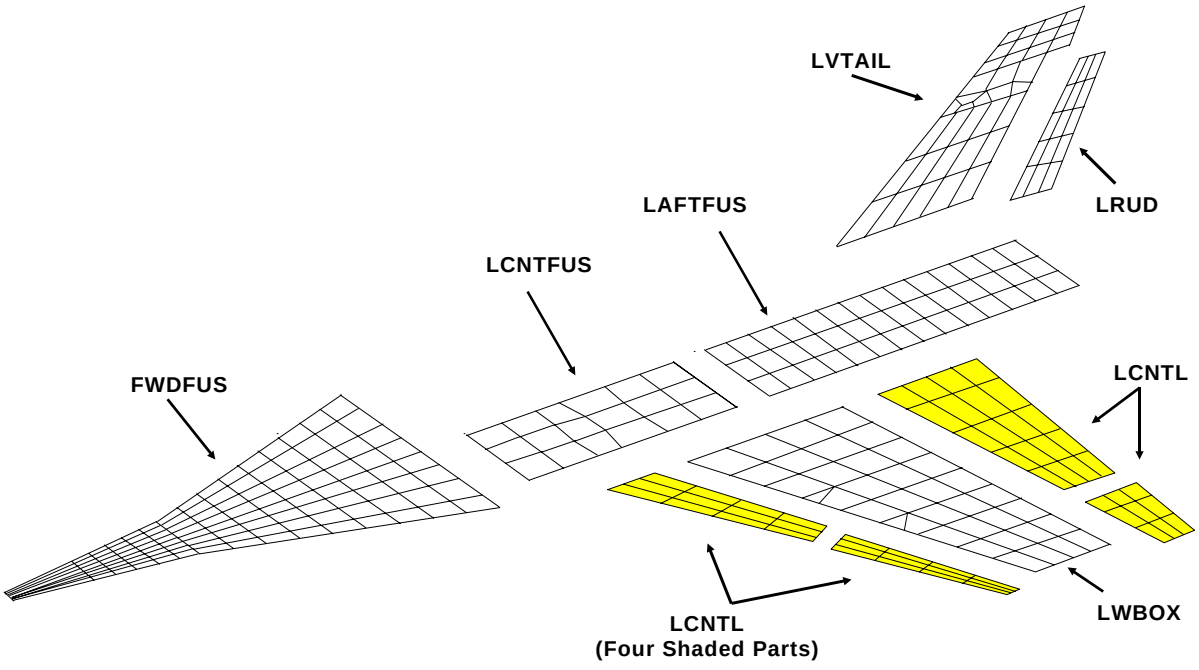


Table 19-8. COMMON DATA FILES

FILE NAME	DESCRIPTION
apafus.blk	Aft fuselage geometry bulk data
apbdys.blk	Boundary set definitions for all component substructures
apcfus.blk	Left center fuselage geometry bulk data
apffus.blk	Forward fuselage geometry bulk data
apiblef.blk	Inboard leading edge flap geometry bulk data
apibtef.blk	Inboard trailing edge flap geometry bulk data
aplwing.blk	Left wing box geometry bulk data
apoblef.blk	Outboard leading edge flap geometry bulk data
apobtef.blk	Outboard trailing edge flap geometry bulk data
aptail.blk	Left vertical fin geometry bulk data
aprudr.blk	Left rudder geometry bulk data
altail.blk	Vertical tail loads bulk data
albend.blk	Wing bending load bulk data
alcntl.blk	Wing control surface load bulk data
altwist.blk	Wing torsion load bulk data
apr10.blk	Vertical tail coordinate system definition
apr3000.blk	Inboard leading edge flap hingeline coordinate system definition
apr4000.blk	Outboard leading edge flap hingeline coordinate system definition
apr5000.blk	Inboard trailing edge flap hingeline coordinate system definition
apr6000.blk	Outboard trailing edge flap hingeline coordinate system definition
apr9000.blk	Rudder hingeline geometry coordinate system definition

The components represent a pilot model that would be useful as an aid in developing and assessing candidate substructuring approaches for a full vehicle aircraft analysis. For purposes of illustrating the features of substructuring, the component *parts* are representative of workshares that might be performed by different organizations in a teaming arrangement, as well as of natural topological boundaries. The ten parts that define basic substructures are a left wing, four wing control surfaces, the left mid-fuselage, the left aft-fuselage, the full nose, a left vertical stabilizer and its rudder. Table 19-8 shows those data files that are shared by all of the analysis scenarios that are described in this section.

☞ ***Warning — These are not stand-alone data files, they contain data components which are used in conjunction with other files defined later!***

Tables 19-9 through 19-12 provide a list of the files for each scenario ***in the order of execution***. In order to execute any of the scenarios, you will need copies of all the files in Table 19-8 and all the files in the scenario of interest. The file names in the scenarios have been chosen such that ASCII collated wildcards (e.g., **a1***) may be used to invoke the example problems in the proper order. All of these tables contain a brief description of the files associated with this series of examples.

Table 19-9. DATA FILES FOR SCENARIO 1

FILE NAME	DESCRIPTION
a1p1af.d	Create left aft fuselage; **laftfus**
a1p1cf.d	Create left center fuselage; **lcntfus**
a1p1ff.d	Create full forward fuselage; **fwdfus**
a1p1rd.d	Create left rudder; **lrud**
a1p1tl.d	Create left vertical stabilizer; **lvtail**
a1p1wb.d	Create left wing box; **lwbox**
a1p1wf.d	Create left wing's 4 surfaces; **lcntl1**
a1p2com.d	Create images and combine to form full vehicle; **plane1**
a1p2sol.d	Solve and recover; **plane1**
a1p3wb.d	Recover internal loads for the wing

Table 19-10. DATA FILES FOR SCENARIO 2

FILE NAME	DESCRIPTION
a2p1af.d	Create left aft fuselage; **laftfus**
a2p1cf.d	Create left center fuselage; **lcntfus**
a2p1ff.d	Create full forward fuselage; **fwdfus**
a2p1rd.d	Create left rudder; **lrud**
a2p1tl.d	Create left vertical stabilizer; **lvtail**
a2p1wb.d	Create left wing box; **lwbox**
a2p1wf1.d	Create left wing's 4 surfaces, configuration 1; **lcntl1**
a2p1wf2.d	Create left wing's 4 surfaces, configuration 2; **lcntl2**
a2p1wf3.d	Create left wing's 4 surfaces, configuration 3; **lcntl3**
a2p2com.d	Create images and combine to form full vehicle, **planenc**, without control surfaces
a2p2sol1.d	Combine flap configuration 1 with rest of aircraft; solve for **plane1**
a2p2sol2.d	Combine flap configuration 2 with rest of aircraft; solve for **plane2**
a2p2sol3.d	Combine flap configuration 3 with rest of aircraft; solve for **plane3**
a2p3wb.d	Recover loads for the wings; all conditions and configurations

Table 19-11. DATA FILES FOR SCENARIO 3

FILE	DESCRIPTION
a3p1af.d	Create left aft fuselage; **laftfus**
a3p1cf.d	Create reduced left center fuselage; **cfusr**
a3p1ff.d	Create full forward fuselage; **fwdfus**
a3p1rd.d	Create reduced left rudder; **rudr**
a3p1tl.d	Create left vertical stabilizer; **lvtail**
a3p1wb.d	Create reduced left wing box; **wboxr**
a3p1wf1.d	Create left wing's 4 surfaces, configuration 1; **lcntl1**
a3p1wf2.d	Create left wing's 4 surfaces, configuration 2; **lcntl2**
a3p1wf3.d	Create left wing's 4 surfaces, configuration 3; **lcntl3**
a3p2com.d	Create images and combine to form full vehicle, **planenc**, without control surfaces
a3p2sol1.d	Combine flap configuration 1 with rest of aircraft; solve for **plane1**
a3p2sol2.d	Combine flap configuration 2 with rest of aircraft; solve for **plane2**
a3p2sol3.d	Combine flap configuration 3 with rest of aircraft; solve for **plane3**
a3p3wb.d	Recover loads for the wings; all conditions and configurations

Table 19.12. DATA FILES FOR SCENARIO 4

FILE	DESCRIPTION
a4pa1afr.d	Create Contractor A's left aft fuselage; **laftfus**
a4pa1cf.d	Create Contractor A's left center fuselage; **lcntfus**
a4pa1rd.d	Create Contractor A's left rudder; **lrud**
a4pa1tl.d	Create Contractor A's left vertical stabilizer; **lvtail**
a4pa2ex.d	Export Contractor A's workshare component; **wrkshra**
a4pb1ff.d	Create Contractor B's full forward fuselage; **fwdfus**
a4pb1wb.d	Create Contractor B's reduced left wing box; **wboxr**
a4pb1wf1.d	Create Contractor B's left wing's 4 surfaces, configuration 1; **lcntl1**
a4pb1wf2.d	Create Contractor B's left wing's 4 surfaces, configuration 2; **lcntl2**
a4pb1wf3.d	Create Contractor B's left wing's 4 surfaces, configuration 3; **lcntl3**
a4pb2bim.d	Import Contractor A's workshare; **wrkshra**
a4pb2com.d	Create images and combine to form full vehicle, **planenc**, without control surfaces
a4pb2so1.d	Combine flap configuration 1 with rest of aircraft; solve for **plane1**
a4pb2so2.d	Combine flap configuration 2 with rest of aircraft; solve for **plane1**
a4pb2so3.d	Combine flap configuration 3 with rest of aircraft; solve for **plane1**
a4pb3wb.d	Recover loads for the wings; all conditions and configurations

While all the **UAI/*NASTRAN*** input files needed for complete execution are available to you, these example problems are more illustrative if considered in sets or *scenarios.* Rather than discuss the details of each input file, the focus of these examples is to illustrate, using the pilot model, a number of *real world* cases that you might want to emulate. Therefore, the focus of the discussion will be on why you might take the illustrated approach, rather than on how the approach is implemented. The supplied input files provide a stand-alone resource to answer the detailed questions on how to use the illustrated substructuring technique.

19.8.2.1 Scenario 1

To form a basis for discussion, the first scenario is a simple substructuring analysis in which each component is created in a PHASE 1 execution and then combined in a one PHASE 2 solution and solved in another. Results are then recovered for the left and right wing components in a example PHASE 3 execution. The files associated with Scenario 1 are listed in Table 19-9 in the order in which they must be executed. During the PHASE 3 recover of the wing box, note that solutions are recovered together for both the left and right hand wing substructures. This is controlled by the multiple **RECOVER** commands for the left and right wing substructures. The **OFFSET** command is used to *create* unique **CASE** identification numbers for the right wing cases. This example illustrates the basic approach to the assembly and solution of a system model in which a group of components have been modeled in their proper geometric orientation, and a single solution in minimum time is desired. Since all components are already in their proper positions on the vehicle, there are no special transformations to relocate components, only *mirror* transformations to produce *righthand* models from *lefthand* models. To achieve mimimum solution time, no reduction techniques are used.

19.8.2.2 Scenario 2

The purpose of the second scenario is to show the power of *subscripted* solutions. Assigning subscripts to the solution results allows you to manage multiple solution items for a single substructure. In versions of **UAI/*NASTRAN*** before Version 11.8, multiple image substructures were created to produce multiple solutions. With subscripted solutions, redundant copies of the substructures are not required. This reduces disk space requirements and is more convenient for data management. You are able to recover multiple solutions for a particular substructure as if the solutions were generated during a single model analysis.

The approach taken in this scenario is to assemble and solve a system model using a group of components; however, during the PHASE 2 **RECOVER** operation, a **SUBSCRIPT** command is included in the Substructure Command Packet to associate the solution data of the Basic substructures with this subscript. This subscript value ***must*** be selected to uniquely identify the recovered results with respect to other solution data that will be generated in later executions. The solution substructure is then destroyed to make disk space available for the next solution substructure.

Several steps must be performed. First, a complete aircraft without any control surfaces (named **planenc**) is assembled in the normal manner. During this process, three distinct substructures are created which could represent three distinct control surface arrangements. Note, however, that in this case, the same control surface models are reused each time to reduce the number of files in the User Guide suite.

Then, three PHASE 2 **COMBINE** and **SOLVE** steps are performed which join the three versions of the control surfaces to the otherwise complete aircraft model. During the PHASE 3 recover of the wing box, note that all three solutions are recovered together for both the left and right hand wing substructures. This is controlled by the multiple **RECOVER** commands for the left

and right wing substructures; both using the **SUBSCRIPTS=ALL** subcommand to recover all subscripted solution vectors. The **OFFSET** command is also used to *create* unique SUBCASE identification numbers for the right wing cases.

In a minor modification to this scenario that is not demonstrated, one might want to model the same control surface configuration using different flap deflections to represent different flight conditions. In substructuring this can be accomplished with the **TRANSFORM** subcommand in the **COMBINE** step. In this case the individual control surfaces would be processed to create basic substructures for each of the control surfaces. For these demonstration problems, the four unique flaps have been included in a single substructure **lcntl**. The geometry has been set up, however, to allow independent processing. It would be most convenient to create each basic control surface substructure in its own basic coordinate system that is aligned with its hingeline. This allows very convenient nesting of the transforms required for the flap deflection and subsequent hingeline location with respect the complete vehicle.

This modeling approach results in a simple tranformation of a basic substructure. Therefore, only one basic control surface substructure is needed, and the *flap deflection* is embodied in the transform operation of the PHASE 2 **COMBINE** step. The remainder of the Scenario 2 would be unaffected by this modification.

19.8.2.3 Scenario 3

This scenario is a minor modification to Scenario 2 to demonstrate the use of reduction approaches in the generation of the basic substructures. In this scenario, reduced models of the center fuselage (**cfusr**), left rudder (**rudr**) and left wing box (**wboxr**) are generated during PHASE 1. These reduced substructures are then used in the **COMBINE** and **SOLVE** steps.

The use of reduction in PHASE 1 is motivated by resource utilization issues. The use of reduction in substructuring is highly dependent on the size of the component and the component's intended use within the analysis process. The reduction process itself takes additional CPU and the storage of the recovery transformations increases the size of the **SOF**. However, if you anticipate using the basic substructure in several analyses and the configuration of the component is stable (i.e., will not need to be reduced again), the cost of reduction and the cost of the additional storage requirements will be compensated by the reduced amount of resources needed in the **SOLVE** steps. This is especially true if you can reduce the component to a very small boundary set.

19.8.2.4 Scenario 4

The final scenario associated with the aircraft model demonstrates the use of **UAI/*NASTRAN*** substructuring in a teaming arrangement in which two contractors have developed models of their respective workshare components and one contractor, denoted as A, now wants to deliver a reduced model of their workshare component to Contractor B. Contractor B will then use the reduced model to perform a detailed analysis of Contractor B's component. In this particular scenario, both contractors are using substructuring approaches within their own workshares, however, all that is required to take advantage of this capability is that both contractors have at least their own workshare as a single component.

To provide Contractor B with the required workshare, Contractor A will create a substructure of its workshare component and **EXPORT** it. The exported component **SOF** is then delivered to Contractor B, who **IMPORT**s it during the PHASE 2 operations leading to the combined analysis. In this case, the result of combining the workshare components is a full vehicle model, but this is not a requirement to use this technique.

At Contractor A's site, the PHASE 1 executions for the left center fuselage, left aft fuselage, left rudder and left vertical stabilizer are run. These components are then equivalenced, tranformed and combined to create the component **wrkshra**.

Independently, at its own site, Contractor B is assembling its workshare component which consists of the the now familiar left wing box and control surface models. These are used to create images of the right hand side equivalents in the normal manner. Upon receipt of the exported **SOF** from Contractor A, a final PHASE 2 solution is performed that imports **wrkshra** from the exported SOF and combines the appropriate components to build the full vehicle model. PHASE 3 recovery operations on Contractor B's components may then be performed in the usual manner.

By a simple extention, you can see that the export operation can proceed in both directions to allow Contractor A to proceed with detailed analyses with a **wrkshrb**. In contrast to each team delivering and receiving numerous individual files defining the components and component loads; in this scenario, the two teams would be transferring a single file containing a single substructure to represent all of that respective contractor's workshare. This approach allows for a *parallel processing* approach between the teams, and significantly reduces the potential for miscommunication among the design teams as new load cases and design modifications are encountered.

This page is intentionally blank.

Chapter 20

AXISYMMETRIC HARMONIC MODELING

Often mechanical and structural components are symmetric about an axis. Examples include tank structures and rotating machinery. **UAI/*NASTRAN*** features three different modeling capabilities for such structures. The first, ***axisymmetric harmonic*** modeling, is described in this Chapter. The two other modeling techniques, ***axisymmetric ring*** modeling and ***cyclic symmetry***, are described in Chapters 21 and 22, respectively.

20.1 TERMINOLOGY

The modeling of axisymmetric structures using harmonic elements has a special nomenclature that differs from the normal structural model. Some important terminology is summarized in this section.

20

Harmonics. Harmonics are Fourier coefficients which are used as generalized coordinates in formulating the axisymmetric harmonic analysis.

Axisymmetric Rings. Unlike general three-dimension structural models, the axisymmetric harmonic model is not defined by GRID points. Instead, it is defined by RINGs. Each RING is located at an (R,Z) coordinate in a cylindrical coordinate system. Although defined by this single point, the RING actually represents motion for the entire circle through the coordinate.

Axisymmetric Elements. As you will see in subsequent sections of this Chapter, there are three axisymmetric harmonic elements. The first, called the CONEAX, is a conical shell element which is defined by two RINGs. The other two are axisymmetric solid elements, the TRIAAX and the TRAPAX. These elements are defined by three and four RINGs, respectively.

Geometric Points. In addition to the RINGs, the axisymmetric harmonic model may contain geometric points. These points may be used to apply loads to the structure or to obtain solution results summed for all of the harmonics.

Detailed descriptions of these concepts are found in the remainder of this Chapter.

20.2 MATHEMATICAL BACKGROUND

The analysis of stress distribution in bodies of revolution, or axisymmetric solids, is of considerable interest in many practical situations. The mathematical problems presented are very similar to those of plane stress and plane strain. By symmetry, the two components of displacements in any plane section of the body along its axis of symmetry completely define the state of strain and, therefore, the state of stress. The axisymmetric formulation is, in a sense, a two-dimensional representation of a solid.

If (r,z) denote the radial and axial coordinates of a point in a cylindrical coordinate system, it can be shown that the displacement functions for plane problems are also applicable for axisymmetric problems. However, any radial displacement automatically induces a strain in the circumferential direction, and as the stresses in this direction are certainly non-zero, this fourth component of strain and the associated stress has to be considered. This is the essential difference in the treatment of axisymmetric problems.

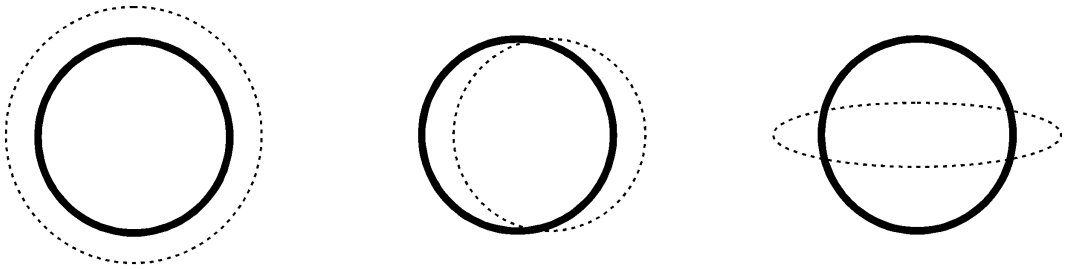
The volume of material associated with an element is that of a body of revolution, and all integrations are performed using this volume. In harmonic modeling, special elements and RINGs, which replace GRID points, are used. In **UAI/*NASTRAN***, generalized coordinates are used to describe the displacement of the structure. Each of these coordinates represents a possible harmonic deformed shape in any of the six component motions at each ring. The loads and responses are ***not*** required to be axisymmetric.

An axisymmetric harmonic model consists of a set of RINGs each of which is located along, and encircles, the Z-axis of the basic coordinate system. Each RING, which is defined by its radial distance from the Z-axis, has six degrees of freedom for each harmonic used in the analysis. Any number of harmonics may be selected. Simply stated, each harmonic represents a possible state of radial deformation. For example, harmonic 0 represents pure symmetric radial motion, harmonic 1 represents pure bending motion of the axisymmetric structure, and harmonics 2 through N represent deformed motions which are symmetric about the Z-axis. Figure 20-1 illustrates the type of deformation that is represented by several generalized harmonic degrees of freedom.

The degrees of freedom defined by each RING 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:

$$u_r(\varphi) = \sum_{n=0}^{N} u_r^n \cos(n\varphi) + \sum_{n=0}^{N} u_r^{n*} \sin(n\varphi) \qquad (20\text{-}1)$$

Figure 20-1. HARMONIC DEGREES OF FREEDOM



where u_r^n and u_r^{n*} are the Fourier coefficients of radial motion for the n^{th} harmonic. When performing an analysis, you specify the highest harmonic to be used. The first sum in (20-1) describes symmetric motion with respect to the $\varphi = 0$ plane. The second sum with the starred (*) superscript describes the antisymmetric motion. The number of harmonics you need to consider depends on the type of loading on the axisymmetric structure. Considering a larger number of harmonics results in an accurate approximation of the deformed shape of the structure, but it is increases computational cost.

Axisymmetric harmonic elements are used to connect the rings. The properties of the axisymmetric harmonic elements are assumed to be symmetric with respect to the Z-axis. However, it is not necessary for the loads to be axisymmetric, because they are expanded in Fourier series of the azimuthal coordinate.

The cylindrical coordinate system for the axisymmetric model, which is automatically defined, has its Z-axis aligned with the basic coordinate system Z-axis, and the R-axis is aligned with the basic X-axis. This allows you to couple a three-dimensional model to an axisymmetric structural component that lies along the Z-axis of the basic coordinate system.

20.3 INPUT DATA REQUIREMENTS

This section provides you with a description of the **UAI/*NASTRAN*** input data required for performing axisymmetric harmonic analysis.

20.3.1 Executive Control Commands

There are no special Executive Control commands required to perform axisymmetric harmonic analysis using **UAI/*NASTRAN***.

20.3.2 Case Control Commands

Two special Case Control commands are used only with axisymmetric harmonic analysis. The first is the **AXISYMMETRIC** command:

$$\textbf{AXISYMMETRIC} = \left\{ \begin{array}{c} \text{SINE} \\ \text{COSINE} \\ \text{FLUID} \end{array} \right\}$$

The **AXISYMMETRIC** command is ***required***. It may be changed from one subcase to another. **AXISYMMETRIC** command enables the selection of symmetric and antisymmetric (i.e., cosine and sine) solutions. For general loading conditions, a combination of symmetric and antisymmetric solutions must be made, using the **SYMCOM** Case Control command. The second command is:

$$\textbf{HARMONICS} = \left\{ \begin{array}{c} \text{ALL} \\ \textit{nharm} \\ \text{NONE} \end{array} \right\}$$

Note that although the **HARMONICS** command is ***optional***, the default is to print solution results ***only*** for harmonic zero. Other Case Control commands are used in the normal manner.

20.3.3 Bulk Data Entries

There is a set of Bulk Data entries that are unique to axisymmetric harmonic modeling and analysis. These are summarized in Table 20-1. Most of the Bulk Data entries have been explained under different sub-sections of this chapter as appropriate.

You indicate the presence of an axisymmetric harmonic model by using the **AXIC** Bulk Data entry. This entry specifies the number of harmonics to be generated for each axisymmetric ring. The use of axisymmetric harmonic elements requires an **AXIC** Bulk Data entry. The format of this entry is:

AXIC	**H**								

where **H** represents the highest harmonic specified for a given analysis. In **UAI/*NASTRAN***, the upper limit on the maximum number of harmonics is 998.

The axisymmetric geometry of the model is described with **RINGAX** entries instead of **GRID** entries. The **RINGAX** data describe concentric circles about the basic Z-axis, with their locations given by radii and z-coordinates. The **RINGAX** entry does not allow a zero radius. However, a small ***hole*** may be defined around the axis of revolution. The format of the **RINGAX** entry is:

20

Table 20-1. BULK DATA FOR AXISYMMETRIC HARMONIC ANALYSI

FUNCTION	BULK DATA
HARMONIC ELEMENT CONNECTIONS	**CCONEAX, PCONEAX** **CTRAPAX, PTRAPAX** **CTRIAAX, PTRIAAX**
HARMONIC RINGS	**RINGAX, SECTAX, POINTAX**
CONSTRAINTS	**ASETAX, ASETAX1** **SPCAX, MPCAX** **OMITAX** **SUPAX**
EXTERNAL LOADS	**FORCEAX, MOMAX** **PRESAX, TEMPAX**
ORTHOTROPIC MATERIAL PROPERTIES	**MAT3, MATT3**
DIRECT MATRIX INPUT	**DMIAX**
ANALYSIS CONTROL PARAMETERS	**AXIC**

RINGAX	**RID**		**R**	**Z**			**PSPC**		

Sections of a ring may be defined using **SECTAX** entry which defines a sector using two angles and a referenced **RINGAX** entry. The **SECTAX** defines six degrees of freedom and its basic coordinate system is the cylindrical coordinate system.

The format of this entry is:

SECTAX	**ID**	**RID**	**R**	**φ1**	**φ2**				

For ***coupling*** the axisymmetric portion of the finite element model to the structural portion, **POINTAX** Bulk Data entries are used. The **POINTAX** entry is used like a **GRID** entry. However, you must remember that the motions of **POINTAX** are ***constrained*** by the number of harmonics specified on the **AXIC** Bulk Data entry. In addition, the **POINTAX** Bulk Data entry is also helpful in defining physical points on the structure for loading and output. **POINTAX** and **SECTAX** components are automatically placed in the *m-set*. Therefore, these entries may ***not*** be constrained in any other manner. The format of the **POINTAX** Bulk Data entry is:

POINTAX	**PID**	**RID**	φ						

UAI/*NASTRAN* also provides direct matrix input for axisymmetric problems. The entries in **DMIAX** are equivalent to **DMIG** entries, with additional capability to specify the harmonic numbers for the degrees of freedom.

20.3.4 Defining Axisymmetric Harmonic Elements

There are three axisymmetric harmonic elements. One is a shell and the others are solids of revolution. The following Sections describe these elements for modeling axisymmetric problems.

20.3.4.1 Conical Shell Element

The conical shell element is defined by two **RINGAX** points. The properties of the conical shell element are specified using the **PCONEAX** Bulk Data entry. The formats of these entries are:

CCONEAX	EID	PID	RID1	RID2					

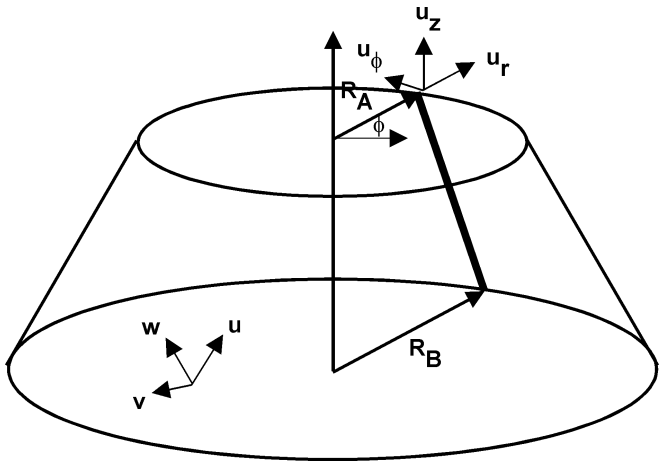
PCONEAX	ID	MID1	T1	MID2	I	MID3	T2	NSM	-cont-
-cont-	Z1	Z2	φ1	φ2	φ3	φ4	φ5	φ6	-cont-
-cont-	φ7	φ8	φ9	φ10	φ11	φ12	φ13	φ14	

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. Figure 20-2 illustrates the geometry of the **CONEAX** element.

20.3.4.2 Solid-of-Revolution Elements

Two solid-of-revolution elements are available to represent non-axisymmetric loadings on axisymmetric structures. These are used for thick or solid cross sections. The first is a triangular element defined by three rings, the TRIAAX, and the second is a trapezoidal element defined by four rings, the TRAPAX. These elements are defined using the Bulk Data entries as shown below:

Figure 20-2. GEOMETRY OF THE CONEAX ELEMENT



CTRIAAX	EID	PID	RID1	RID2	RID3	θm			

PTRIAAX	PID		MID	φ1	φ2	φ3	φ4	φ5	-cont-
-cont-	φ6	φ7	φ8	φ9	φ10	φ11	φ12	φ13	-cont-
-cont-	φ14								

CTRAPAX	EID	PID	RID1	RID2	RID3	RID4	θm		

PTRAPAX	PID		MID	φ1	φ2	φ3	φ4	φ5	-cont-
-cont-	φ6	φ7	φ8	φ9	φ10	φ11	φ12	φ13	-cont-
-cont-	φ14								

As in the case of the CONEAX, up to fourteen stress recovery points may be defined.

20.3.5 Specifying Constraints

The axisymmetric component degrees of freedom defined by **RINGAX** entries ***must*** be constrained in a special manner. All harmonics may be constrained for a particular degree of freedom on a ring by specifying permanent single-point constraints (**PSPC**) on the **RINGAX** entries.

Individual harmonics of each degree of freedom on a ring may be constrained using **SPCAX** entries. This entry is analogous to the usual **SPC** entry except that a harmonic is specified. The format of the **SPCAX** entry is:

SPCAX	SID	RID	HID	DOF	δ				

Similarly, the **ASETAX**, **ASETAX1**, **MPCAX**, **OMITAX**, and **SUPAX** entries correspond to the **ASET**, **ASET1**, **MPC**, **OMIT**, and **SUPORT** data except that harmonics are specified. **SPCADD** and **MPCADD** Bulk Data may be used to combine constraint sets in the usual manner. The formats of **ASETAX**, **ASETAX1**, **MPCAX**, **OMITAX**, and **SUPAX** Bulk Data entries are:

ASETAX	RID	HID1	DOF1	RID2	HID2	DOF2			
		HCODE1			HCODE2				

ASETAX1	HID	DOF	RID1	RID2	RID3	RID4	RID5	RID6	
	HCODE								
	HID	DOF	RIDF	"THRU"	RIDL				
	HCODE								

MPCAX	SID				RIDD	HIDD	DOFD	B	-cont-
-cont-	RID1	HID1	DOF1	A1	*CONTINUES IN GROUPS OF 4*				-cont-

OMITAX	RID	HID	DOF						

SUPAX	RID1	HID1	DOF1	RID2	HID2	DOF2			

Note that axisymmetric harmonic input data contains a special Harmonic Code, **HCODE**. This code allows you to specify a range of harmonics of the form:

Sn1Tn2

where *n1* is the first harmonic and *n2* the last harmonic in a sequence. For example, to select harmonics 0 through 10, you would use **S0T10**

The stiffness matrix for the CONEAX element includes ***five*** degrees of freedom per ring for each harmonic when ***transverse shear*** flexibility is ***included.*** Because the rotation about the normal to the surface is not included, the fourth component degree of freedom (for the cosine series) or the sixth component (for the sine series) must be constrained to zero when the angle between the meridional generators of two adjacent elements is zero. When the ***transverse shear*** flexibility is ***not included***, only ***four*** independent degrees of freedom exist at each ring connected to CONEAX elements. In this case, both the fourth and sixth components of motion must be constrained. The **AUTOSPC** Case Control command provides a convenient way to apply these constraints. Similar constraints are required for all rotational components of rings connected only to the solid axisymmetric elements.

20.3.6 Applying Loads

The axisymmetric 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 on the **PRESAX** data entry which defines an area bounded by two rings and two angles. The format of the **PRESAX** entry is:

PRESAX	**LID**	**P**	**RID1**	**RID2**	**φ1**	**φ2**			

Temperature fields are described by a paired list of angles and temperatures around a ring as required by the **TEMPAX** entry.

TEMPAX	**SID**	**RID**	**φ**	**TEMP**					

Direct loads on the harmonics of a **RINGAX** point are given by the **FORCEAX** and **MOMAX** entries, shown below:

FORCEAX	**LID**	**RID**	**HID**	**F**	**V1**	**V2**	**V3**		

MOMAX	**SID**	**RID**	**HID**	**M**	**V1**	**V2**	**V3**		

The value of a harmonic load F_n is the total load on the whole ring of radius *r*. For example, if a sinusoidal load per unit length of maximum value a_n is given, the magnitude of force entered on the **FORCEAX** entry must be:

For Harmonic HID = 0 $\quad F_n = 2\pi r a_n$

For Harmonic HID > 0 $\quad F_n = \pi r a_n$

20.4 SOLUTION RESULTS

20

Harmonic output is requested by RING number for displacements and element number for element stresses and forces. The number of harmonics that will be output for any request is specified by the Case Control command **HARMONICS**. The solution results may be obtained either as magnitudes of the harmonic generalized coordinates, or as physical quantities computed by the summation of all harmonics. Physical results are obtained by using **POINTAX** and **SECTAX** Bulk Data for displacement quantities. Element forces and stresses are output as physical quantities at points around the circumference of the element identified on the element property Bulk Data entries. Table 20-2 shows an example of displacement results. Point **11** represents a **POINTAX**. You will note that there is no entry under the heading HARMONIC. On the other hand, point **111** is a **RINGAX**. It results in displacement contributions for each harmonic in the analysis. In this case, point **11** is the sum of the displacements for **RINGAX 111**.

For CONEAX element, the element forces per unit width are obtained either as harmonic coefficients or at specified locations. The element stresses are calculated at two specified points (**Z1** and **Z2** entries) on the PCONEAX Bulk Data entry. Element forces per unit width may be obtained for the CONEAX either as harmonic coefficients or at specified locations. The available forces are the bending moments on the *u* and *v* faces, twisting moments, and shearing forces on the *u* and *v* faces.

Element stresses are calculated at two specified points on the cross-section of the element and may also be obtained either as harmonic coefficients or at specified locations. The available stresses are the normal stresses in the u and v directions and the shear stress on the u face in the v direction.

For TRIAAX and TRAPAX elements, the element forces are also given for each harmonic, or at physical points around the circumference. Element forces for these solid axisymmetric elements consist of total radial, circumferential, and axial quantities acting at the centroid of the element. These data are given for each harmonic and, if specified by the element property entry, they are given at physical points around the circumference.

The stresses are calculated at the centroid of the cross section for the TRIAAX element. For the TRAPAX element, the stresses are calculated at the four corners as well as at a point located at an average radius and height. The stress output for these elements is similar to that for the CONEAX element previously described. Solution results for each element include the Fourier coefficients of stress for each harmonic followed by the stresses at the angular locations specified by the element's property entry. Stresses are calculated at the ***centroid*** of the cross section

Table 20-2. DISPLACEMENT RESULTS

D I S P L A C E M E N T V E C T O R

SECTOR-ID / POINT-ID / RING-ID	HARMONIC	T1	T2	T3	R1	R2	R3
11		-2.737490E-02	0.0	0.0	0.0	0.0	0.0
111	0	-5.503183E-04	0.0	0.0	0.0	0.0	0.0
111	1	-2.031799E-03	1.084670E-03	0.0	0.0	0.0	0.0
111	2	-1.176937E-03	3.645807E-03	0.0	0.0	0.0	0.0
111	3	-8.373315E-03	3.417965E-03	0.0	0.0	0.0	0.0
111	4	-4.121537E-03	1.481668E-03	0.0	0.0	0.0	0.0
111	5	-2.144611E-03	7.382611E-04	0.0	0.0	0.0	0.0
111	6	-1.222256E-03	4.213114E-04	0.0	0.0	0.0	0.0
111	7	-7.439750E-04	2.626879E-04	0.0	0.0	0.0	0.0
111	8	-4.762427E-04	1.742432E-04	0.0	0.0	0.0	0.0
111	9	-3.168369E-04	1.207640E-04	0.0	0.0	0.0	0.0
111	10	-2.170748E-04	8.635721E-05	0.0	0.0	0.0	0.0

111 is a RINGAX **11 is a POINTAX**

for the TRIAAX element. For the TRAPAX element stresses are calculated at the four corners, as well as at a point which is located at an average radius and height.

Table 20-3 illustrates the element force and stress output for the TRAPAX element. This example shows the stress components printed for each harmonic and how the results may be obtained a specified azimuthal angles by defining stress recovery points.

Table 20-3. HARMONIC ELEMENT SOLUTION RESULTS

a. Element Forces

F O R C E S I N A X I S - S Y M M E T R I C T R A P E Z O I D A L R I N G E L E M E N T S (TRAPAX)

ELEMENT ID.	HARMONIC NUMBER	POINT ANGLE	RADIAL (R)	CIRCUMFERENTIAL (THETA-T)	AXIAL (Z)
111	0		-1.403322E-13	0.0	-2.500498E+01
			-9.426515E-01	0.0	-1.552611E+01
			-4.343546E-01	0.0	1.606551E+01
			2.142603E-01	0.0	2.446559E+01
111	1		5.102585E-13	-3.106876E-12	-1.794263E+01
			-9.090184E-01	5.073786E-01	-7.839157E+00
			-4.018128E-01	-5.088683E-01	8.395775E+00
			2.208542E-01	-1.088487E+00	1.741305E+01
111		0.0000	-4.649392E-12	0.0	-2.221671E+02
			-1.792078E+00	0.0	-1.059951E+02
			2.718485E+00	0.0	1.122878E+02
			2.539801E+00	0.0	2.181004E+02
111		7.1000	-4.389357E-12	-2.597407E-12	-1.765620E+02
			-1.540176E+00	3.271304E+01	-8.064883E+01
			2.125978E+00	2.383652E+01	8.557304E+01
			1.980796E+00	-1.031115E+01	1.731622E+02

b. Element Stresses

S T R E S S E S I N A X I S - S Y M M E T R I C T R A P E Z O I D A L R I N G E L E M E N T S (TRAPAX)

ELEMENT ID.	HARMONIC NUMBER	POINT ANGLE	RADIAL (R)	AXIAL (Z)	CIRCUMFERENTIAL (THETA-T)	SHEAR (ZR)	SHEAR (RT)	SHEAR (ZT)
111	0		2.657005E-01	2.472376E+00	-1.708777E+00	3.263068E-01	0.0	0.0
			-3.326821E-01	1.042685E+00	-2.273694E+00	3.284751E-01	0.0	0.0
			-3.087723E-01	1.060090E+00	-2.239587E+00	-2.950068E-01	0.0	0.0
			2.897422E-01	2.489913E+00	-1.674362E+00	-2.971751E-01	0.0	0.0
			-2.139594E-02	1.766373E+00	-1.973855E+00	1.565000E-02	0.0	0.0
111	1		5.906742E-01	3.763975E+00	-3.048223E+00	6.863518E-01	3.529457E-02	-8.019427E-02
			-7.187436E-01	7.025689E-01	-4.351542E+00	6.907307E-01	3.224559E-02	-7.019072E-02
			-6.720896E-01	7.360861E-01	-4.286472E+00	-6.232604E-01	3.147447E-02	-2.719917E+00
			6.371782E-01	3.797342E+00	-2.983502E+00	-6.276394E-01	3.462341E-02	-2.752950E+00
			-4.072536E-02	2.250013E+00	-3.667389E+00	3.154567E-02	3.339627E-02	-1.405764E+00
111		0.0000	1.313529E+01	4.710047E+01	5.753896E+01	5.531201E+00	0.0	0.0
			-1.315031E+01	6.769316E+00	1.025145E+01	5.583740E+00	0.0	0.0
			-1.283482E+01	6.927187E+00	1.046220E+01	-4.950428E+00	0.0	0.0
			1.337699E+01	4.718456E+01	5.757754E+01	-5.002966E+00	0.0	0.0
			6.463294E-02	2.692823E+01	3.380084E+01	2.903869E-01	0.0	0.0
111		7.1000	1.037557E+01	3.702735E+01	4.153640E+01	4.652960E+00	1.946362E+00	-7.129729E-01
			-1.041030E+01	4.392392E+00	4.885123E+00	4.692707E+00	2.407175E+00	-5.096024E-01
			-1.014813E+01	4.535322E+00	5.099382E+00	-4.194110E+00	2.382852E+00	-5.057777E+01
			1.059009E+01	3.712262E+01	4.163946E+01	-4.233857E+00	1.926544E+00	-5.121476E+01
			5.105946E-02	2.071867E+01	2.317160E+01	2.294247E-01	2.167693E+00	-2.575285E+01

Summed element stresses at requested azimuthal coordinates.

Stress components for each harmonic used in the analysis.

20.5 MODELING GUIDELINES

The axisymmetric harmonic modeling feature of **UAI/*NASTRAN*** described to you so far, can be applied to many practical engineering problems. These include solid and hollow structural components one frequently encounters in aerospace and automobile industries. Other applications include structures such as reactor vessels, cooling towers etc., subjected to unsymmetric loads such as wind forces. The example discussed in this section deals with an axisymmetric shell subjected to ***asymmetric*** pressure loading.

The geometry definition for analyses of axisymmetric problems is made very simple in **UAI/*NASTRAN***. You start by defining rings on **RINGAX** Bulk Data entries which essentially define the locus of the axisymmetric element nodes. The axisymmetric elements are then defined using either shell or solid elements, depending on the type of structure. The loads are defined either with spatial data or in a harmonic form, whereas the constraints must be defined by the particular Fourier harmonics.

The number of harmonics to be considered in the analysis depends on the problem at hand. If the loading applied is such that only the symmetric radial motions occur, then you must use 0th harmonic ***only***. For a general loading, more number of harmonics result in an accurate computation of the deformations of the structure. However, if you use too many harmonics in your solution, then computational cost increases.

20.6 EXAMPLE PROBLEM

This example demonstrates the use of solid-of-revolution elements subjected to ***asymmetric*** loading. A cylindrical shell subjected to uniform external pressure over a small area is analyzed using **UAI/*NASTRAN***. Trapezoidal axisymmetric solids (TRAPAX) are used to model this structure.

Example Problem 20-1

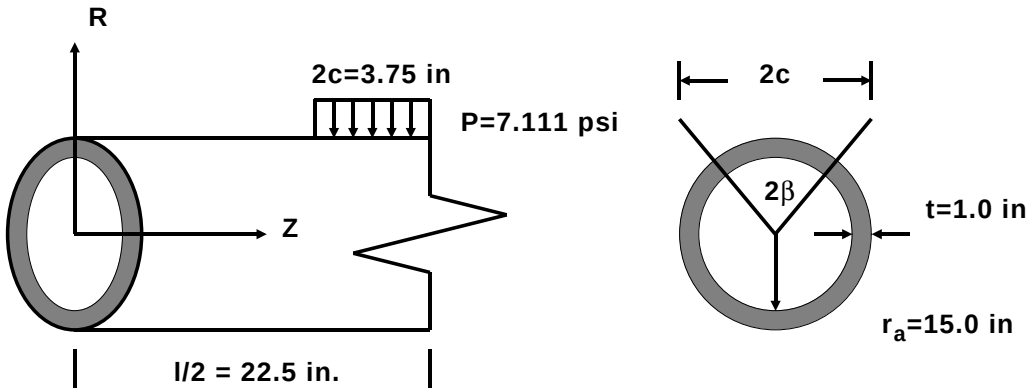
A simply supported cylindrical shell of radius r_a an length *l*, shown in Figure 20-3, is subjected to asymmetric external pressure load P acting over a length and width both of which are 2c. You wish to compute the displacements and stresses under the center of the load. Represent the deformations of the cylinder by its first ten harmonics.

The cylinder and its physical characteristics are shown in Figure 20-3. The input data stream for this problem is found in file **AXIH**. In developing the finite element model for this problem, notice that you can take advantage of the load symmetry of the cylinder along its length, in addition to the axisymmetric geometry. The cylinder in its upright position has the Z-axis along the length, and the radial R-axis in the horizontal direction. The finite element model of the upper half of the structure is shown in Figure 20-4. Trapezoidal elements have been used, and the mesh has been made coarser away from the loaded region.

Since the cylinder is simply supported at the ends, the radial and circumferential displacements and bending moments at the supports are zero, i.e, $u_r = u_\varphi = 0$. At the plane of symmetry, the z-displacements are zero, i.e., $u_z = 0$. You can now use the rings to define axisymmetric elements.

In the **PTRAPAX** Bulk Data entry, you can also input azimuthal coordinates (φi) for stress recovery. For this problem, the angle subtended at the center of the cylinder and the ends of the load, 2β, are selected for φi .

Figure 20-3. FINITE ELEMENT MODEL OF THE SHELL



E = 6.667x10^4 **psi**
ν = 0.3

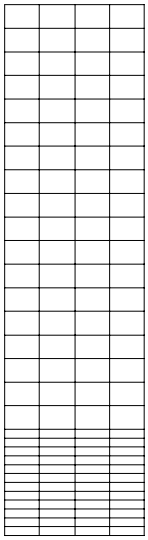
The input for the **PTRAPAX** Bulk Data entry is therefore:

PTRAPAX	5		15	.0	7.162				

The loads and deflections are represented by Fourier series expansions with respect to the azimuthal axis. The deflections are represented by a Cosine series using the **AXISYMMETRIC** Case Control command, since the loaded structure has only one plane of symmetry ($\varphi = 0$). The **AXIC** Bulk Data entry is used to specify the number of harmonics.

The pressure load is defined using the **PRESAX** Bulk Data entries, referenced by **LOAD = 20** Case Control command shown above. The given pressure loading of p = 7.11111 psi, is applied on the rings using **PRESAX** Bulk Data entries.

Figure 20-4. FINITE ELEMENT MODEL



In the Case Control packet, notice that we requested output for all harmonics. In order to obtain results under the center of the applied load, you need to define physical points with **POINTAX** Bulk Data entries.

The input data is then executed in **UAI/*NASTRAN***. The theoretical displacement and stresses, given in [Bjilaard54], which occur at the center of the load, $z = l/2$, $\varphi = 0$, are:

u_r = 0.0272 in (Inward Radial Deflection)

$\sigma_z = -15.47 \pm 63.42$ psi

$\sigma_\varphi = -17.42 \pm 79.44$ psi

where the stress values are given at the inside wall (+) and the outside wall (-). The radial deformation computed by **UAI/*NASTRAN*** is 0.0275 in, which differs from the theoretical solution by about 1%.

Figure 20-6 and 20-7 show the axial and circumferential stresses for the elements at the centerline of the structure under the center of the load plotted as a function of radial position. Agreement with the theoretical stresses is good.

Note that for this problem, ten harmonics were used. The number of harmonics effects the solution. The more complex the expected deformation field, the more harmonics are needed to adequately represent the shape.

Figure 20-5. AXIAL STRESSES UNDER THE LOAD

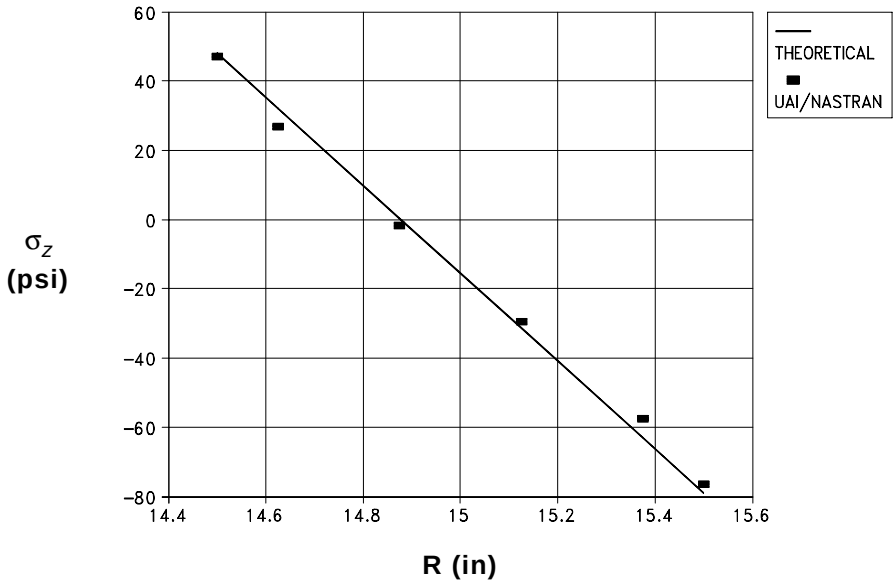
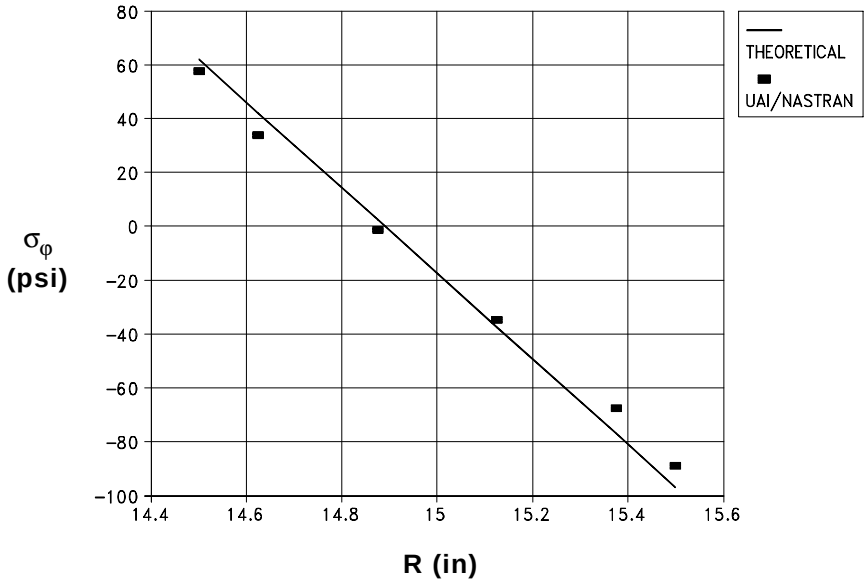


Figure 20-6. CIRCUMFERENTIAL STRESSES UNDER THE LOAD



20

This page is intentionally blank.

Chapter 21

21

AXISYMMETRIC RING MODELING

UAI/*NASTRAN* features three different modeling capabilities for structures that are symmetric with respect to one or more axes. The previous chapter explained the Axisymmetric Harmonic modeling capability. In special cases when all loads and boundary conditions are symmetric, you may use the axisymmetric ring modeling feature effectively. Furthermore, the axisymmetric ring modeling may be used when you perform Heat Transfer analyses, described in Chapter 24, whereas axisymmetric harmonic modeling may not.

21.1 BACKGROUND

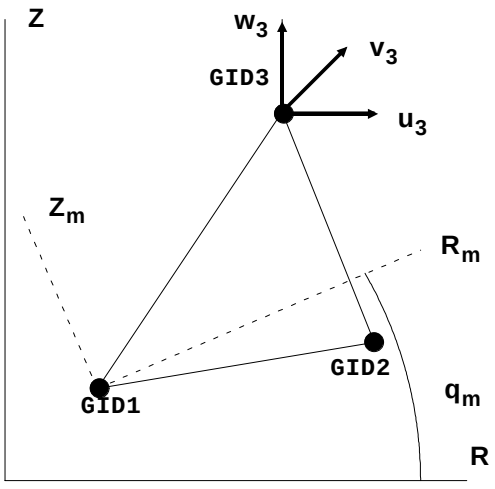
Models of axisymmetric structures which have only symmetric loads may be constructed using three special axisymmetric ring elements. Other required data such as constraints and loads, are defined using the normal Bulk Data entries. The model is completely defined by a planar section in the R-Z plane and care must be taken to describe loads and boundary conditions with this in mind.

There are three available Axisymmetric Ring elements: a shell of revolution element, the TORDRG, and two solids of revolution, TRIARG and TRAPRG. Together, these elements provide you the flexibility needed to model axisymmetric components with different sectional shapes. You may not combine general structural elements with these special elements. These elements as well as the Bulk Data entries required, are presented in the following sections.

Note that unlike the harmonic elements, the ring elements are connected by GRID points in the normal fashion. Properties, loads, and responses of axisymmetric ring elements are assumed to be symmetric with respect to the Z-axis. For axisymmetric ring models, the basic coordinate system is the cylindrical coordinate system described. As a result, the definition of the component motions for these elements is different than for general three dimensional models. These elements are described in more detail in the following Sections.

21.1.1 Triangular Ring Element

Figure 21-1. TRIARG GEOMETRY



The coordinate system for the triangular ring element, TRIARG, is illustrated in Figure 21-1. Deformations of the element are described in terms of translations in the **r** and **z** directions at each of the three GRID points defining the element. All other degrees of freedom are undefined and as a result should be constrained in your model.

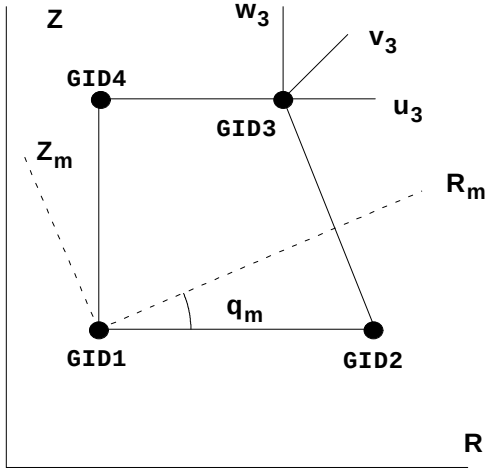
The triangular ring element is defined with a **CTRIARG** Bulk Data entry. No property entry is used for this element. The material property reference is given directly on the connection entry. The three GRID points, **GID1**, **GID2**, and **GID3**, must be specified in a counter-clockwise fashion around the element. They must also lie in the r-z plane of the basic cylindrical coordinate system on, or to the right of, the axis of symmetry. You may use cylindrically orthotropic materials, those defined with **MAT3** Bulk Data entries, with this element. When you do so, the orientation of the orthotropic axes in the r-z plane is specified by the angle θ_m.

21.1.2 Trapezoidal Ring Element

The coordinate system for the trapezoidal ring element is shown in Figure 21-2. This element is similar to the triangular ring element, but is has the ***additional restriction*** that the element numbering must begin at the lower left hand corner of the element. Also, the parallel sides of the trapezoid must be perpendicular to the axis of symmetry. The TRAPRG can be used in the limiting case where the r-coordinates associated with GRID points **GID1** and **GID4** are zero. In this special case, the element is referred to as a ***core element***.

The trapezoidal ring element is defined with a **CTRAPRG** Bulk Data entry and, again, has no special property entry.

Figure 21-2. TRAPRG GEOMETRY



21.1.3 Toroidal Ring Element

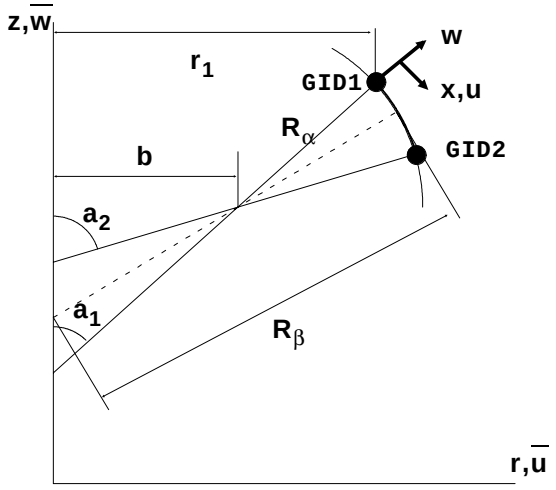
The toroidal ring element is formulated in a curvilinear coordinate system as shown in Figure 21-3. The motions of GRID points defining TORDRGs are limited to ***five*** degrees of freedom. The two translations, $\bar{u}$ and $\bar{w}$ are measured in the radial and axial directions of the cylindrical systems. The three rotational degrees of freedom have a special meaning. The R1 component represents the slope, u', in the element ξ direction, the R2 component represents the strain in the ξ direction, and the R3 component is a measure of the curvature in the z-ξ plane.

When you create an Axisymmetric Ring model, you must constrain the second translational degree of freedom at each GRID point. Any GRID points of the TORDRG element which are connected to TRIARG and TRAPRG, must also have the rotational degrees of freedom constrained. The displacements $\bar{u}$ and $\bar{w}$ may be obtained in the basic coordinate system, or you may select some other coordinate system in the usual manner. However, GRID point solution quantities for the three rotational degrees of freedom are always output in the element coordinate system.

When you use the axisymmetric elements, **UAI/*NASTRAN*** always uses a cylindrical coordinate system, so it is not necessary for you to define an explicit cylindrical system. The toroidal element may use isotropic, **MAT1**, or orthotropic, **MAT3**, materials. The axes of orthotropy are assumed to coincide with the element coordinate axes.

The toroidal ring element is completely defined by the **CTORDRG** and **PTORDRG** Bulk Data entries. The element is defined by two GRID points, **GID1** and **GID2**. The GRID points must lie in the r-z plane of the basic cylindrical coordinate system and they must lie on or to the right of the axis of symmetry. $\alpha 1$ and $\alpha 2$ are the angles of curvature of the element. They are defined as the angle, in degrees, from the axis of symmetry to a line which is perpendicular to the tangent to the surface at GRID points **GID1** and **GID2** respectively. For conic rings, $\alpha 1 = \alpha 2$, and for cylindrical rings $\alpha 1 = \alpha 2 = 90^o$. Toroidal ring elements may be connected to form closed figures in the r-z plane, but slope discontinuities are not permitted at connection points.

Figure 21-3. TORDRG GEOMETRY



21.2 INPUT DATA REQUIREMENTS

This section describes the input data necessary for performing an Axisymmetric Ring Analysis.

21.2.1 Executive Control Commands

There are no special Executive Control commands required for axisymmetric ring analysis. Note that the axisymmetric elements may not be used when performing Substructuring, Cyclic Symmetry, Axisymmetric Harmonic Analysis, and Nonlinear Material Analysis.

21.2.2 Case Control Commands

Axisymmetric Ring Analysis does not require any special Case Control commands. Output requests are made with the usual Case Control commands.

21.2.3 Bulk Data

As introduced earlier, the only Bulk Data entries specific to Axisymmetric Ring modeling are those required to define the ring elements themselves.

21.3 SOLUTION RESULTS

The usual Case Control commands such as **DISP**, **FORCE**, **STRESS**, are used to request output results. The following sections describe the interpretation of solution results.

21.3.1 Displacement

For the models using the TRIARG and TRAPRG elements the output displacements are interpreted in the normal manner. If your model contains TORDRG elements, then all GRID points connected to these elements are interpreted in the manner described earlier.

21.3.2 Element Forces

For the solid ring elements, TRIARG and TRAPRG, the radial and axial forces at each connected GRID point are available. The positive directions for these forces are shown in Figures 21-4 and 21-5. These are apparent element forces and they include any equivalent thermal loads. The element forces for the TORDRG element, and their sign conventions, are shown in Figure 21-6. The results, evaluated at each end of the element, are the radial force, axial force, meridional moment, a generalized force which corresponds to the strain degree of freedom, and a generalized force which corresponds to the curvature degree of freedom. Remember that the first three forces are in the global coordinate system and the two generalized forces are in the element coordinate system. The element force results are shown in Table 21-1.

21.3.3 Element Stresses

Table 21-2 shows sample stress output for the Axisymmetric Ring elements. For each TRIARG element the stresses results are printed at the centroid of the element. For the TRAPRG element, the stresses are provided at the four connected GRID points as well as the centroid. The stresses include the normal stresses in the radial, circumferential, and axial directions, and the shear stress on the radial face in the axial direction. The element stresses for the TORDRG element, and their sign conventions, are also shown in Figure 21-6. These stresses, which are

Figure 21-4. TRIARG ELEMENT FORCES

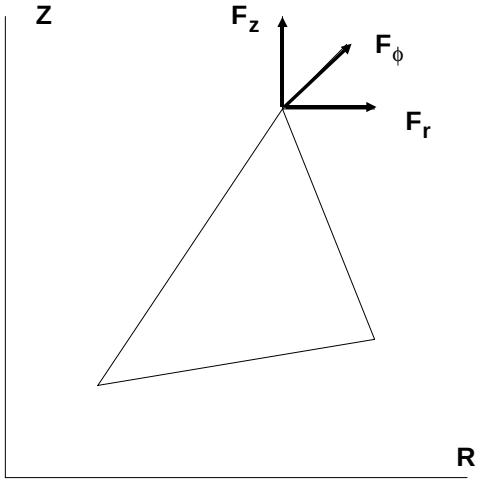


Figure 21-5. TRAPRG ELEMENT FORCES

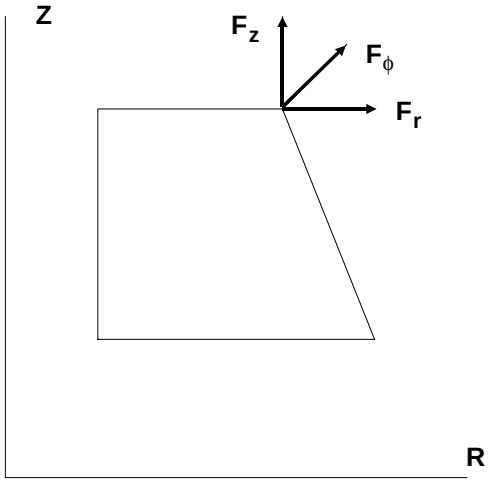
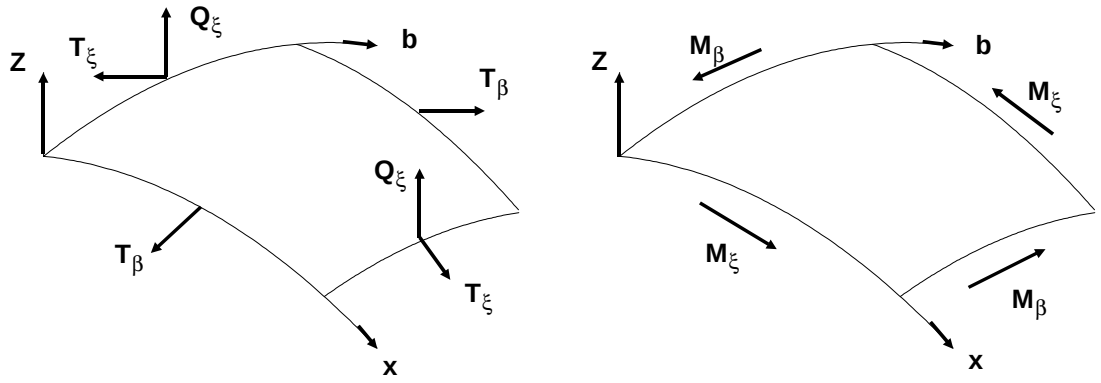


Figure 21-6. TORDRG STRESS RESULTANTS



evaluated at both ends and the midspan of each element, are the tangential membrane stress, given as force per unit length, circumferential membrane stress, also force per unit length, tangential bending stress, moment per unit length, circumferential bending stress, moment per unit length, and shear stress, again force per unit length.

Table 21-1. AXISYMMETRIC RING ELEMENT FORCES

a. Element Forces for the TRIARG

F O R C E S F O R T H E T R I A N G U L A R R I N G S (C T R I A R G)

EL ID	CORNER POINT	RADIAL (X)	CIRCUMFERENTIAL (THETA)	AXIAL (Z)
101	1	2.294324E+02	0.0	3.605169E-01
	2	-2.159320E+02	0.0	8.169623E-01
	3	6.479801E+00	0.0	-1.177479E+00

b. Element Forces for the TRAPRG

F O R C E S F O R T H E T R A P E Z O I D A L R I N G S (C T R A P R G)

EL ID	CORNER POINT	RADIAL (X)	CIRCUMFERENTIAL (THETA)	AXIAL (Z)
101	1	2.356000E+02	0.0	6.056666E-11
	2	-2.148304E+02	0.0	1.535598E+00
	3	-2.157339E+02	0.0	-2.614426E+00
	4	2.354213E+02	0.0	1.078828E+00

c. Element Forces for the TORDRG

F O R C E S F O R T H E T O R O I D A L R I N G S (C T O R D R G)

EL ID	CORNER POINT	RADIAL (X)	CIRCUMFERENTIAL (THETA)	AXIAL (Z)	MOMENT (ZX)	DIRECT STRAIN (XI)	CURVATURE (XI,XI)
1	1	0.0	0.0	-8.857395E+00	0.0	-2.208491E-04	2.073072E-03
	2	-9.359089E+02	0.0	8.895553E+00	-1.208687E+01	3.972204E-03	-2.250222E+00
2	1	9.337456E+02	0.0	-7.085145E+01	2.695217E+01	-3.603190E-03	-7.947493E+00
	2	-1.870682E+03	0.0	7.088524E+01	-4.478873E+01	4.961243E-02	-8.136202E+00
3	1	1.862039E+03	0.0	-1.944918E+02	5.958523E+01	-4.919587E-02	-1.224862E+01
	2	-2.799089E+03	0.0	1.945188E+02	-1.015593E+02	-3.891885E-03	-1.308389E+01

Table 21-2. AXISYMMETRIC RING ELEMENT STRESSES

a. Element Stresses for the TRIARG

S T R E S S E S F O R T H E T R I A N G U L A R R I N G S (C T R I A R G)

EL ID	RADIAL (X)	CIRCUMFERENTIAL (THETA)	AXIAL (Z)	SHEAR (ZX)
101	-9.199218E+01	2.035164E+02	-9.724625E-01	-1.488728E-01
102	-8.088476E+01	1.922748E+02	-8.320206E-01	2.340684E-02
103	-7.104300E+01	1.822314E+02	-7.778947E-01	6.109322E-02
104	-6.222620E+01	1.732510E+02	-7.457213E-01	5.183109E-02
105	-5.427525E+01	1.652002E+02	-7.149724E-01	2.814834E-02

b. Element Stresses for the TRAPRG

S T R E S S E S F O R T H E T R A P E Z O I D A L R I N G S (C T R A P R G)

EL ID	STRESS POINT	RADIAL (X)	CIRCUMFERENTIAL (THETA)	AXIAL (Z)	SHEAR (ZX)
101	1	-8.902100E+01	2.166734E+02	3.489161E+00	-5.483640E-01
	2	-9.765656E+01	1.958101E+02	-4.432712E+00	-6.134093E-01
	3	-9.807610E+01	1.956984E+02	-4.592065E+00	1.002820E-01
	4	-8.943092E+01	2.165842E+02	3.339430E+00	1.653272E-01
	5	-9.363977E+01	2.059731E+02	-6.426756E-01	-2.240410E-01

c. Element Stresses for the TORDRG

S T R E S S R E S U L T A N T S F O R T H E T O R O I D A L R I N G S (C T O R D R G)

EL ID	STRESS POINT	MEMBRANE (FORCES) TANGENTIAL	MEMBRANE (FORCES) CIRCUMFERENTIAL	FLEXURE (MOMENTS) TANGENTIAL	FLEXURE (MOMENTS) CIRCUMFERENTIAL	SHEAR (FORCE)
1	1	-3.803334E+01	8.833459E+00	1.841608E+00	-1.841608E+00	6.118080E+00
	2	-4.740751E+01	-4.740755E+01	-3.820555E-01	-6.670842E-02	-2.762749E-01
	3	-4.740788E+01	-4.741441E+01	-1.462554E-01	-3.919497E-02	1.320863E+00
2	1	-4.740997E+01	-4.745204E+01	-1.462544E-01	-3.919544E-02	-3.589890E-01
	2	-4.742402E+01	-4.745127E+01	1.607639E-01	3.351545E-02	-1.024575E-01
	3	-4.742612E+01	-4.745460E+01	3.496951E-01	1.183697E-01	-2.864588E-01
3	1	-4.743487E+01	-4.745604E+01	3.496955E-01	1.183708E-01	-1.367781E-01
	2	-4.743534E+01	-4.745527E+01	5.680695E-01	2.268141E-01	-2.124462E-01
	3	-4.743887E+01	-4.744284E+01	8.559441E-01	3.623493E-01	-2.404461E-01

21.4 MODELING GUIDELINES

Since the loads and deformations must be symmetric about the longitudinal axis, the input data preparation for Axisymmetric Ring Analysis is quite simple. The basic finite element model is simply built in the r-z plane using all of the techniques which apply to normal two dimensional finite element modeling. The ring model is then completely defined by this planar section. You must remember that, by definition, all of the loads on the structure and all of the boundary conditions may only be axisymmetric. For similar structures whose loadings or boundary conditions are asymmetric, you should use either Axisymmetric Harmonic Analysis, described in Chapter 20, or Cyclic Symmetry Analysis, described in Chapter 22.

21.5 EXAMPLE PROBLEM

This Section provides an example which illustrates how you perform an analysis using Axisymmetric Ring elements.

Example Problem 21-1

A thick-walled cylindrical shell having an outer radius of 5 in and an inner radius of 3 in is subjected to a uniform pressure of 100 psi acting outward on its inner surface. You wish to determine the radial and circumferential stresses in the structure.

The cylinder and its physical characteristics are shown in Figure 21-7. The input data stream for this problem is found in file **AXIR**. The finite element model is comprised of TRAPRG elements, as shown in Figure 21-8. There are no special Bulk Data entries for defining pressure on an Axisymmetric Ring model. Therefore, the load must be expressed in terms of GRID point loads. This is done by computing the area of the inner surface of the cylinder, determining the total load required for that area, and distributing it to the GRID points. The resulting load distribution is also shown in Figure 21-8.

In order to fix this model in space, the axial displacements at the midplane have been constrained. The model was then executed in **UAI/*NASTRAN*** Figure 21-9 shows the resulting radial and circumferential stresses plotted as a function of the radial position through the wall of the cylinder. Agreement with the theoretical results, taken from [Young89] are excellent.

Figure 21-7. THICK CYLINDER

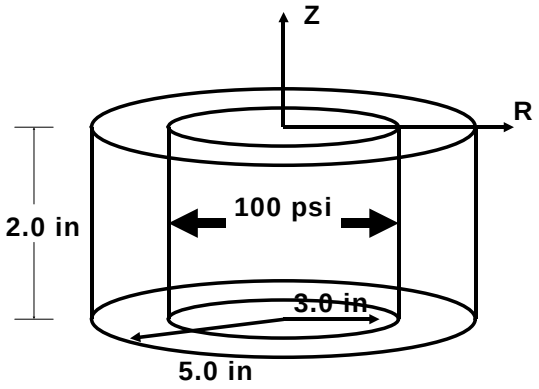


Figure 21-8. FINITE ELEMENT MODEL

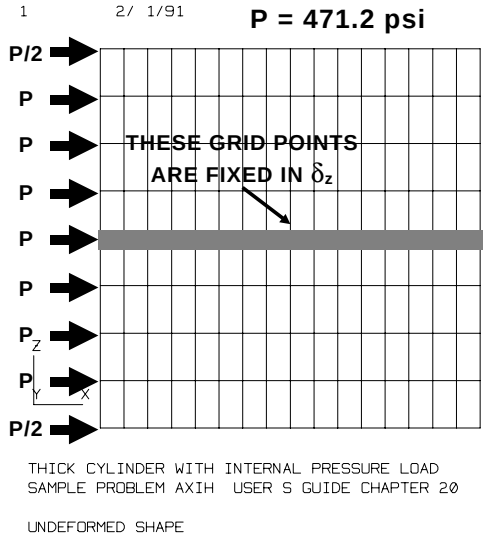
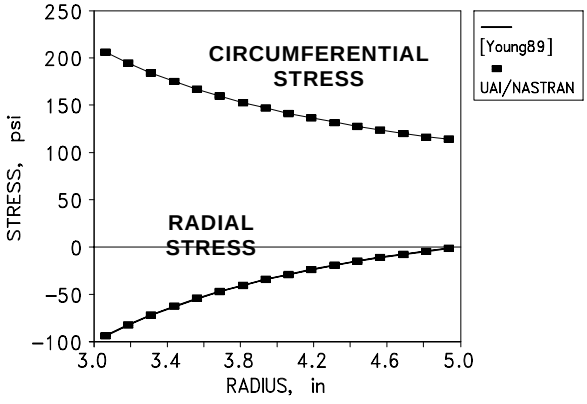


Figure 21-9. SOLUTION RESULTS



21

This page is intentionally blank.

Chapter 22

CYCLIC SYMMETRY ANALYSIS

Many structures, including pressure vessels, rotating machines, and antennae for space communications, are made up of virtually identical segments that are symmetrically arranged with respect to an axis. This type of geometry is said to have ***Cyclic Symmetry***. **UAI/*NASTRAN*** provides you the capability to analyze these practical structures in three analytical disciplines which are summarized in Table 22-1. This chapter describes how Cyclic Symmetry Analysis is performed.

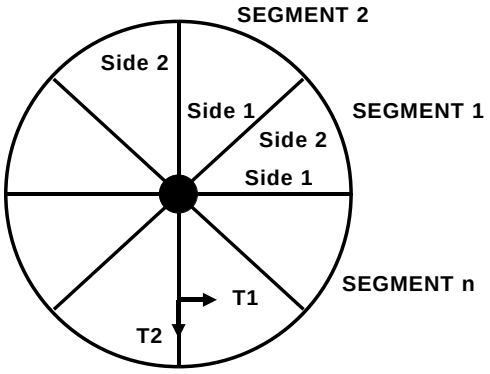
Table 22-1. SUMMARY OF STATICS RIGID FORMATS

RIGID FORMAT	DESCRIPTION
14	Static Analysis with Cyclic Symmetry
15	Normal Modes with Cyclic Symmetry
16	Buckling Analysis with Cyclic Symmetry

22.1 TERMINOLOGY

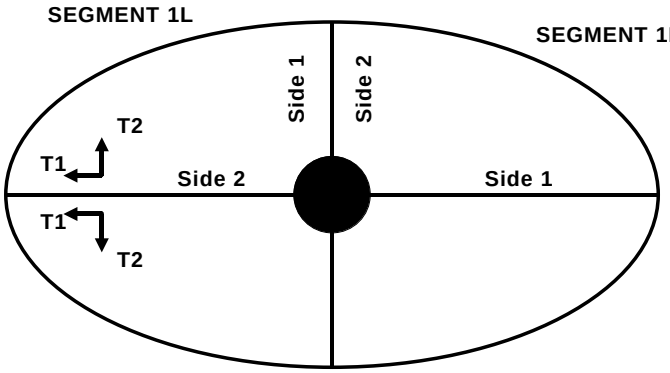
There are two types of Cyclic Symmetry. The first, called ***rotational symmetry,*** is shown in Figure 22-1. Rotationally symmetric structures are those whose geometry and physical properties repeat at even angular intervals about an axis. As a result, structures which display this symmetry only require one segment to be modeled. Each segment has its own coordinate system that rotates with it. The boundaries of the segment may be curved, but the local coordinate systems must conform where segments join as shown in the Figure. The segments are defined by their sides, as noted.

Figure 22-1. ROTATIONAL SYMMETRY



The second type, called ***dihedral symmetry***, is a special form of rotational symmetry. In this case, each of the segments has an internal plane of reflective symmetry as illustrated in Figure 22-2. For structures which exhibit this property, ***only the right half-segment*** of the structure is modeled, a right or **R** segment. The left or **L** segments are mirror images of the right segments. Again, each segment has its own coordinate system that rotates with it. The left segments have left handed coordinate systems. Segment boundaries must be planar, and local displacement coordinate systems between left and right half segments must be in, or normal to, the plane of reflection.

Figure 22-2. DIHEDRAL SYMMETRY



22.2 MATHEMATICAL BACKGROUND

As introduced in the previous section, Side 2 of segment *n* is by definition connected to Side 1 of segment *n+1*. Thus, the displacement components on the adjoining sides satisfy the relationship:

$$u_1^{n+1} = u_2^n \quad \text{for } n = 1,\ldots,N \tag{22-1}$$

where the subscript indicates the Side number and the superscript the segment number. All responses, *r*, including deformations, forces, and stresses, of the structure may then be defined by a Fourier series of the form:

$$r^n = \begin{cases} \bar{r}^0 + \sum_{k=1}^{(n-1)/2} \bar{r}^{ks}\sin(k\theta_n) + \bar{r}^{kc}\sin(k\theta_n) & \text{for } N \text{ odd} \\ \bar{r}^0 + \sum_{k=1}^{(n-2)/2} \bar{r}^{ks}\sin(k\theta_n) + \bar{r}^{kc}\sin(k\theta_n) + (-1)^{(n-2)/2}\,\bar{r}^{N/2} & \text{for } N \text{ even} \end{cases} \tag{22-2}$$

where:

$$\theta_n = \frac{2\pi(n-1)}{N} \quad \text{for } n = 1,\ldots,N \tag{22-3}$$

The coefficients $\bar{r}^0$, $\bar{r}^{kc}$, $\bar{r}^{ks}$ and $\bar{r}^{N/2}$ are called symmetrical components. These components are used as generalized coordinates when performing the analysis. You may notice that the upper limit of the summation is determined by the number of segments. Since each segment is represented by both a *sin* and *cos* term, the maximum number of harmonics is approximately one-half of the number of segments.

The equations for dihedral symmetry differ from those for the rotational case. For right segments, they are:

$$r^{nR} = \begin{cases} \sum_{k=1}^{(n-1)/2} (\bar{r}^{kc} + \bar{r}^{kc*})\cos(k\theta_n) + (\bar{r}^{ks} + \bar{r}^{ks*})\sin(k\theta_n) & \text{for } N \text{ odd} \\ \sum_{k=1}^{(n-2)/2} (\bar{r}^{kc} + \bar{r}^{kc*})\cos(k\theta_n) + (\bar{r}^{ks} + \bar{r}^{ks*})\sin(k\theta_n) & \text{for } N \text{ even} \end{cases} \tag{22-4}$$

and for Left segments, they are:

$$r^{nL} = \begin{cases} \displaystyle\sum_{k=1}^{(n-1)/2} (\bar{r}^{kc} - \bar{r}^{kc*}) \cos\left(k\left(\theta_n + \frac{2\pi}{N}\right)\right) \\ \qquad + (\bar{r}^{ks} - \bar{r}^{ks*}) \sin\left(k\left(\theta_n + \frac{2\pi}{N}\right)\right) \quad \text{for } N \text{ odd} \\ \displaystyle\sum_{k=1}^{(n-2)/2} (\bar{r}^{kc} - \bar{r}^{kc*}) \cos\left(k\left(\theta_n + \frac{2\pi}{N}\right)\right) \\ \qquad + (\bar{r}^{ks} - \bar{r}^{ks*}) \sin\left(k\left(\theta_n + \frac{2\pi}{N}\right)\right) \quad \text{for } N \text{ even} \end{cases} \tag{22-5}$$

22

As before, the coefficients $\bar{r}^{kc}$, $\bar{r}^{kc*}$, $\bar{r}^{ks}$ and $\bar{r}^{ks*}$ are used as generalized coordinates. In both cases, the symmetrical components are uncoupled In the latter case, the starred terms, $\bar{r}^{kc*}$ and $\bar{r}^{ks*}$, represent antisymmetrical motion, while the unstarred terms, $\bar{r}^{kc}$ and $\bar{r}^{ks}$, represent symmetric motion.

You will also note from all of the equations presented, that as the number of harmonic increases, the magnitude of the terms in the Fourier series representation become smaller. As a result, accurate answers may be obtained using only a modest number of harmonics. The number required, however, is highly dependent on the structure and its anticipated behavior. By analogy, the number of harmonics used is similar to the number of harmonics used in Axisymmetric Harmonic analysis, described in Chapter 20, and the modal reduction procedure used in dynamic response analyses.

22.3 INPUT DATA REQUIREMENTS

This section deals with the input data requirements for performing Cyclic Symmetry analyses with **UAI/*NASTRAN***.

22.3.1 Executive Control Commands

The Cyclic Symmetry capability is selected with the **SOL** Executive Control Command. There are three Rigid Formats available. For static analysis, Rigid Format 14 is used, for normal modes, Rigid Format 15 is used, and for buckling, Rigid Format 16 is used. The Executive Control commands are specified as,

```
SOL 14   (Static Cyclic Symmetry)
SOL 15   (Modal Cyclic Symmetry)
SOL 16   (Buckling Cyclic Symmetry)
```

Cyclic Symmetry data may not be used with any other Rigid Formats, nor will it be accepted when using the Automated Substructuring feature.

22.3.2 Case Control Commands

Cyclic Symmetry uses four special Case Control commands as shown in Table 22-2. The command **HARMONICS** is ***required*** when performing Cyclic Symmetry analysis. The format of this command is:

```
HARMONICS = { nharm }
            { ALL   }
```

For Cyclic Symmetry, the number of harmonics, *nharm*, corresponds to the number of terms retained in the Fourier series expansion. Alternately, you may select **ALL** harmonics for the analysis.

The selection of solution results uses a combination of the normal output requests, such as **DISP**, **SPCF** and **STRESS**, and two special Cyclic Symmetry Case Control commands **HOUTPUT** and **NOUTPUT**. You use the **HOUTPUT** command:

```
HOUTPUT ( { C  } ) = { ALL }
        ( { S  } )   { sid }
        ( { C* } )
        ( { S* } )
```

to select the specific harmonics and symmetrical components which will be output for each request. You may select **ALL** harmonics, or you may define a **SET**, *sid*, which lists the specific

Table 22-2. CASE CONTROL COMMANDS FOR CYCLIC SYMMETRY

COMMAND	FUNCTION
HARMONICS	Specifies harmonics to be used
DSYM	Defines dihedral planes of symmetry
NOUTPUT	Print request for physical solution results
HOUTPUT	Print request for harmonic solution results

harmonics that will be output. The optional command argument further allows you to select some or all of the symmetrical components of the specified harmonics. If the argument is not present, than all components are printed: **C** and **S** in the case of **ROT**ational symmetry; and **C**, **S**, **C**, **S** in the case of **DIH**edral symmetry. If your **HOUTPUT** command specifies any component, then you must use an additional command for all of the symmetrical components for which output is desired.

Having selected specific harmonics, you may also specify particular segments for output using the **NOUTPUT** command:

NOUTPUT $\left[\left(\left\{\begin{matrix}\text{R}\\\text{L}\end{matrix}\right\}\right)\right] = \left\{\begin{matrix}\text{ALL}\\\textit{sid}\end{matrix}\right\}$

This time, the **SET** refers to a list of segment identification numbers for which you want output. When using **DIH**edral symmetry, you may also select right or left segments with the arguments **R** and **L**. Naturally, both half-segments will be output if you do not specify otherwise.

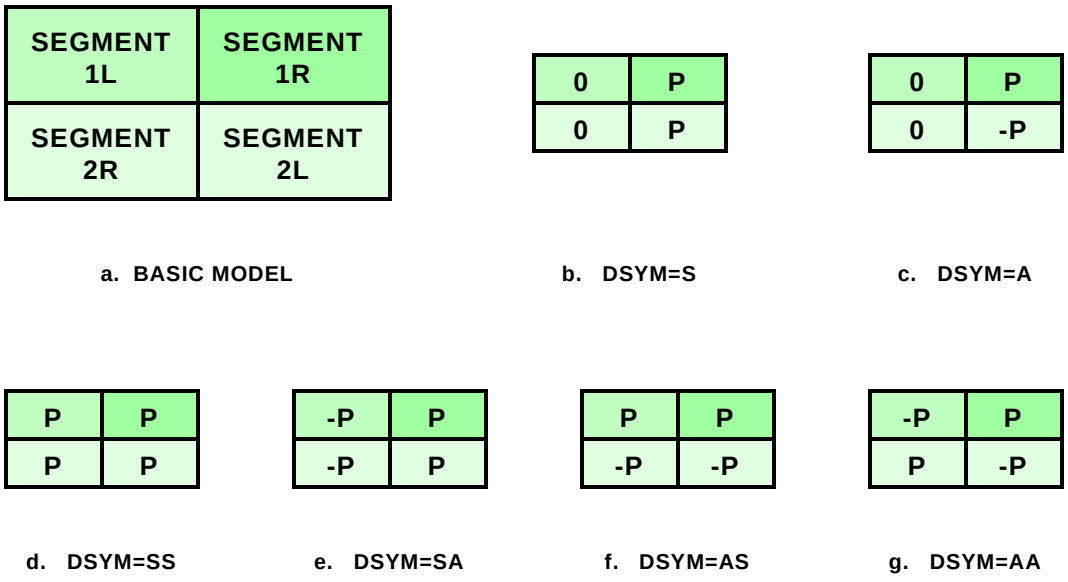
22

One further option is available for **DIH**edral symmetry which allows you to easily specify symmetric or antisymmetric boundary conditions or loadings when your structure has one or two planes of symmetry. This situation is indicated with the **DSYM** Case Control command:

DSYM $= \left\{\begin{matrix}\text{S}\\\text{A}\\\text{SS}\\\text{SA}\\\text{AS}\\\text{AA}\end{matrix}\right\}$

You may select a single symmetry plane, which may be **S**ymmetric or **A**ntisymmetric. This single plane of symmetry is ***always*** with respect to side 1, the boundary between segments. If you specify a ***second*** symmetry plane, it must be ***perpendicular*** to Side 1 and your model must contain an ***even number*** of segments. To understand how this works, consider Figure 22-3. The basic dihedral model is shown in (a). Suppose that you apply a load, P, to the basic segment 1R. If you specify one plane of symmetry, the total load applied to the structure will result in that shown in (b). On the other hand, if you specify antisymmetric loads, (c) will result. The situation becomes even more interesting with two planes of symmetry or antisymmetry or mixtures of each. All cases are shown in (d), (e), (f), and (g).

Figure 22-3. EXAMPLES OF THE DSYM COMMAND



The definition of SUBCASEs for a Cyclic Symmetry static analysis is generally the same as in statics. Any of the standard load request Case Control commands, such as **LOAD**, **TEMP(LOAD)**, or **DEFORM** may be used in each SUBCASE. For example, the Case control packet for a typical analysis is:

```
DISP = ALL
NOUTPUT = ALL
HARMONICS = ALL
SUBCASE 1
   LOAD = 101
   STRESS = ALL
SUBCASE 2
   TEMP(LOAD) = 201
   DISP = 301
BEGIN BULK
```

There are two restrictions to Case Control which you must remember:

- There can be no boundary condition changes from one SUBCASE to another. In other words, any **SPC** and **MPC** requests ***must*** be placed above the SUBCASE level.
- The **HARMONICS** command must appear above the SUBCASE level.

22.3.3 Bulk Data Entries

There are several Bulk Data entries that are used only for Cyclic Symmetry analyses. These are summarized in Table 22-3. A description of the use of each of these is presented in the following sections.

22.3.3.1 Defining the Model

The Cyclic Symmetry model requires the normal finite element model data and a special Bulk Data entry **CYSYM**:

CYSYM	NSEG	STYPE							

is used to define the number of segments in the model, **NSEG**, and specify the type of symmetry, **STYPE**, as either **ROT**ational or **DIH**edral.

22.3.3.2 Defining Segment Boundaries

Two Bulk Data entries are used to define the boundaries of a segment, **CYAX** and **CYJOIN**. The **CYAX** entry:

CYAX	GID1	GID2	GID3	GID4	GID5	GID6	GID7	GID8	

is used to define all GRID points which lie on the axis of symmetry. The **CYJOIN** Bulk Data entry is used to define all other boundary points of the segment:

CYJOIN	SIDE	NORM	GID1	GID2	GID3	GID4	GID5	GID6	-cont-
			GIDF	"THRU"	GIDL				
-cont-	GID7	GID8	*CONTINUES WITH LIST OF VALUES*						-cont-

The use of the **CYJOIN** entry varies depending on the type of symmetry. In the case of rotational symmetry, separate **CYJOIN** entries are used to define Sides 1 and 2 of the segment as illustrated in Figure 22-4. Note that GRID points lying on the axis of symmetry may not be included on a **CYJOIN** entry, but ***must*** be defined using **CYAX** Bulk Data as described above.

When using the dihedral symmetry option, the use of **CYJOIN** data is different. For this type of symmetry, Side 1 refers to the boundary between segments and Side 2 refers to the boundary in the middle of the segment. An example of this is shown in Figure 22-5.

For dihedral problems, the third field of the **CYJOIN** entry defines the coordinate direction that is normal to the common boundary.

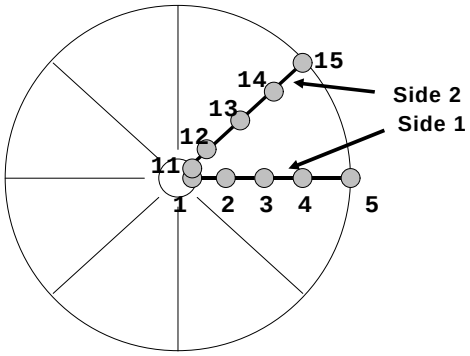
22.3.3.3 Defining Loads

The Cyclic Symmetry load requests do not point directly to the load bulk Data entries directly as they normally do. Instead, they point to special Bulk Data; **LOADCYH**, in the case of physical loads, and **LOADCYN**, in the case of harmonic loads.

Table 22-3. BULK DATA FOR CYCLIC SYMMETRY

BULK DATA	TO SPECIFY
CYSYM	Solution parameters
CYAX	Points on the axis of symmetry
CYJOIN	Segment boundary points
LOADCYN	Loads at physical points
LOADCYH	Harmonic loads

Figure 22-4. BOUNDARY DEFINITION, ROTATIONAL SYMMETRY



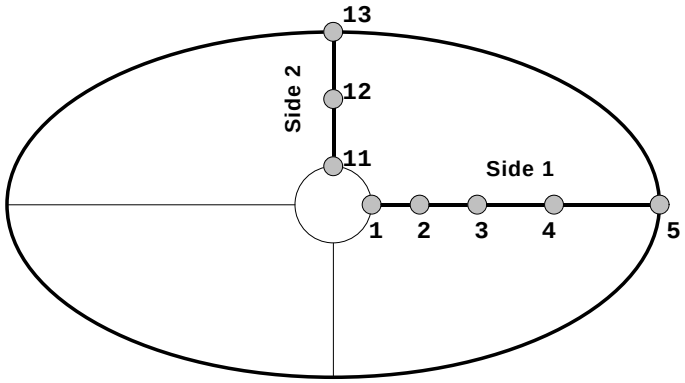
CYSYM	8	ROT							

The **LOADCYN** Bulk Data entry:

LOADCYN	SID	S	SEGID	SEGTYPE	S1	LID1	S2	LID2	

is used to select one or two loads which are applied to a specific segment which is defined by its segment identification number, **SEGID**. Actual physical loads, such as forces, moments or pressures, are referenced by their identification numbers, **LID1** and **LID2**. Note that you may also define individual scalar multipliers, **S1** and **S2**, and an overall multiplier, **S**, to be applied

Figure 22-5. BOUNDARY DEFINITION, DIHEDRAL SYMMETRY



CYSYM	2	DIH							

to the actual loads. The two inertial loads, **GRAV** and **RFORCE** may not be referenced by the **LOADCYN** entry. Instead, they must be selected with the harmonic load entry **LOADCYH**:

LOADCYH	SID	S	HID	HTYPE	S1	LID1	S2	LID2	

These Bulk Data are similar to the **LOADCYH**, but you must specify a single harmonic identification number, **HID**, for which the load will be applied. Additionally, you must specify which symmetric component, **HTYPE**, will be loaded. These types are indicated by **C**, **S**, **CSTAR**, and **SSTAR**. If you leave the **HTYPE** field blank, the load will be applied to all relevant components. The **HTYPE** field is also used to apply the **GRAV** and **RFORCE** inertial loads. To do this, you enter the appropriate keyword. When using this option, the **HID** field is not used — the inertia load is automatically applied to all harmonics.

You must remember that coordinate systems vary when using dihedral symmetry and that left half segments use left-handed coordinate systems. Special attention to cyclic transformation must be used when defining loads with directional vectors such as the **FORCE** and **MOMENT** Bulk Data entries.

22.4 SOLUTION RESULTS

As you saw in Section 22.4.2, **UAI/*NASTRAN*** allows you to select a combination of normal output requests and the two special Cyclic Symmetry Case Control commands, **HOUTPUT** and **NOUTPUT**. For a complete specification of output, you select the desired harmonics with **HOUTPUT**, the segments with **NOUTPUT**, and the specific GRID point or element response quantities within a segment in the usual manner with commands such as **DISP** and **STRESS**. Because each segment has identical numbering and either two or four symmetrical components of the solution, the ordering of the printed output results is quite different than normal. Table 22-3 shows the output sort order for the two types of symmetry. The table indicates the maximum output if all segments and harmonics are selected. If you reduce the amount of output with either or both **HOUTPUT** and **NOUTPUT** commands, then only the requested data will appear, but it will retain the order shown. Note that **SORT2** output is not available in Cyclic Symmetry analyses.

Because of the many levels of data identification, a special convention must be used when plotting solution results for individual segments. This convention defines ***coded*** SUBCASE identification numbers for each segment, and, for dihedral symmetry, each half-segment, and each harmonic. The general form of the coded identification numbers is:

```
coded_id = part_1 + part_2 + part_3
```

It is constructed from three terms which are different for physical and harmonic output. The terms are shown in the following table.

OUTPUT TYPE	SYMMETRY TYPE	*part_1*	*part_2*	*part_3*
PHYSICAL	ROTATIONAL	10000**sub_seq_num*	10**seg_num*	0
	DIHEDRAL			1 for Right Segment 2 for Left Segment
HARMONIC	BOTH	10000**sub_seq_num*	10**harm_num*	1 for Component C 2 for Component S 3 for Component C* 4 for Component S*

Note that the actual SUBCASE identification number is not used, but rather the *sub_seq_num*, or SUBCASE sequence number. This number maps the SUBCASEs to consecutive integers beginning with one. The other terms are the segment identification number, *seg_num* and the harmonic identification number, *harm_num*. The examples shown below illustrate how you may request plots using this notation.

OUTPUT TYPE	ROTATIONAL SYMMETRY	DIHEDRAL SYMMETRY
PHYSICAL (NOUTPUT)	SUBCASE 1 SEGMENT 1 RESPONSE 1 RESPONSE 2 ... SEGMENT 2 RESPONSE 1 RESPONSE 2 ... ... SEGMENT N ...	SUBCASE 1 SEGMENT 1R RESPONSE 1 RESPONSE 2 ... SEGMENT 1L RESPONSE 1 RESPONSE 2 ... ... SEGMENT NR ... SEGMENT NL ...
HARMONIC (HOUTPUT)	SUBCASE 1 SEGMENT 1,C RESPONSE 1 RESPONSE 2 ... SEGMENT 1,S RESPONSE 1 RESPONSE 2 ... SEGMENT 2,C ... SEGMENT 2,S ... ... SEGMENT N,C ... SEGMENT N,S ...	SUBCASE 1 SEGMENT 1R,C SEGMENT 1R,S SEGMENT 1R,C* SEGMENT 1R,S* SEGMENT 1L,C SEGMENT 1L,S SEGMENT 1L,C* SEGMENT 1L,S* SEGMENT 2R,C ... SEGMENT 2R,S* SEGMENT 2L,C ... SEGMENT 2L,S* ... SEGMENT NR,C ... SEGMENT NR,S* SEGMENT NL,C ... SEGMENT NL,S* ...

1. Assume that your model has rotational symmetry. Request deformed structural plots for the third and fourth SUBCASEs for segments 1 and 6.

```
PLOT STATIC DEFORMATION 30010,40060
```

2. Now, assume that you have requested your results in harmonic form. Request plots for the **S** component only for harmonics 3 and 6.

```
PLOT STATIC DEFORMATION -30032,-30062
```

3. This time, you have a model with dihedral symmetry. Request plots for the second and third SUBCASEs, segment 4, right and left halves.

```
PLOT STATIC DEFORMATION 20041,20042,30041,30042
```

4. Finally, assume you request harmonic results for your dihedral model. Request plots for the **C*** and **S*** components of harmonics 0 and 1, for the first SUBCASE.

```
PLOT STATIC DEFORMATION -10003,-10004,-10013,-10014
```

22.5 MODELING GUIDELINES

The use of the **UAI/*NASTRAN*** Cyclic Symmetry feature can, in many cases, significantly reduce computer time. It should be considered whenever highly symmetric structures are to be analyzed. Several specific guidelines to improve your use of Cyclic Symmetry are presented in this section.

22.5.1 Static Condensation (Guyan Reduction)

It is strongly recommended that a static condensation be performed on the fundamental region of your model whenever the number of boundary degrees of freedom are significantly less than the number of interior degrees of freedom. The greater the ratio of interior to boundary degrees of freedom, the greater the savings of execution time. The reduction is accomplished in the usual manner using **ASET** or **OMIT** bulk Data entries. All points on the axis of symmetry, those defined by **CYAX** data, must be retained during reduction. This reduction is also recommended for Normal Modes analysis. Remember, however, that the reduction in dynamics is not an exact representation, as it is for statics. You must select your retained degrees of freedom in a manner that models the dynamic behavior of the structure correctly.

22.5.2 Coordinate Systems

Remember that the coordinate systems vary when using dihedral symmetry and that left half-segments use left-handed coordinate systems. This must be considered when reviewing solution results.

22.5.3 Specifying Loads

The above warning relative to coordinate systems also applies to defining loads. It is recommended that loads of types **FORCE1**, **FORCE2**, **MOMENT1**, and **MOMENT2** be used because their orientation is defined by points fixed in space. Special attention to cyclic transformation must be used when defining loads with vectors such as **FORCE** and **MOMENT**.

22.5.4 Static Cyclic Symmetry Modeling

For structures with Cyclic Symmetry, obtaining theoretical solutions is often an unimaginable task. Therefore, it is important for the analyst to use engineering judgement in interpreting and verifying the analysis results. It is often possible to visualize the deformed configuration of the structure under static loading. The locations where the stresses are likely to be the highest can also be intuitively determined. You should make use of these checks to judge whether the finite element results are accurate.

Care should be taken to see that modeling is done correctly. When using **CYJOIN** Bulk Data entry, attention should be given to the definition of Sides 1 and 2, as this definition is different for rotational and dihedral symmetries. The Case Control command **DSYM** also needs to be applied carefully.

There are many types of practical structures where the static Cyclic Symmetry feature of **UAI/*NASTRAN*** can be efficiently applied. Analysis of static loading effects on the roofs of circular storage structures, silos, gas and steam turbines and similar structures may be performed efficiently using the Cyclic Symmetry feature.

22.6 EXAMPLE PROBLEMS

This Section presents two example problems which illustrate how you may use Cyclic Symmetry.

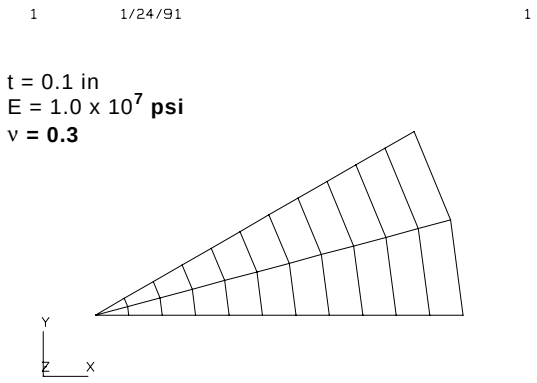
22.6.1 Static Analysis — Rotational Symmetry

The first example solves a very simple statics problem and demonstrates the use of rotational symmetry.

Example Problem 22-1
A circular plate of radius 11 in is fixed along its circumference and subjected to an unsymmetric load of 1 psi over a sector of various angles. You wish to determine the maximum normal displacement.

The finite element model is constructed for a 30° segment of the plate. The model, which is composed of QUAD4 and TRIA3 elements, and its physical characteristics are shown in Figure 22-6. This model is found in file **CYC1**. Six SUBCASEs are defined which apply the load over different surface areas of the plate. The first SUBCASE loads a 30° sector, the second a 60° sector, and so on. The application of the load is accomplished by loading one additional segment in each of the SUBCASEs. The **LOADCYN** Bulk Data for the first two loads are:

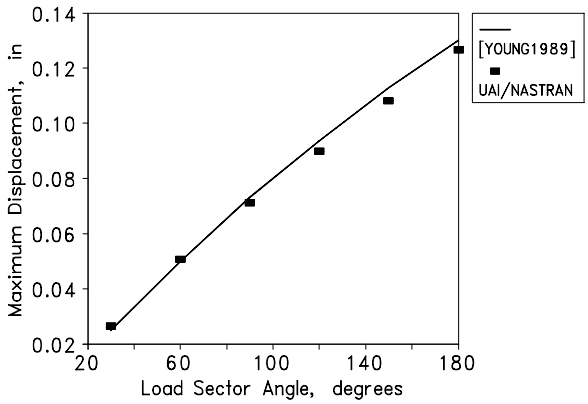
Figure 22-6. CIRCULAR PLATE MODEL



LOADCYN	**30**	**1.**	**1**		**1.**	**20**			
LOADCYN	**60**	**1.**	**1**		**1.**	**20**			
LOADCYN	**60**	**1.**	**2**		**1.**	**20**			
PLOAD4	**20**	**201**	**-1**				**THRU**	**310**	
PLOAD4	**20**	**101**	**-1**				**THRU**	**102**	

Also shown are the **PLOAD4** entries used to define the pressure load. Remember that the **LOADCYN** data must reference other load entries. A theoretical solution to this problem is found in [Young89]. Figure 22-7 compares these solutions against those of **UAI/*NASTRAN***. Note that the maximum deformation occurs away from the center of the circle. Therefore, in each SUBCASE, the maximum deformation may change from one node to another closer to the center. While the results are quite close, they would be nearly exact for a finer mesh refinement.

Figure 22-7. SOLUTION FOR VARIOUS LOADINGS



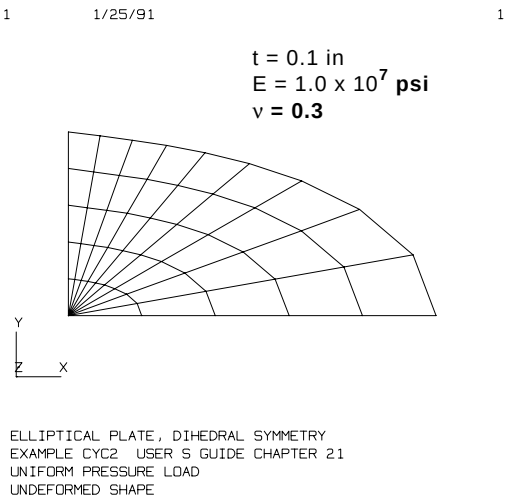
22.6.2 Static Analysis — Dihedral Symmetry

This example again solves a simple statics problem which demonstrates the use of dihedral symmetry.

Example Problem 22-2
A elliptical plate with semi-major axis of 10 in and a semi-minor axis of 5 in is fixed along its circumference and subjected to a uniform load of 1 psi over its surface. You wish to determine the maximum normal displacement.

The finite element model is constructed for a 90° Right segment of the plate. The model, again composed of QUAD4 and TRIA3 elements, and its physical characteristics are shown in Figure 22-8. This model is found in file **CYC2**.

Figure 22-8. ELLIPTICAL PLATE MODEL



As in Example 1, the load is applied with **LOADCYN** Bulk Data entries. Because of the dihedral symmetry, field 5 of the entry must specify whether the load is applied to the Right, **R**, or Left, **L**, segment.

LOADCYN	1	1.	1	R	1.	1			
LOADCYN	1	1.	1	L	1.	1			
LOADCYN	1	1.	2	R	1.	1			
LOADCYN	1	1.	2	L	1.	1			
PLOAD4	1	11	-1.				THRU	49	
PLOAD4	1	1	-1.				THRU	9	

Again, the **PLOAD4** entries used are used to define the actual pressure load and these are referenced by the **LOADCYN** data. A theoretical solution to this problem is found in [Young89]. The maximum vertical deformation, which occurs at the center of the plate is 0.0231 in. The **UAI/*NASTRAN*** result is 0.0202 in. Again, it is expected that a finer mesh would result in a solution which is close to the theoretical result.

Chapter 23

THREE DIMENSIONAL FLUID-STRUCTURE INTERACTION

UAI/*NASTRAN* provides you with the capability to perform real eigenvalue analysis and both the Direct and Modal solutions for coupled fluid-structure interaction (FSI) dynamic response analyses. The methodology is applicable to incompressible as well as compressible fluids, subject to constraints which presently limit its use to contained fluids and to low frequency acoustic problems and hydroelastic analyses. This Chapter describes the methodology for performing FSI analyses.

23.1 MATHEMATICAL BACKGROUND

This Section provides a synopsis of the theoretical development of the **UAI/*NASTRAN*** Fluid-Structure analysis capability. It is not a complete formulation, but rather presents the important equations and concepts that you may need to successfully build models and perform analyses.

23.1.1 Nomenclature

The following symbols are used in this Section:

SYMBOL	DESCRIPTION
$\boldsymbol{u}$	Vector of structural displacements.
$\boldsymbol{p}$	Vector of fluid pressures.
ρ	Fluid mass density.
$\boldsymbol{g}$	Gravity vector.
B	Bulk Modulus of the fluid.
$\boldsymbol{q}$	Vector of generalized forces acting on the fluid.
$\boldsymbol{M}_f$	Fluid mass matrix.
$\boldsymbol{K}_f$	Fluid stiffness matrix.
$\boldsymbol{A}$	Area matrix.
$\boldsymbol{M}_s$	Structural mass matrix.
$\boldsymbol{B}_s$	Structural damping matrix.
$\boldsymbol{K}_s$	Structural stiffness matrix.
$\boldsymbol{F}_f$	External forces acting on the fluid.
$\boldsymbol{F}_s$	External forces acting on the structure.

23.1.2 Governing Equations

The basic governing equations of fluid flow are developed from the Navier-Stokes equations. These are simplified using four basic assumptions:

- The fluid is homogeneous, inviscid, and irrotational.
- Only small motions of the fluid and small deformations of the structure occur.
- Pressure and density behavior is linear.
- Convective momentum terms are negligible.

These assumptions allow the fluid equations to be written as:

$$\ddot{\boldsymbol{u}} = -\frac{1}{\rho}\nabla \boldsymbol{p} + \boldsymbol{g} \tag{23-1}$$

$$\frac{1}{B}\boldsymbol{p} = -\nabla \boldsymbol{u} \tag{23-2}$$

where (23-1) is a form of the momentum conservation equations, and (23-2) is the continuity equation. These equations may be manipulated to derive an expression in which the fluid pressures are the independent variables:

$$\boldsymbol{q} = \frac{1}{B}\ddot{\boldsymbol{p}} - \nabla\frac{1}{\rho}\nabla\boldsymbol{p} \tag{23-3}$$

The finite element method allows the discretization of these equations resulting in:

$$\boldsymbol{M}_f\ddot{\boldsymbol{p}} + \boldsymbol{K}_f\boldsymbol{p} + \boldsymbol{A}^T\ddot{\boldsymbol{u}} = 0 \tag{23-4}$$

The matrix $\boldsymbol{A}$ is very important and is discussed in detail in the next Section.

23.1.3 Boundary Conditions

The matrix $\boldsymbol{A}^T$ in (23-4) is called the ***Area*** matrix because it represents the surface area that encloses the fluid region. The equivalent force acting on the fluid region:

$$\boldsymbol{P} = -\boldsymbol{A}^T\ddot{\boldsymbol{u}} \tag{23-5}$$

represents the total flux of volumetric acceleration through the boundary of the fluid region. This boundary may be divided into three portions:

- The rigid wall boundary
- The fluid and flexible surface interface
- The free surface of the fluid

For either the rigid wall or any wall moving at constant velocity, the flux of volumetric acceleration is zero, i.e. $\boldsymbol{P} = \boldsymbol{0}$. Therefore, no special treatment is required. At the fluid-struc-

ture interface and at the free surface, the flux of volumetric acceleration is non-zero. Each of these aspects is treated separately, therefore, the area matrix is partitioned into two parts:

$$\boldsymbol{A}^T \ddot{\boldsymbol{u}} = \boldsymbol{A}_s^T \ddot{\boldsymbol{u}}_s + \boldsymbol{A}_f^T \ddot{\boldsymbol{u}}_f \tag{23-6}$$

where the subscripts *s* and *f* indicate the fluid-structure interface and free surface of the fluid, respectively.

23.1.3.1 Fluid-Structure Interface

The structural dynamic response may be written as:

$$\boldsymbol{M}_s \ddot{\boldsymbol{u}} + \boldsymbol{B}_s \dot{\boldsymbol{u}} + \boldsymbol{K}_s \boldsymbol{u} = \boldsymbol{F}_f + \boldsymbol{F}_s \tag{23-7}$$

where the external loads have been added. Note that (23-1) and (23-2) represent an Eulerian formulation while (23-7) is formulated from the Lagrangian point-of-view. As a result, the pressure at the fluid-structure interface defined in (23-1) is not the pressure acting on the structure in (23-7). The actual relationship is:

$$\boldsymbol{p}_s = \boldsymbol{p} + \boldsymbol{u} \nabla \boldsymbol{p} \tag{23-8}$$

which, when combined with (23-1) results in:

$$\boldsymbol{p}_s = \boldsymbol{p} + \rho \boldsymbol{u} \boldsymbol{g} \tag{23-9}$$

For a small variation of the displacement on the boundary, this may be written in matrix form as:

$$\boldsymbol{F}_f = \boldsymbol{B}_s \boldsymbol{p} - \Delta \boldsymbol{K}_{sg} \boldsymbol{u} \tag{23-10}$$

where $\boldsymbol{K}_{sg}$ represents the gravity effects at the fluid-structure interface as described in the next section. This allows (23-7) to be rewritten as:

$$\boldsymbol{M}_s \ddot{\boldsymbol{u}} + \boldsymbol{B}_s \dot{\boldsymbol{u}} + (\boldsymbol{K}_s + \Delta \boldsymbol{K}_{sg}) \boldsymbol{u} - \boldsymbol{A}_s \boldsymbol{p} = \boldsymbol{F}_s \tag{23-11}$$

23.1.3.2 Free Surface of the Fluid

If the effect of gravity at the free surface is neglected, then the boundary condition is simply a single point constraint which sets $\boldsymbol{p} = 0$ at the free surface. If, on the other hand, gravity is considered, then a load will exist at the free surface. This may be expressed as:

$$\Delta \boldsymbol{K}_{fg} \boldsymbol{u}_f - \boldsymbol{A}_f \boldsymbol{p} = 0 \tag{23-12}$$

Note that the matrix $\Delta \boldsymbol{K}_{fg}$ is non-singular and thus invertible. Differentiation of (23-12) results in:

$$\boldsymbol{A}_f^T \ddot{\boldsymbol{u}} - \Delta \boldsymbol{M}_{fg} \ddot{\boldsymbol{p}} = 0 \tag{23-13}$$

where $\Delta \boldsymbol{M}_{fg} = \boldsymbol{A}_f^T \Delta \boldsymbol{K}_{fg}^{-1} \boldsymbol{A}^f$.

23.1.4 System Equations

The equations of motion for the coupled fluid-structure problem are given in this section.

23.1.4.1 Compressible Fluid Model

For a compressible fluid, the equations of motion are:

$$\begin{bmatrix} \boldsymbol{M}_s & \boldsymbol{0} \\ \boldsymbol{A}_s^T & \boldsymbol{M}_f + \Delta \boldsymbol{M}_{fg} \end{bmatrix} \begin{Bmatrix} \ddot{\boldsymbol{u}}_s \\ \ddot{\boldsymbol{p}} \end{Bmatrix} + \begin{bmatrix} \boldsymbol{B}_s & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \dot{\boldsymbol{u}}_s \\ \dot{\boldsymbol{p}} \end{Bmatrix} + \begin{bmatrix} \boldsymbol{K}_s + \Delta \boldsymbol{K}_{sg} & -\boldsymbol{A}_s \\ \boldsymbol{0} & \boldsymbol{K}_f \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}_s \\ \boldsymbol{p} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{F} \\ \ddot{\boldsymbol{Q}} \end{Bmatrix} \tag{23-14}$$

where $\boldsymbol{M}_{fg}$ and $\boldsymbol{K}_{sg}$ represent the gravity effects at the free surface and the fluid-structure interface, respectively, $\boldsymbol{F}$ are the external forces on the structure, and $\ddot{\boldsymbol{Q}}$ are the volumetric accelerations produced by the flow sources.

23.1.4.2 Incompressible Fluid Model

For incompressible fluid, $\frac{1}{\boldsymbol{B}} = 0$ and thus $\boldsymbol{M}_f = 0$ in (23-14). However, to implement the capability, the equations of motion are cast in a slightly different form compared to the compressible case. Combining (23-4), (23-6), (23-11), and (23-12), results in:

$$\begin{bmatrix} \boldsymbol{M}_s & \boldsymbol{0} \\ \boldsymbol{A}_s^T & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \ddot{\boldsymbol{u}} \\ \ddot{\boldsymbol{p}} \end{Bmatrix} + \begin{bmatrix} \boldsymbol{B}_s & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \dot{\boldsymbol{u}} \\ \dot{\boldsymbol{p}} \end{Bmatrix} + \begin{bmatrix} \boldsymbol{K}_s + \Delta \boldsymbol{K}_{sg} + \Delta \boldsymbol{K}_{fg} & -\boldsymbol{A}_s \\ \boldsymbol{0} & \boldsymbol{K}_f \end{bmatrix} \begin{Bmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{F} \\ \boldsymbol{0} \end{Bmatrix} \tag{23-15}$$

where $\boldsymbol{u}$ are the displacements for both the free surface degrees of freedom and the structural degrees of freedom as shown in (23-6), and $\Delta \boldsymbol{K}_{fg}$ represents the gravity effect at the free surface. You will note that $\ddot{\boldsymbol{Q}} = 0$, so that you may not have flow sources in the incompressible case. The major difference between the equations for the compressible and incompressible cases is that in the former, the free surface degrees of freedom are included in $\boldsymbol{p}$ as pressures, while in the latter, they are included in $\boldsymbol{u}$ as displacements.

23.1.5 Eigenvalue Extraction

Although the systems of equations in (23-14) and (23-15) are unsymmetric, they have real eigenvalues. Even though the eigenvalues are real, the eigensolution must be performed using complex numbers with the matrices in their unsymmetric form. Complex eigenextraction is very expensive relative to real eigensolvers. Therefore, **UAI/*NASTRAN*** provides two approaches which provide efficient solutions. The first is a special purpose eigensolver which uses subspace iteration. The second approach is to formulate the problem using symmetric matrices which can then be addressed using a real eigensolver such as Lanczos. These approaches are described in the following sections.

23.1.5.1 Symmetric Matrix Formulation

In order to facilitate solution using real eigenextraction methods, it is necessary to transform these equations to a symmetric form. The same method is used in both the subspace iteration and direct symmetric eigenvalue techniques. Let $\boldsymbol{v}$ represent the volumetric flux at the fluid GRID points:

$$\boldsymbol{v} = \boldsymbol{M}_f \boldsymbol{p} \tag{23-16}$$

Then the lower partition of (23-14) may be written:

$$\boldsymbol{P} = \boldsymbol{K}^{-1}\left(-\ddot{\boldsymbol{v}} - \boldsymbol{A}^T \ddot{\boldsymbol{u}} + \ddot{\boldsymbol{Q}} \right) \tag{23-17}$$

Combining (23-14), (23-16), and (23-17) results in:

$$\left[\overline{\boldsymbol{M}}_s + \overline{\boldsymbol{A}}\, \boldsymbol{K}_f^{-1} \overline{\boldsymbol{A}}^T \right] \ddot{\overline{\boldsymbol{u}}} + \boldsymbol{B} \dot{\overline{\boldsymbol{u}}} + \boldsymbol{K} \overline{\boldsymbol{u}} = \left\{ \boldsymbol{F} + \overline{\boldsymbol{A}}\, \boldsymbol{K}_f^{-1} \overline{\boldsymbol{A}}^T \ddot{\boldsymbol{Q}} \right\} \tag{23-18}$$

where:

$$\overline{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{I} \end{bmatrix} ; \; \overline{\boldsymbol{u}} = \begin{Bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{Bmatrix}$$

$$\overline{\boldsymbol{M}}_s = \begin{bmatrix} \boldsymbol{M}_s & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} ; \; \boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_s & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} ; \; \boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}_s & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}_f^{-1} \end{bmatrix}$$

This represents the compressible case, which reduces to the incompressible case when:

$$\overline{\boldsymbol{A}} = \boldsymbol{A} \; ; \; \overline{\boldsymbol{u}} = \boldsymbol{u} \; ; \; \boldsymbol{M}_f = \boldsymbol{0} \; ; \text{ and } \ddot{\boldsymbol{Q}} = \boldsymbol{0} \tag{23-19}$$

In the following sections, there will be no further separation of the two cases. Also, to simplify the equations, $\boldsymbol{A}$ will be used for either $\boldsymbol{A}$ or $\overline{\boldsymbol{A}}$, and, $\boldsymbol{u}$ will be used for either $\boldsymbol{u}$ or $\overline{\boldsymbol{u}}$, as needed.

Because the mass matrix in (23-18), $\left[\boldsymbol{M}_s + \boldsymbol{A}\, \boldsymbol{K}_f^{-1} \boldsymbol{A}^T \right]$, may often be very dense due to the area coupling matrix, $\boldsymbol{A}$, special methods have been implemented to minimize the cost of evaluating (23-18). The **FSIEPS** field on the **FSIDATA** Bulk Data entry controls the grounding of matrix $\boldsymbol{K}_f$ when it contains a constant pressure mode. This temporary grounding avoids the need to perform an exact inversion of $\boldsymbol{K}_f$, thus reducing both disk space requirements and CPU time.

23.1.5.2 Subspace Iteration

The real, symmetric eigenextraction methods available in **UAI/*NASTRAN*** require the evaluation of this matrix. Its density requires that large amounts of memory be used to extract the eigenvalues. As a result, only small FSI models may be considered. To overcome this computational problem, a special-purpose eigensolver based on a block subspace iteration algorithm has been implemented in **UAI/*NASTRAN*** to perform this operation using:

$$\boldsymbol{K\Phi} = \left[\boldsymbol{M}_s + \boldsymbol{A}\,\boldsymbol{K}_f^{-1}\,\boldsymbol{A}^T \right] \boldsymbol{\Phi\lambda} \tag{23-20}$$

where $\boldsymbol{\lambda}$ and $\boldsymbol{\Phi}$ are the eigenvalues and eigenvectors of the system, respectively. Initially, $\boldsymbol{\Phi}$ is a set of random vectors, and $\boldsymbol{\lambda}$ is an identity matrix. In this method, the matrix $\boldsymbol{A}\,\boldsymbol{K}_f^{-1}\,\boldsymbol{A}^T$ is not explicitly evaluated, instead the matrix $\boldsymbol{A}^T\boldsymbol{\Phi\lambda}$ is first computed. This results in a small matrix on the right-hand-side of (23-21) even for a very large FSI model. Note that because the mass matrix is not explicitly evaluated, it is not possible to perform shifting during the eigenextraction. This results in greater computational cost when extracting higher eigenvalues.

In general, for small FSI models any of the eigenextraction methods may be used. For moderate size problems, Subspace Iteration is often more efficient, and for very large problems, it is usually required to obtain a solution depending on the disk and memory resources available on your host computer.

23.1.6 Fluid Elements

FSI models are simple to construct. The structure may include any of the standard **UAI/*NASTRAN*** one-, two-, and three-dimensional elements described in Chapter 4. The fluid may be modeled using only the solid elements HEXA, PENTA and TETRA. These elements reference a corresponding **PSOLID** property entry in the usual manner. The only difference is that the material property identification number may only reference a **MATF** Bulk Data entry. More details of modeling are presented in section 23.2.

23.1.7 Acoustic Elements

One of the major applications of FSI is the solution of sound transmission problems for noise level prediction in passenger aircraft and automobiles. To support this important analytical regime, **UAI/*NASTRAN*** includes two fluid interface elements: the acoustic barrier, and the acoustic absorber. Each of these is discussed in this section.

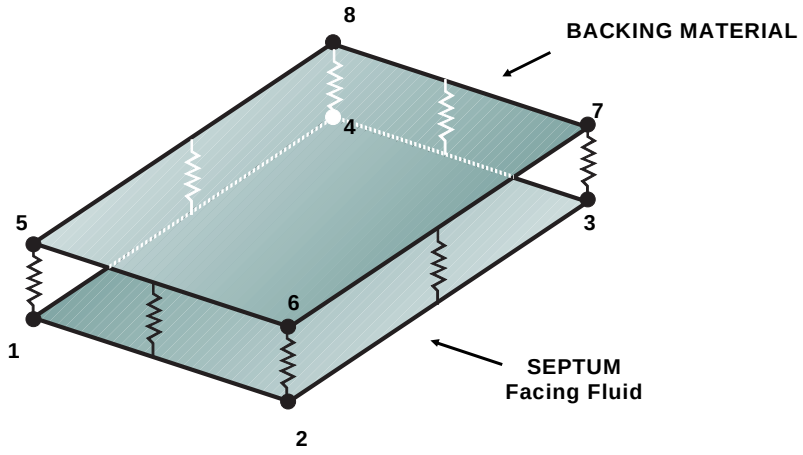
23.1.7.1 The Acoustic Barrier Element

The Barrier element is used to represent material that reduces the transmission of sound between acoustic fields, or volumes. This element is defined as a two-mass system connected by a spring. A schematic representation is shown in Figure 23-1. The properties of the element are:

- The mass per unit area of the backing material, m_b
- The mass per unit area of the septum, m_s
- The effective stiffness per unit area, k_e
- The resonant frequency, f_r

The values m_b and m_s must be supplied for each element. It is necessary to specify only one of either k_e or f_r. The resonant frequency, which can be obtained from standard ASTM tests, is defined to be the frequency at which the transmission loss is at a minimum. The effective stiffness and resonant frequency are related by:

Figure 23-1. THE ACOUSTIC BARRIER ELEMENT



$$k_e = 4\pi^2 f_r^2 \frac{m_s m_b}{m_s + m_b} \tag{23-21}$$

23

Once the mass and stiffness per unit area are known, element stiffness and mass matrices are determined using a hexahedral isoparametric finite element formulation. Finally, the element mass and stiffness matrices are assembled into the structural mass and stiffness matrices of the system equations (23-14) or(23-15).

23.1.7.2 The Acoustic Absorber Element

Absorber elements can be placed in the acoustic field to reduce the strength of reflected sound. Acoustic absorbing materials are characterized by their specific acoustic impedance. The impedance is measured by a standard ASTM standing wave tube test, defined by:

$$Z_a(f) = \frac{P}{\dot{U}} = R(f) + iX(f) \tag{23-22}$$

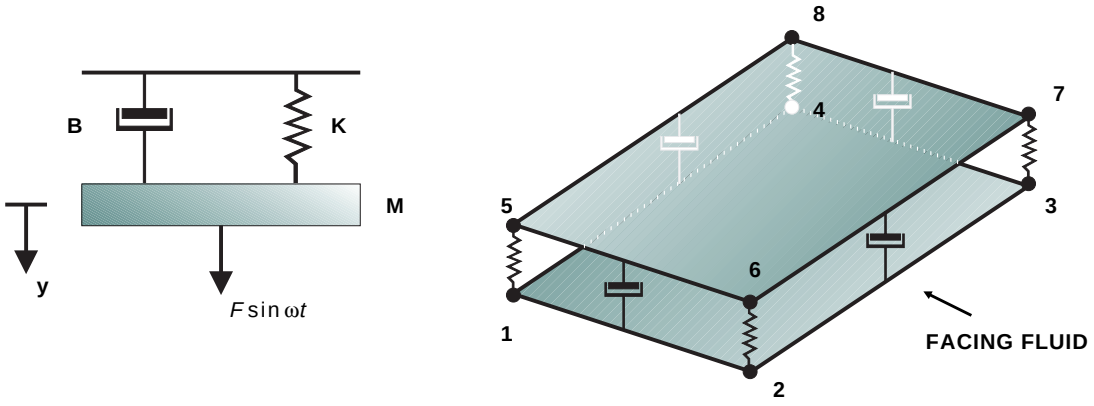
where P is the pressure applied to the standing wave tube test, $\dot{U}$ is the particle velocity, R is the resistance, and X is the reactance. In most cases, the resistance and reactance are functions of frequency, f.

The impedance of the absorbing material is simulated by a mechanical system as shown in Figure 23-2a. The mechanical impedance of this system is given by:

$$Z_m = \frac{F}{\dot{y}} = B + i\left(\omega M - \frac{1}{\omega}K\right) \tag{23-23}$$

where M is the mass, K is a grounded spring, and B is a damper. In order to use Z_m to simulate Z_a, the damping is defined as the average value of acoustic resistance:

Figure 23-2. THE ACOUSTIC ABSORBER ELEMENT



a. Equivalent Mechanical System **b. HACABS Element**

$$B = \frac{1}{n} \sum_{j=1}^{n} R(f_j) \tag{23-24}$$

You may input the resistance, *R*, as a table defined using the **PACABS** Bulk Data entry with a cutoff frequency. When this is done, *n* equal distance frequency points from **0.0** to **FCUTOFF** are taken to evaluate the damping matrix above. **UAI/*NASTRAN*** uses **10** for the value of *n*. A least squares approximation may then be used to evaluate ***M*** and ***K***.

Let:

$$\xi = \sum_{j=1}^{n} \left[\omega_j M - \frac{1}{\omega} K - X(f_j) \right]^2 W_j \tag{23-25}$$

where W_j is a weighting function. Minimizing ξ results in the following equations:

$$\begin{bmatrix} \sum \omega_j^2 W_j & -\sum W_j \\ -\sum W_j & \sum \frac{1}{\omega_j^2} W_j \end{bmatrix} \begin{Bmatrix} M \\ K \end{Bmatrix} = \begin{Bmatrix} \sum X(f_j)\, \omega_j\, W_j \\ \sum \frac{X(f_j)}{\omega_j} W_j \end{Bmatrix} \tag{23-26}$$

which can be solved for ***M*** and ***K***. Again, element stiffness, damping and mass matrices are determined using a hexahedral isoparametric finite element formulation. Finally, the element mass and stiffness matrices are assembled into the structural mass and stiffness matrices of the system equations (23-14) or(23-15).

23.1.8 Acoustic Source

An acoustic source is assumed to be a sphere which oscillates in an infinite acoustic field. The strength of the source is characterized by a frequency-dependent flux of volumetric velocity defined by:

$$\dot{Q}\, e^{i\omega t} = \int_S \dot{\boldsymbol{u}}^T \boldsymbol{r}\, dS \tag{23-27}$$

where $\dot{Q}$ is the strength of the source, and $\boldsymbol{r}$ is the unit radial vector. The power, H, radiated from the source is related to its strength by:

$$H = \frac{\pi}{2} \rho\, c \left(\frac{\dot{\theta}}{\lambda} \right)^2 \tag{23-28}$$

where ρ is the density of the fluid, c is the speed of sound, and λ is the wave length. Equation (23-28) may be inverted to yield:

$$\dot{Q} = \frac{1}{\omega} \left(\frac{8\, \pi\, c\, H}{\rho} \right)^{1/2} \tag{23-29}$$

In **UAI/*NASTRAN***, power as a function of frequency is specified using **ACSRCE** Bulk Data entries. The source strength is then computed using (23-29) and applied to the fluid in the model as an external load.

23.1.9 Acoustic Pressure

Several forms of acoustic pressures are computed during FSI analyses. This section defines the meaning of these.

23.1.9.1 Frequency Band and Center Frequency

A time-history of sound pressure, $P(t)$, can be partitioned into a frequency band using:

$$P(t) = \sum_i P_i(t) = \sum_i \left[\int_{f_{i-1}}^{f_i} P_o e^{i(2\pi f t + \varphi)}\, df \right] \tag{23-30}$$

The term in braces can be considered as a frequency-band component pressure over the range:

$$f_{i-1} < f \le f_i \tag{23-31}$$

with center frequency, f_c, defined by:

$$f_{c,i} = \sqrt{f_i f_{i-1}} \tag{23-32}$$

An octave band, a 1/3 octave band, and a 1/n-octave band are, respectively:

$$f_i = 2 f_{i-1} \; ; \; f_i = 2^{1/3} f_{i-1} \; ; \text{ and } f_i = 2^{1/n} f_{i-1} \tag{23-33}$$

Frequency Response analyses with **UAI/*NASTRAN*** may be performed either at the boundaries of the frequency band, f_i, or at the center frequencies, $f_{c,i}$. You select this option with the **CFREQ** field of the **FSIDATA** Bulk Data entry.

23.1.9.2 Mean Square Pressure and RMS Value

The mean square and RMS values of pressure are defined by:

$$P_m^2 = \lim_{T \to \infty} \frac{1}{T} \int_{t_c - T/2}^{t_c + T/2} P^2(t)\, dt \; ; \; P_{rms} = \sqrt{P_m^2} \tag{23-34}$$

where $P(t)$ may be the total pressure, a frequency-band component, or a single frequency pressure. In the case of a single frequency pressure, (23-34) becomes:

$$P_{rms} = \frac{1}{\sqrt{2}} P_o \tag{23-35}$$

where P_o is the peak value of the single frequency pressure. When performing Transient Response analyses, (23-35) is used to compute the RMS values at the current time step t with $T = t$ and $t_c = t/2$. Note that only the output time steps are used in performing the integration of (23-34). When performing Frequency Response analyses, (23-35) is used for the computation.

23.1.9.3 Sound Pressure Level

Generally, sound pressure is expressed on a logarithmic scale as the Sound Pressure Level (SPL) in decibels (dB). The SPL is defined by:

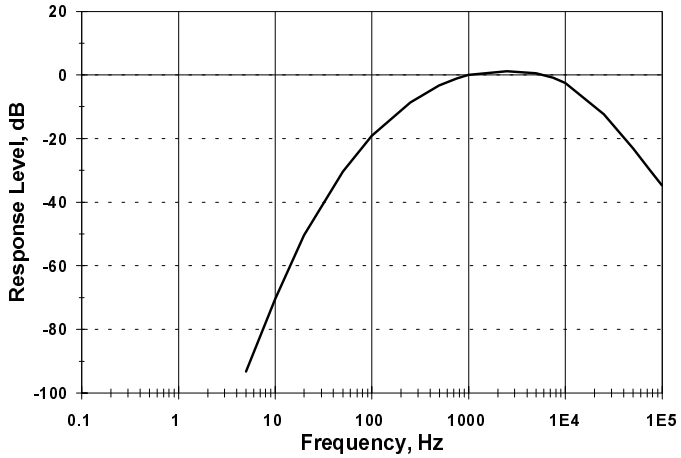
$$SPL = 10 \log \frac{p_m^2}{P_{ref}^2} = 20 \log \frac{P_{rms}}{P_{ref}} \tag{23-36}$$

where P_{ref} is the reference pressure. For air, $P_{ref} = 20 \times 10^{-6}$ pascal. (23-36) shows the rms-based SPL, you may also select the peak SPL value by using the **DBOUT** field of the **FSIDATA** Bulk Data entry.

23.1.9.4 Frequency Weighting Function

Psychoacoustic studies have determined that the perception of loudness in the human ear is frequency dependent. The SPL defined in (23-36) treats the pressure at all frequencies with the same weight. In order to compensate for the sensory perception of the noise, a weighting function is used:

Figure 23-3. RESPONSE LEVEL OF A-WEIGHTING NETWORK



$$\left(P_m^2 \right)_w = \sum_{i=1}^{n} W(f_i) P_m^2(f_i) \qquad (23\text{-}37)$$

23

which, in terms of dB, may be rewritten:

$$\mathrm{SPL}_w(f_i) = \mathrm{SPL}_i + \Delta \mathrm{SPL}_w(f_i) \qquad (23\text{-}38)$$

where W and $\Delta \mathrm{SPL}_w$ are the weighting functions. The most commonly used weighting function is the ANSI standard A-Weighting Network. An A-Weighted SPL is denoted by its units, dB(A). Figure 23-3 shows the A-Weighted SPL in dB(A) for an unweighted 0 dB SPL from 5 Hz. to 100KHz.

23.1.9.5 Combination of Sound Pressure Level

The SPL given by (23-38) is measured in dB(A) at each frequency. Because the human ear hears all of the sound frequencies simultaneously, it is necessary to combine the dB(A) values for all frequencies. **UAI*/NASTRAN*** performs this combination under the following assumptions.

- The computed SPL is a spectrum density function. This is true when the input loads are spectrum density functions. For example, if the input loads are acoustic sources, then the computed SPL is a spectrum density function.
- The frequency range of the analysis is divided into frequency bands.
- Within each frequency band, the SPL is assumed to be a constant value taken at the center frequency specified in (23-32).

Under these assumptions, the mean square pressure for the frequency band i is:

$$P_m^2 = P_{c,i}^2 \Delta f_i \tag{23-39}$$

where $\Delta f_i = f_i - f_{i-1}$ is the bandwidth, and $P_{c,i}$ is the mean square pressure at the center frequency. This may be written in terms of dB(A) as:

$$\mathrm{SPL_i} = \mathrm{SPL}_{c,i} + 10 \log \Delta f_i \tag{23-40}$$

Now, the combined dB(A) for the first n bands may be expressed as:

$$\mathrm{SPL}_n = 10 \log \left[\sum_{i=1}^{n} 10^{\frac{\mathrm{SPL}_i}{10}} \right] \tag{23-41}$$

More accurate combined dB(A) values will be obtained if you set the **CFREQ** field to **YES** on the **FSIDATA** Bulk Data entry, and the frequencies used in the analysis are center frequencies of the $\frac{1}{n}$-octave band.

23.2 THE FINITE ELEMENT IDEALIZATION

The finite element representation of a three dimensional fluid-structure interaction model is similar to that employed in conventional structural systems, but differs by the definition of the fluid-structure interface and the inclusion of the fluid model, as discussed below.

The finite element model of the 3D structure is constructed in the same manner as for the usual structural system; i.e., by a set of elements connected to GRID points, defined in various coordinate systems. All of the element types employed in conventional static and dynamic analyses are available for use.

The modeling of the fluid is performed similarly to that of the structure, but the element types are restricted to HEXA, PENTA and TETRA solids. The fluid elements must be connected to a set of GRID points which have no connection with the structural model. Unlike structural GRIDs which have displacements as degrees of freedom, the GRIDs associated with fluid elements have only one degree of freedom as the unknown: the pressure. The solid fluid elements reference **PSOLID** data in the usual manner. These data, in turn, reference the **MATF** fluid material Bulk Data entry which defines the fluid density and, optionally, the Bulk Modulus. For incompressible fluids, the Bulk Modulus should not be defined. When the Modulus is specified, the fluid is treated as compressible and may be either a liquid or a gas — there is no mathematical distinction.

Compressible and incompressible fluid behavior is controlled only by the presence or absence of the Bulk Modulus on the **MATF** Bulk Data entry.

In addition, two special structural elements, the acoustic barrier (HACBR) and the acoustic absorber (HACAB) are available for modeling acoustic control devices.

The fluid-structure interface, discussed in detail below, may be modeled so that the structural GRID points on the interface are coincident with those in the fluid model. When this is the case, the model has what is called a *matching mesh*. Otherwise, it is has a *non-matching mesh*. When using a non-matching mesh, it is recommended that the structural mesh be refined at least as much as the fluid. While all fluid elements may be used at the interface, only 2-D (QUAD4, QUADR, TRIA3 and TRIAR), 3-D (HEXA, PENTA, and TETRA), and acoustic elements (HACBR and HACAB) are allowed as structural interface elements.

23.2.1 Boundary Conditions

You apply boundary conditions on the structural model in the normal manner. For the fluid model, there are three types of boundary conditions: the free surface; the fluid-structure interface; and the rigid wall boundary. Each of these is described below.

23.2.1.1 Fluid-Structure Interface

The fluid-structure interface is described by **FLSTR** Bulk Data entries, which identify the structural elements which touch each fluid element. If no **FLSTR** entries are supplied, then the **FLSTR** data are automatically generated. (Note: You may obtain a listing of the internally generated fluid-structure connections by including **DIAG 70** in your Executive Control packet.) It is possible to control the generation of these with the **TOLER** field of the **FSIDATA** Bulk Data entry.

23.2.1.2 Rigid Wall

Unlike the structural model, boundaries of the fluid that represent a reflecting surface, such as the natural boundary associated with a plane of symmetry, are not constrained by **SPC** Bulk Data entries. Instead, they are left free. In the case of fluid GRID points, an SPC produces a zero pressure condition. This is detailed in the following Section.

23.2.1.3 Free Surface

An SPC is used at a fluid GRID point to impose a zero pressure constraint. This is correct for representing the far-field boundary in external acoustics problems. You may also apply SPC constraints to GRID points on the free surface of an incompressible fluid if you are not including gravity effects to produce sloshing modes. If such modes are required, then GRID points on the free surface of the fluid must be identified by using **FLFREE** Bulk Data entries. In addition, you must specify the gravity field that is acting on the fluid boundary by using **FSIDATA** and **GRAV** Bulk Data entries.

23.2.2 Available Analyses

You may perform fluid-structure interaction analyses for three disciplines: Normal Modes analysis (Rigid Format 3), Frequency Response analysis (Both Direct and Modal, Rigid Formats 8 and 11) and Transient Response analyses (Both Direct and Modal, Rigid Formats 9 and 12). Additionally, you may use the Multidisciplinary APEX program (**APP APEX, SOL MULTI**). However, you may only use the Normal Modes and Direct Frequency Response disiciplines.

23.2.3 External Loads

In the normal fashion, the **LOAD** or **DLOAD** Case Control commands are used to select environmental loads to the structural system, and these in turn refer to any of the load entries such as **FORCE**, **LOAD**, and **PRESSURE**. Gravity fields acting on the system are defined with **FSIDATA** and **GRAV** Bulk Data entries, which apply their field to the fluid elements. For compressible fluid models, you may apply the following external loads.

- Enforced constant pressure at a fluid GRID point by using the **SPC** Case Control command and **SPCD** Bulk Data entries.
- Time-dependent or frequency dependent enforced pressure, volumetric acceleration, volumetric velocity, or volumetric flux by selecting appropriate values for the **DYNEX** field of referenced **RLOADi** or **TLOADi** Bulk Data entries.
- Acoustic sources which are characterized by the flux of volumetric velocities which, in turn, are computed from the power spectrum density functions that you provide using **ACSRCE** Bulk Data entries.

Note that for incompressible fluid models, only the enforced constant pressure is available.

23.2.4 Reduction Procedures

The Direct method dynamic response analyses allow reduction of the active degrees of freedom using **ASET** or **OMIT** Bulk Data entries or dynamic reduction for more economical processing of large order systems. With the sparse matrix algorithms available, it may now be more economical to forego reduction and, for Frequency and Transient Response analyses,

solve the unsymmetric system described in Section 23.1.4 directly. You may force this option by using the Bulk Data entry:

PARAM	**NONSYM**	**YES**							

If you do not use this entry, **UAI/*NASTRAN*** will create the symmetric system as described in Section 23.1.5.

For Modal solution procedures, **UAI/*NASTRAN*** always creates the symmetric system in order to extract the real eigenvectors used for the modal reduction. The normal static condensation and dynamic reductions may also be used. As discussed in Section 23.1.6, the Subspace Iteration method of eigenextraction may be used to improve performance. This method, denoted by **SUBS**, is requested with the **EIGR** Bulk Data entry in the usual manner. When using Subspace Iteration, or Lanczos, static condensation and dynamic reductions are not recommended.

23.2.5 Free-Free FSI Model

A free-free FSI model is one that has not been constrained by single point constraints, and, as a result, it may move in space as a free body. In **UAI/*NASTRAN***, such a model has six rigid body modes and one constant pressure mode. When a normal modes analysis is performed, the frequencies of these seven modes are nearly zero.

If there is no free fluid surface in the free-free model, then these rigid body modes are automatically computed during a normal modes analysis. However, if there is a free surface in the model, the results depend on the manner in which you prescribe the boundary conditions of the free surface. As described in Section 23.2.1, the boundary conditions at a surface which is not bounded by structural elements may be prescribed in one of three ways: as a rigid wall; by imposing a zero pressure constraint; or by defining it as a free surface by using **FLFREE** Bulk data entries. For free-free FSI models, you must prescribe the boundary condition for the free fluid surface using **FLFREE** Bulk Data entries. In addition, you must specify a structural GRID point (**GID**) and degree of freedom (**DOF**) on the **FSIDATA** Bulk Data entry:

FSIDATA								**GID**	**-cont-**
-cont-	**DOF**								

If **GID** and **DOF** are not specified, the motion of the free surface degrees of freedom are measured relative to ground, thus the rigid body mode in the direction of gravity is constrained. When **GID** and **DOF** are specified, the motion of the free surface is measured relative to **DOF**. The GRID point **GID** must be located on the line of the center of gravity of the free surface, and the direction of **DOF** must be in the same direction as the gravity vector. ***This feature may only be used for incompressible fluid models.***

23.2.6 Modal Synthesis with APEX

FSI analyses may be performed using the multidisciplinary APEX program. When using APEX, a modal synthesis approach is available for performing Normal Modes, Modal Frequency Response, and Modal Transient Response analyses. Modal synthesis can often result in a savings in CPU time because, by using generalized coordinates based on normal modes, it is a more computationally efficient approach for large FEM models. This feature is activated by using the Case Control command **CASE**. The general form of this command is:

```
CASE  case_id  [ STRUCTURAL ][   MODES    ]
               [   FLUID    ][ FREQUENCY ]
               [  COUPLED   ][ TRANSIENT ]
```

The **CASE** command selects the portion of the FSI model to process, and the analytical discipline to be used. In addition to the **CASE** command, the modal synthesis approach relies on the **USING** concept of the Modal Approach found in the APEX program. The general form of the **USING MODES** command is:

```
USING [ STRUCTURAL ] MODES [ CASE ] case_id
      [   FLUID    ]
      [  COUPLED   ]
```

The **USING** command allows one discipline to use the results computed in a previous **CASE**.

23.2.6.1 Normal Modes Analysis

When performing Normal Modes analysis for FSI problems, the available options for the **CASE** command are **COUPLED MODES**, **STRUCTURAL MODES**, or **FLUID MODES**. The first option requests that the solution be performed for the coupled fluid-structure model. The second option indicates the analysis will be performed for only the dry structure. Finally, the third option indicates that an analysis of rigid-wall fluid modes will be performed. For incompressible fluid, only the first two types of analysis are available. For example, to compute the modes for the structure and fluid separately, the Case Control packet is:

```
CASE 1 STRUCTURAL MODES
...
CASE 2 FLUID MODES
...
```

These modes may then be used as generalized coordinates in a coupled solution. This is done with:

```
CASE 1 STRUCTURAL MODES
...
CASE 2 FLUID MODES
...
CASE 3 COUPLED MODES
   USING STRUCTURAL MODES CASE 1
   USING FLUID MODES CASE 2
   ...
```

Again, after performing the the first two **CASE**s to extract the structural and fluid modes separately, a third **CASE** is run to perform a **COUPLED MODES** analysis using the results of the first two analyses. This is accomplished with the **USING STRUCTURAL MODES** and **USING FLUID MODES** commands. This analysis will be performed using modal synthesis. If the **COUPLED MODES** analysis only refers to the results of a previous **STRUCTURAL MODES** analysis, then the generalized coordinates representing the structural modes will be used in combination with the physical degrees of freedom of the fluid. Similarly, if the **COUPLED MODES** analysis only refers to the results of a previous **FLUID MODES** analysis, then the generalized coordinates representing the fluid modes will be used in combination with the physical degrees of freedom of the structure.

23.2.6.2 Frequency Response and Transient Response Analyses

If a dynamic response analysis, either Frequency Response or Transient Response, is to be performed on the coupled system, then the system model may be defined using all of the same options as the coupled modes case. When the **USING** command references the results of a **COUPLED MODES** analysis, the resulting solution is performed exactly the same as a standard modal approach solution.

The modal approach may also be performed for either the dry structure, or for the rigid wall fluid. For the structure, you may use the Case Control commands **STRUCTURAL FREQUENCY** or **STRUCTURAL TRANSIENT**, and you may only reference the modes of a previous **STRUCTURAL MODES** case. For the fluid, you may use the Case Control commands **FLUID FREQUENCY** or **FLUID TRANSIENT**, and you may only reference the modes of a previous **STRUCTURAL MODES** case.

23.2.6.3 Design Optimization

When using APEX, you may also perform optimization during FSI analyses. When performing Frequency Response analyses, three specific design constraints are available. These are pressure, dB, and db(A) as defined in Section 23.1. Details for performing MDO are found in Chapter 25.

23.2.7 Substructuring Capabilities

Fluid-structure interaction analyses may be performed using the **UAI/*NASTRAN*** substructuring feature. This may be useful, for example, to create a system model of a missile launch vehicle containing several fluid filled tanks, or to create a system model of an automotive type structure containing air or liquid filled cavities.

There are three modeling, or operational, restrictions which must be observed very carefully when using fluid-structure models with substructuring. These restrictions are discussed next.

The substructuring database, the SOF, does not have provision for storing the area coupling matrix which defines the boundary between fluid elements and structural elements. Therefore, the first restriction is that it is not possible to model the fluid as one substructure and the fluid's structural container as another substructure. Instead, substructures which are associated with a fluid must completely contain the fluid, and such substructures must have their boundary GRID points as structural GRID points only, i.e., no pressure points are allowed on a substructure boundary.

The second restriction is related to the first. If the following modeling conditions exist:

- A ***wetted*** GRID point, i.e, a structural GRID point which is in contact with fluid, is also a boundary GRID point
- There is a fluid GRID point coincident with that structural GRID point
- The fluid is compressible such that the fluid GRID point is not eliminated by static condensation
- The structural GRID point will be part of a substructure **COMBINE** operation

then the **RELEASE** feature of the automatic substructuring **COMBINE** operation must be used to avoid the inadvertent connection of the structural GRID point motions in direction T1 with the fluid pressure of the pressure GRID point.

The third restriction also follows from the first. This restriction is that the subspace iteration technique will not have the same beneficial operations for either a modal reduction process or a normal modes solution with automated substructuring as it usually has in a standard **UAI/*NASTRAN*** fluid-structure interaction analysis. The special techniques of subspace iteration for fluid-structure interaction models require that the fluid and structure equations be uncoupled and that the coupling area matrix terms be available separately. Since the area coupling matrices are not stored on the SOF, this aspect of subspace iteration is not available in this situation.

23.3 INPUT DATA REQUIREMENTS

This section describes the input data requirements for coupled fluid-structure interaction analyses using **UAI/*NASTRAN***.

23.3.1 Executive Control Commands

No special Executive Control commands are necessary in order to include the effect of the fluid elements in the finite element model and conduct the coupled analysis. The conventional Modal or Direct Rigid Format solutions are called as usual with the **SOL** Executive Control command.

23.3.2 Substructuring Commands

Chapter 19 describes the commands required for substructuring analysis. As noted in section 23.2.5, the are limitations on the usual substructuring modeling and analysis practices due to the inclusion of the fluid model. Model creation, reduction, and coupling of substructures can be conducted as specified in Chapter 19.

23.3.3 Case Control Commands

You use the standard Case Control commands for the Modal and Direct approach Rigid Format solutions using the fluid-structure interaction capability. Single and multipoint constraints (SPC and MPC) may be imposed on the structural model, SUBCASEs defined, eigenvalue extraction methods specified, dynamic reduction selected, sets defined, solution results printed for GRIDs and elements, and plot output specified exactly as in conventional analyses. However, damping of the fluid motion is not available. Hence, any selection of damping will influence only the structural system.

To obtain output of the pressures in the fluid at the fluid GRID points, you use the **PRESSURE** Case Control command. As described in Section 23.1.4, the manner in which free surface GRID points are treated in the compressible and incompressible cases is different. For the incompressible case, the degrees of freedom are displacements, while in the compressible case, they are pressures. As a result, you request output in the first case with the **DISP** command, and in the second case with the **PRESSURE** command.

Finally, Section 23.2.5 describes the special Case Control commands used when performing FSI using the modal synthesis approach in the APEX program.

23.3.4 Bulk Data Entries

This section describes the Bulk Data entries which are used in performing fluid-structure interaction analyses.

23.3.4.1 Fluid Elements

GRID Bulk Data entries are used to specify the geometric coordinates and coordinate systems of the fluid degrees of freedom, as in the usual practice. The solid elements **CHEXA**, **CPENTA** and **CTETRA** are used to model the fluid. The fluid connection entry refers to a **PSOLID** Bulk Data entry, as follows:

PSOLID	**PID**	**MATF**							

where **PID** is the property identification number and the **MATF** refers to a fluid material Bulk Data entry, **MATF**. The format of the **MATF** is:

MATF	MID	ρ	B						

where **MID** is the material identification number, ρ refers to the mass density, and the parameter **B** is the bulk modulus of the fluid. For an incompressible fluid, **B** is set to zero. You may also specify the ρ be considered a weight density by including the **WTMASS** parameter in your Bulk Data packet.

23.3.4.2 Special Acoustic Data

The acoustic absorber element was described in Section 23.1.6.2. These elements are defined by **CHACAB** Bulk Data entries:

CHACAB	EID	PID	GID1	GID2	GID3	GID4	GID5	GID6	-cont-
-cont-	GID7	GID8	GID9	GID10	GID11	GID12			-cont-
-cont-			GID17	GID18	GID19	GID20			

Each element has an identification number, **EID**, and a property identification number, **PID**. The element is defined with either eight or 16 GRID points, **GIDi**. Note that the format of this entry is similar to the **CHEXA** for consistency. The **PID** selects a corresponding **PACABS** Bulk Data entry:

PACABS	PID	OP	RDEF	XDEF	WDEF	TSTAREA	FCUTOFF	B	-cont-
-cont-	M	K							

Where **PID** is the property identification number. The field **OP** specifies whether the properties will be defined by tabular data (**YES**) or derived from an equivalent structural model (**NO**). The default value is **YES**. **RDEF** and **XDEF** are identification numbers of **TABLEDi** entries which define the resistance and reactance, respectively. Both data items are required. **WDEF** is the identification number of a **TABLEDi** entry which defines the weighting function. Its default value is **1.0**. The **TSTAREA** is the area of the test specimen. Its default value is also **1.0**. The cutoff frequency of the least squares fit of the impedance curves defined by **RDEF**, **XDEF**, and **WDEF** is specified by **FCUTOFF**. Its default value is **500.0**. Otherwise, **B**, **M**, and **K** are the equivalent damping, mass, and stiffness values used if **OP** is **NO**.

The second element, the acoustic barrier, is described in Section 23.1.6.1. These elements are defined by **CHACBR** Bulk Data entries:

CHACBR	EID	PID	GID1	GID2	GID3	GID4	GID5	GID6	-cont-
-cont-	GID7	GID8	GID9	GID10	GID11	GID12			-cont-
-cont-			GID17	GID18	GID19	GID20			

Each element has an identification number, **EID**, and a property identification number, **PID**. The element is defined with either eight or 16 GRID points, **GIDi**. Note that the format of this entry is similar to the **CHEXA** for consistency. The **PID** selects a corresponding **PACBAR** Bulk Data entry:

PACBAR	PID	MBACK	MSEPTM	FRESON	KRESON				

Where **PID** is the property identification number. **MBACK** specifies the mass per unit area of the backing material and **MSEPTM** the mass per unit area of the septum. **FRESON** and **KRESON**

are used to define the resonant frequency at which the transmission loss is a minimum. **FRESON**, which can be obtained from the standard ASTM test, is used to compute an equivalent spring value. This spring constant may also be entered directly as **KRESON**, in which case the **FRESON** field is left blank.

The final special data is the acoustic source which is defined by the Bulk Data entry:

ACSRCE	LID	ADEF	τDEF	θDEF	CTAB	RHO	B		

As with other frequency-dependent loads, the input includes a load set identification number, **LID**, and set identification numbers (**ADEF**,τ**DEF**, and θ**DEF**) which reference **DAREA, DELAY and DPHASE** entries. The field **CTAB** references **TABLEDi** Bulk Data entries which defines power vs. frequency for the source. Finally, **RHO** and **B** give the density and Bulk Modulus of the fluid, respectively.

23.3.4.3 Other Data

In addition to the modeling the fluid with GRID entries and solid elements, the fluid-structure interaction analysis has several additional data options. If the fluid in your model has a free surface and you desire slosh modes, then you must identify the fluid GRIDs on the surface using **FLFREE** Bulk Data entries. The form of this entry is:

FLFREE	GID1	GID2	GID3	GID4	GID5	GID6	GID7	GID8	-cont-

where the **GIDi** entries refer to the GRIDs on the free surface of the fluid. As many of these entry cards are allowed as necessary to describe the free surface. A gravity field must also be specified on the **FSIDATA** entry.

The Bulk Data entry **FLSTR** may be used to optionally specify the elements which make up the fluid-structure interface:

FLSTR	EIDF	EIDS1	EIDS2	EIDS3	*CONTINUES WITH LIST OF ELEMENT IDS*	-cont-

where **EIDF** is the identification number of a fluid element and **EIDSi** are the structural elements which are in surface contact with the fluid element. The recommended approach, however, is to allow **UAI/*NASTRAN*** to automatically generate this interface.

One final bulk data entry may be required:

FSIDATA	GRAVID	TOLER	PREFDB	FSIEPS	DBOUT	CFREQ	NTANKS	GID	-cont-
-cont-	DOF								

You use this entry to select and control computational processes specific to the Fluid-Structure Interaction problem. **GRAVID** identifies a **GRAV** Bulk Data entry that identifies a gravity field to be applied to the fluid model. This is required when **FLFREE** Bulk Data entries are present and the fluid slosh modes are desired.

TOLER allows you to adjust the tolerance used to identify the fluid-structure interface points, when the **FLSTR** Bulk Data entries are not included. An interface between a fluid element and a structural element is created when the projected distance from a face of the structural element to a face of the fluid element is within **TOLER**.

PREFDB specifies the reference pressure used to compute Sound Pressure Levels in relative dB. You must supply **PREFDB** in the appropriate units to have relative SPL's computed correctly. For example, the standard reference Pressure for air is 20.0×10^{-6} Pascal.

FSIEPS is the epsilon value used when inverting the fluid stiffness matrix to avoid the singularities due to constant pressure modes.

DBOUT selects the SPL output as either an **RMS** or **PEAK** value.

The **CFREQ** field is used to specify whether the center frequencies of $1/n$ octave bands are to be used when computing the combined SPL in frequency response analyses. If you specify **CFREQ=YES**, then the frequency list selected by the Case Control command **FREQ** must be the center frequencies of $1/n$ octave bands.

NTANKS defines the number of fluid tanks, or compartments, in your FSI model. A tank is defined as a volume of fluid completely enclosed by structural elements or a free surface. **NTANKS** does not include enclosed volumes for which any fluid degree of freedom has been constrained by an SPC.

Finally, **GID** and **DOF** specify a reference degree of freedom for free surface motion as described in detail in Section 23.2.5.

When using APEX to perform Design Optimization, you may also specify design constraints on acoustic responses. This is done using the **DCDYNRG** Bulk Data entry which is described in detail in Chapter 25.

23.3.5 Solution Results

The solution results depend on the type of analysis conducted. For Normal Modes analyses, free surface, sloshing modes and structural interaction modes are summarized. Selected eigenvectors (displacements of structural degrees of freedom and pressures at fluid grids) are printed. For both Transient Response and Frequency Response analyses, response versus time for quantities specified in Case Control are output.

A special fluid-structure interaction solution quantity is the pressure at the fluid GRID points. This quantity is selected with the **PRESSURE** Case Control command. The results are presented in tabular format as shown in Table 23-1. The following table describes the meaning of the columns which were defined in Section 23.1.9.

Table 23-1. PRESSURE SOLUTION RESULTS

MODE NUMBER = 4 FREQUENCY = 1.222874E+02 HZ. EIGENVALUE = 5.903686E+05

R E A L E I G E N V E C T O R (PRESSURE) N O . 4

POINT ID.	TYPE	P	P(RMS)	DB	DB(A)	COMB. DB(A)
1	GRID	5.222277E-02	3.692707E-02	-2.865310E+01	-4.511881E+01	-7.380379E+01
2	GRID	-2.761615E-02	-1.952757E-02	-3.418704E+01	-5.065274E+01	-6.811077E+01
3	GRID	-2.156816E-02	-1.525099E-02	-3.633404E+01	-5.279975E+01	-6.918403E+01
4	GRID	-2.073830E-02	-1.466419E-02	-3.667484E+01	-5.314054E+01	-7.781104E+01
5	GRID	-4.922258E-04	-3.480562E-04	-6.916702E+01	-8.563272E+01	-7.586577E+01
6	GRID	4.592928E-04	3.247690E-04	-6.976851E+01	-8.623421E+01	-7.917668E+01
7	GRID	-3.903903E-04	-2.760476E-04	-7.118032E+01	-8.764603E+01	-7.988207E+01
8	GRID	3.837255E-04	2.713349E-04	-7.132989E+01	-8.779559E+01	-7.694676E+01
9	GRID	1.000000E+00	7.071068E-01	-3.010300E+00	-1.947601E+01	-4.893164E+01
10	GRID	-9.437832E-01	-6.673555E-01	-3.512855E+00	-1.997856E+01	-5.219176E+01
12	GRID	-7.913811E-01	-5.595909E-01	-5.042587E+00	-2.150830E+01	-4.994757E+01
211	GRID	7.415584E-01	5.243610E-01	-5.607392E+00	-2.207310E+01	-5.323871E+01

A solution quantity specific to FSI analysis is the pressure at the fluid GRID points. This quantity is selected with the **PRESSURE** Case Control command. The table below describes the results from a normal modes analysis of a compressible fluid-structure model.

COLUMN	DESCRIPTION
P	peak pressure value
P(RMS)	the root mean square valued of P. For Normal Modes and Frequency Response this is $\frac{P}{\sqrt{2}}$. For Transient Response this is the RMS value of P from the first time step to the current time step.
DB	Sound Pressure Level in peak or RMS pressure in dB (relative to PREFDB)
DB(A)	A-network weighted Sound Pressure Level peak or RMS pressure in dB (relative to PREFDB)
COMB DB(A)	Combined A-weighted Sound Pressure Level

23.4 MODELING GUIDELINES

The fluid portion of an FSI model is discretized in the same manner as the structural portion. In general, the fluid mesh may be less dense than the structural mesh, and the fluid-structure interface in **UAI/*NASTRAN*** is designed to allow for this incompatible mesh condition. However, bad results may be obtained when the fluid mesh is more dense than the structural mesh. Therefore it is highly recommended that the structural grid density be as great or greater than the fluid grid density at the interface. Please note that the recommendation of eight or more elements per wavelength is applicable to both the fluid and structural mesh.

The use of any static or dynamic reduction technique to reduce the active degrees of freedom can result in a constraint on the resulting system matrices. This can have the effect of increased frequencies of predicted modes or of missing modes entirely. This is true for any structural model, however for Fluid-Structure Interaction problems the solution matrices are very dense, making some form of reduction required to produce a tractable problem.

A special effort has been made to avoid this difficulty for modal solutions. The Subspace Iteration method will perform an eigensolution of medium to large problems with minimum memory and disk space required. The use of the Lanczos method will usually require much greater memory and disk space, and it may be less efficient.

For small to medium problems, you may use direct solutions. In this case, dynamic reduction is the recommended approach.

23.5 EXAMPLE PROBLEM

This section presents an example problem which illustrates some of the important features of coupled fluid-structure interaction analyses. The problem consists of the prediction of axisymmetric sloshing and hydroelastic modes of a full hemispherical tank. This example has previously been documented in the literature(e.g. [Guyan69] and [SRI75]).

Example Problem 23-1
A hemispherical tank of radius 200 in. is full of a non-viscous liquid. A 15 degree sector of the tank is modeled in order to obtain the axisymmetric slosh and coupled fluid-structure hydroelastic modes.

The physical model of the structural container and fluid is shown in Figures 23-1 and 23-2, respectively. The finite element model of the structure is composed of QUAD4 shell elements, and 8-noded HEXA, and 6-noded PENTA elements are to model the fluid. The physical data is given in the adjacent table. The input data stream for this problem is found in file **FSI01**.

STRUCTURE	**Radius**	**200 in**
	Thickness	**0.1 in**
	Young's Modulus	**1.0E7 psi**
	Poisson's Ratio	**0.33**
	Mass Density	**2.59×10^{-4}**
FLUID	**Mass Density**	**1.06×10^{-4}**
OTHER	**Gravity Constant**	**386.9 in/sec^2**

Figure 23-4. MODEL OF THE TANK

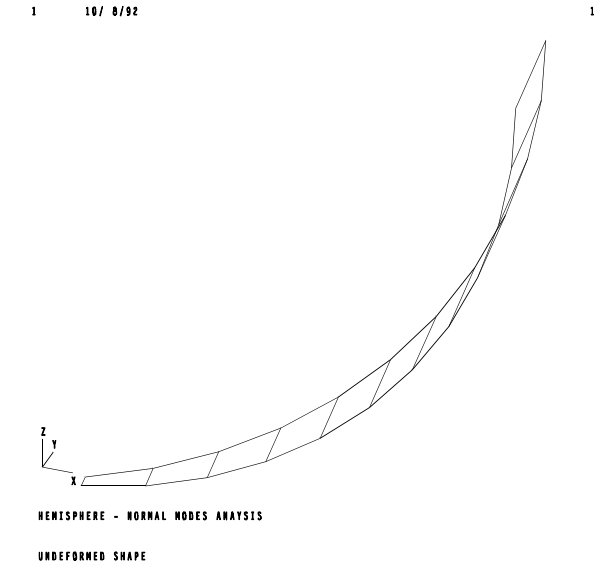


Figure 23-5. MODEL OF THE FLUID

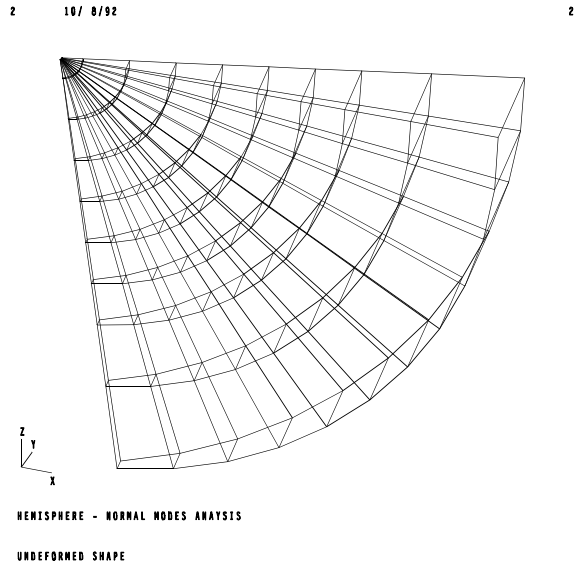


Table 23-1 provides a comparison of the results obtained using the **UAI/*NASTRAN*** fluid-structure interaction capability with those obtained by previous investigators. The results show that close agreement is achieved with the predictions obtained by [Guyan69] as well as those of other investigators (0 - 9% differences).

TYPE	MODE	NATURAL FREQUENCY — HZ			
		[Guyan69]	[Hsiung73]	[Rush75]	UAI/*NASTRAN*
SLOSH	1	0.46	0.43	0.43	0.43
	2	0.62	0.62	0.62	0.60
	3	0.75	0.81	0.76	0.74
	4	0.86	1.00	0.90	0.87
BULGE	1	6.69	7.26	6.62	6.70
	2	9.92	11.47	10.25	10.29
	3	12.59	14.56	12.35	13.13

23.6 REFERENCES

[Guyan76]: Guyan, R. J., Ujihara, B. H., and Welch, P. W., "Hydroelastic Analysis of Axisymmetric Systems by a Finite Element Method," Proceedings of the 2nd Conference on Matrix Methods in Structural Mechanics, Air Force Flight Dynamics Lab, AFFDL-TR-68-150, 1969.

[Hsiung73]: Hsiung, H., "Dynamic Analysis of Hydroelastic Systems Using the Finite Element Method," PhD Dissertation, University of Southern California, Los Angeles, California, 1973.

[Rush75]: Rush, R. H., "A Modal Analysis of the Fluid Shell Dynamics Interaction Problem," ME Thesis. University of Alabama, Huntsville, 1975.

[SRI75]: "A Study of Axisymmetric Modes in Thin-Walled Uniform Thickness Hemispherical and Hemispherical-Spherical Tanks Containing Liquid," Final Report, Contract NAS8-30270, Southwest Research Institute, December 1975.

Chapter 24

HEAT TRANSFER ANALYSIS

UAI/*NASTRAN* provides three analysis disciplines for solving heat transfer problems. These capabilities are built on the power and flexibility of the structural analysis methodology, thus allowing the representation of complex geometries using any **UAI/*NASTRAN*** finite elements except those for axisymmetric harmonic analysis. In addition, special heat transfer elements have been formulated for modeling convection and radiation. Scalar heat conduction elements, which you may use to model lumped thermal conduction or capacitance, allow the simulation of the thermal models typically used with older finite difference solution schemes.

24.1 TERMINOLOGY

All systems which are affected by temperature gradients result in a transfer of energy which is called ***Heat Transfer***. There are three recognized modes of heat transmission. These are called ***Conduction***, ***Convection*** and ***Radiation***.

Conduction is the process by which heat flows from a region of higher temperature to one of lower temperature. ***Convection*** is a process in which heat is transferred through a combination of physical phenomena. Convection is the most important factor in energy transfer between solid surfaces and fluids in either the gaseous or liquid state. Finally, ***Radiation*** is a transfer of heat between two objects of differing temperatures which are separated in space.

The energy exchange may depend on time in a transient or unsteady manner, or it may be independent of time, in which case it is most often called steady-state behavior. The steady-state behavior may be nonlinear if there is radiation exchange.

Many sources of thermal energy may be represented in a heat transfer analysis. You may prescribe temperatures at GRID points in certain types of analyses. Temperatures at the surrounding, or ambient, points may also be defined. The ambient points contribute to the convective flux, which is a form of surface heat transfer. Other specific forms of surface heat transfer include prescribed flux and flux directed from a distant source. In general, surface heat exchange involves the heat boundary element, HBDY. In addition to these external sources of thermal energy, you may also model the internal generation of heat with the heat transfer finite elements.

24.2 MATHEMATICAL BACKGROUND

In UAI/*NASTRAN*, the formulation of heat transfer problems is analogous to the formulation developed for structural analysis.

24.2.1 General Formulation

The general governing equation for the unsteady, or transient, heat transfer analysis is:

$$\boldsymbol{C}\dot{\boldsymbol{T}}(t)+\boldsymbol{K}\boldsymbol{T}(t)+\boldsymbol{R}(\boldsymbol{T}(t)-\boldsymbol{T}_{amb})^4=\boldsymbol{P}(t) \tag{24-1}$$

where $\boldsymbol{C}$ is the heat capacity matrix, $\boldsymbol{K}$ is the heat conductivity matrix, $\boldsymbol{R}$ the radiation exchange matrix, and $\boldsymbol{P}$ the heat flow vector due to applied heat flux. $\boldsymbol{T}$ is the temperature vector, and $\boldsymbol{T}_{amb}$ the ambient temperature vector. Both $\boldsymbol{T}$ and $\boldsymbol{P}$ are time-dependent.

When the heat transfer phenomena is not time dependent, (24-1) reduces to the nonlinear time-independent, or steady-state, equation of heat transfer:

$$\boldsymbol{K}\boldsymbol{T}+\boldsymbol{R}(\boldsymbol{T}-\boldsymbol{T}_{amb})^4=\boldsymbol{P} \tag{24-2}$$

Finally, if there are no radiation effects:

$$\boldsymbol{K}\boldsymbol{T}=\boldsymbol{P} \tag{24-3}$$

which is the linear steady-state heat transfer formulation.

24.2.2 Heat Flux Formulation

The right-hand-sides of all of the above equations is the heat flow due to the applied heat flux. Depending on the source of this flux, various formulas are used to calculate the heat flow applied to the GRID points.

24.2.2.1 Prescribed Heat Flux

You may model heat flux in several different ways. For constant heat flux, Q, the heat flow into each GRID point i of a given area or element is defined as:

$$P_i = Q\,A_i \tag{24-4}$$

where A_i is the area attributed to the GRID point. Alternately, you may define the flux Q_i at each GRID:

$$P_i = Q_i\,A_i \tag{24-5}$$

24.2.2.2 Convective Heat Flux

The heat flow into a given heat boundary element resulting from convective heat flux, is described by:

$$\boldsymbol{P} = \boldsymbol{S}(\boldsymbol{T}_{amb} - \boldsymbol{T}_e) \tag{24-6}$$

where $\boldsymbol{S}$ is the element surface conduction matrix, $\boldsymbol{T}_{amb}$ is the vector of ambient temperatures, and $\boldsymbol{T}_e$ is the temperatures at the element GRID points. The surface conduction matrices are automatically calculated for each element based on its type.

24.2.2.3 Radiation from a Distant Source

When a heat boundary element is irradiated from a distant source, the heat flow into each GRID point *i* of the element is:

$$P_i = \alpha f Q A_i \tag{24-7}$$

where α is the absorptivity, and

$$f = -\boldsymbol{u} \cdot \boldsymbol{n} \tag{24-8}$$

is a measure of the fraction of the radiating beam reaching the surface of the element. In this equation, $\boldsymbol{u}$ is the unit vector of the radiating beam, and $\boldsymbol{n}$ is the outward unit normal to the surface of the element. Note that *f* is set to zero when the two vectors are pointing in the same direction. This would be the case where the radiating beam approaches the active surface of the element from behind.

24

24.2.2.4 Radiation Exchange between Surfaces

The governing relationship between the vector of heat flow due to radiation at all of the points in the model, $\boldsymbol{Q}_g$, and nodal temperatures in absolute scale, $\boldsymbol{T}_g$, is:

$$\boldsymbol{Q}_g = -\boldsymbol{R}_g \boldsymbol{T}_g^4 \tag{24-9}$$

$\boldsymbol{R}_g$ is the radiation exchange matrix which is calculated from the emissivity and absorptivity, and the radiation exchange coefficients. The latter may be automatically calculated in **UAI*/NASTRAN*** as explained later in this chapter.

24.3 THE HEAT TRANSFER BOUNDARY ELEMENT

Unlike structural analysis which deals with continua, there are a number of different heat transfer phenomena which relate to surface effects. These include heat flux into a surface, heat flux from a distant source, convection from a surrounding medium, and radiation exchange between surfaces. Rather than using the normal structural finite elements to define such effects, **UAI/*NASTRAN*** provides the heat transfer boundary element, HBDY.

The HBDY elements overlay the structural elements and thus define an *active* surface for them. To define more than one active surface, you must use multiple HBDY elements. For example, if both sides of a QUAD4 element are capable of radiating, you must define two HBDY elements with opposite numbering orders using the same GRID points.

Each boundary element is defined using a **CHBDY** Bulk Data entry:

CHBDY	EID	PID	TYPE	GID1	GID2	GID3	GID4	VIEWID	-cont-
-cont-	GIDA1	GIDA2	GIDA3	GIDA4	V1	V2	V3		

As with other elements, **EID** is the element identification number, and **PID** is the identification number of a boundary element property entry defined by **PHBDY** Bulk Data entries. The third field, **TYPE**, is used to specify a boundary type on which the thermal effect will act. These types, along with the specific uses of the remaining fields on the **CHBDY** entry are described in the next section.

The **PHBDY** Bulk Data entry is required for all boundary elements having the **TYPE**s **POINT**, **LINE** and **ELCYL**, and is otherwise used only when you are modeling convection, heat flux from a distant source, or radiation effects. The format of the **PHBDY** entry is:

PHBDY	PID	MID	AF	E	α	R1	R2		

The **MID** field selects a convective material property identification number. This field is used only when you are modeling convection, and it may only reference a **MAT4** Bulk Data entry. The **E** field defines the emissivity which is used only when modeling radiation effect, and the α field defines the absorptivity, which is used when modeling heat flux due to radiation from a distant source. The other fields are only used for some of the different types as discussed below. The definition of the various types of **CHBDY** elements are shown in Figure 24-1.

24.3.1 The POINT Boundary Element

The HBDY element of **TYPE POINT** allows you to define heat transfer effects at a single point. This is defined using a Bulk Data entry of the form:

CHBDY	EID	PID	'POINT'	GID				VIEWID	-cont-
-cont-	GIDA				V1	V2	V3		

The **GID** specifies the point of application. When you are modeling convection, the GRID or SCALAR point given by **GIDA** defines the ambient temperature. Finally, if you are modeling a heat flux from a distant source, you must define the components of a vector *V* which defines the normal vector. The normal vector always points outward from the active surface of the element. The components of this vector must be in the basic coordinate system. An HBDY of **TYPE POINT** must always reference a **PHBDY** entry:

PHBDY	PID	MID	AF	E	α				

This entry must include an area factor, **AF**, which defines the area over which all heat transfer effects will be applied. As described above, the **MID**, **E** and α fields are used only for selected effects.

24.3.2 The LINE Boundary Element

The **LINE** element connects two GRID points. The effective area of the surface is then specified using the referenced **PHBDY** Bulk Data entry. The **CHBDY** Bulk Data entry for this element is:

CHBDY	EID	PID	'LINE'	GID1	GID2			VIEWID	-cont-
-cont-	GIDA1	GIDA2			V1	V2	V3		

When you are modeling convection, the GRID or SCALAR points given by **GIDA1** and **GIDA2** define the ambient temperature. Finally, if you are modeling a heat flux from a distant source, you must define the components of a vector ***V*** which defines the normal vector. The components of this vector must be in the basic coordinate system.

An HBDY element having a **TYPE** of **LINE** must always reference a **PHBDY** entry. The form of this entry is the same as **TYPE POINT**. In this case, the area factor, **AF**, which defines the area over which heat transfer effects will be applied may be used in several ways. For example, for a line which represents a circular cylinder, the area factor is equal to the circumference of the cylinder.

24.3.3 The ELCYL Boundary Element

The **ELCYL**, or elliptic cylinder, is defined by two GRID points, much like the **LINE** element. Its surface area, however, is calculated using the principal radii of the cross-section which are supplied on its **PHBDY** entry. The **CHBDY** Bulk Data entry is:

CHBDY	EID	PID	'ELCYL'	GID1	GID2				-cont-
-cont-	GIDA1	GIDA2			V1	V2	V3		

As you can see, the **ELCYL** element may not reference a **VIEW** Bulk Data entry. Hence, you must input the radiation exchange coefficient calculations involving this **TYPE**. Furthermore, you may only prescribe heat flux from a distant source for this **TYPE** using the **QVECT** Bulk Data entries. **QBDY1** and **QBDY2** entries are not supported for this **TYPE**.

When you are modeling convection, the GRID or SCALAR points given by **GIDA1** and **GIDA2** define the ambient temperature. Finally, if you are modeling a heat flux from a distant source, you must define the components of a vector ***V*** which defines the normal vector. The components of this vector must be in the basic coordinate system.

An HBDY element having a **TYPE** of **ELCYL** must always reference a **PHBDY** entry. The form of this entry is:

PHBDY	PID	MID	AF	E	α	R1	R2		

As described above, **R1** and **R2** are the principal radii of the cross-section. The **MID**, **E** and α fields are used only for selected effects described earlier.

24.3.4 The REV Boundary Element

When you are using the axisymmetric ring elements, TRIARG and TRAPRG, you may define boundary effects using an HBDY element with a **TYPE** of **REV**. The Bulk Data entry defining this element is:

CHBDY	EID	PID	'REV'	GID1	GID2			VIEWID	

The points **GID1** and **GID2**, which are part of the axisymmetric ring model, are defined in the x-z plane. The outward normal to the element is defined by the line perpendicular to the line segment connecting the two points, going away from the z-axis as shown in Figure 24-1d.

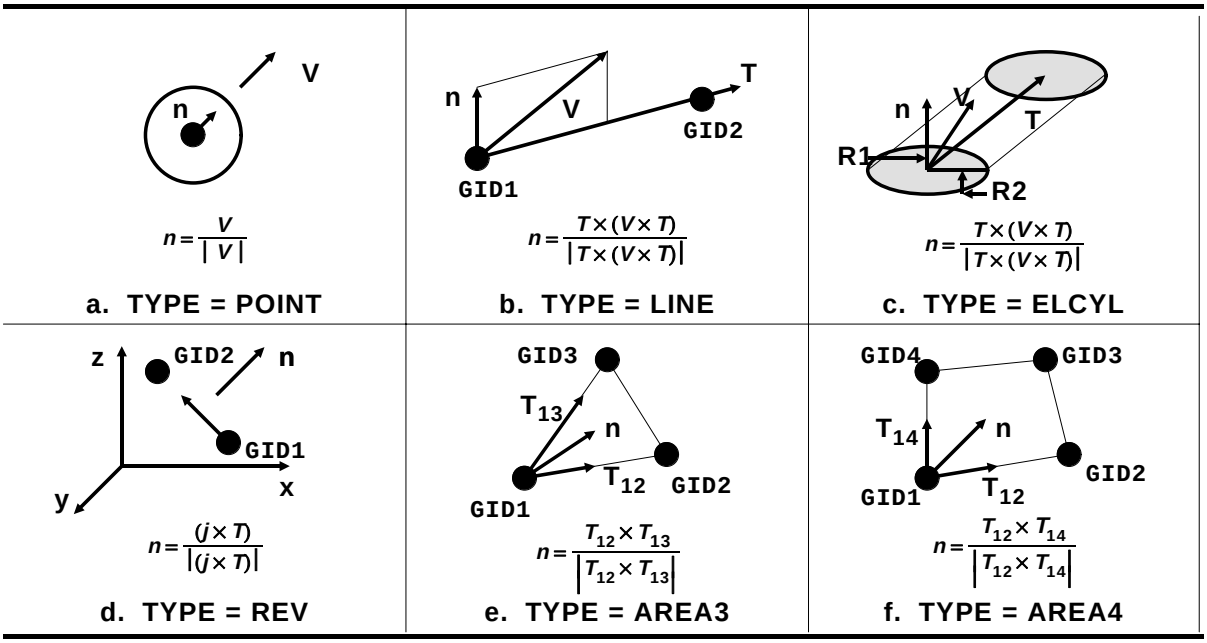
24.3.5 The AREA3 and AREA4 Boundary Elements

Often your finite element model will represent a plate structure or a three-dimensional solid structure which has thermal effects on its surfaces. You define such effects by using the boundary elements with a **TYPE** of **AREA3**, or a **TYPE** of **AREA4**. The former is used with either the TRIA3 and TRIA6 plate and shell elements, or on the triangular faces of the PENTA or TETRA solid elements. The latter is used with either the QUAD4 and QUAD8 plate and shell elements, or on the quadrilateral faces of the HEXA or PENTA solid elements. The Bulk Data entry for these two types is:

CHBDY	EID	PID	'AREA3' 'AREA4'	GID1	GID2	GID3	GID4	VIEWID	
	GIDA1	GIDA2	GIDA3	GIDA4					

The outward normal to theses element is defined in the same manner as for the corresponding structural element as shown in Figures 24-1e and 24-1f.

Figure 24-1. THE BOUNDARY ELEMENT



24.4 RADIATION EXCHANGE COEFFICIENT CALCULATIONS

In the earlier sections, it was described that how the radiation exchange coefficient matrix is used to define the relationship among all of the elements capable of either radiating, or being irradiated, or both. You may define this matrix using **RADLST** and **RADMTX** Bulk Data entries, having calculated its terms externally, or use the VIEW capability in **UAI/*NASTRAN*** to automatically define the matrix.

The automated calculations for the radiation exchange coefficients are performed whenever you include the necessary information in a nonlinear or transient heat transfer analysis model. To trigger these calculations, you must first identify the HBDY elements involved in the radiation exchange. All HBDY **TYPE**s except **ELCYL** may participate in these calculations. You then use the **VIEWID** fields of the corresponding **CHBDY** Bulk Data entries to specify the type of radiation desired.

When you enter the character string **SPACE** in the **VIEWID** field, the element will radiate to a black body. You may, however, reference the identification number of a **VIEW** Bulk Data entry which most appropriately defines the shading and subelement meshing information for that element. If you negate the **VIEW** identification number in the **VIEWID** field of an HBDY element of **TYPE AREA3** or **AREA4**, the active surface of the element is switched.

The **VIEW** entry has the following form:

VIEW	**VID**	**SHDR**	**SHDE**	**NB**	**NG**	**DLIN**			

The **SHDR** and **SHDE** fields define whether an element can be a **SH**a**D**e**R** of other elements, or can be a **SH**a**D**e**E**. The next two fields allow you to specify a subelement mesh size in the predefined directions for the HBDY element. These will assist in defining more clearly the fraction of each element which can view, or can be viewed from, other elements. The last field is a measure of offset which is only used with an HBDY element of **TYPE LINE**.

You may further control the method and output from these calculations by including a **VIEWOP** entry in the Bulk Data packet. Note that only one of these entries is allowed in a model.

24.5 INPUT DATA REQUIREMENTS

This section describes the **UAI/*NASTRAN*** input data requirements for performing heat transfer analyses.

24.5.1 Executive Control Commands

The only Executive Control commands used for heat transfer analyses are the approach selection:

```
APP HEAT
```

and the Rigid Format selection:

```
SOL 1   (Steady-State Conduction)
SOL 3   (Nonlinear Steady-State)
SOL 9   (Transient)
```

24.5.2 Case Control Commands

There are two special Case Control commands which may be used in heat transfer analyses to control the output. These are **FLUX**, which requests the output of element flux and gradient information, and **THERMAL**, which requests the output of GRID point temperatures. You may use the **THERMAL** command to request that the temperature results be written to the **BULK** file with a distinct identification number. This information may then be directly used in a subsequent thermal stress analysis model as a temperature load distribution.

Some of the other Case Control commands take on a special meaning for heat transfer analyses. For example, the **DISP** command is synonymous with **THERMAL**, and the **VELOCITY** command in a transient heat transfer analysis requests the output of rate of change in temperature at GRID points with respect to time. In contrast, however, the **ACCELERATION** command is rendered meaningless. In general, the quantities associated with displacements in a structural analysis may be associated with temperatures in a heat transfer analysis. Similarly, forces are analogous to heat flow. Therefore, forces of constraint, **SPCF**, would be a measure of heat flow lost to the environment.

24.5.3 Bulk Data Entries

There are four groups of Bulk Data entries which are used specifically in heat transfer analyses. These are discussed in the following sections.

24.5.3.1 Material Properties

Thermophysical properties used by the **UAI/*NASTRAN*** finite element library, except for the HBDY element, include thermal conductivity and thermal capacity. These properties are defined using the material property definition Bulk Data entries **MAT4** for isotropic conductivity, and **MAT5** for anisotropic conductivity. In both cases, the thermal capacity is unique. The most general form of material property specification is with the **MAT5** entry:

MAT5	MID	KXX	KXY	KXZ	KYY	KYZ	KZZ	CP	

With this entry you may define a matrix of thermal conductivity which is variant in all three dimensions. If this is not the case, it suffices to use the **MAT4** Bulk Data entry which has only one value for thermal conductivity:

MAT4	**MID**	**K**	**CP**						

In both entries, **CP** is the thermal capacity per unit volume. This value is obtained by multiplying the material density into its specific heat.

MAT4 Bulk Data entries may also be used to define the convective properties of a heat boundary, HBDY, element. These include convective thermal conductivity, sometimes called the convective film coefficient, and convective thermal capacity, or film capacity. When used for this purpose, the data is specified with respect to the unit surface area of the element rather than its volume.

The conductivity properties defined on the **MAT4** and **MAT5** entries may be made temperature-dependent by using **MATT4** and **MATT5** Bulk Data entries, respectively, sharing the same identification numbers with the entries which reference them. These entries, in turn, refer to **TABLEM1**, **TABLEM2**, **TABLEM3**, and **TABLEM4** entries to define a tabular dependency.

24.5.3.2 Applied Heat

Five special Bulk Data entries may be used to apply heat to your heat transfer model. **QBDY1** is designed for application of a constant flux to an HBDY element. You may vary the flux at each GRID point of the element by using **QBDY2** Bulk Data entries. The application of a constant flux to GRID points does not require the presence of an HBDY element. By using the **QHBDY** entry you may remove this requirement.

QVECT Bulk Data entries allow you to direct radiating flux from a distant source into an HBDY element. The rays are assumed to be parallel. Only the active surface of the element may be irradiated.

You may also model internal heat generation using the **QVOL** Bulk Data entry. This entry generates heat at a specified rate, and references standard library of finite elements.

24.5.3.3 Radiation Exchange Coefficients

The effects of radiation in nonlinear steady-state and transient analyses require the presence of a matrix of exchange coefficients. This matrix may be either generated automatically by using the VIEW capability, or calculated external to **UAI*/NASTRAN*** and input using one **RADLST** entry and multiple **RADMTX** entries. The **RADLST** entry consists of a list of HBDY element identification numbers. The position of each number in the list corresponds to a column number specified on one of the **RADMTX** entries. Each number may appear more than once in this list to account for possible participation of both surfaces of an HBDY element in the radiation exchange. The **RADMTX** entries define each column of the matrix. Since the matrix is symmetric, only the lower triangular portion of it need be specified.

The recommended approach to specifying this matrix is to choose the VIEW capability by using **VIEW** and **VIEWOP** Bulk Data entries. This option has been discussed in an earlier section.

24.5.3.4 Parameters

The steady-state discipline does not use any special parameters. There are two parameters that may be optionally specified when using the nonlinear or transient heat transfer disciplines. These are:

PARAM	**SIGMA**	σ							

PARAM	**TABS**	**value**							

SIGMA represents the Stefan-Boltzman constant, and **TABS** the absolute temperature. The default for both of these parameters is **0.0**.

Under severe circumstances, you may need to tighten the convergence criterion or increase the number of iterations by using the following two parameters in your nonlinear analysis:

PARAM	**EPSHT**	**value**							

PARAM	**MAXIT**	**value**							

EPSHT is the convergence test value with a default of **0.001**, and **MAXIT** the maximum number of iterations. By default, **4** iterations will be performed before a solution is abandoned.

The **BETAH** parameter is the transient integration stability and accuracy control parameter. It is not recommended that you change the value of this parameter in a normal transient analysis from its default of **0.55**. When performing transient analysis, you may linearize the effects of radiation with the following **PARAM** Bulk Data entry:

PARAM	**RADLIN**	**1**							

24.6 SOLUTION RESULTS

The major solution results for Heat Transfer analyses are temperature profiles through the structure and temperature gradients and fluxes within the elements. Because the temperature at each GRID point in your model is a scalar quantity, the temperature output, requested with the **THERMAL** Case Control command, is in the same format as displacements for SCALAR points. This means that up to six temperature values are printed on a single line of results, as shown in Table 24-1a. The temperature gradient and flux results for each element, both of which are requested with the Case Control command **FLUX**, are shown in Table 24-1b.

24.6.1 Interface to Structural Analysis

You can use the temperature results obtained in Heat Transfer Analysis to define loads for structural analyses. This is done by requesting that the resultant temperatures be written as high-precision **TEMP** Bulk Data entries using the command:

```
THERMAL (...,BULK,SID=tempid,...) = sid
```

where *tempid* is the identification number of the resulting **TEMP** Bulk Data entries. You may use these entries in a structural analysis job by including them in your input data stream and then selecting them as thermal loads in the usual manner using the Case Control command:

```
TEMP(LOAD) = tempid
```

In the case of transient Heat Transfer, a separate set of temperatures is written for each time step. The *tempid* that you specify is used as a base value, and additional identification numbers are automatically generated using the rule:

```
tempid(step) = tempid * 1000 + step
```

where *step* is the integration step number.

Table 24-1. HEAT TRANSFER SOLUTION RESULTS

a. Temperatures

T E M P E R A T U R E V E C T O R

POINT ID.	TYPE	ID	VALUE	ID+1 VALUE	ID+2 VALUE	ID+3 VALUE	ID+4 VALUE	ID+5 VALUE
1	TEMP	0.0		0.0	0.0	0.0	0.0	0.0
7	TEMP	0.0						
101	TEMP	0.0		3.286702E+01	5.692711E+01	6.573403E+01	5.692711E+01	3.286702E+01
107	TEMP	0.0						
201	TEMP	0.0		6.714314E+01	1.162947E+02	1.342863E+02	1.162947E+02	6.714314E+01
207	TEMP	0.0						

b. Element Temperature Gradients and Fluxes

F I N I T E E L E M E N T T E M P E R A T U R E G R A D I E N T S A N D F L U X E S

ELEMENT-ID	EL-TYPE	X-GRADIENT	Y-GRADIENT	Z-GRADIENT	X-FLUX	Y-FLUX	Z-FLUX
1	QUAD4	3.286701E+00	8.216754E+00		-3.286701E+00	-8.216754E+00	
2	QUAD4	2.406010E+00	2.244853E+01		-2.406010E+00	-2.244853E+01	
3	QUAD4	8.806917E-01	3.066529E+01		-8.806917E-01	-3.066529E+01	
4	QUAD4	-8.806917E-01	3.066529E+01		8.806917E-01	-3.066529E+01	

24.7 MODELING GUIDELINES

This Section provides you with a number of modeling guidelines that may prove useful when performing Heat Transfer analyses. One of the most used modeling requirements is the specification of prescribed, or constant, temperatures at points within your thermal model. These differ depending on the analysis discipline that you are using. The methods of input are described in the following sections.

24.7.1 Steady-state Conduction

The definition of prescribed temperatures when performing steady-state conduction analyses is completely analogous to prescribed displacements in static structural analysis. The first way to fix the temperature of a point is to define its value using an **SPC** Bulk Data entry:

SPC	SID	GID1	DOF1	d1	GID2	DOF2	d2		

As usual, you must select the single-point constraints with the **SPC** Case Control command. Remember that when performing heat transfer analyses you may specify only the DOF Codes 0 or 1 in all Bulk Data entries. A second method, which you will find useful if your analysis has several fixed temperature conditions, is to define the temperature values using **SPCD** Bulk Data entries:

SPCD	LID	GID	DOF	TFIX					

By processing these constraints as loads, **UAI/*NASTRAN*** performs the time-consuming matrix decomposition only once. Note that when you use the **SPCD** entry that you must also place the degree of freedom on an **SPC** entry, and that you must use the **LOAD** Case Control command to select the **SPCD** values.

24.7.2 Nonlinear Steady-state Analysis

You may define a set of *estimated* temperatures for your nonlinear model. This temperature distribution is defined on the **TEMP** Bulk Data entries, and must be selected in the Case Control packet with either a **TEMP(ESTIMATED)** or **TEMP(MATERIAL)** command. Any degrees of freedom with prescribed temperature must be both included in this set and appear on **SPC** or **SPC1** Bulk Data entries. If you use the **SPC** entries, the value on those entries are ignored and replaced with the values on the corresponding **TEMP** entries.

24.7.3 Transient Analysis

When performing transient heat transfer analyses, you may not define fixed temperatures using **SPC** or **SPCD** data as discussed above. Instead, it is necessary to proceed using the following two step process. First, define a scalar element which connects the GRID point for which you wish constant temperature, T, to ground. This element must have very large conductivity, k, and is most easily specified using a **CELAS2** Bulk Data entry. Second, apply a load, q, to the GRID point using an **SLOAD** Bulk Data entry. These values must obey the relation:

$$T = \frac{q}{k}$$

Only **SLOAD** Bulk Data entries may be used to apply a heat flux to a GRID point. The values of k and q should be very large relative to other heat fluxes applied to your model. For most industrial heat transfer applications the basic guideline is that the values should be five orders of magnitude larger than the largest actual flux. This method may also be applied in steady-state conduction, but it requires considerably more input data than using **SPC** entries.

24.7.4 The Thermal-Structural Analogy — Units

The differential equations of heat transfer and structural analysis take the same form. As a result, there is a natural analogy between structural and heat transfer engineering quantities. These are summarized for you convenience in Table 24-2.

Table 24-2. SUMMARY OF ENGINEERING UNITS

STRUCTURAL QUANTITIES			HEAT TRANSFER QUANTITIES		
PHYSICAL QUANTITY	**U.S. ENGINEERING UNITS**	**S.I. UNITS**	**PHYSICAL QUANTITY**	**U.S. ENGINEERING UNITS**	**S.I. UNITS**
Displacement	**in** **ft**	**m**	Temperature	**°F** **°R**	**°K**
Force	**lb**	**N**	Heat Flow Rate	**Btu/hr**	**J/sec** **W**
Stress	**lb/in^2 (psi)** **lb/ft^2**	**N/m^2**	Heat Flux	**Btu/hr-in^2** **Btu/hr-ft^2**	**J/sec-m^2**
Strain			Gradient	**°F/in** **°F/ft**	**°K/m**
Young's Modulus	**lb/in^2 (psi)** **lb/ft^2**	**N/m^2**	Thermal Conductivity	**Btu/hr-in^2-°F** **Btu/hr-ft^2-°F**	**J/sec-m-°K** **W/m-°K**
			Heat Transfer Coefficient	**Btu/hr-in^2-°F** **Btu/hr-ft^2-°F**	**J/sec-m-°K** **W/m-°K**
			Specific Heat Capacity	**Btu/hr-in-°F** **Btu/hr-ft-°F**	**J/kg-°K**
Weight	**lb**	**N**			
Mass	**lb-sec^2/in** **lb-sec^2/ft**	**Kg** **N-sec^2/m**			

Btu	British Thermal Unit	°K	Kelvin degree
Ft	Foot	m	Meter
°F	Fahrenheit degree	N	Newton
hr	Hour	psi	Pounds per square inch
in	Inch	°R	Rankine degree
J	Joule	sec	Second
Kg	Kilogram	W	Watt
lb	Pound (force)		

24.8 EXAMPLE PROBLEMS

This section presents four example problems which illustrate the use of the Heat Transfer analysis discipline. They have been selected to cover a broad range of capability and data options.

24.8.1 LINEAR STEADY-STATE ANALYSES

Three examples of linear steady-state heat analyses are provided in this section. The first illustrates simple conduction, the second analyzes a system which includes internal heat generation, and the third explores convection.

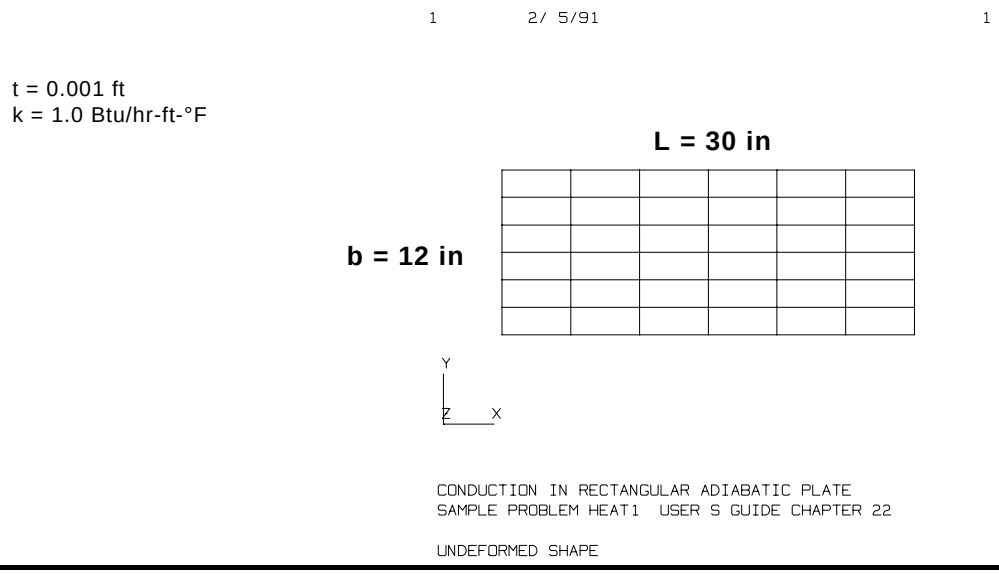
24.8.1.1 Conduction

The first example illustrates a simple steady-state conduction analysis of a flat plate subject to a prescribed temperature boundary condition and no external heat flow.

Example Problem 24-1
A thin rectangular adiabatic plate 30 in long and 12 in wide is insulated on three sides. The fourth side is maintained at a temperature that varies sinusoidally along the length of the plate as $T = 500 \sin(\pi x / 30)$. You wish to determine the temperature distribution throughout the plate.

The finite element model is composed of 36 QUAD4 elements as shown in Figure 24-2. The input data stream for this problem is found in file **HEAT1**. The heat transfer model requires very little special input data. The first data which differs from a static structural analysis is the material property definition. A **MAT4** Bulk Data entry is used to define the heat transfer characteristics of temperature-independent, isotropic materials.

Figure 24-2. ADIABATIC PLATE MODEL



For this problem, the following data is used:

MAT4	**1**	**1.0**	**1.0**						

Note that the conductivity, defined in the second field, is given a value of unity because it is uniform and, as a result, does not affect the temperature distribution. The only other special Bulk Data is the use of **SPC** entries to define the fixed boundary temperatures. The three insulated sides are constrained while the varying side has the temperatures defined with:

SPC	**100**	**602**	**1**	**250.0**					
SPC	**100**	**603**	**1**	**433.0**					
SPC	**100**	**604**	**1**	**500.0**					
SPC	**100**	**605**	**1**	**433.0**					
SPC	**100**	**606**	**1**	**250.0**					

Three Case Control commands were used to request solution results:

```
THERMAL = ALL
FLUX    = ALL
SPCF    = ALL
```

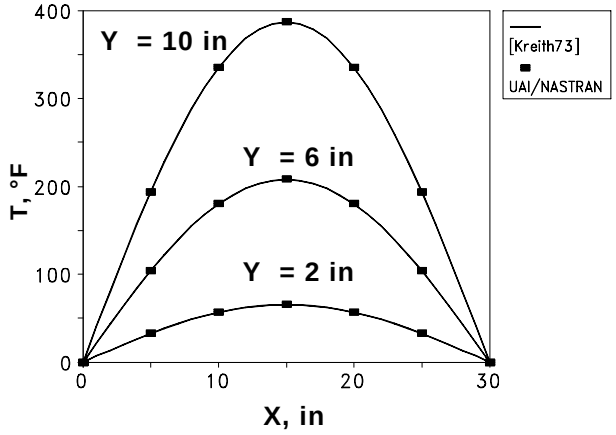
As you saw earlier, the **THERMAL** command requests the resulting temperatures, **FLUX** the resulting temperature gradients and heat flux, or heat flow per unit area, for the elements in the model, and the **SPCF** the heat flows required to maintain the constant temperatures defined by the **SPC** data.

The closed form solution to this problem, given in [Kreith73] is:

$$T(x,y) = T_m \frac{\sinh(\pi y/L)}{\sinh(\pi b/L)} \sin(\pi \frac{x}{L}) = 500 \frac{\sinh(\pi y/30)}{\sinh(12\pi/30)} \sin\left(\pi \frac{x}{30}\right)$$

A comparison of the **UAI/*NASTRAN*** results and the theoretical temperature profiles along three y-coordinate lines is shown in Figure 24-3. Equally good agreement is obtained for the temperature gradients and heat flux.

Figure 24-3. TEMPERATURE PROFILES FOR PLATE



24.8.1.2 Uniform Internal Heat Generation

The modeling of uniform internal heat generation, such as that often found in applications including electric coils, nuclear fuel elements and resistance heaters, is very simple using **UAI/*NASTRAN*** as demonstrated in the next example.

Example Problem 24-2

A long uranium fuel rod 2 in in diameter is jacketed by an annular tube in which water is flowing. The rod generates heat at a rate of 7.2 x 10^6 Btu/hr-ft^3-°F. The coolant is able to maintain a temperature of 300°F on the surface of the rod. Determine the radial temperature distribution in the fuel rod. To simplify your model, use axisymmetric element.

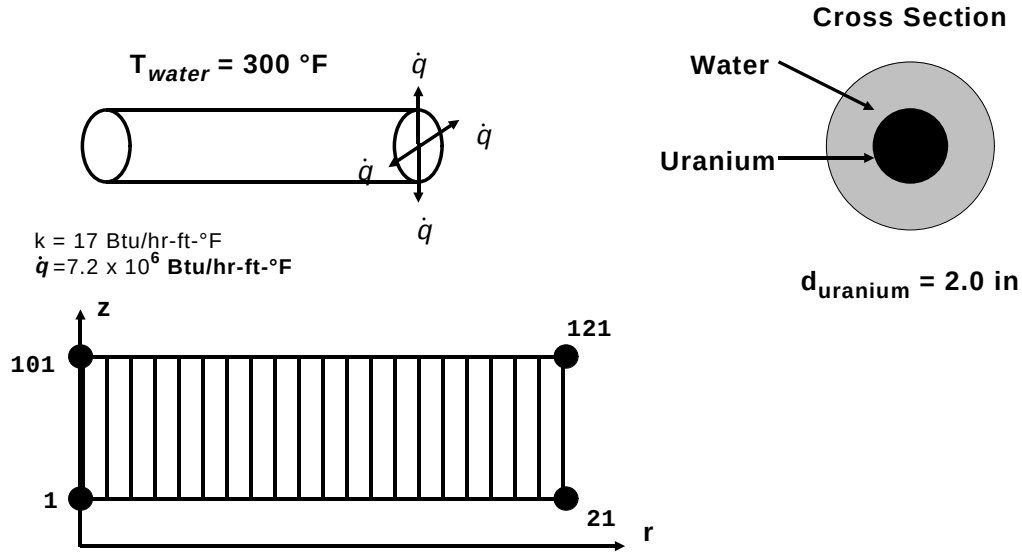
The finite element model and its engineering data are shown in Figure 24-4. The model, which is composed of 20 TRAPRG elements, is found in file **HEAT2**. Only a short vertical slice of the rod must be modeled because the heat flow is one dimensional. The only Bulk Data required to define the appropriate heat transfer conditions are **SPC** entries, which are used to enforce the prescribed temperature of 300°F at the grid points on the surface of the rod, and a **QVOL** Bulk Data entry which defines the internal heat generation:

QVOL	**200**	**7.2+6**	**1**	**THRU**	**20**				

Recall that the heat generation rate is always expressed in terms of a unit volume. The volumetric heat generation represents an external heat source, or load. As such, it must be requested with the Case Control command:

```
LOAD = 200
```

Figure 24-4. URANIUM FUEL ROD MODEL

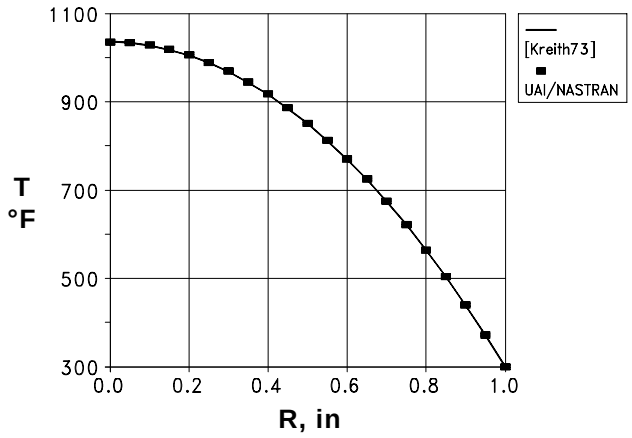


The exact solution to this problem, presented in [Kreith73], is:

$$T(r) = T_0 + \frac{\dot{q}\, r_0^2}{4k}\left[\, 1 - \left(\frac{r}{r_0}\right)^2 \,\right]$$

Figure 24-5 shows the theoretical temperature profile through the radius of the rod and the **UAI/*NASTRAN*** results. You will note the excellent agreement.

Figure 24-5. TEMPERATURE PROFILE IN FUEL ROD



24.8.1.3 Convection

The final steady-state heat conduction example illustrates the transfer of heat from a pin fin heat exchanger by convection.

Example Problem 24-3

An aluminum pin fin, shaped like a rod, is attached to a wall whose temperature is held at 600°F. The fin has a circular cross-section with a diameter of 0.5 in. The fin is cooled convectively by air at an ambient temperature of 70°F. Determine the temperature profile as a function of the axial coordinate and the temperature gradient and heat flow per unit area.

The pin fin geometry and engineering data are shown in Figure 24-6. The input data are found in file **HEAT3**. The finite element model, shown in Figure 24-7, is comprised of 10 BAR elements. Additionally, ten heat boundary elements are used to model the convective heat flow. The first special modeling technique requires that you specify the ambient fluid temperature. This is most easily done by defining a SCALAR point which is then constrained to maintain the necessary temperature. This is done with the Bulk Data entries:

SPOINT	**5000**								
SPC	**100**	**5000**	**1**	**70.0**					

It is now necessary to model the convective heat flow from the pin to the air. To do this, you must use the heat boundary element, HBDY. For this one-dimensional problem, boundary elements of type **LINE** are used. Each ROD element has a corresponding **HBDY** element. The first of these elements is defined by:

CHBDY	**11**	**1000**	**LINE**	**1**	**2**				**+CH1**
+CH1	**5000**	**5000**							

The values on the continuation entry (**5000,5000**) specify the identification numbers of GRID or SCALAR points which are held at the ambient temperatures at the ends of the **LINE**. The HBDY element references a **PHBDY** Bulk Data entry that selects heat transfer properties and defines an effective area:

PHBDY	**1000**	**2000**	**.1309**						

Figure 24-6. PIN FIN HEAT EXCHANGER

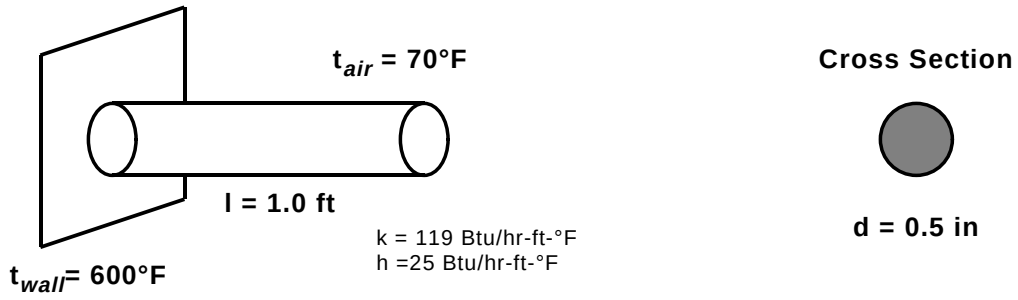
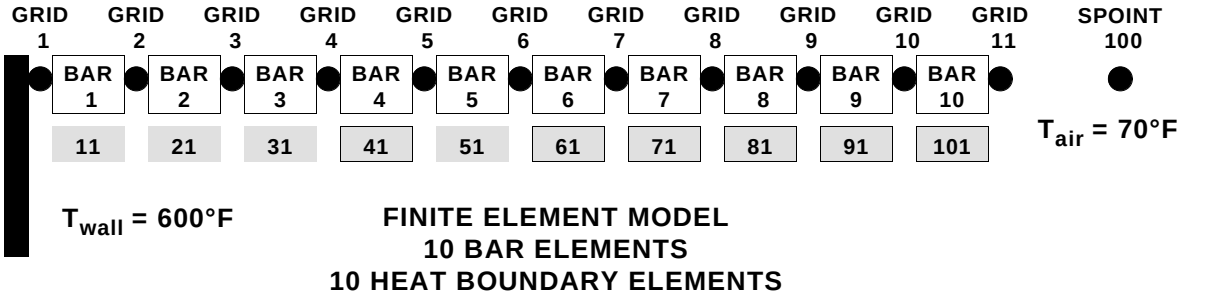


Figure 24-7. PIN FIN HEAT TRANSFER MODEL



MAT4	2000	25.							

The fourth field of the **PHBDY** entry represents the area factor. This factor defines the surface through which heat will flow. In this case, the value specified is the circumference of the fin. The **PHBDY** also references a material property which is also shown. The interpretation of the **MAT4** fields is different for boundary elements. The third field now represents the heat transfer coefficient which is also called the convective film coefficient.

The general solution to this problem is derived in [Kreith73] and is given by:

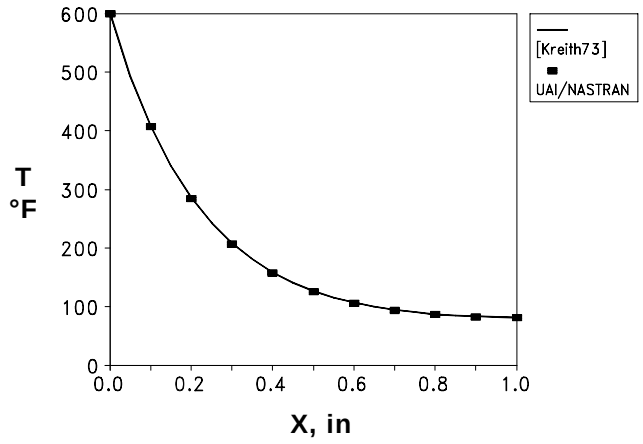
$$T(x) = (T_s - T_\infty)\frac{\cosh(mL - mx)}{\cosh(mL)} + T_\infty$$

where T_s represents the surface temperature of the wall, T_∞ represents the temperature of the fluid, and L is the length of the pin. Finally, m is:

$$m = \left[\frac{\bar{h}\,P}{k\,A}\right]^{1/2}$$

where k is the conductivity of the pin, A its cross-sectional area, P its perimeter, and $\bar{h}$ the heat transfer coefficient between the fluid and the surface of the pin. For the example problem, the area of the pin is given as 0.5 in, the pin perimeter is, for a circular cross section, its circumference.

Figure 24-8. TEMPERATURE PROFILE



The temperature gradient may be obtained by explicit differentiation:

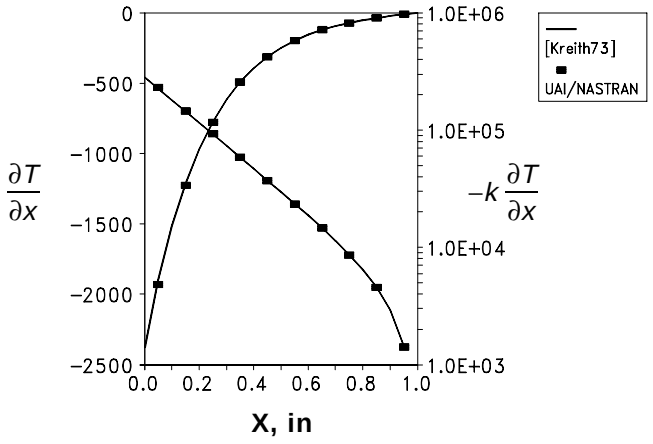
$$\frac{\partial T}{\partial x} = -m\,(T_s - T_\infty)\,\frac{\sinh(mL - mx)}{\cosh(mL)}$$

the flux is then defined simply as:

$$q = -k\,\frac{\partial T}{\partial x}$$

The **UAI/*NASTRAN*** temperature results, compared with the theory, are shown in Figure 24-8. The temperature gradient and flux are shown in Figure 24-9.

Figure 24-9. GRADIENT AND FLUX RESULTS



24.8.2 TRANSIENT ANALYSES

This section illustrates how transient heat transfer analyses are performed.

Example Problem 24-4

A large 2 in thick sheet of plastic, initially at 100°F, is suddenly immersed in boiling water so that its temperature increases to 212°F. You wish to determine the temperature history on a plane 0.5 in from the surface.

The finite element model is composed of 4 HEXA elements which, along with its engineering data, is shown in Figure 24-10. The input data stream for this problem is found in file **HEAT4**. The first important data is the specification of the initial conditions. The plate temperature is defined using:

TEMPD	**200**	**100.**							

This defines a default temperature of 100°F everywhere in the model. You then select this initial condition with the Case Control command:

```
IC = 100
```

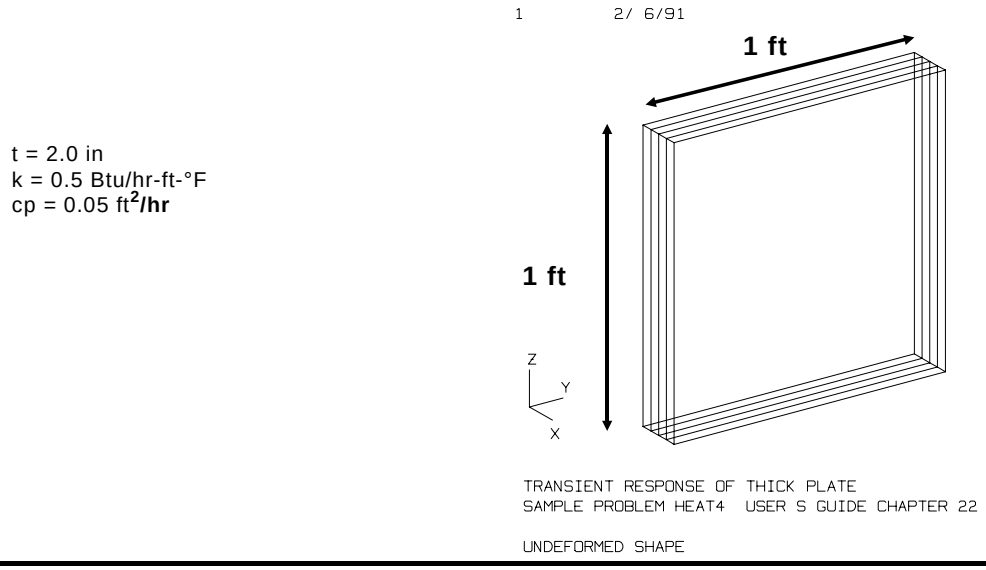
The next requirement is to maintain the surface temperature at 212°F. This cannot be done using **SPC** data as in previous examples. Instead, you must use an artifice to enforce this condition. First, data which define SCALAR springs are defined using **CELAS2** Bulk Data entries:

CELAS2	**10**	**1.+10**		**10**	**1**				

Note that the *spring constant*, which in the thermal analog represents conductivity, is set to a very large value. This is equivalent to using a large mass in a dynamics model. These springs

24

Figure 24-10. THICK PLASTIC SHEET MODEL



are connected from all of the GRID points on both external faces of the plate to ground. Next, a scalar load is applied to each point using entries of the form:

SLOAD	**500**	**10**	**212.+10**						

The actual value of the load is set to the temperature that you wish to enforce multiplied by the same large factor as the conductivity of the associated springs. This method works because of the simple relationship:

$$q = kT$$

This allows a temperature to be specified by defining a heat flow, or load, q such that:

$$T = \frac{q}{k}$$

You must now select these loads using the Case Control command:

```
DLOAD = 100
```

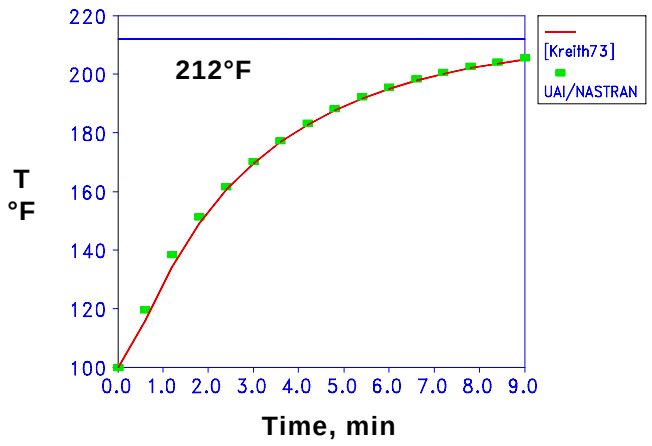
Note that the load request does not directly reference the **SLOAD** data. Instead, it must reference one of the time domain load entries, **TLOADi**. In this case, the input is:

TLOAD1	**100**	**500**		**0**	**1**				
TABLED1	**1**								
	0	**1.**	**100.**	**1.**	**ENDT**				

The **TLOAD1** entry selects the **SLOAD** data by referencing its identification number in the third field, and points to a table with defines the time dependence of these loads. In this case, the **TABLED1** Bulk Data entry was used. The table data itself indicates that the loads will be multiplied by **1.0** for all times between **1.0** and **100.0**. Chapter 9 of this manual contains detailed explanations of how you may specify transient loads.

The example is then executed in **UAI/*NASTRAN***. A comparison of the temperature profiles computed and the theoretical solution given in [Kreith73], both for the first nine minutes, are shown in Figure 24-10.

Figure 24-11. TEMPERATURE PROFILE



This page is intentionally blank.

Chapter 25

MULTIDISCIPLINARY DESIGN OPTIMIZATION

UAI/*NASTRAN* provides you with a fully automated Multidisciplinary Design Optimization (MDO) capability suitable for the design of mechanical structures. The design process can simultaneously encompass the strength, vibration, and frequency response requirements of the model. It does this within the context of all of the normal **UAI/*NASTRAN*** features such as multiple boundary conditions and loads. This feature allows you to define an objective function, design variables, and response constraints for the structure. The objective function may be the weight, mass, or volume of the structure, or it may be a single response quantity.

This chapter describes how MDO is performed and how its results may be used to assist you in improving your structural designs.

25.1 WHAT IS MDO?

Conceptually, the optimization process is quite simple. It attempts to determine the best possible value of a specified objective function, typically subject to one or more restrictions. In structural design, the objective is most often the weight or volume of the model. This is simply because for many products there is a strong correlation between weight and product cost or performance. Aircraft and aerospace structures are the most dramatic examples of this. In other cases, the objective may be to attain a specified structural response in order to *tune* the model to a specified behavior, or to correlate the model with test results. The constraints on the structure are usually those which relate to the response characteristics of the product, such as stress or strain limits, restricting specific motion, or bounding the response to an external excitation. Also, it is necessary to restrict the physical characteristics of the model, such as ensuring kinematic admissibility (e.g. the parts fit together, fit within the outer mold lines, and will not contact each other under load) and manufacturability (i.e. that the parts can be built).

Multidisciplinary Design Optimization is the process through which a structural system may be optimized simultaneously for design constraints on responses which arise from several analytical regimes, or disciplines. For example, nearly all products are constrained by strength requirements — the product cannot break — and many are further constrained by dynamic response characteristics. MDO thus considers the design constraints for all of the disciplines such that the resulting solution simultaneously satisfies all of the performance requirements. Older techniques which attempted to use the optimum results from individual disciplines or single conditions to find an MDO optimum have been shown to yield suboptimal results [Sobieski89] or results that are infeasible for other conditions that are experienced in normal operation.

25.1.1 The Nondeterministic Nature of MDO

The typical finite element analysis discipline formulates and solves a deterministic set of equations of motion. Subject to the actual approximations inherent in the discretization of the structure to form the FE model, these equations are solved exactly, as in the case of linear static analysis, or by stable iterative methods which also converge to exact results, such as eigensolutions and transient integrations. There are, however, other disciplines where the solutions are not deterministic. These include most nonlinear analyses, which must use heuristic, adaptive algorithms to arrive at some solution to a problem. Design optimization falls into the latter category.

Like the current **UAI/*NASTRAN*** nonlinear analyses, design optimization can converge to a nonunique solution: this means that in multiple executions to solve the "same" optimization problem, different answers may be found. Typically these answers will not be radically different, but they may be noticeably different. The solution to these nonlinear problems, controlled by heuristic algorithms, display an extreme sensitivity to initial conditions. This problem is further exacerbated by the heuristics themselves in that the responses of the system are used to control the subsequent iterations. Thus, there is a potential to obtain a seemingly very different answer for the same problem statement.

In examining the optimal solution, then, the designer must take care to evaluate the answer for engineering relevance. Be aware that there may really be many "optimal" answers (those that lead to very nearly the same objective function value — this is known as a *flat design space*) or additional constraints may need to be imposed on the problem to preclude the optimizer from accepting certain solutions (an underdetermined problem — this is known as a *poorly posed optimization problem*). Lastly, there may be a numerical sensitivity that yields slightly different answers over a series of design iterations.

25.2 THE DESIGN MODEL

A design model is composed of three sets of input that are different than, but related directly to, the analysis model. These are:

- The Design Variables
- The Design Constraints
- The Objective Function

The first of these is required, the latter two are optional depending on the kind of answer the designer is seeking. Typically, a constrained optimum is sought in which case all three sets of data are required. Under special circumstances, however, the designer may choose not to limit any other system responses and seek only the "best" single response. The designer may also choose to seek ***any*** answer that satisfies the plethora of requirements. The former special case is rare and is used primarily for model tuning, the latter scenario is often useful for gaining insight into the model before choosing which response is the best candidate to be placed at an extreme. Each of these sets of inputs is briefly described in the following sections.

25.2.1 Design Variables

The first step in MDO is to define some number of ***design variables***. A design variable is a characteristic of your structure that may be changed during the design process. Typically, these characteristics have been identified in the finite element model as distinct element or material properties during the process of ***model development***. The model's design variables, then, may represent simple ***physical attributes*** of the model such as the thickness of a plate, the cross sectional area of a bar, or the location of a GRID point. It may also represent material characteristics such as Young's Modulus, Poisson's Ratio, or the damping coefficient. Finally, a design variable may be defined as ***a set of relationships between two or more physical properties***. This is called ***linking***.

In order to differentiate between the physical characteristics of the structure, i.e. the model's physical attributes, and the linked design variables, the concept of a ***mathematical design variable*** is used. A mathematical design variable is an independent variable that controls one or more physical attributes of your model. These attributes are called ***physical design variables***. The actual design variables of the structural system remain the exclusive domain of the design engineer. The physical design variables are selected during model development to be representative of the design freedoms in a manner similar to that in which the FE discretization is chosen to represent the structural degrees of freedom. In all cases, mathematical design variables are used during the optimization process. The set of all possible values of the mathematical design variables is called the ***design space***, and a specific set of the variables is called a ***design point***.

25.2.2 Design Constraints

In addition to the design variables, you may also define ***constraints*** on some number of structural response quantities or model characteristics. Examples include requiring: that element stresses do not exceed their allowables; that a specific displacement does not exceed a value which would cause the modeled product to fail in its operational environment; and that the first natural frequency of the structure exceed a value that would result in a destructive excitation during operation. The response quantities that may be constrained in **UAI/*NASTRAN*** depend on the discipline, or analysis, that you are performing, whereas the model characteristics are independent of the discipline. All constraints are summarized in Table 25-1.

Table 25-1. DESIGN CONSTRAINTS

RESPONSE CONSTRAINTS			
RESPONSE QUANTITY	DISCIPLINE		
	STATICS	NORMAL MODES	FREQUENCY RESPONSE
GRID Point Displacements	✔		✔
GRID Point Velocities			✔
GRID Point Accelerations			✔
Eigenvectors		✔	
Element Stress Components	✔		
Element Strain Components	✔		
Element Force Components	✔		
Natural Frequencies		✔	
MODEL CONSTRAINTS			
Weight, Mass, or Volume	Global Constraints, Discipline Independent		

Note that there are also constraints on the values allowed for each design variable. These ***side constraints*** typically reflect fabrication requirements for a structure.

A constraint on the response quantity R can represent either a maximum value or a minimum value, or both:

$$R_{min} \leq R \leq R_{max} \tag{25-1}$$

Constraints can be either ***violated***, implying the the design point is ***infeasible*** or they can be ***satisfied***. If all constraints are satisfied, the design point is ***feasible***. Further, the term ***critical*** is applied to satisfied constraints that are near their limiting value. Within **UAI/*NASTRAN***, response constraint values, g_r, are normalized and rearranged such that a positive value ($g_r > 0.0$) indicates that the constraint is violated. This is often done, for example, using:

$$g_r = \frac{R - R_{max}}{\left|R_{max}\right|} \leq 0 \quad \text{or} \quad g_R = \frac{R_{min} - R}{\left|R_{min}\right|} \leq 0 \tag{25-2}$$

If you specify R_{max} or R_{min} to be zero, then this normalization is not performed. Also, certain specialized normalizations are used to tailor the constraint function for better function approximation, described later in the Chapter.

If there is a violated constraint, a different design point must be found to satisfy it — to place the response value within the specified limit. The change in a constraint value with respect to a design variable is called its ***sensitivity*** to that variable. Because changes in the design variables cause changes in the structural response, a finite element analysis must be performed to determine the sensitivities. These responses, in turn, depend on the analytical discipline. The handling of constraints, a significant part of solving the optimization problem, is described in

detail in the next section of this Chapter.

The optimization process then requires that the minimization, maximization or constraint satisfaction be performed subject to all of the constraints that you define. The mathematics of this process are presented in the next section. Because the objective and constraint functions are generally nonlinear in the mathematical design variables, performing the optimization requires evaluations of these functions at a large number of design points. Re-evaluating the FE model completely at each candidate design point is not computationally feasible. Therefore, an ***approximate problem*** is formulated. This problem uses analysis and sensitivity information from the set of constrained analyses to build approximations to the objective and constraint functions that are valid in a neighborhood of the current design point. Typically, this is done by linearizing the responses or formulating intermediate responses using Taylor Series Expansions. In **UAI/*NASTRAN***, a ***mathematical programming*** technique, called the ***method of robust feasible directions***, is then used to solve the approximate problem. The best answer within the neighborhood is found and the design point is updated to these values. This step is often called ***scaling*** or ***resizing***. Complete finite element analyses are then performed to determine the actual responses and function values at the proposed design point. This procedure continues iteratively until convergence is achieved.

25.2.3 Objective Function

The ***objective function*** in optimization is that function that is to be placed at an extremum during the optimization process. In structural optimization, the extreme value is typically bounded implicitly by the constraints, but the extreme value may be located at some design point where there are no critical constraints. The objective function in **UAI/*NASTRAN*** may be a model characteristic or a structural response. Of the former type, the weight, mass, or volume of the structural model may be selected. The objective may also be any single structural response quantity provided that it is a member of the set of constrainable responses. Using an analytical response as the objective is useful in formulating frequency response design problems and in performing what is often called System Identification — correlating model results with test data.

In either case, the objective function may be ***minimized*** or ***maximized*** depending on your design goal. The extremum is reached by modifying the design variables you have defined to arrive at the design point at which the objective is at an extreme value and all the design constraints are satisfied. This implies that any other design point will either have a less satisfactory value of the objective or will be infeasible.

25.2.4 MDO and Analysis — Different Goals

The goals of automated structural design and FE analysis are subtly, but importantly different. In FE analysis, the engineer attempts to build a mathematical model of a physical system. Both the inputs to the system and the physical system itself are modeled. The responses of the model are then obtained. A design engineer then compares the FE results to known requirements. Unacceptable responses are used to review both the model and the physical system. Depending on the outcome of the review, the model or both the physical system and its model are changed to *improve* the model behavior. It is assumed at each step that the model correctly simulates the behavior of the system. Subsequently, further analyses are done to verify the changes. Automated structural design attempts to allow the computational tools some freedom to perform the latter parts of this task.

The finite element approach is used to model many different phenomena in structures, e.g., static responses, dynamic responses in the time and frequency domains, normal modes, responses to random excitation and many others. In some cases, specialized FE models are built

to capture just the important structural characteristics relevant to the analysis in question. This is done both for historical reasons and for efficiency. Each domain of engineering has its own unique way of building an FE-based model of the physical system.

Optimization, on the other hand, requires a model of the physical system which not only captures the physics of its behavior, but also models the acceptable ways in which the system can be changed to improve its performance. It is the latter that is termed the ***design model***. This seemingly simple additional model, however, requires a profound change in outlook. Modeling the "acceptable ways" to change a structural system requires a formalization of engineering rules-of-thumb that all design engineers use, but rarely must enumerate.

☞ ***Limiting the optimizer to acceptable designs while allowing it sufficient freedom to find a new and potentially counterintuitive design is the key to successful use of MDO as a design tool.***

Further, there is a new burden placed on the analytical model: the ability to model all the relevant responses. This new burden comes about because the optimizer can only improve or satisfy requirements for those characteristics of the system of which it is aware. Unlike a design engineer who "knows," for example, that the dynamic characteristics require that some parts of the structure be overdesigned for statics, an optimizer shown only static results will simply remove the "extra" material to minimize the weight. To ensure that this does not happen, structural optimization has been made "multidisciplinary." What have traditionally been specialty models must now be folded into a single mathematical representation. Limits are then placed on all the responses that are deemed by the design engineer to be important. Only then can the optimizer be unleashed to solve the problem.

This is a broad unification of effort that typically spans many traditionally distinct engineering organizations. This unification represents the most profound distinction between successful analysis models and successful MDO models. A design engineer must embody a great deal more information in the MDO model than has heretofore been necessary. Certainly, MDO tools can yield useful insights without radical changes to the traditional FE model, but the true power of MDO requires posing the correct question. The results of the MDO/FE model are NOT the responses of the system — those have been dictated as requirements. Rather, the results from MDO are the characteristics (design variable values) of a physical system that meets the requirements. Notice that it is "a" physical system, not "the" system. There may exist many sets of variables which satisfy the requirements. At this stage in the state-of-the-art, only the design engineer can decide whether the resulting design is one that truly is best or one that simply represents an incomplete specification of the requirements.

☞ ***MDO is a tool which provides design insight and better solutions. The ability of an FE-based design system to provide an optimal solution is predicated on the fidelity of the represention of the physical system by the FE model.***

25.3 MATHEMATICAL BACKGROUND

This section describes the manner in which Multidisciplinary Design Optimization is performed in **UAI/*NASTRAN***. It includes five major sections:

- The Optimization Problem Formulation
- The Approximate Problem Formulation
- Side Constraints
- The Optimization Process
- Sensitivity Computation

The intent of this section is to provide you with an overview of the structural optimization process. For those interested in complete details of the process, there are a number of comprehensive texts including [Vander84] and [Haftka90].

25.3.1 The Optimization Problem Formulation

The general structural optimization problem may be expressed as:

$$\begin{aligned} &\text{minimize} \quad F(\boldsymbol{X}) \\ &\text{such that} \quad g_j(\boldsymbol{X}) \le 0 \qquad j = 1,...,n_{con} \\ &\qquad\qquad x_i \in D_i \qquad\quad i = 1,...,n_{dv} \end{aligned} \tag{25-3}$$

where $\boldsymbol{X}$ is a vector of n_{dv} design variables, F is the objective function, and the g_i are n_{con} constraints. Additionally, each of the design variables is subject to side constraints defined by a specific domain, D. There are three types of side constraints which are valuable in MDO:

$$D_c = \left\{ x \mid x^{lower} \le x \le x^{upper} \right\} \tag{25-4a}$$

$$D_d = \left\{ x_1, x_2, ..., x_m \right\} \tag{25-4b}$$

$$D_i = \left\{ i_1, i_2, ..., i_m \right\} \tag{25-4c}$$

The first domain, D_c, is used to define ***continuous design variables*** which may take on any real value in a specified range. These variables allow the most design freedom, but the results may not be easily implemented. Thus, the second domain, D_d, may be used to defined ***discrete design variables***. Such variables may only have one of the enumerated values. The final domain, D_i, is a set of ***integer design variables*** which must assume a discrete integer value. Such variables are most often used to design laminated composites. In **UAI/*NASTRAN*** Version 11.7, only continuous constraints of the form (25-4a) are available.

The problem defined by (25-3) is called the ***nonlinear constrained optimization problem***. This well-studied problem [Vander84] has been solved in a number of ways. The most frequently used methods are called ***gradient search methods***. Generally, these algorithms attempt to reduce the objective function by using its gradient and those of *critical* constraints to determine a search direction that reduces the objective while retaining or achieving feasibility. The best possible value along this search direction is found in a process called the *one-dimensional search*. A new search direction is then determined and the algorithm continues until the objective cannot be improved. These techniques have proven quite successful in solving closed-form mathematical problems. However, finite element-based MDO presents a larger challenge. The search methods depend on the continuous evaluation of both the objective function and the constraints. Because each evaluation in the MDO environment represents the solution of several complete finite element analyses, the problem is not computationally effective.

25.3.2 The Approximate Problem Formulation

The major breakthrough that has allowed pragmatic MDO to be successful was first formulated by [Schmidt76] and has been expanded upon ever since. Rather than solving the exact problem given in (25-3), an approximation to it is formulated. The approximate problem is computationally smaller and makes possible the continual reanalyses with the new caveat that the validity of its analyses is limited depending on the nature of the approximations made in its formulation. The major aspects of the problem formulation fall into two categories: building approximations and limiting problem size. The former includes:

- Function Approximation
- Intervening Variables
- Intervening Responses
- Region of Approximation Validity

while the latter includes:

- Design Variable Linking
- Constraint Screening

each of which is described in the following sections.

25

25.3.2.1 Function Approximation

Suppose that the objective function and constraints are expanded as first-order Taylor series about the current design point $\boldsymbol{x}_o$ as:

$$\bar{F}(\boldsymbol{x}) = F(\boldsymbol{x}_0) + \sum_{k=1}^{n_{dv}} \frac{\partial F}{\partial x_k}\left(x_k - x_k^0\right) = F(\boldsymbol{x}_0) + \nabla \boldsymbol{F}^T \cdot (\boldsymbol{x} - \boldsymbol{x}_0) \tag{25-5}$$

$$\bar{g}_j(\boldsymbol{x}) = g_j(\boldsymbol{x}_0) + \sum_{k=1}^{n_{dv}} \frac{\partial g_j}{\partial x_k}\left(x_k - x_k^0\right) = g_j(\boldsymbol{x}_0) + \nabla \boldsymbol{g}^T \cdot (\boldsymbol{x} - \boldsymbol{x}_0) \tag{25-6}$$

These expansions can then be used to formulate a new optimization problem:

$$\begin{aligned} &\text{minimize} && \bar{F}(\boldsymbol{X}) && \\ &\text{such that} && \bar{g}_j(\boldsymbol{X}) \le 0 && j = 1,\ldots,n_{con} \\ & && x_i^{lower} \le x_{i,ml}^{lower} \le x_i \le x_{i,ml}^{upper} \le x_i^{upper} && i = 1,\ldots,n_{dv} \end{aligned} \tag{25-7}$$

In (25-7), $x_{i,ml}^{lower}$ and $x_{i,ml}^{upper}$ represent the actual move limits placed on the design variables during the solution of this problem. While over-simplified, this is the basically the optimization problem solved by **UAI/*NASTRAN***. It is called the ***Approximate Optimization Problem***. If the gradients of all of the exact functions are computed only once during the finite element analyses and only first-order Taylor Series Approximations are used then (25-7) is a pure Linear Programming problem. If other approximations are used, or more than simple first-order gradient information is used, then the Approximate Problem is Nonlinear. **UAI/*NASTRAN*** formulates an Approximate Problem that is nonlinear in the design variables.

25.3.2.2 Intervening Variables

Because of the inherent nonlinearity of structural response quantities, such as element stress and strain, to the set of physical design variables, the linear approximation is inaccurate even for neighborhoods close to $\boldsymbol{x}_o$. One way to improve the fidelity of the approximation is to introduce ***Intervening Variables*** in which the functions are more linear. While there are many possible choices for such variables, **UAI/*NASTRAN*** uses a reciprocal approximation which results in:

$$y_k = \frac{1}{x_k} \tag{25-8a}$$

$$\bar{g}_j(\boldsymbol{y}) = g_j(\boldsymbol{y}_0) + \sum_{k=1}^{m} \frac{\partial g_j}{\partial x_k}\left(\frac{1}{x_i} - \frac{1}{x_i^o}\right) \tag{25-8b}$$

$$\frac{\partial \bar{g}_j}{\partial y_k} = -\frac{1}{x_i^2}\frac{\partial \bar{g}_j}{\partial x_k} \tag{25-8c}$$

The use of the reciprocal variable is rooted in the early design of structural members, such as trusses, where responses were inversely related to, for example, cross-sectional area. **UAI/*NASTRAN*** selects a direct or inverse approximation automatically based on an assessment of which will result in the greatest accuracy.

25.3.2.3 Intervening Responses

It is also possible to define constraints in terms of ***Intervening Responses*** which may result in higher quality approximations. An example of this in **UAI/*NASTRAN*** is used in normal modes design. Rather than approximating the eigenvalue, λ , by a simple first-order Taylor Series, the Rayleigh Quotient Approximation [Canfield88] is used:

$$\lambda = \frac{\bar{k}}{\bar{m}} = \frac{\boldsymbol{\Phi}^T \boldsymbol{K} \boldsymbol{\Phi}}{\boldsymbol{\Phi}^T \boldsymbol{M} \boldsymbol{\Phi}} \tag{25-9}$$

the sensitivites $\frac{\partial \bar{k}}{\partial x}$ and $\frac{\partial \bar{m}}{\partial x}$ are used to build independent Taylor Series approximations to the numerator and denominator of the Rayleigh Quotient. While the generalized mass and stiffness are approximated linearly, the eigenvalue approximation is nonlinear in its behavior.

25.3.2.4 Region of Approximation Validity

Recall that (25-4a) defines a set of continuous side constraints on the design variables for your optimization problem. These values are often quite far apart. The function approximations described in the previous sections result in functions that behave like the actual functions only in a neighborhood of $\boldsymbol{x}_0$. The size of the neighborhood depends on the sophistication of the approximation and is not easily computed. One of the heuristic aspects of the MDO problem is the control of the size of the neighborhood that the optimizer will be allowed to use in solving the Approximate Optimization Problem.

Typically, when formulating the approximate problem, ***Move Limits*** are established and imposed as side constraints to ensure that the design stays within a *reasonable* neighborhood of the current design point. The lower and upper bound side constraints for the approximate problem are:

$$x^{lower} \leq x_{ml}^{lower} \leq x \leq x_{ml}^{upper} \leq x^{upper} \tag{25-10}$$

UAI/*NASTRAN* computes the move limits automatically using an adaptive algorithm which allows the maximum movement for each design variable during each design cycle while maintaining good quality in the approximations.

25.3.2.5 Design Variable Linking

While the preceding discussion has focussed on the development of approximations to avoid the computational expense of complete FE analyses during the optimization process, the most important way in which the design problem is made tractable is by limiting the number of independent design variables that are used and the number of constraint functions that must be approximated.

Design variable linking is the process by which several, or many, physical design variables are controlled by a single mathematical variable. The mathematical variable is then the only one which participates in the optimization step (the independent variable) while the physical variables are determined *a posteriori* from the mathematical variables.

Linking is simply a transformation which relates the physical design variables, $\boldsymbol{v}_p$, to the mathematical design variables, $\boldsymbol{v}_m$ which you have defined. **UAI/*NASTRAN*** allows you to specify a relationship of the form:

$$\boldsymbol{v}_p = \boldsymbol{v}_p^{inv} + \boldsymbol{T}\,\boldsymbol{v}_m \tag{25-11}$$

Figure 25-1. LINKING DESIGN VARIABLES

	QUAD4 1 Thickness t_1	QUAD4 2 Thickness t_2	QUAD4 3 Thickness t_3
Independent Physical Design Variables	$v_1 = t_1$	$v_2 = t_2$	$v_3 = t_3$
Single Linked Design Variable Controlling Three Physical Design Variables	$t_1 = v_1$	$t_2 = 0.667v_1$	$t_3 = 0.333v_1$

The vector $\boldsymbol{v}_p^{inv}$ represents the ***invariant portion*** of the physical design variable and the matrix $\boldsymbol{T}$ represents a ***dependency linking relationship*** between the physical design variables and the mathematical ones. To illustrate these concepts, consider Figure 25-1. This is a small finite element model which contains three **QUAD4** elements. Suppose that you wish to define the thicknesses of each plate, t_1, t_2, and t_3 as independent physical design variables which may vary in any manner. Then (25-11) could be written:

$$\begin{Bmatrix} t_1 \\ t_2 \\ t_3 \end{Bmatrix} = \boldsymbol{v}_p^{inv} + \boldsymbol{Tv} = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \end{bmatrix} + \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \begin{Bmatrix} v_1 \\ v_2 \\ v_3 \end{Bmatrix} \tag{25-12}$$

In other words, each of the physical design variables corresponds to a separate mathematical design variable and there is no invariant portion of the thicknesses.

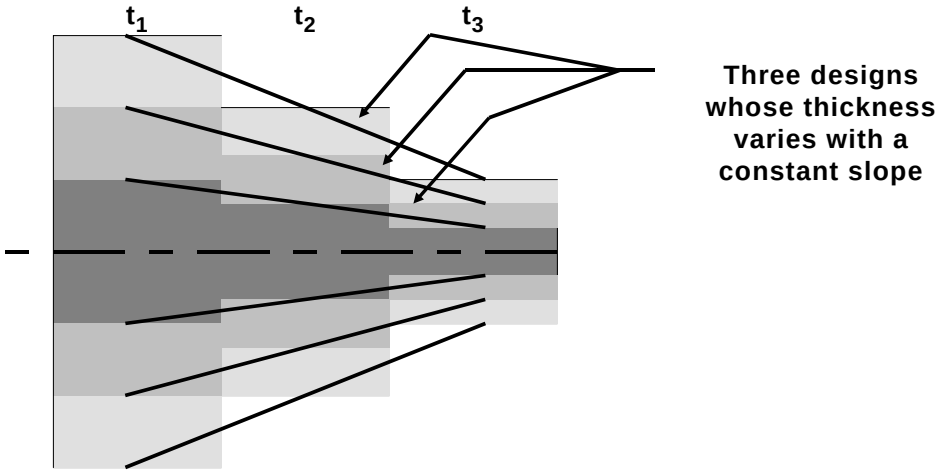
Now, suppose that you wish to require that the plate thickness vary linearly such that:

$$\begin{aligned} t_1 &= t_1 \\ t_2 &= 0.667 t_1 \\ t_3 &= 0.333 t_1 \end{aligned}$$

Figure 25-2 illustrates the thicknesses for each of the three plates in the example and a family of possible thicknesses which satisfy these dependency relations. Clearly, in this case, only one mathematical design variable is needed, $v_1 = t_1$ in which case (25-11) can be written as:

$$\begin{Bmatrix} t_1 \\ t_2 \\ t_3 \end{Bmatrix} = \boldsymbol{v}_p^{inv} + \boldsymbol{Tv} = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \end{bmatrix} + \begin{bmatrix} 1.000 \\ 0.667 \\ 0.333 \end{bmatrix} \{ v_1 \}$$

Figure 25-2. FAMILY OF LINKED SHAPES



Next, consider a linking relationship that specifies a fixed value for one or more of the physical variables. Figure 25-3 illustrates such a case. The thickness of the first element, t_1, is fixed at 8.0 in., but you wish to link the three thicknesses such that they obey a linear tapering rule. In this case, the mathematical design variable can be defined to be the slope of the thickness defined by the equation:

$$t_i = 8.0 + m\,(x_i - 5.0)$$

where x_i represents the x coordinate at the centroid of the i^{th} element. Figure 25-3 shows three possible designs which obey your linear taper rule. For this case, (25-11) can be written as:

$$\begin{Bmatrix} t_1 \\ t_2 \\ t_3 \end{Bmatrix} = \boldsymbol{v}_p^{inv} + \boldsymbol{T}\boldsymbol{v} = \begin{bmatrix} 8.0 \\ 8.0 \\ 8.0 \end{bmatrix} + \begin{bmatrix} 0.0 \\ 10.0 \\ 20.0 \end{bmatrix} \{ m \}$$

25

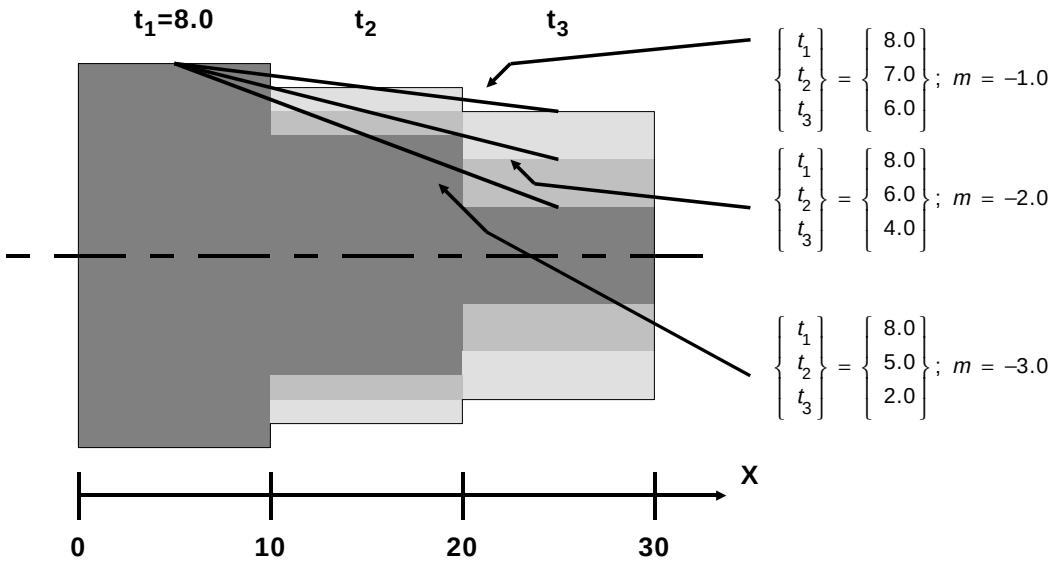
Finally, consider a linking relationship that seeks to allow *any* linear taper for the physical variables. In this case, the mathematical design variables can be defined to be the slope of the thickness, *m*, and the intercept, *b*, as in the equation:

$$t_i = b + m\,(x_i - 5.0)$$

where x_i again represents the x coordinate at the centroid of the i^{th} element. For this case, (25-11) can be written as:

$$\begin{Bmatrix} t_1 \\ t_2 \\ t_3 \end{Bmatrix} = \boldsymbol{T}\boldsymbol{v} = \begin{bmatrix} 1.0 & 0.0 \\ 1.0 & 10.0 \\ 1.0 & 20.0 \end{bmatrix} \begin{Bmatrix} b \\ m \end{Bmatrix} \qquad (25\text{-}13)$$

Figure 25-3. LINKING WITH A FIXED VALUE



Notice that, in this case, none of the independent variables have an *a priori* physical significance. However, by using this sophisticated technique, known as ***reduced basis linking***, all possible linear tapers exist in the Design Space with the elimination of a third of the independent variables. Notice also that this technique imposes a commonly desired manufacturing limit solely by virtue of the linking relationship!

25.3.2.6 Constraint Screening

The large number of design constraints that may be specified in a typical problem makes it extremely expensive to compute all possible sensitivities. It has been demonstrated that good results may be achieved when only those constraints which are ***violated***, ***active***, or ***near active***, are considered during an optimization cycle. Choosing the subset of constraints to be considered as part of the current Approximate Problem is called ***Constraint Screening*** and the constraints which are used are called ***Retained Constraints***. Constraint screening is the process by which the entire set of design constraints, *G*, is partitioning into three disjoint subsets:

$$G = G_{violated} \cup G_{active} \cup G_{inactive} \tag{25-14}$$

These subsets are defined as:

$$\begin{aligned} G_{violated} &= \left\{ g \in G \mid g \geq g_{maxfeasible} \right\} \\ G_{active} &= \left\{ g \in G \mid g_{maxfeasible} > g \geq g_{minactive} \right\} \\ G_{inactive} &= \left\{ g \in G \mid g < g_{minactive} \right\} \end{aligned} \tag{25-15}$$

The first subset, $G_{violated}$, represents constraints which are violated. The default value for constraint violation (remembering that the constraints are normalized), $g_{maxfeasible}$, is 0.001. The second subset, G_{active}, represents constraints which are active, or near their satisfaction value. The default value, $g_{minactive}$, is -0.001. The inactive constraint set, $G_{inactive}$ is simply all other constraints.

The purpose of screening is to reduce the number of constraints used in the Approximate Problem. This technique has proven to be very successful for reducing the computational requirements of this step. At the same time, it has been shown that the inactive constraints play only a minor role in the determination of the optimal design point and they can severely degrade the numerical stability of the search process by introducing large numbers of nearly linearly dependent constraints. The retained constraint set can be thought of as:

$$G_{retained} = G_{violated} \cup G_{active} \tag{25-16}$$

however, there are certain pathologies which may occur. The first is the case in which many constraints are violated — the design is very infeasible. In this case, the retained set is limited to a maximum number of the violated and active contraints. As you will see later in the Chapter, you may specify this number using the **MAXRETAIN** subcommand of the **OBJECTIVE** Case Control command. The second case is when the design is well within the feasible region and there are no violated or active constraints, again the retained set is limited to a maximum number of the constraints most likely to become active.

25.3.3 Side Constraints

The concepts of design variable linking and move limits introduce a number of complexities to the handling of side constraints. Generally, side constraints are limits to the set of values that a design variable can take. There are, as you have now seen, two different classes of design variables: physical variables and mathematical variables. In general, you must only deal with the bounds on the physical design variables. The appropriate bounds to the mathematical variables are then automatically determined from the set of associated physical bounds. As illustrated in the following, this computation cannot be done in the case of reduced basis linking. Instead, other techniques are used which allow the side constraints to be handled automatically in almost all cases.

Revisiting the first linking example of (25-12), suppose that you define the thicknesses of each plate, t_1, t_2, and t_3 as independent physical design variables which may vary in any manner:

$$\begin{Bmatrix} t_1 \\ t_2 \\ t_3 \end{Bmatrix} = \boldsymbol{v}_p^{inv} + \boldsymbol{T}\boldsymbol{v} = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \end{bmatrix} + \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \begin{Bmatrix} v_1 \\ v_2 \\ v_3 \end{Bmatrix}$$

Then the upper bound side constraints on the mathematical variables v_1^{max}, v_2^{max}, and v_3^{max} defined by this expression can be determined from the expression:

$$\begin{Bmatrix} t_1^{max} \\ t_2^{max} \\ t_3^{max} \end{Bmatrix} = \boldsymbol{v}_p^{inv} + \boldsymbol{T}\boldsymbol{v}_{\mathbf{max}} = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \end{bmatrix} + \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \begin{Bmatrix} v_1^{max} \\ v_2^{max} \\ v_3^{max} \end{Bmatrix}$$

and the lower bound side constraints, v_i^{min}, can be similarly determined. Since $\boldsymbol{T}$ is invertible, explicit values of v_i^{min} or v_i^{max}, collectively denoted as v_i^{side}, can be determined from the physical variable bounds. This is the most straightforward case.

Looking instead at the third linking example:

$$\begin{Bmatrix} t_1 \\ t_2 \\ t_3 \end{Bmatrix} = \boldsymbol{v}_p^{inv} + \boldsymbol{Tv} = \begin{bmatrix} 8.0 \\ 8.0 \\ 8.0 \end{bmatrix} + \begin{bmatrix} 0.0 \\ 10.0 \\ 20.0 \end{bmatrix} \{ m \}$$

it can be seen that the bounds on m can again be determined uniquely:

$$m_1^{side} = \textbf{\textit{undefined}}$$

$$m_2^{side} = \frac{(t_2^{side} - 8.0)}{10.0}$$

$$m^{side}{}_3 = \frac{(t_3^{side} - 8.0)}{20.0}$$

The range of m is then determined from the most restrictive upper and lower bounds computed from the above expressions. Notice that the first linking expression yields no value for the bounds of m. This is because t_1 is not really a design variable: it is constrained to be 8.0 for all values of m.

Finally, consider once again the example of reduced basis linking, (25-13):

$$\begin{Bmatrix} t_1 \\ t_2 \\ t_3 \end{Bmatrix} = \boldsymbol{Tv} = \begin{bmatrix} 1.0 & 0.0 \\ 1.0 & 10.0 \\ 1.0 & 20.0 \end{bmatrix} \begin{Bmatrix} b \\ m \end{Bmatrix}$$

Now $\boldsymbol{T}$ is not invertible and no unique selection of the mathematical bounds can be determined from the physical bounds. The mathematical variables, however, must be controlled to ensure that the physical bounds are met. To ensure that these vital constraints are handled, **UAI/*NASTRAN*** automatically generates a *true constraint* to act as the side constraint. These constraints are termed ***pseudo-side constraints***:

$$g_{ps}^{min} = \frac{t}{t_{min}} - 1.0$$

$$g_{ps}^{max} = 1.0 - \frac{t}{t_{max}}$$

The mathematical variables themselves now ***require*** no side constraints at all. The correct limiting behavior is fully described by the set of true constraints and the mathematical variables will be forced to take on "correct" values simply to satisfy the constraints. There are, however, certain instances when you ***might*** want these reduced basis variables to be explicitly

bounded. The most common example is in limiting the slope of a thickness taper due to a manufacturing limit.

Again, in the reduced basis linking example, suppose that the milling machine is limited to a rate of change in thickness that is no more than 0.25 inch every 10 linear inches. This implies that $|m|$ must never by greater than 0.25, since m explicitly represents the rate of change in thickness in each 10 inch segment. By default, **UAI/*NASTRAN*** has unbounded reduced basis variables, but allows you to place explicit side constraints on them in order to handle this type of manufacturing limit.

The second complication with side constraints is the definition of move limits. Since these are temporary modifications to the side constraints, however, no new technology is introduced. Instead, you must only be aware that the side constraints and pseudo-side constraints may be more restrictive on any given design cycle than your input value. **UAI/*NASTRAN*** will automatically handle the side constraints appropriately during the design cycles with no user intervention.

25.3.4 The Optimization Process

As introduced in Section 25.3.1, the optimization process solves the problem:

$$\begin{aligned} &\text{minimize} \quad \bar{F}(\boldsymbol{X}) \\ &\text{such that} \quad \bar{g}_j(\boldsymbol{X}) \le 0 \qquad j = 1,...,n_{con} \\ &\qquad\qquad x_i \in D_i \qquad\quad i = 1,...,n_{dv} \end{aligned} \tag{25-7}$$

UAI/*NASTRAN* employs the method of robust feasible directions for this purpose. More than ten years of large-scale MDO has been performed in this manner [Herendeen86], [Neill90], [Kamat93]. It is important to bear in mind the hierarchic nature of the optimization solution that takes place in **UAI/*NASTRAN***. The numerical optimization problem (25-7) is the ***Approximate Problem***, **NOT** the full problem to be solved. That means that the iterations that are part of the optimizer are done using approximations formulated to allow rapid, approximate, reanalyses. There are thus **TWO** levels of iteration that take place: the ***design cycle*** which iterates to create approximate problems from exact FE analyses and the ***optimization iteration*** which iterates to find the constrained optimum expressed in (25-7).

25

25.3.4.1 The Method of Robust Feasible Directions

The Robust Feasible Directions method is one of several gradient methods for handling nonlinear constrained optimization problems. It can be considered a hybrid method that attempts to take advantage of the most powerful aspects of the reduced gradient method while retaining the efficiency of the feasible directions method. The literature provides excellent discussions of these methods [Vander84] and only a brief synopsis is presented here for completeness.

The first step in the solution of the constrained optimization problem is to pick from the reduced set of constraints in the Approximate Problem, a further subset that can be considered *active*. The current gradients of the objective function and the active constraints are then computed using the approximate problem formulation to provide rapid analysis. The method then computes a ***search direction*** — a vector in the design space — that is both ***usable*** (i.e. the objective function is reduced along the direction) and ***feasible*** (i.e. the active constraints are improved or remain satisifed along the direction).

A ***one-dimensional search*** is then performed along the search direction to find the distance to move ***in that one direction*** from the current design point to a new one. The distance is computed such that it makes the best possible improvement in the objective function. Special processing takes place to handle the movement from an infeasible solution to a feasible one. Once a feasible solution is found, the method essentially attempts to follow the active constraint surfaces to the optimum.

This process continues, using the approximate problem to provide the requisite reanalysis, until convergence is achieved. Each optimization iteration involves numerous re-evaluations of the approximate objective and approximate constraint values, re-evaluations of new active subsets and computation of new search directions. Only when no further improvement can be made using the approximate problem are the iterations terminated. The final design point is then saved to be used in the next design cycle to build the next approximate problem.

25.3.4.2 Problem Convergence

Since the optimization process itself is hierarchical, the convergence criteria are also. First, a pair of criteria are imposed on the approximate problem to determine its convergence. Convergence of the approximate problem means that the Approximate Problem (25-7) cannot be moved from the initial design point $\boldsymbol{X_0}$ used to form the approximations. (It does NOT imply that a feasible, optimal solution was found.)

Then, once convergence has been achieved for the approximate problem, final convergence criteria are imposed on the exact reanalysis. At this point, the solution must be feasible and optimal to be considered converged. These levels of criteria, which are necessary because of the inexactness of the Approximate Problem, are described in the following sections.

Approximate Problem Convergence. Two criteria are imposed to determine that convergence has been achieved in the solution of the current approximation to the full optimization problem. Both criteria attempt to determine if a design point was found that is different from the one about which the Approximate Problem was formed. The first is a limit on the percent change in the objective function, and the second is a limit on the length of the $\delta \boldsymbol{X}$ vector as a percentage of the length of $\boldsymbol{X}$.

- ***Relative Change in the Objective.*** The first check determines if the final prediction of the *approximate* objective function value is unchanged relative to the initial (*exact*) function value. Mathematically, this requires:

$$\left| \bar{F}(\boldsymbol{X}^n) - F(\boldsymbol{X}^{n-1}) \right| \leq \left| \varepsilon_O \cdot F(\boldsymbol{X}^{n-1}) \right| \tag{25-17}$$

- ***Relative Design Variable Move.*** The second check determines the change in the final design point relative to the point about which the approximate problem was built. This is computed from the Euclidean Norm of the *change in the design point* and is normalized by the length of the design vector $\boldsymbol{X_0}$ to ensure that poorly scaled mathematical design variables still satisfy the criterion. Mathematically, the ratio of the two Euclidean Norms must satisfy:

$$\frac{||\, \delta \boldsymbol{X} \,||}{||\, \boldsymbol{X} \,||} \leq \varepsilon_X \tag{25-18}$$

  Both of these conditions must be met before the Approximate Problem is considered to have converged. The values ε_O and ε_X are computed from the value given by the **CONVERGE** subcommand of the **OBJECTIVE** Case Control command as:

$$\varepsilon_O = \frac{\mathbf{CONVERGE}}{100.0} \tag{25-19}$$

$$\varepsilon_X = \frac{\mathbf{3 \cdot CONVERGE}}{100.0} \tag{25-20}$$

Final Problem Convergence. Convergence of the full optimization process depends on the nature of the optimization problem. For the typical constrained optimization problem, the criteria are:

- The Approximate Problem must converge
- The final exact analyses must result in no violated constraints
- There must be at least one constraint in G_{active}

For special cases in which no objective function exists or no constraints are defined, the appropriate adjustments to these criteria are made.

Maximum Iteration Failsafe. The final termination check is on the maximum number of design iterations, **MAXITER**. This parameter is used as a failsafe mechanism to avoid excessive computer costs in the event that convergence cannot be attained, for example when there is no feasible solution, or in the case that the design problem has been poorly posed. The default value for **MAXITER** is 10.

25.3.4.3 Controlling the Optimization

The literature, including [Vander84], [Hafka90], and [Kamat93], presents literally hundreds of optimization algorithms with thousands of controlling parameters that have been *tuned* to maximize the performance of a given algorithm for a given problem. All these developments are of limited value in addressing the generality of the FEM-based MDO design problem that is the target problem for **UAI/*NASTRAN***. Instead, **UAI/*NASTRAN***'s MDO philosophy is that ***domain expertise should be imbedded internally and controlled easily***, without *requiring* domain expertise on the part of the user. Only in limited circumstances should the algorithmic control variables be useful, and they should never be needed.

Algorithmic control of the optimization problem is limited to the control of the highest level: the optimization problem, the design cycles and approximate problem formulation. The resizing step itself, with its solution of the nonlinear constrained optimization problem, is totally self contained and needs no intervention. The optimization problem is defined through the specification of the design variables, the objective function, the analysis cases to be performed and their attendant constraints. The next section of this Chapter deals extensively with these inputs. You may control the design cycles in three areas:

- The constraint deletion algorithm
- The convergence tolerance
- The Maximum Iteration Failsafe

These areas are now discussed relating the input data to the **UAI/*NASTRAN*** optimization process just presented.

The constraint deletion algorithm is controlled by three interrelated command options. The first two are the **MAXFEASIBLE** and **MINACTIVE** subcommands of the **OBJECTIVE** Case Con-

trol command. They relate to the partitioning of the constraint set, *G*, into its three component subsets:

$$G = G_{violated} \cup G_{active} \cup G_{inactive} \tag{25-21}$$

These subsets are defined as:

$$\begin{aligned} G_{violated} &= \{\, g \in G \mid g \geq \mathbf{MAXFEASIBLE} \,\} \\ G_{active} &= \{\, g \in G \mid \mathbf{MAXFEASIBLE} > g \geq \mathbf{MINACTIVE} \,\} \\ G_{inactive} &= \{\, g \in G \mid g < \mathbf{MINACTIVE} \,\} \end{aligned} \tag{25-22}$$

The final constraint screening control is the **MAXRETAIN** Case Control command. This command can provide a value for the constraint screening override in the event that $G_{violated} \cup G_{active}$ is either too small or too large. In either event, the largest **MAXRETAIN** constraints are kept for use in solving the Approximate Problem.

The convergence tolerance is controlled by selecting a value for the **CONVERGE** subcommand of the **OBJECTIVE** command. This value represents the percentage change that will be used in the measures to determine approximate problem convergence.

The maximum iteration failsafe is controlled by selecting a value for the **MAXITER** subcommand of the **OBJECTIVE** command. This value represents the maximum number of *design cycles* that will be performed. If *final* convergence is achieved more quickly, fewer cycles will be performed. Remember that the initial start-up analysis that is needed to form the first approximate problem is ***not*** counted: the maximum number of ***FE analyses*** is **MAXITER + 1**.

25.3.5 Sensitivity Computation

This section describes the manner in which sensitivities of structural response quantities and constraints are computed in **UAI/*NASTRAN***. The system matrix sensitivities are handled differently than are those of the solution results. Further, nodal responses and element responses are treated slightly differently as are the specific analysis disciplines. For these discussions, $V = \{ v_1, v_2, \ldots, v_n \}$ is a set of n design variables and $G = \{ g_1, g_2, \ldots, g_m \}$ is a set of m constraints. For sensitivity computations, the objective function can be considered to be exactly like a constraint.

25.3.5.1 System Matrix Sensitivities

UAI/*NASTRAN* uses a technique called ***Semi-Analytic*** (SA) gradient calculation to evaluate the sensitivities of the constraints and the objective function. This term means that the constraints are differentiated analytically to yield an expression for the sensitivity in terms of the analysis quantities, design variables and other sensitivities. Then, ***Finite Difference*** (FD) methods are used to solve for just those constituent sensitivities that must be known to evaluate the analytic expression for the derivative of merit.

In **UAI/*NASTRAN***, the FD method is used to compute the following terms:

$$\frac{\partial \boldsymbol{K}^e}{\partial \boldsymbol{v}}, \frac{\partial \boldsymbol{M}^e}{\partial \boldsymbol{v}}, \frac{\partial \boldsymbol{B}^e}{\partial \boldsymbol{v}}, \frac{\partial \boldsymbol{P}}{\partial \boldsymbol{v}}, \frac{\partial \boldsymbol{G}_{mn}}{\partial \boldsymbol{v}}$$

These constitute the basis for all subsequent analytic gradient computations.

☞ Note that the last term, $\frac{\partial \boldsymbol{G}_{mn}}{\partial v}$, relating to MPC reductions, is unlike the others. It arises only in the special case of GRID point coordinate design variables that are connected to rigid elements. These terms will be omitted for clarity from the following discussions, but they introduce additional terms into all the sensitivities that are computed by **UAI/*NASTRAN***.

As part of the normal **UAI/*NASTRAN*** analysis, the system matrices of the model for design vector $\boldsymbol{v}$, called $\boldsymbol{K}(\boldsymbol{v})$, are computed. Then, during sensitivity analysis, each v_j is perturbed by a small fraction, $\delta\beta$, which results in an incremental perturbation of $\delta v_j = \delta\beta \cdot v_j$. The elemental matrices associated with δv_j are recomputed and a finite difference computation is performed

$$\frac{\partial \boldsymbol{K}(\boldsymbol{v})}{\partial \boldsymbol{v}} = \frac{\boldsymbol{K}(\boldsymbol{v} + \delta\boldsymbol{v}) - \boldsymbol{K}(\boldsymbol{v})}{\delta\beta \cdot \boldsymbol{v}} \qquad (25\text{-}23)$$

In the unique case where $v_j = 0.0$, **UAI/*NASTRAN*** sets the increment $\delta v_j = \delta\beta$. In this case, (25-20) becomes:

$$\frac{\partial \boldsymbol{K}(\boldsymbol{v})}{\partial \boldsymbol{v}} = \frac{\boldsymbol{K}(\boldsymbol{v} + \delta\boldsymbol{v}) - \boldsymbol{K}(\boldsymbol{v})}{\delta\beta} \qquad (25\text{-}24)$$

Finally, the incremental perturbation, δv_j, is taken as 0.1% of the current value of the design variable v_j, that is, $\delta\beta$ is 0.001 by default. You may change this value by using the Bulk Data entry:

PARAM	**'DELTAB'**	$\delta\beta$							

25

Subsequent sensitivity analysis is performed in a manner that is specialized for each discipline as outlined in the following sections.

25.3.5.2 Static Analysis

As you know from previous chapters, the equations of motion for linear static analysis are given by:

$$\boldsymbol{Ku} = \boldsymbol{P} \qquad (25\text{-}25)$$

These equations of motion may be differentiated and solved to determine the displacement sensitivity which, in turn, may be used to determine the sensitivities of other response quantities, as shown in the following sections.

Nodal Displacement Sensitivity. Equation, (25-25) may be explicitly differentiated with respect to the i^{th} design variable yielding:

$$\frac{\partial \boldsymbol{K}}{\partial v_i}\boldsymbol{u} + \boldsymbol{K}\frac{\partial \boldsymbol{u}}{\partial v_i} = \frac{\partial \boldsymbol{P}}{\partial v_i} \tag{25-26}$$

from which the nodal displacement sensitivities are:

$$\frac{\partial \boldsymbol{u}}{\partial v_i} = \boldsymbol{K}^{-1}\left[\frac{\partial \boldsymbol{P}}{\partial v_i} - \frac{\partial \boldsymbol{K}}{\partial v_i}\boldsymbol{u}\right] \tag{25-27}$$

Element Strain Sensitivity. The general strain-displacement for a finite element is represented as:

$$\varepsilon = \boldsymbol{B}\boldsymbol{u} \tag{25-28}$$

The strain sensitivities can therefore be determined by differentiation using:

$$\frac{\partial \varepsilon}{\partial v_i} = \boldsymbol{B}\frac{\partial \boldsymbol{u}}{\partial v_i} + \frac{\partial \boldsymbol{B}}{\partial v_i}\boldsymbol{u} \tag{25-29}$$

Element Stress Sensitivity. Similarly, the element stress sensitivities are determined by differentiating the element stress-strain relationship:

$$\sigma = \boldsymbol{G}\varepsilon \tag{25-30}$$

to yield:

$$\frac{\partial \sigma}{\partial v_i} = \boldsymbol{G}\frac{\partial \boldsymbol{\varepsilon}}{\partial v_i} + \frac{\partial \boldsymbol{G}}{\partial v_i}\boldsymbol{\varepsilon} \tag{25-31}$$

Element Force Sensitivity. Element force sensitivities are determined by differentiating the element force relationship:

$$\boldsymbol{F}^e = \boldsymbol{A}\,\sigma \tag{25-32}$$

to yield:

$$\frac{\partial \boldsymbol{F}^e}{\partial v_i} = \boldsymbol{A}\frac{\partial \boldsymbol{\sigma}}{\partial v_i} + \frac{\partial \boldsymbol{A}}{\partial v_i}\boldsymbol{\sigma} \tag{25-33}$$

The matrices ***B***, ***G***, and ***A*** depend on the specific finite element and material being used. Their derivatives are also computed using the FD method.

25.3.5.3 Normal Modes Analysis

When performing a normal modes extraction, the equations of motion are transformed into the standard eigenproblem form:

$$\left[\boldsymbol{K} - \lambda_j \boldsymbol{M} \right] \varphi_j = \boldsymbol{0} \tag{25-34}$$

where the eigenvector φ_j corresponds to the eigenvalue λ_j. Once again, the equations may be explicitly differentiated with respect to the i^{th} design variable yielding:

$$\left[\frac{\partial \boldsymbol{K}}{\partial v_i} - \frac{\partial \lambda_j}{\partial v_i} \boldsymbol{M} - \lambda_j \frac{\partial \boldsymbol{M}}{\partial v_i} \right] \varphi_j + \left[\boldsymbol{K} - \lambda_j \boldsymbol{M} \right] \frac{\partial \varphi_j}{\partial v_i} = \boldsymbol{0} \tag{25-35}$$

Eigenvalue Sensitivity. The response sensitivity of the eigenvalue to a given design variable, derived from (25-35) is:

$$\frac{\partial \lambda_j}{\partial v_i} = \frac{\varphi_j^T \left[\dfrac{\partial \boldsymbol{K}}{\partial v_i} - \lambda_j \dfrac{\partial \boldsymbol{M}}{\partial v_i} \right] \varphi_j}{\varphi_j^T \boldsymbol{M} \varphi_j} \tag{25-36}$$

Frequency Sensitivity. The cyclic frequency, f_j is related to its eigenvalue by:

$$f_j = \frac{1}{2\pi} \sqrt{\lambda_j} \tag{25-37}$$

The frequency constraint may then be simply recast as an equivalent constraint on λ. This is done in **UAI/*NASTRAN*** to improve the constraint approximations.

Eigenvector Sensitivity. Note that (25-35) also contains the eigenvector response sensitivity. This equation can be written as:

$$\frac{\partial \varphi_j}{\partial v_i} = -\left[\boldsymbol{K} - \lambda_j \boldsymbol{M} \right]^{-1} \left[\frac{\partial \boldsymbol{K}}{\partial v_i} - \frac{\partial \lambda_j}{\partial v_i} \boldsymbol{M} - \lambda_j \frac{\partial \boldsymbol{M}}{\partial v_i} \right] \varphi_j \tag{25-38}$$

It is necessary to reduce the rank of the term $[\boldsymbol{K} - \lambda_j \boldsymbol{M}]$ by one in order to solve this equation. This is performed automatically using Nelson's Method [Haftka90] with attendant special code to handle the sensitivity of the eigenvector normalizations.

25.3.5.4 Frequency Response Analysis

Direct Frequency Response. The equations of motion for direct frequency response are:

$$(-\omega^2 \boldsymbol{M}_{dd} + i\,\omega \boldsymbol{B}_{dd} + \boldsymbol{K}_{dd})\, u_d\, e^{i\omega t} = \boldsymbol{P}_d\, e^{i\omega t} \tag{25-39}$$

where:

$$\boldsymbol{M}_{dd} = \boldsymbol{M}^{1}_{dd} + \boldsymbol{M}^{2}_{dd} \tag{25-40a}$$

$$\boldsymbol{B}_{dd} = \boldsymbol{B}^{1}_{dd} + \boldsymbol{B}^{2}_{dd} \tag{25-40b}$$

$$\boldsymbol{K}_{dd} = (1 + i\,G)\,\boldsymbol{K}^{1}_{dd} + \boldsymbol{K}^{2}_{dd} + i\,\boldsymbol{K}^{4}_{dd} \tag{25-40c}$$

$$\boldsymbol{K}^{4} = \mathbf{A}\,(g_e)_i\,\boldsymbol{k}^{e}_{ij} \tag{25-40d}$$

Note that $\boldsymbol{K}^4$ is assembled from its elemental components.[1] Differentiation of (25-39) assuming that $\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{v}} = \mathbf{0}$ yields:

$$\frac{\partial \boldsymbol{u}_d}{\partial \boldsymbol{v}} = (-\omega^2\,\boldsymbol{M}_{dd} + i\,\omega\,\boldsymbol{B}_{dd} + \boldsymbol{K}_{dd})^{-1}\left[-\omega^2\,\frac{\partial \boldsymbol{M}_{dd}}{\partial v} + i\,\omega\,\frac{\partial \boldsymbol{B}_{dd}}{\partial v} + \frac{\partial \boldsymbol{K}_{dd}}{\partial v}\right] u_d \tag{25-41}$$

where the final sensitivity term $\frac{\partial \boldsymbol{K}_{dd}}{\partial v}$ is assembled using:

$$\frac{\partial \boldsymbol{K}_{dd}}{\partial \boldsymbol{v}} = (1 + i\,g)\,\mathbf{A}\,\frac{\partial \boldsymbol{K}^{e}}{\partial \boldsymbol{v}} + i\left[\mathbf{A}\left[\frac{\partial g_e}{\partial \boldsymbol{v}}\,\boldsymbol{K}^{e}\right] + \mathbf{A}\left[g_e\,\frac{\partial \boldsymbol{K}^{e}}{\partial \boldsymbol{v}}\right]\right] \tag{25-42}$$

and the other terms are computed directly from the elemental FD system matrix sensitivities and the solution vector.

Modal Frequency Response. The frequency response equations may also be solved using generalized, or modal, coordinates. In this case, the equations of motion are:

$$(-\omega^2\,\boldsymbol{M}_{hh} + i\,\omega\,\boldsymbol{B}_{hh} + \boldsymbol{K}_{hh})\,u_h\,e^{i\,\omega\,t} = \boldsymbol{P}_h\,e^{i\,\omega\,t} \tag{25-43}$$

where

$$\boldsymbol{M}_{hh} = \boldsymbol{m} + \boldsymbol{\Phi}^{T}_{dh}\boldsymbol{M}^{2}_{dd}\boldsymbol{\Phi}_{dh} \tag{25-44a}$$

$$\boldsymbol{B}_{hh} = \boldsymbol{b} + \boldsymbol{\Phi}^{T}_{dh}\boldsymbol{B}^{1}_{dd}\boldsymbol{\Phi}_{dh} + \boldsymbol{\Phi}^{T}_{dh}\boldsymbol{B}^{2}_{dd}\boldsymbol{\Phi}_{dh} \tag{25-44b}$$

$$\boldsymbol{K}_{hh} = (1 + i\,G)\,\boldsymbol{k} + \boldsymbol{\Phi}^{T}_{dh}\boldsymbol{K}^{2}_{dd}\boldsymbol{\Phi}_{dh} + \boldsymbol{\Phi}^{T}_{dh}\boldsymbol{K}^{4}_{dd}\boldsymbol{\Phi}_{dh} \tag{25-44c}$$

1 $\mathbf{A}$ is an operator denoting matrix assembly.

$$\boldsymbol{P}_h = \boldsymbol{\Phi}_{dh}^T \boldsymbol{P}_d \tag{25-44d}$$

$\boldsymbol{m}$ and $\boldsymbol{k}$ represent the diagonal modal mass and stiffness matrices, and

$$b_i = (2\pi f_i)\, g(f_i)\, m_i \tag{25-44e}$$

is the modal damping.

Differentiating (25-43), and assuming that $\frac{\partial \boldsymbol{\Phi}_{dh}}{\partial v} \equiv 0$, results in:

$$\frac{\partial \boldsymbol{u}_d}{\partial v} = \boldsymbol{\Phi}_{dh} \frac{\partial \boldsymbol{u}_h}{\partial v} = \boldsymbol{\Phi}_{dh}\left[-\omega^2 \boldsymbol{M}_{hh} + i\,\omega \boldsymbol{B}_{hh} + \boldsymbol{K}_{hh} \right]^{-1} \left[-\omega^2 \boldsymbol{\Phi}_{dh}^T \frac{\partial \boldsymbol{M}_{dd}}{\partial v} \boldsymbol{\Phi}_{dh} + i\,\omega \boldsymbol{\Phi}_{dh}^T \frac{\partial \boldsymbol{B}_{dd}}{\partial v} \boldsymbol{\Phi}_{dh} + \boldsymbol{\Phi}_{dh}^T \frac{\partial \boldsymbol{K}_{dd}}{\partial v} \boldsymbol{\Phi}_{dh} \right] \tag{25-45}$$

where the derivatives of the *d-set* matrices are identical to their counterparts in the direct method. The assumption that the normal modes, $\boldsymbol{\Phi}$, are invarient — and, by extension, the modal damping — implies that modal truncation effects are at least as pronounced in the computation of the derivative as in the solution. It is always a good practice to retain additional modes, or to use residual flexibility vectors, to improve accuracy when using the modal approach.

Equations (25-41) or (25-45) are solved only for the frequency steps that are associated with *active* constraints for each frequency response subcase. Given the displacement sensitivities, the GRID point response constraint sensitivities are obtained in a straightforward manner.

25.3.5.5 Mode Tracking

During the MDO iterations, changes in design variables may easily cause changes in the order of the modes of the finite element model. Thus, if you have applied a constraint to a specific frequency or eigenvector you may no longer be applying it to the correct mode. To accomodate this condition, **UAI/*NASTRAN*** performs automatic mode tracking. This means that when you constrain a specific mode shape, the constraint is always applied to that specific shape. This is done by computing the cross-orthogonality between the eigenvectors of successive iterations, *k-1* and *k*:

$$\boldsymbol{XORTHO} = (\boldsymbol{\Phi}^{\boldsymbol{k-1}})^T \boldsymbol{M}\, \boldsymbol{\Phi}^{\boldsymbol{k}} \tag{25-46}$$

From the ***XORTHO*** matrix, **UAI/*NASTRAN*** determines which mode shapse in the new analysis matches those of the previous analysis. This tracking sequence is used to apply constraints to the initial mode shape that you specify in your Bulk Data regardless of its order of extraction in the updated model.

25.4 INPUT DATA REQUIREMENTS

This section provides you with a description of the **UAI/*NASTRAN*** input data required for performing Multidisciplinary Design Optimization.

25.4.1 Executive Control Commands

The MDO solution sequence is selected using the Executive Control command:

```
SOL MULTI
```

This command invokes the multidisciplinary analysis and optimization APEX sequence. Optimization will then be performed if design variables exist in the Bulk Data ***and*** the **OBJECTIVE** command appears in the Case Control.

25.4.2 Case Control Commands

There are three special Case Control commands used for MDO. These commands are used to select an objective function, design constraints, and to request optional solution results which will be written to the Archive database for use with ***eShell***. Each is described in the following sections.

25.4.2.1 Selecting the Objective Function

The first Case Control command, and it subcommands, specifies the objective function and optional parameters for the optimization. These commands are:

```
                 { { MINIMIZE } { WEIGHT                    } }
                 { { MAXIMIZE } { MASS                      } }
OBJECTIVE        {              { VOLUME                    } }
                 {              { single_value_constraint_id } }
                 {                                            }
                 { NONE                                       }

   MAXITER     = niter
   CONVERGE    = per_cent
   MAXFEASIBLE = gval
   MINACTIVE   = gval
```

You may select a specific function which will be either minimized or maximized. The objective function may be the total model **WEIGHT**, the total model **MASS**, the total material **VOLUME**, or a design constraint which results in a single contraint value. This constraint is selected by specifiying the *single_value_constraint_id*. The objective can also be **NONE**, indicating that optimization is desired, but only to achieve feasibility: there will be no objective function during the optimization cycles. The **OBJECTIVE** command ***must*** be used to cause the MDO solution sequence to iterate. In its absence, a single set of multidisciplinary analyses are performed. Sensitivity analysis is performed if design variables and constraints are in the input stream.

The **OBJECTIVE** subcommands are used to control the optimization procedure. The **MAXITER** option specifies the maximum number of redesigns that will be performed. Note, the base analysis is not included in the iteration count. The default for *niter*, which must be a positive value, is **10**. A value of **0** is useful to allow just the base analysis to take place and the summary of the first constraint screening to be echoed. The **CONVERGE** command specifies the convergence limit, as a

percentage, *per_cent*. The default value is **1%**. The final two subcommands, **MAXFEASIBLE** and **MINACTIVE**, specify the constraint values required for a constraint to be considered feasible and to be considered active, respectively. The meaning of these is defined in (25-22). See the ***User's Reference Manual*** for more information.

25.4.2.2 Selecting Design Constraints

There are three Case Control commands used to select design constraints. The first is used to select overall model constraints (those that are independent of analysis case):

```
[SET rcsid = set_specifier]
...
MODDESCON = { ALL  }
            { rcsid }
```

with this command, you may select all **DCMODEL** Bulk Data entries, or you may specify a **SET** of entries referenced by *rcsid*. This command must appear above all analysis **CASE**s and the optional **SET**s must be defined before they are referenced. The second command is **DESCON** which is used to select response constraint Bulk Data entries on a case-by-case basis. The format of this command is:

```
[SET rcsid = set_specifier]
...
DESCON = { ALL  }
         { rcsid }
```

Note that an optional **SET** command is again shown. The principal use of the **DESCON** command is to select some, or all, of the constraint data from your Bulk Data packet. If only some of the data are to be selected, then the set *rcsid* contains one or more identification numbers of constraint Bulk Data entries as described in the next section. Note that if you specify an *rcsid*, then a corresponding **SET** with the same identification number must appear in your Case Control command packet before it is referenced. When the *rcsid* is replaced by the keyword **ALL**, then all of the design constraints which appear in your Bulk Data packet will be used in the associated analysis case.

The final command controls the screening of the design constraints. This command is:

```
MAXRETAIN = num
```

where *num* indicates the maximum number of constraints that will be considered active during the formation of the approximate problem. The default value for *num* is 100. In addition to these constraints, all pseudo-side constraints are also retained.

25.4.2.3 Defining the Multidisciplinary Analysis Sequence

The MDO capability allows structural models to be designed subject to constraints spanning a number of analytical disciplines. In Version 11.7, these include statics, normal modes, and both direct and modal frequency response. Each discipline and loading condition is indicated by the **CASE** Case Control command:

```
CASE id [ { STATICS    } ]
        [ { MODES      } ]
        [ { DFREQUENCY } ]
        [ { MFREQUENCY } ]
```

Each **CASE** may represent a different discipline, or, for statics and direct frequency response, a different loading condition. Each **CASE** may also have a different boundary condition, including Normal Modes analyses. This allows a structure to be optimized for several different operating conditions simultaneously.

25.4.2.4 Saving MDO Results

The MDO process creates very large volumes of intermediate data. These data include sensitivities and design variable values for each iteration in the design cycle. To request that these intermediate results be placed on the ARCHIVE Database, you may use the special options in the **ARCHIVE** command:

```
ARCHIVE [ { SENSITIVITY } ] [ TO logical_name [ :path_name ] ]
        [ { OPTIMIZATION} ]
```

One or both commands may be used. They are intended to be placed above the discipline level, because only one command of each type will, in fact, be honored.

25.4.3 Bulk Data Entries — Design Variables

Five Bulk Data entries are used to define design variables in **UAI/*NASTRAN***. They fall into three categories:

- Physical Properties: **DVPROP** and **DVPROPS**
- Grid Point Coordinates: **DVGRID** and **DVGRIDS**
- Reduced-Basis Linking: **DVLINK**

Each of these is described in this section and examples of their use are given.

25.4.3.1 Defining Design Variables which are Physical Properties

As you have seen, a physical design variable, or property, is a characteristic of your finite element model that may be changed during the design process. The physical property design variables that you may select in **UAI/*NASTRAN*** are from three groups:

- Element Property Entries.
- Element Connection Entries.
- Material Property and Modal Damping Entries

The specific physical properties which may be used as design variables are presented in Table 25-2. When you are defining a design variable that corresponds to a physical design variable, you use the **DVPROP** Bulk Data entry. The format of this entry is:

DVPROP	**DVNAME**	**PTYPE**	**PRPID**	**LBOUND**	**UBOUND**	**PSYM**			

Each design variable has a unique character name, **DVNAME**, which is used to identify the variable in the **UAI/*NASTRAN*** solution results. The third and fourth fields specify a pair of values which represent either an element connection or property type and identification number, or a material type and identification number. The side constraints on the *physical* design

Table 25-2. ALLOWABLE PHYSICAL DESIGN VARIABLES

GROUP	BULK DATA ENTRY	PHYSICAL PROPERTY SYMBOL	DESCRIPTION
PROPERTIES	PBAR	A	Cross Sectional Area
		I1,I2,I12	Area Moments of Inertia
		J	Torsional Constant
		NSM	Nonstructural Mass
		K1,K2	Shear Area Factors
	PBAR1	D1,D2,D3,D4,D5,D6	Cross Sectional Area Parameters
		NSM	Nonstructural Mass
	PBEAM	AA, A1,...,A9, AB	Cross Sectional Areas
		I1A,I11,...,I19,I1B	Area Moments Of Inertia
		I2A,I21,...,I29,I2B	Area Moments Of Inertia
		I12A,I121,...,I129,I12B	Area Products Of Inertia
		JA,J1,...,J9,JB	Torsional Stiffness Parameters
		NSMA, NSM1,...,NSM9, NSMB	Nonstructural Masses
		K1, K2	Shear Stiffness Factors
		S1, S2	Shear Relief Coefficients
		NSIA, NSIB	Moments of Inertia about Nonstructural Mass CG
		CWA, CWB	Warping Coefficients
		YMA, ZMA, YMB, ZMB	Coordinates of Nonstructural Mass CG
		YNA, ZNA, YNB, ZNB	Coordinates of Neutral Axis
	PBEAM1	D1A,D2A,D3A,D4A,D5A	Cross Section Shape Parameters
		D1B,D2B,D3B,D4B,D5B	Cross Section Shape Parameters
		NSMA, NSMB	Nonstructural Mass
	PBUSH	K1, K2,...,K6	Bushing Spring Stiffnesses
		B1, B2,...,B6	Bushing Dampers
	PDAMP	B	Damping Value
	PELAS	K	Stiffness
	PMASS	M	Mass
	PPILE	A,T,D	Cross Sectional Area Parameters
		NSM	Nonstructural Mass
	PPILE1	A	Cross Sectional Area
		I1,I2	Area Moments of Inertia
		J	Torsional Constant
		NSM	Nonstructural Mass
	PROD	A	Cross Sectional Area
		J	Torsional Constant
		NSM	Nonstructural Mass
	PSHEAR	T	Plate Thickness
		NSM	Nonstructural Mass
	PSHELL	T	Plate Thickness
		IFACT	Bending Stiffness Parameter
		TFACT	Transverse Shear Parameter
		NSM	Nonstructural Mass
		Z0	Offset from Reference Plane

CONTINUED ON NEXT PAGE

Table 25-2. ALLOWABLE PHYSICAL DESIGN VARIABLES(Cont'd)

GROUP	BULK DATA ENTRY	PHYSICAL PROPERTY SYMBOL	DESCRIPTION
ELEMENTS	CDAMP2,4	B	Damping Value
	CELAS2,4	K	Stiffness
	CMASS2,4	M	Mass
	CONM1	Mij	Mass
	CONM2	M	Mass
		Iij	Moments of Inertia
MATERIALS	MAT1	E, G, NU	Young's Shear Moduli and Poisson's Ratio
		RHO	Density
		ALPHA	Thermal Expansion Coefficient
		GE	Structural Damping Coefficient
	MAT2	Gij	Material Property Matrix
		RHO	Density
		ALPHA1,ALPHA2,ALPHA12	Thermal Expansion Coefficients
		GE	Structural Damping Coefficient
	MAT8	E1,E2	Young's Moduli
		NU12	Poisson's Ratio
		Gij	Shear Moduli
		RHO	Density
		ALPHA1, ALPHA2	Thermal Expansion Coefficients
		GE	Structural Damping Coefficient
	MAT9	Gij	Material Property Matrix
		RHO	Density
		ALPHAi	Thermal Expansion Coefficients
		GE	Structural Damping Coefficient
MODAL DAMPING	TABDMP1	d1,d2,...	Modal damping as a function of frequency
	TABDMP2	d1,d2,...	Modal damping at specific modes

variable are specified by **LBOUND** and **UBOUND**. These physical variable side constraints are automatically used to determine the *mathematical* design variable side constraints whenever possible. Under certain linking options, however, the physical side constraints are used to create *pseudo-side constraints*, which are explicit constraints formulated such that satisfaction of the constraint satisfies the **LBOUND** or **UBOUND** physical bound. Finally, the **PSYM** field selects the symbol for one of the allowable physical properties from the selected **PTYPE** that specifies the design variable. Two examples of the use of the **DVPROP** Bulk Data entry are given below.

1. Define design variable **THICK** as the thickness of all plate elements which reference the **PSHELL** Bulk Data entry with identification number **99**, assume that the thickness must range between **0.01** and **100.0**.

DVPROP	**THICK**	**PSHELL**	**99**	**0.01**	**100.0**	**T**			

2. Define design variable **YOUNG** as Young's Modulus for all element properties which reference the **MAT1** Bulk Data entry with identification number **10**. Specify that the modulus must be between $\mathbf{1.5x10^6}$ and $\mathbf{3.0x10^7}$.

DVPROP	**YOUNG**	**MAT1**	**10**	**1.5+6**	**3.+7**	**E**			

Every **DVPROP** Bulk Data entry which you place in the Bulk Data packet is used during the design process. Each data entry may represent a single unique physical element property, or, if the **PRPID** is referenced more than once, a group of element properties which assume the same value. This is called ***Shared Property Linking***.

Two or more physical property *entries* may be linked to form a mathematical design variable by using **DVPROPS** entries. A third form of linking, using the **DVLINK** entries, allows physical variables defined by **DVPROP** entries to be linked together to form reduced bases. The manner in which you define these linking options is discussed in the next section. Those **DVPROP** entries which are linked together using **DVLINK** will not also be used as separate design variables. Those physical property entries that appear on **DVPROPS** may not be used on a **DVLINK** entry. A single **DVPROP** may, and typically will, participate in more than one **DVLINK** linking relationship.

25.4.3.2 Simple Physical Linking

You will note that the **DVPROP** entry does not allow you to control multiple property *entries* as one mathematical variable without defining a linking relationship using **DVLINK**. When defining a design model, however, one commonly wants design variables that link elemental properties together independently from the Bulk Data property entries. To accommodate this common linking, you may define a fixed relationship among two or more property entries. This is referred to as ***Physical Linking*** since the mathematical variable that results is still typically a physical value and the **LBOUND** and **UBOUND** values can directly yield the bounds on the mathematical variable. This is done with the Bulk Data entry

DVPROPS	**DVNAME**	**PTYPE1**	**PRPID1**	**LB1**	**UB1**	**PSYM1**			**-cont-**
-cont-		**PTYPE2**	**PRPID2**	**LB2**	**UB2**	**PSYM2**			**-cont-**

Two or more properties are assigned to the single design variable, **DVNAME**. This variable, *v*, is then defined as a single linear function of the specified properties:

$$p = p_o \cdot v$$

where $\boldsymbol{p}_o$ is a vector which contains the initial value of the properties selected. For example, consider the following Bulk Data entries:

PSHELL	**101**	**100**	**0.5**						
PSHELL	**201**	**100**	**0.75**						
DVPROPS	**SHAPE1**	**PSHELL**	**101**	**0.1**	**1.0**	**T**			**-cont-**
-cont-		**PSHELL**	**201**	**0.1**	**1.0**	**T**		,	

In this example, two **PSHELL** thicknesses are linked as a single design variable, **SHAPE1**. The initial values from the referenced **PSHELL** identification numbers (**101** and **102**) define the vector $\boldsymbol{p}_o$ so that:

$$\left\{ \begin{array}{c} \mathbf{T_{101}} \\ \mathbf{T_{102}} \end{array} \right\} = \left[\begin{array}{c} 0.5 \\ 0.75 \end{array} \right] \mathbf{SHAPE1}$$

Notice that **SHAPE1** is still physical, that is, it represents a thickness value. **UAI/*NASTRAN*** will automatically create this relationship from the **DVPROPS** and assign **SHAPE1** an initial value of **1.0**. Further, the bounds on **SHAPE1** will be computed:

$$\mathbf{SHAPE1}_{min} = \max\left(\left\{ \frac{0.1}{0.5} , \frac{0.1}{0.75} \right\} \right) = 0.2000$$

$$\mathbf{SHAPE1}_{max} = \min\left(\left\{ \frac{1.0}{0.5} , \frac{1.0}{0.75} \right\} \right) = 1.3333$$

The value of **SHAPE1** is not be the exact thickness of any element, but it will only be different by a ***single*** scalar multiplier. That same multiplier is used to establish the bounding values of **SHAPE1**. The most restrictive physical bounds will control the final mathematical bounds. While this example shows the most common use: linking like property entries/symbols together, there is no requirement that the properties be of the same type. Thus, the **DVPROPS** entry allows a great deal of flexibility in defining physical linking — where the ratios of the designed quantities remain fixed. Still more complex linking, referred to as Reduced Basis Linking, allows nonphysical mathematical design variables to be defined.

25.4.3.3 Reduced Basis Linking of Physical Properties

As discussed in Section 25.2.5, a mathematical design variable may control any number of physical properties. The linking function is defined by specifying a vector of invariant physical properties, $\boldsymbol{v}_p^{inv}$, and a transformation matrix, $\boldsymbol{T}$, defining the dependencies of the physical properties, $\boldsymbol{v}_p$ to the mathematical design variables, $\boldsymbol{v}_m$:

$$\boldsymbol{v}_p = \boldsymbol{v}_p^{inv} + \boldsymbol{T}\boldsymbol{v}_m$$

A linked, or mathematical, design variable is defined with the **DVLINK** Bulk Data entry:

DVLINK	DVNAME	VINIT	PINV1	C1	PDV1	PINV2	C2	PDV2	-cont-
-cont-			PINV3	C3	PDV3	*CONTINUES IN GROUPS OF 3*			-cont-

As was the case with the **DVPROP** Bulk Data entry, a linked design variable is also assigned a unique name which is specified by the **DVNAME** field. The initial value of the mathematical design variable is then given by **VINIT**. The actual design variable linking is then defined by entering triples of data which define the rows of $\boldsymbol{v}_p^{inv}$ and $\boldsymbol{T}$ as written in (25-11). The **PINVi** fields define the invariant portion of the physical design variable whose name is **PDVi**. The **Ci** are the constant values defining the rows of the $\boldsymbol{T}$ matrix column for the mathematical variable being defined.

By itself, a single **DVLINK** does not result in ***Reduced Basis Linking***. Only when a single **DVPROP** entry appears in ***more than one*** **DVLINK** are two or more reduced basis mathematical design variables defined. Whenever a physical design variable, **DVPROP**, is controlled by *one* mathematical variable, the linking can be viewed as a simple physical linking. The only difference between a single **DVLINK** and a **DVPROPS** is that the coefficients of the $\boldsymbol{T}$ matrix for the latter input are predefined to be the initial property values and **VINIT** is set to **1.0**. ***UAI/NASTRAN*** recognizes this case and will not generate pseudo-side constraints unless true reduced basis linking has been defined.

The following examples show how you may define both reduced basis mathematical variables and specialized simple physical linking.

3. Consider the first example presented in Section 25.2.5. The mathematical design variable represents the thickness of element 1, T_1, in the model and the linking relationship is defined as:

$$\begin{Bmatrix} t_1 \\ t_2 \\ t_3 \end{Bmatrix} = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \end{bmatrix} + \begin{bmatrix} 1.000 \\ 0.667 \\ 0.333 \end{bmatrix} \{ T_{LINK} \}$$

Specify the appropriate Bulk Data entries to define the design variable and linking rules. Assume that the elements are numbered 1,2,3 and that each references a **PSHELL** entry with the same ID number.

DVLINK	TLINK	1.0	0.0	1.0	T1	0.0	0.667	T2	+DVL1
+DVL1			0.0	0.333	T3				
DVPROP	T1	PSHELL	1			T			
DVPROP	T2	PSHELL	2			T			
DVPROP	T3	PSHELL	3			T			

Notice that, in this case, the variable **TLINK** is easily related to a physical variable. It is, in fact, the multiplier to a linearly tapered thickness distribution. Also, the bounding mathematical variable values can be (and are) determined explicitly from the T_i and the $L/_{UBOUND_i}$ values. In fact, this use of **DVLINK** is nothing more than a **DVPROPS** with coefficients in T that are not derived from the property entries.

4. Consider a second example. Here the linking rule allows all possible linear tapers. An invarient thickness is used to provide an constant offset based on expected results:

$$\begin{Bmatrix} t_1 \\ t_2 \\ t_3 \end{Bmatrix} = \begin{bmatrix} 8.0 \\ 8.0 \\ 8.0 \end{bmatrix} + \begin{bmatrix} 1.0 & 0.0 \\ 1.0 & 10.0 \\ 1.0 & 20.0 \end{bmatrix} \begin{Bmatrix} b \\ m \end{Bmatrix}$$

Specify the Bulk Data entries to solve this problem using the nomenclature of the previous example.

DVLINK	**B**	**0.0**	**8.0**	**1.0**	**T1**	**8.0**	**1.0**	**T2**	**+DVL1**
+DVL1			**8.0**	**1.0**	**T3**				
DVLINK	**M**	**0.1**	**8.0**	**0.0**	**T1**	**8.0**	**10.0**	**T2**	**+DVL2**
+DVL2			**8.0**	**20.0**	**T3**				
DVPROP	**T1**	**PSHELL**	**1**			**T**			
DVPROP	**T2**	**PSHELL**	**2**			**T**			
DVPROP	**T3**	**PSHELL**	**3**			**T**			

You will note that while the **DVLINK** entry includes the initial value of the mathematical design variables, **VINIT**, it does not include a provision for upper and lower bound side constraints. These are defined to be -10^{20} and 10^{20}, respectively. These defaults are used because the mathematical variable defined by **DVLINK** has no *a priori* physical interpretation and, therefore, no easily predetermined bounds. The defaults result in an unbounded mathematical variable. The physical bounds on the **DVPROP** entries are imposed by the automatic generation of *pseudo-side constraints* as discussed in Section 25.3.3.

You may override the default values of the mathematical bounds by using the **DVMATH** entry:

DVMATH	**DVNAME**	**FNAME**	**VINIT**	**VMIN**	**VMAX**				

Here, the **FNAME** references the name of a mathematical variable defined by a **DVLINK** entry. The fields **VINIT**, **VMIN**, and **VMAX** then allow you to override any or all of these values. Remember that this entry is optional and it is only required if you must change the side constraints.

25.4.3.4 Defining Design Variables which are Grid Point Coordinates

You may define the locations of GRID point coordinates in your model to be design variables. There are two forms of input to do this. The first is:

DVGRID	**DVNAME**	**GID**		**LBOUND**	**UBOUND**	**COORD**			

Again, the design variable has a unique character name, **DVNAME**, which is followed by the identification number of a GRID point, **GID**, one of whose coordinates will be a design variable. The side constraints on the physical location are again specified by **LBOUND** and **UBOUND**. Finally, the **COORD** field selects the symbol for the coordinate which is to be designed. The allowable values of **COORD** depend on the ***input coordinate system*** of the GRID point. These are shown in the following table.

TYPE OF INPUT COORDINATE SYSTEM	SYMBOL FOR COORDINATE TRANLATION:		
	X1	X2	X3
RECTANGULAR	X	Y	Z
CYLINDRICAL	R	THETA	Z
SPHERICAL	R	THETA	PHI

You may also define a simple physical linking which maintains a fixed ratio between two or more coordinates using the Bulk Data entry:

DVGRIDS	DVNAME	PTYPE		LBOUND	UBOUND	COORD			-cont-
-cont-		PTYPE		LBOUND	UBOUND	COORD			-cont-

Two or more coordinates are assigned to the single design variable, **DVNAME**. The **DVLINK** and **DVMATH** Bulk Data entries may also be used to perform more general linking of the geometric design variables. All the same rules for side constraints and pseudo-side constraints that were enumerated for properties apply equally to these geometric variables.

25.4.3.5 Defining Side Constraints on Design Variables

As you have seen, there are many bounds specifications on the various Bulk Data entries defining physical and mathematical design variables. As discussed earlier in Section 25.3.3, **UAI/*NASTRAN*** uses the physical bounds, **LBOUND** and **UBOUND** on the **DVPROP**, **DVPROPS**, **DVGRID**, and **DVGRIDS** entries to compute the corresponding bounds on the mathematical design variables. The only exception is that made in the case of true reduced basis linking where the corresponding bounds cannot be determined from the linking expression. In this case, **UAI/*NASTRAN*** defines the mathematical variables to be unbounded and generates *pseudo-side constraints* to replace the physical bound side constraints with true constraints. In general, then, you do not ever need to specify bounds for the mathematical variables. The **DVMATH** entry, however, can be used to override the computed mathematical bounds in the event that the user wishes to do so.

25.4.3.6 The Initial Design Point

When **UAI/*NASTRAN*** performs its first analysis, the physical design variables are those defined by Bulk Data. The Bulk Data Packet contains values for property and GRID entries, and for design variables. These data are combined internally to create the ***initial design***. The following describes the steps taken to determine the initial design from the Bulk Data.

Recall the linking relationship:

$$\boldsymbol{v}_p = \boldsymbol{v}_p^{inv} + \boldsymbol{T}\boldsymbol{v}_m$$

The ***contents*** of $\boldsymbol{v}_p$ have been defined from **DVPROP**, **DVPROPS**, **DVGRID**, and **DVGRIDS** Bulk Data entries. Indirectly, the existence of $\boldsymbol{v}_m$ has been implied. There are two distinct scenarios discussed below:

- ***When there are no* DVLINK *or* DVMATH *entries***. Then, $\boldsymbol{v}_p^{inv}$ is null and the $\boldsymbol{T}$ matrix contains either unit values (if **DVPROP** or **DVGRID**) or values from the Bulk Data Field (if **DVPROPS** or **DVGRIDS**) and similarly the $v_m^{initial}$ vector is the value from the Bulk Data field or a unit value.
- ***When there are physical design variables and* DVLINK *and/or* DVMATH *entries***. Then, $\boldsymbol{v}_p^{inv}$ is null for all cases except where **DVLINK** contains values. The $\boldsymbol{T}$ matrix is as above except where **DVLINK** explicitly overrides the **DVPROP** Bulk Data field value with its own input. The $v_m^{initial}$ vector is as above except where **DVLINK** or **DVMATH** entries provides another value.

In all cases, the $v_p^{initial}$ vector is ***computed*** from these data and ***the original data from the Bulk Data fields are overridden!*** Mathematically this is represented:

$$\boldsymbol{v}_p^{initial} \equiv \boldsymbol{v}_p^{inv} + \boldsymbol{T}\boldsymbol{v}_m^{initial}$$

25.4.4 Bulk Data — Defining Response Constraints

As described earlier, you may define any number of response constraints during your analysis. These are taken to be inequality constraints which are applied as upper bounds, lower bounds, or both. You specify these constraints with **DCGRID**, **DCGRIDM**, **DCELEM**, **DCFREQ**, **DCMODR**, or **DCDYNRG** Bulk Data entries, depending upon the type of analytical response to be constrained.

25.4.4.1 Static Displacement Constraints

The **DCGRID** entry is used for defining static displacement constraints. Its format is:

DCGRID	**DCSID**	**DCNAME**	**'DISP'**	**COMP**	**LLIM**	**ULIM**		**GID1**	**-cont-**
-cont-	**GID2**	**GID3**	*CONTINUES WITH LIST OF GRID POINTS*						**-cont-**

The **DCSID** field is the constraint set identification number that was a member of a **SET** that you selected in Case Control using the **DESCON** command ***of a STATICS case***. The **DCNAME** field allows you to specify a name for the constraint that is used to label printed output. The **LLIM** field specifies a lower bound constraint, and the **ULIM** field an upper bound constraint. Both may be selected on a single entry. ***Remember, if you are only defining a single bound, leave the other field blank.*** You may select from the usual six degrees of freedom for which displacements are computed. The codes used are: **T1**, **T2**, **T3**, **R1**, **R2**, and **R3**. Finally, you list one or more grid point identification numbers to which the specified constraints apply.

A second form of displacement constraint may be defined using:

DCGRIDM	**DCSID**	**DCNAME**	**'DISP'**		**LLIM**	**ULIM**			**-cont-**
-cont-	**GID1**	**COMP1**	**A1**	**GID2**	**COMP2**	**A2**			**-cont-**
-cont-	**GID3**	**COMP3**	**A3**	***CONTINUES IN GROUPS OF 3***					**-cont-**

This defines a single constraint on a linear combination of displacement components defined by:

$$\mathrm{DC} = \sum_{i} \mathrm{A}_i \left(\mathrm{GID}_i \, , \, \mathrm{COMP}_i \right)$$

The following table presents the equations used for computing the constraint value or objective function value for the **DCGRID** and **DCGRIDM** entries.

CONSTRAINT SPECIFICATION	FORM OF DESIGN CONSTRAINT	FORM AS OBJECTIVE FUNCTION
$R_g = \sum_i \mathbf{Ai}\, u_i \geq R_{llim}$	$g = \dfrac{R_{llim} - R_g}{\lvert R_{llim} \rvert}$	$F = R_g$
$R_g = \sum_i \mathbf{Ai}\, u_i \leq R_{ulim}$	$g = \dfrac{R_g - R_{ulim}}{\lvert R_{ulim} \rvert}$	

5. Define a constraint on the negative z-displacement of GRID Point **200** as **D200** and specify that it cannot exceed **0.1**. Assume the constraint set identification number is **100**.

DCGRID	**100**	**D200**	**DISP**	**T3**		**0.1**		**200**	

6. Specify a constraint, called **RELD**, that requires the average of the x-displacements of Grid Points **101**, **201** and **301** is less than **2.0**. Assume the constraint set identification number is **101**.

DCGRIDM	**101**	**RELD**	**DISP**			**2.0**			
	101	**T1**	**0.3333**	**201**	**T1**	**0.3333**			
	301	**T1**	**0.3333**						

25.4.4.2 Static Element Response Constraints

When you define constraints on element related responses ***for STATICS analyses***, you use one of the formats:

DCELEM	**DCSID**	**DCNAME**	**'STRESS'** **'STRAIN'** **'FORCE'**	**COMP**	**LLIM**	**ULIM**		**EID1**	**-cont-**
-cont-	**EID2**	**EID3**	*CONTINUES WITH LIST OF ELEMENTS*						**-cont-**

As in the case of the static displacement response, you must specify a component of element stress, strain, or force. These **COMP**s, which depend on the element type, are given in Table 25-3.

The following table presents the equations used for computing the constraint value or objective function value for the **DCELEM** entry.

CONSTRAINT SPECIFICATION	FORM OF DESIGN CONSTRAINT	FORM AS OBJECTIVE FUNCTION
$R_e = \left\{ \begin{matrix} R_\sigma \\ R_\varepsilon \\ R_p \end{matrix} \right\} \geq R_{llim}$	$g = \dfrac{R_{llim} - R_e}{\left\lvert R_{llim} \right\rvert} \leq 0$	$F = R_e$
$R_e = \left\{ \begin{matrix} R_\sigma \\ R_\varepsilon \\ R_p \end{matrix} \right\} \leq R_{ulim}$	$g = \dfrac{R_e - R_{ulim}}{\left\lvert R_{ulim} \right\rvert} \leq 0.0$	

7. Place stress constraints on all **QUAD4** elements in a model such that the stress in the x-direction does not exceed a tensile allowable of 450Ksi and a compressive allowable of 350Ksi. Assume that the elements are numbered from 1 to 100. Use a set ID of 10 and a **LABEL** of **SIGLIM**.

DCELEM	**10**	**SIGLIM**	**STRESS**	**SIGX**	**-3.5+5**	**4.5+5**		**1**	
	THRU	**100**							

Table 25-3. ELEMENT RESPONSE CONSTRAINT SYMBOLS

RESPONSE TYPE	RESPONSE SYMBOL	ELEMENT CLASS														
		SCLR	ONE-D				TWO-D							THREE-D		
		ELASi	BAR	BEAM	PILE	ROD	QUAD4	TRIA3	QUADR	TRIAR	QUAD8	TRIA6	SHEAR	TETRA	PENTA	HEXA
ELEMENT STRESS	SIGX	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔		✔	✔	✔
	SIGY						✔	✔	✔	✔	✔	✔		✔	✔	✔
	SIGZ													✔	✔	✔
	TAUXY						✔	✔	✔	✔	✔	✔		✔	✔	✔
	TAUYZ													✔	✔	✔
	TAUZX													✔	✔	✔
	SIG1		✔	✔	✔	✔	✔	✔	✔	✔	✔	✔		✔	✔	✔
	SIG2						✔	✔	✔	✔	✔	✔		✔	✔	✔
	SIG3													✔	✔	✔
	MEANSTRS													✔	✔	✔
	MAXSHEAR					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
	VONMISES		✔	✔	✔	✔	✔	✔	✔	✔	✔	✔		✔	✔	✔
ELEMENT STRAIN	EPSX						✔	✔	✔	✔	✔	✔				
	EPSY						✔	✔	✔	✔	✔	✔				
	EPSXY						✔	✔	✔	✔	✔	✔				
	EPS1						✔	✔	✔	✔	✔	✔				
	EPS2						✔	✔	✔	✔	✔	✔				
	MAXSHEAR						✔	✔	✔	✔	✔	✔	✔			
	VONMISES						✔	✔	✔	✔	✔	✔				
ELEMENT FORCE	AXIAL	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔				
	SHEAR		✔	✔	✔		✔	✔	✔	✔	✔	✔	✔			
	BENDING		✔	✔	✔		✔	✔	✔	✔	✔	✔				
	TWIST		✔	✔	✔	✔	✔	✔	✔	✔	✔	✔				

25.4.4.3 Frequency Constraints

You may constrain one or more of the natural frequencies ***of a MODES analysis*** to be either above or below a specified threshhold. These constraints are specifed with the entry:

DCFREQ	DCSID	DCNAME	UPPER LOWER	LIM	MODEID				

Where you define a single mode identification number, **MODEID**, which is to have a either an **UPPER** or **LOWER** frequency bound of **LIM** cycles per second (Hz). Recall from Section 25.3.5.5 that this constraint will be tracked and applied to the frequency of the mode representing the initial shape, even if modes have switched during the design process.

The following table presents the equations used for computing the constraint value or objective function value for the **DCFREQ** entry.

CONSTRAINT SPECIFICATION	FORM OF DESIGN CONSTRAINT	FORM AS OBJECTIVE FUNCTION
$f \leq$ **LIM** if **UPLOW** = **UPPER**	$g = \dfrac{\lambda}{4\pi^2 \mathbf{LIM}^2} - 1 \leq 0$	$F = \lambda$
$f \geq$ **LIM** if **UPLOW** = **LOWER**	$g = 1 - \dfrac{\lambda}{4\pi^2 \mathbf{LIM}^2} \leq 0$	

8. Constrain the first and second natural frequencies of a model so that they are both less than **5.0** Hz. Assume the constraint set identification number is **99** and the **DVNAME** is **F1AND2**.

DCFREQ	99	F1AND2	UPPER	5.0	2				

25.4.4.4 Real Eigenvector Constraints

The **DCMODE** entry, which is used in sensitivity analysis, is ignored in MDO. Instead, a required mode shape constraint ***for a MODES analysis***, very useful for System Identification, is defined with the entry:

DCMODR	DCSID	DCNAME	MODENUM	'COMP' 'RMS'	LIM	NORM	GNORM	CNORM	-cont-
-cont-		GID1	COMP1	DISP1	GID2	COMP2	DISP2		-cont-
-cont-		GID3	COMP3	DISP3	*CONTINUES IN GROUPS OF 3*				-cont-

In the usual manner, the constraint is assigned an identification number, **DCSID**, and name, **DCNAME**. Skipping ahead briefly, the groups of three values (**GIDi, COMPi, DISPi**) define the target mode shape at some number of GRID points for a given **MODENUM**, the mode sequence number that you wish to constrain. As in the case of the **DCFREQ**, this constraint will be tracked and applied to the mode representing the initial shape, even if modes have switched during the design process. The eigenvector components, **DISPi** are entered in the normalization defined in the **NORM** field. The FE modal analysis vectors are converted to this normalization *from whatever normalization is used in the analysis method* prior to computation of the constraint.

This entry may generate either one constraint for each triple or it may generate a single constraint for the entire shape. The former is selected by entering **COMP** in the fifth field, and the latter by entering **RMS**. Finally, **LIM**, a positive number, represents the maximum allowable (individual or RMS) difference between the computed mode shape and the desired

mode shape. Only those components appearing on this entry will be constrained, so sufficient terms must be used to characterize the mode shape.

When **RMS** is selected, the constraint formed is:

$$g = \frac{1}{nc}\left[\sum_{j=1}^{nc}\left[(u_i - \mathbf{VALi})^2\right]^{1/2}\right] - \mathbf{LIM} \le 0.0$$

and the objective is:

$$F = \frac{1}{nc}\left[\sum_{j=1}^{nc}\left[(u_i - \mathbf{VALi})^2\right]^{1/2}\right]$$

and when **COMP** is selected, the *nc* constraints are formed using:

$$g_i = \left[t(u_i - \mathbf{VALi})^2\right]^{1/2} - \mathbf{LIM} \le 0.0 \quad i = 1, \ldots, nc$$

and the objective is:

$$F = \left[(u_1 - \mathbf{VAL1})^2\right]^{1/2}$$

In each case, *nc* is the number of components, **(GIDi,COMPi)**, specified for the mode shape.

9. Constrain the first mode to be the same as that measured in test to within 1% RMS. The measured modal data are point normalized such that the mode has unit value at a point equivalent to GRID **10025**, component **T3** of the FE model. Assume the constraint set identification number is **99** and the **DVNAME** is **MODR1**.

DCMODR	**99**	**MODR1**	**1**	**RMS**	**0.01**	**POINT**	**10025**	**T3**	**+A**
+A		**1**	**T1**	**0.03**	**2**	**T1**	**0.06**		**+B**
+B		**...**			**...**				**...**

25.4.4.5 Dynamic Response Constraints

When performing ***Frequency Response analyses***, you may, for structural models, constrain the displacement, velocity and acceleration responses at specific GRID points and components. Additionally, when performing Fluid-Structure Interaction analyses, you may constrain the pressure, dB, and dB(A) responses (See Chapter 23 for a description of Fluid-Structure Interaction analyses). This is done with the Bulk Data entry:

DCDYNRG	**DCSID**	**DCNAME**	**'DISP' 'VELO' 'ACCE' 'PRES' 'DB' 'DBA'**	**'AVG' 'PEAK'**	**LLIM** **LLTAB**	**ULIM** **ULTAB**	**'FMIN'**	**'FMAX'**	**-cont-**
-cont-	**GID1**	**COMP1**	**A1**	**GID2**	**COMP2**	**A2**			**-cont-**
-cont-	**GID3**	**COMP3**	**A3**	***CONTINUES IN GROUPS OF 3***					**-cont-**

In the usual manner, the constraint is assigned an identification number, **DCSID**, and name, **DCNAME**. The **DCDYNRG** entry then constrains a response amplitude, $R(\omega)$, over a range of frequencies. The response is built from the complex displacement (u_i), velocity, or acceleration responses at one or more degrees of freedom using:

$$R(\omega) = \left[\sum_{i=1}^{ng} \mathbf{VALi} \left[mag \left\{ \begin{matrix} u_i \\ i\,\omega\,u_i \\ -\omega^2 u_i \end{matrix} \right\} \right] \right]$$

Where the selection **DISP**, **VELO**, **ACCE**, **PRES**, **DB**, or **DBA** selects which component will be used to form the response. The user then selects either an **AVeraGe** of the response over the range **FMIN** to **FMAX** or the **PEAK** response over the range.

When **AVG** is selected, the one upper and/or one lower bound constraint is formed:

$$\text{for upper bounds: } g = \frac{1}{nf}\left[\sum_{j=1}^{nf} \frac{R(\omega_j)}{\mathbf{LIM}_j^U} \right] - 1.0 \le 0.0$$

$$\text{for lower bounds: } g = 1.0 - \frac{1}{nf}\left[\sum_{j=1}^{nf} \frac{R(\omega_j)}{\mathbf{LIM}_j^L} \right] \le 0.0$$

where $\mathbf{LIM}_j^L$ and $\mathbf{LIM}_j^U$ are taken either from **LLIM** or **ULIM**, or from **TABLEDi** data referenced by **LLTAB** or **ULTAB**. When the table data are used, the limit values become functions of frequency. When referenced by **OBJECTIVE**, the corresponding objective function is:

$$F = \frac{1}{nf}\left[\sum_{j=1}^{nf} R(\omega_j) \right]$$

and when **PEAK** is selected, *nf* upper and lower bound constraints are formed using:

$$\text{for upper bounds: } g_j = \frac{R(\omega_j)}{\mathbf{LIM_j^U}} - 1.0 \le 0.0 \quad j = 1,...,nf$$

$$\text{for lower bounds: } g_j = 1.0 - \frac{R(\omega_j)}{\mathbf{LIM_j^L}} \le 0.0 \quad j = 1,...,nf$$

and in all cases for the objective:

$$F = R(\omega_i)$$

In each case, *nf* is the number of frequencies specified by **FREQi** Bulk Data entries in the range:

LRANGE ≤ f ≤ **URANGE**

The frequency range is given in Hertz. Any number of constraints can be placed over the range of analysis frequencies, combinatorial responses, displacements, velocities, accelerations and/or fluid responses.

10. Constrain the peak value of the average vertical acceleration among GRID points **101** and **102**, to be less that **1000** in/sec/sec over the frequency range **0** to **50** Hertz. Assume the constraint set identification number is 99 and the **DVNAME** is **101RD102**.

DCDYNRG	99	101RD102	ACCE	PEAK		1000.0	0.0	50.0	+A
+A	101	T3	0.50	102	T3	0.50			

25.4.5 Defining Model Characteristic Constraints

You may also place side constraints on global model characteristics including weight, mass, and volume. This is done with the entry:

DCMODEL	DCSID	DCNAME	'WEIGHT' 'MASS' 'VOLUME'		LLIM	ULIM			

In general, the model characteristics are used as the objective function. There are special options to the **OBJECTIVE** command to automatically call for these quantities. However, when the objective function is a specific response quantity, you may want to constrain these model characteristics or you may choose to call out a **DCMODEL** as the "response" constraint identifier option on the **OBJECTIVE** command. The use of the **DCMODEL** entry is not precluded in any case. Remember that the entire model's weight, mass or volume are included in the computation so that your allowable, **LLIM** or **ULIM** must be scaled appropriately to include all these contributions.

The following table presents the equations used for computing the constraint value or objective function value for the **DCMODEL** entry. Note that C_g indicates one of the constrained model characteristics.

CONSTRAINT SPECIFICATION	FORM OF DESIGN CONSTRAINT	FORM AS OBJECTIVE FUNCTION
$C_g \geq C_{llim}$	$g = \dfrac{C_{llim} - C_g}{\lvert C_{llim} \rvert}$	$F = C_g$
$C_g \leq C_{ulim}$	$g = \dfrac{C_g - C_{ulim}}{\lvert C_{ulim} \rvert}$	

25.5 SOLUTION RESULTS

There are two basic types of output for MDO results. The first is the output from the selected analyses, and the second is the output from the optimization process itself. These are described in the following section.

25.5.1 Analysis Output

Each of the analytical disciplines in the MDO job may have its own output requests for displacement, stress, and any other selected solution results that are available for that discipline. All control mechanisms are the same.

25.5.2 Design Output

In addition to the analytical solution results, there are a number of MDO summaries that are automatically printed. These summaries trace the convergence status of the design problem and are useful in determining how the model is behaving. The different types of output are described in the following sections.

25.5.2.1 The Optimization Status Summary

The optimization status summary is given by the data shown in Table 25-4. It includes information regarding the status of both the global and approximate problem convergence. It also gives a count of the violated, critical, and satisfied constraints at the current design point, as well as the value of the largest constraint. Finally, it indicates whether the current design point is feasible.

25.5.2.2 The Objective Function

There are two output tables which summarize the status of the objective function for each design iteration. The first, shown in Table 25-5a, gives information regarding the current and predicted values of the objective function and the difference between them. This provides a measure of the accuracy of the approximate problem in modeling the actual design problem. It also indicates the previous value of the objective and the improvement made during the current iteration.

Table 25-4. OPTIMIZATION SUMMARY

```
S U M M A R Y   O F   O P T I M I Z A T I O N   P R O B L E M
            AFTER ANALYSIS 2 OF A MAXIMUM OF 21
           10 CONSTRAINTS RETAINED OF 10 APPLIED

    ************************************************
         GLOBAL CONVERGENCE HAS NOT BEEN ACHIEVED

            THE APPROXIMATE PROBLEM WAS NOT CONVERGED
            NUMBER OF VIOLATED CONSTRAINTS    :      0
            NUMBER OF CRITICAL CONSTRAINTS    :      0
            NUMBER OF SATISFIED CONSTRAINTS   :     10
            MAXIMUM CONSTRAINT VALUE     :  -9.051E-02

         THE CURRENT DESIGN POINT IS FEASIBLE
    ************************************************
```

Table 25-5. OBJECTIVE FUNCTION STATUS

a. Objective Function Summary

```
********************************
OBJECTIVE FUNCTION SUMMARY:

CURRENT VALUE    :    3.77682E+03

PREDICTED VALUE  :    3.77703E+03
PREDICTION DIFF. :       .00547 %

PREVIOUS VALUE   :    1.25894E+04
DELTA            :   -8.81258E+03
********************************
```

b. Detailed Objective Function Description

*** OBJECTIVE FUNCTION WEIGHT ***

RANK	CONID	SET-ID	LABEL	TYPE	BOUND	STATUS	ACTIVE	VALUE	PRED. VALUE	% DIFF.	ABS. DIFF.
N/A	1	N/A	WEIGHT	N/A	OBJECTIV	RET'D		3.7768E+03	3.7770E+03	5.4687E-03	2.0654E-01

The second, shown in Table 25-5b, provides a detailed summary of the nature of the objective function. It depends on the type of objective function that was selected. The form of this table is identical, except for the header, to the Design Constraint Summary Tables discussed in the next section. The header simply identifies that this summary relates to the objective function.

25.5.2.3 The Design Constraints

There are two levels of output regarding design constraints: an overall summary of the constraint screening and a detailed summary of the active constraints. An example of the former is shown in Table 25-6. This summary shows the value of **MAXRETAIN** and the breakdown of the number of constraints retained. The number retained within the target is the number of constraints retained because they are the most critical **MAXRETAIN** constraints. Usually this number will be the lesser of **MAXRETAIN** or all the applied constraints. It also shows the number of constraints that are retained because **PREVIOUSLY ACTIVE**. These constraints no longer fall in the top **MAXRETAIN** window, but did so in a recent iteration. These constraints are retained for **AGE** and are so labelled in the **STATE** field of the detailed constraint summary tables that are discussed next. The table then shows the number of pseudo-side constraints that **UAI/*NASTRAN*** generated due to reduced basis linking. Finally, the **MAXFEASIBLE** and **MINACTIVE** values are echoed for completeness.

Table 25-6. CONSTRAINT SUMMARY

```
S U M M A R Y   O F   O P T I M I Z A T I O N   P R O B L E M
            AFTER ANALYSIS 2 OF A MAXIMUM OF 16
           290 CONSTRAINTS RETAINED OF 560 APPLIED

 ********************************************************
 CONSTRAINT SCREENING SUMMARY: MAX RETENTION TARGET =   100

 NUMBER RETAINED WITHIN THE RETENTION TARGET         :  100
 NUMBER RETAINED BECAUSE PREVIOUSLY ACTIVE           :   16
 NUMBER RETAINED AS PHYSICAL PSEUDO-SIDE CONSTRAINTS :  174

 CONSTRAINTS ARE "   OK   " IF LESS THAN          :  -.0010
 LARGER CONSTRAINTS ARE "CRITICAL" IF LESS THAN   :   .0010
 CONSTRAINTS ARE "VIOLATED" IF GREATER THAN       :   .0010
 ********************************************************
```

In specifying the design constraints, the user input is tailored to allow a minimum number of input entries to generate many constraints. Since one Bulk Data entry can generate an open-ended number of constraints, you cannot typically give each constraint a unique identifier. The ability to uniquely identify the critical constraints, however, is key to gaining insight into the physical system and how the optimizer is attempting to improve its behavior. Consequently, **UAI/*NASTRAN*** gives a detailed description of each active constraint to allow you to identify which particular responses are key participants in the optimization process.

For output presentation purposes, the design constraints are separated into nine groups for detailed identification. On any given design cycle, only those constraints that are both ***applied and active*** will be summarized in the print file:

- ❒ Model
- ❒ Pseudo-side Constraints
- ❒ Displacement
- ❒ Element Stress
- ❒ Element Strain
- ❒ Element Force
- ❒ Mode Shape
- ❒ Frequency
- ❒ GRID Point Dynamic Response

The basic structure of each detailed constraint summary table is the same. It consists of a ranking value, the constraint identification data (which varies among constraint types), the current status and state of the constraint and, finally, measures of the prediction accuracy.

Table 25-7 describes in detail the common attributes of the constraint summary tables and Table 25-8 documents the constraint-type dependent and specialized data for all the constraint types. In the subsequent descriptions, only exceptions and special features will be noted in the examples.

Model: Table 25-9a shows an example of the **DCMODEL** constraint summary. Notice that, in this example, the "constraint summary" denotes that this is the objective function. Since it was called out by **OBJECTIVE MINIMIZE WEIGHT**, there is no **SET-ID** or **LABEL** and the **STATUS** is **OBJECTIV**.

This example also shows the behavior of the predictive accuracy attributes on the first iteration. Since no predictions exist, **N/A** is used in all associated columns.

Pseudo-side Constraints: Table 25-9b gives an example of the output for pseudo-side constraints: internally called **PROPCON** by **UAI/*NASTRAN***. These constraints are ***generated automatically*** when reduced basis linking is used. Because there may be many thousands of physical variables, only the critical and violated constraints are echoed. ***All pseudo-side constraints are retained during each design cycle to ensure that your side constraints are satisfied***.

Displacement: Table 25-10a presents an example of the output for constraints from **DCGRID** and **DCGRIDM**.

Element Stress: Table 25-10b provides an example of the output for constraints from **DCELEM** entries of type **STRESS**. Notice that the constraints with **CONID** 45 and 53 have been retained for **AGE**. These data in the **STATE** field imply that constraint 45 is now below the **MAXRETAIN**

Table 25-7. COMMON CONSTRAINT SUMMARY ATTRIBUTES

Shared Elements	**RANK**	Ranking from 1 to number of retained constraints in decreasing magnitude of constraint value. N/A implies that the constraint is retained for AGE limit or is now INACTIVE and echoed only as a final summary (see **STATE**). N/A is also used for the objective function.
	CONID	An internally assigned constraint identifier. In general it is a single value from 1 to ncon. For dynamic constraints in which the number of generated constraints in the frequency range is unknown, a two tiered system is used.
	CASE	Case identification number of the associated case, if appropriate
	SETID	**DCSID** Set identification number (if appropriate)
	LABEL	User label
	BOUND	When both **UPPER** and **LOWER** bounds are allowed, this field will signify which bound is represented.
	STATUS	One of **OBJECTIV**, **OK**, **VIOLATED** or **CRITICAL**. **OK** implies that the the constraint is well satisified. **CRITICAL** implies that it is satisfied but near its limit value. **VIOLATED** implies that the bound is not satisfied. **OBJECTIV** is used to denote that this entry is the user-selected objective function.
	STATE	One of **RET'D**, **AGE** = *intval*, or **INACTIV**. A **RET'D** constraint is one that is retained due to a simple application of the constraint screening algorithm. It is among the MAXRETAIN most critical constraints. **AGE** = *intval* implies that the constraint has been retained in the current approximate problem only because it was active in earlier iterations. *intval* denotes the number of times the constraint has been considered inactive but has still been retained. **INACTIV** implies that the constraint has been dropped from the active set, but is summarized here to show the dynamic of the screening process.
	VALUE	The current constraint value (normalized)
	PRED. VALUE	The value that the Approximate Problem predicted for the constraint at the current design point. **N/A** is used for the first summary.
	% DIFF.	The difference between the pred. value and value as a percentage of value.
	ABS. DIFF	The absolute difference between the pred. value and value.

Table 25-8. SPECIFIC CONSTRAINT SUMMARY ATTRIBUTES

Special elements	**DCGRID**	**1ST GID**	The **GID** of the first grid point participating in the constraint equation. For **DCGRID** there is only one, for **DCGRIDM** there may be any number, only the first is shown.
		COMP	The **COMP**onent of the first **GID** in the constraint equation.
	DCFREQ	**MODE**	The mode sequence number of the constrained mode
	DCMODEL	**TYPE**	One of **WEIGHT**, **VOLUME** or **MASS** for the type of model constraint applied.
	DCMODR	**MODE**	The mode sequence number of the constrained mode
		FORM	The form of the applied constraint: either **RMS** or **COMP**
		1ST GID	The **GID** of the first grid point participating in the constrained mode shape.
		COMP	The **COMP**onent of the first **GID** in the constraint.
	DCELEM	**ELEM-ID**	The identification number of the constrained element.
		COMP	The elemental response component that is constrained.
	DCDYNRG	**FREQ. RANGE**	One of two forms is used. 1. The particular frequency value associated with this **PEAK** response constraint. Note that the **CONID** field uses the two tiered numbering scheme. 2. The frequency range over which the **AVG** response is constrained.
		COMP	The **COMP**onent of the first **GID** in the constraints AND the label of the response component: **D** for displacement, **V** for velocity and **A** for acceleration.
Special Constraints	**PROPCON**		This table summarizes the violated and critical pseudo-side constraints. These constraints are internally generated as discussed in Section 25.3.3. Only the critical and violated constraints are echoed to save space.
		DVNAME	Name of the physical design variable
		CON. VALUE	Normalized constraint value
		PHYS. VALUE	The value of the physical design variable.
		PHYS. BOUND	The value of the physical bound.

Table 25-9. DESIGN CONSTRAINTS - I

a. Model Constraints

*** OBJECTIVE FUNCTION WEIGHT ***

RANK	CONID	SET-ID	LABEL	TYPE	BOUND	STATUS	STATE	VALUE	PRED. VALUE	% DIFF.	ABS. DIFF.
N/A	1	N/A		WEIGHT	N/A	OBJECTIV	RET'D	7.7335E+01	N/A	N/A	N/A

b. Pseudo-side Constraints

*** CRITICAL OR VIOLATED PHYSICAL VARIABLE BOUNDS (PROPCON) ***

RANK	CONID	SET-ID	DVNAME	BOUND	STATUS	STATE	CON. VALUE	PHYS. VALUE	PHYS. BOUND
10	400	N/A	QD409	LOWER	CRITICAL	RET'D	-3.3492E-05	5.2502E-02	5.2500E-02
9	402	N/A	QD410	LOWER	CRITICAL	RET'D	-3.3492E-05	5.2502E-02	5.2500E-02
8	542	N/A	SH417	LOWER	CRITICAL	RET'D	7.4506E-07	4.0000E-02	4.0000E-02

Table 25-10. DESIGN CONSTRAINTS - II

a. Displacement Constraints

*** ACTIVE DCGRID DISP RESPONSES ***

RANK	CONID	CASE	SET-ID	LABEL	1ST GID	COMP	BOUND	STATUS	STATE	VALUE	PRED. VALUE	% DIFF.	ABS. DIFF.
77	2	1	100		1	T3	UPPER	OK	RET'D	-2.7298E-01	-2.8201E-01	3.3075E+00	-9.0288E-03
76	3	1	100		2	T3	UPPER	OK	RET'D	-2.7298E-01	-2.8202E-01	3.3098E+00	-9.0352E-03
58	4	1	100		3	T3	UPPER	OK	RET'D	-2.2323E-01	-2.3347E-01	4.5870E+00	-1.0240E-02
57	5	1	100		4	T3	UPPER	OK	RET'D	-2.2323E-01	-2.3348E-01	4.5894E+00	-1.0245E-02
47	6	1	100		5	T3	UPPER	OK	RET'D	-1.7510E-01	-1.8623E-01	6.3559E+00	-1.1129E-02

b. Stress Constraints

*** ACTIVE DCELEM STRESS RESPONSES ***

RANK	CONID	CASE	SET-ID	LABEL	ELEM-ID	COMP	BOUND	STATUS	STATE	VALUE	PRED. VALUE	% DIFF.	ABS. DIFF.
N/A	45	1	100		98	MAXSHEAR	UPPER	OK	AGE = 1	-5.5092E-01	-5.6070E-01	1.7758E+00	-9.7832E-03
122	47	1	100		100	MAXSHEAR	UPPER	OK	RET'D	-4.5663E-01	-4.6939E-01	2.7930E+00	-1.2754E-02
119	48	1	100		101	MAXSHEAR	UPPER	OK	RET'D	-4.4605E-01	-4.6277E-01	3.7479E+00	-1.6718E-02
N/A	53	1	100		106	MAXSHEAR	UPPER	OK	AGE = 2	-5.8809E-01	-6.2776E-01	6.7453E+00	-3.9668E-02
123	54	1	100		107	MAXSHEAR	UPPER	OK	RET'D	-4.8565E-01	-5.3161E-01	9.4628E+00	-4.5956E-02

c. Strain Constraints

*** ACTIVE DCELEM STRAIN RESPONSES ***

RANK	CONID	CASE	SET-ID	LABEL	ELEM-ID	COMP	BOUND	STATUS	STATE	VALUE	PRED. VALUE	% DIFF.	ABS. DIFF.
98	67	1	100		1	EPSX	UPPER	OK	RET'D	-3.5256E-01	-3.9723E-01	1.2671E+01	-4.4673E-02
N/A	69	1	100		2	EPSY	UPPER	OK	AGE = 1	-5.3799E-01	-5.7813E-01	7.4616E+00	-4.0143E-02
N/A	70	1	100		2	EPSX	UPPER	OK	AGE = 3	-7.2495E-01	-7.6203E-01	5.1138E+00	-3.7072E-02
N/A	74	1	100		4	EPSX	UPPER	OK	AGE = 2	-6.5161E-01	-6.5916E-01	1.1583E+00	-7.5476E-03
N/A	82	1	100		8	EPSX	UPPER	OK	AGE = 2	-6.7328E-01	-6.9693E-01	3.5137E+00	-2.3657E-02
1	122	1	100		28	EPSY	UPPER	VIOLATED	RET'D	2.2975E-02	-2.8238E-05	1.0012E+02	-2.3004E-02
4	126	1	100		30	EPSY	UPPER	VIOLATED	RET'D	1.5915E-02	-1.8945E-02	2.1904E+02	-3.4860E-02
13	129	1	100		32	EPSY	UPPER	OK	RET'D	-2.8635E-02	-5.0554E-02	7.6546E+01	-2.1919E-02
6	142	1	100		38	EPSY	UPPER	VIOLATED	RET'D	3.7893E-03	-2.1675E-02	6.7200E+02	-2.5464E-02
N/A	144	1	100		39	EPSX	UPPER	OK	INACTIVE	-7.1347E-01	-7.1589E-01	3.3908E-01	-2.4192E-03

d. Element Force Constraints

*** ACTIVE DCELEM FORCE RESPONSES ***

RANK	CONID	CASE	SET-ID	LABEL	ELEM-ID	COMP	BOUND	STATUS	STATE	VALUE	PRED. VALUE	% DIFF.	ABS. DIFF.
2	2	1	100	TIPF	200	AXIAL	UPPER	CRITICAL	RET'D	-2.7676E-04	-2.7676E-04	0.0000E+00	0.0000E+00
1	3	1	100	TIPF	300	AXIAL	UPPER	CRITICAL	RET'D	-2.7676E-04	-2.7676E-04	0.0000E+00	0.0000E+00

most critical constraints but was within that threshold during the previous iteration. Constraint 53 was within the threshold two iterations ago. For now, however, they will be considered active.

Element Strain: Table 25-10c shows an example of the output for constraints from **DCELEM** entries of type **STRAIN**. Notice that constraint 122 is ***violated*** and, since its **RANK** is 1, it is currently the *largest, most violated* constraint. Notice also, that constraint 144 is **INACTIV**, denoting that it has fallen below the **MAXRETAIN** retention threshold enough times to be dropped. It will not be considered in the upcoming resizing step.

Element Force: Table 25-10d presents an example of the output for constraints from **DCELEM** entries of type **FORCE**.

Mode Shape: Table 25-11a gives an example of the output for constraints from **DCMODR** entries.

Frequency: Table 25-11b shows an example of the output for constraints from **DCFREQ** entries.

Grid Point Dynamic Response: Table 25-11c provides an example of the output for constraints from **DCDYNRG** entries. Notice that the **FREQ. RANGE** field has two distinct forms. The first form has a single frequency value and is associated with **PEAK** response constraints. The frequency value is the associated output frequency for this particular constraint. The **CONID** is multi-valued since the number of applied constraints cannot be determined until after the $FREQ_i$ data are processed. The second form has a pair of frequency values denoting the range, **FMIN** and **FMAX**, on the Bulk Data entry for this **AVG** response constraint.

Table 25-11. DESIGN CONSTRAINTS - III

a. Eigenvector Constraints

*** ACTIVE EIGENVECTOR DIFFERENCES (DCMODR) ***

RANK	CONID	CASE	SET-ID	LABEL	MODE FORM	1ST GID	COMP	STATUS	STATE	VALUE	PRED. VALUE	% DIFF.	ABS. DIFF.
2	4	1001	2000	MODE3	3 RMS	3	T1	VIOLATED	RET'D	3.9892E-02	-6.1732E-02	2.5475E+02	-1.0162E-01

b. Frequency Constraints

*** ACTIVE DCFREQ RESPONSES ***

RANK	CONID	CASE	SET-ID	LABEL	MODE BOUND	STATUS	STATE	VALUE	PRED. VALUE	% DIFF.	ABS. DIFF.
1	1	1001	2000	MODE1	1 LOWER	VIOLATED	RET'D	1.1448E-01	3.3721E-01	1.9457E+02	2.2273E-01
3	2	1001	2000	MODE1	1 UPPER	OK	RET'D	-1.1453E-01	-3.3725E-01	1.9446E+02	-2.2272E-01

c. Dynamic Response Constraints

*** ACTIVE GRID POINT DYNAMIC RESPONSES ***

RANK	CONIDS	CASE	SET-ID	LABEL	FREQ. RANGE	1ST GID	COMP	BOUND	STATUS	STATE	VALUE	PRED. VAL	ABS. DIFF
47	6/35	1000	1001	A110X	1.0400E+01	110	T1/A	UPPER	OK	RET'D	-4.112E-01	N/A	N/A
15	6/38	1000	1001	A110X	1.0700E+01	110	T1/A	UPPER	OK	RET'D	-1.679E-01	N/A	N/A
9	6/39	1000	1001	A110X	1.0800E+01	110	T1/A	UPPER	OK	RET'D	-9.328E-02	N/A	N/A
1	6/42	1000	1001	A110X	1.1100E+01	110	T1/A	UPPER	OK	RET'D	-2.569E-03	N/A	N/A
4	6/43	1000	1001	A110X	1.1200E+01	110	T1/A	UPPER	OK	RET'D	-1.870E-02	N/A	N/A
8	6/45	1000	1001	A110X	1.1400E+01	110	T1/A	UPPER	OK	RET'D	-8.088E-02	N/A	N/A
25	5/1	1000	1000	A110Z	7.000E+0/1.500E+1	110	T3/A	UPPER	OK	RET'D	-5.133E-01	N/A	N/A
26	6/1	1000	1001	A110X	7.000E+0/1.500E+1	110	T1/A	UPPER	OK	RET'D	-5.504E-01	N/A	N/A
27	7/1	1000	1001	A110Y	5.000E+0/1.000E+1	110	T2/A	UPPER	OK	RET'D	-7.088E-01	N/A	N/A

Table 25-12. DESIGN VARIABLE SUMMARY

a. Mathematical Design Variables

MATHEMATICAL DESIGN VARIABLE VALUES
DURING ANALYSIS 2

ASSIGNED ID	NAME	CURRENT VALUE	INITIAL VALUE	MINIMUM VALUE	MAXIMUM VALUE
1	ROD001	9.00000E+00	3.00000E+01	1.00000E-01	1.00000E+04
2	ROD002	9.00000E+00	3.00000E+01	1.00000E-01	1.00000E+04
3	ROD003	9.00000E+00	3.00000E+01	1.00000E-01	1.00000E+04
4	ROD004	9.00000E+00	3.00000E+01	1.00000E-01	1.00000E+04
5	ROD005	9.00000E+00	3.00000E+01	1.00000E-01	1.00000E+04
6	ROD006	9.00000E+00	3.00000E+01	1.00000E-01	1.00000E+04
7	ROD007	9.00000E+00	3.00000E+01	1.00000E-01	1.00000E+04
8	ROD008	9.00000E+00	3.00000E+01	1.00000E-01	1.00000E+04
9	ROD009	9.00000E+00	3.00000E+01	1.00000E-01	1.00000E+04
10	ROD010	9.00000E+00	3.00000E+01	1.00000E-01	1.00000E+04

b. Physical Design Variables

PHYSICAL DESIGN VARIABLE VALUES
DURING ANALYSIS 2

ASSIGNED NUMBER	NAME	TYPE	ID	SYMBOL	CURRENT VALUE	INITIAL VALUE	LOWER BOUND	UPPER BOUND
1	ROD001	PROD	1	A	9.00000E+00	3.00000E+01	1.00000E-01	1.00000E+04
2	ROD002	PROD	2	A	9.00000E+00	3.00000E+01	1.00000E-01	1.00000E+04
3	ROD003	PROD	3	A	9.00000E+00	3.00000E+01	1.00000E-01	1.00000E+04
4	ROD004	PROD	4	A	9.00000E+00	3.00000E+01	1.00000E-01	1.00000E+04
5	ROD005	PROD	5	A	9.00000E+00	3.00000E+01	1.00000E-01	1.00000E+04
6	ROD006	PROD	6	A	9.00000E+00	3.00000E+01	1.00000E-01	1.00000E+04
7	ROD007	PROD	7	A	9.00000E+00	3.00000E+01	1.00000E-01	1.00000E+04
8	ROD008	PROD	8	A	9.00000E+00	3.00000E+01	1.00000E-01	1.00000E+04
9	ROD009	PROD	9	A	9.00000E+00	3.00000E+01	1.00000E-01	1.00000E+04
10	ROD010	PROD	10	A	9.00000E+00	3.00000E+01	1.00000E-01	1.00000E+04

25.5.2.4 The Design Variables

There are two tables which give the values of the design variables at each iteration. The first, shown in Table 25-12a, details the values of each of the mathematical variables which are participating in the design. Each variable is described by its **ASSIGNED ID**, or reference number, its name as specified in the Bulk Data packet, its **CURRENT VALUE** for this design iteration, its **INITIAL VALUE** at the start of the design process, and the overall bounds, called **MAXIMUM VALUE** and **MINMUM VALUE**, as determined automatically by **UAI/*NASTRAN***.

The physical design variable values are also presented as shown in Table 25-12b. Again, the description includes **ASSIGNED ID**, or reference number, and its name as specified in the Bulk Data packet. The specific reference to a property, material or GRID point is then given. This is then followed by its **CURRENT VALUE** for this design iteration, its **INITIAL VALUE** at the start of the design process, and the physical side constraints, called **LOWER BOUND** and **UPPER BOUND**, that were defined by the Bulk Data.

25.5.2.5 The Approximate Optimization Summary

The results of the solution of the Approximate Optimization Problem are summarized by the data shown in Table 25-13. It includes information regarding the status of the two criteria for approximate problem convergence. It shows the numerical values of the **CURRENT VALUE** of

Table 25-13. APPROXIMATE PROBLEM SUMMARY

```
****              APPROXIMATE OPTIMIZATION SUMMARY              ****
***                        ITERATION  1                         ***
**    RESIZING METHOD          = MATHEMATICAL PROGRAMMING         **
*                                                                 *
*     CRITERION 1:  OBJECTIVE CHANGE                              *
*     UPPER BOUND PERCENT MOVE =      .600000 PERCENT             *
*                    CURRENT VALUE        =   2.5903E+03          *
*                    PREVIOUS VALUE       =   2.6768E+03          *
*                    DELTA                =  -8.6484E+01          *
*                    PERCENT MOVE         =     -3.2309           *
*     CRITERION 2:  DIRECT DESIGN VECTOR MOVE                     *
*     UPPER BOUND PERCENT MOVE =     1.800000 PERCENT             *
*                    NORM OF X-X0         =   1.7792E+02          *
*                    EUCLIDEAN NORM OF X0 =   5.9450E+03          *
*                    PERCENT MOVE         =      2.9928          *
*                                                                 *
*                    SCALED DESIGN VECTOR MOVE                    *
*                    NORM OF V-V0         =   5.9327E-01          *
*                    EUCLIDEAN NORM OF V0 =   6.0000E+00          *
*                    PERCENT MOVE         =      9.8879          *
*                                                                 *
*     THE APPROXIMATE PROBLEM IS NOT CONVERGED                    *
*******************************************************************
```

the approximate objective function, the **PREVIOUS VALUE** of the ***exact*** objective function and the absolute and percentage changes. It shows the numerical value of the **CONVERGE**-based **UPPER BOUND PERCENT MOVE** of the objective function.

The second criteria is summarized in a similar manner with the **NORM OF X-X0** being the Euclidean Norm of the $\delta \boldsymbol{X}$ vector, **EUCLIDEAN NORM OF X0** is the norm of the $\boldsymbol{X}$ vector about which the approximations were built and the **PERCENT MOVE** is the percentage ratio. Again, the numerical value of the **CONVERGE**-based **UPPER BOUND PERCENT MOVE** is shown as the limiting value. Finally, the overall approximate problem convergence status is shown. This status is derived from these two criteria.

25.5.2.6 The Design History

The design history, shown in Table 25-14, includes the **ANALYSIS NUMBER**, the **OBJECTIVE VALUE** computed during the analysis cycle, and the **PREDICTED VALUE** of the objective from

Table 25-14. ITERATION HISTORY

O P T I M I Z A T I O N H I S T O R Y

CONVERGED SOLUTION REACHED AFTER 17 ANALYSES

ANALYSIS	OBJECTIVE VALUE	PREDICTED VALUE	MAXIMUM CONSTRAINT VALUE	MAXIMUM CONSTRAINT IDENTIFICATION	APPROXIMATE CONVERGENCE
1	1.25894E+04	N/A	-7.27153E-01	4	NOT CONVERGED
2	3.77682E+03	3.77703E+03	-9.05111E-02	4	NOT CONVERGED
3	1.93410E+03	1.93412E+03	1.66502E-02	2	NOT CONVERGED
4	1.80119E+03	1.80119E+03	1.09689E-02	2	NOT CONVERGED
5	1.71411E+03	1.71411E+03	3.88438E-04	4	NOT CONVERGED
6	1.68508E+03	1.68507E+03	2.28437E-04	4	NOT CONVERGED
7	1.65889E+03	1.65889E+03	8.51641E-04	8	NOT CONVERGED
8	1.63868E+03	1.63868E+03	1.72547E-03	8	NOT CONVERGED
9	1.62704E+03	1.62704E+03	2.84844E-03	8	NOT CONVERGED
10	1.61912E+03	1.61912E+03	3.29508E-03	8	NOT CONVERGED
11	1.61236E+03	1.61236E+03	2.88273E-03	8	NOT CONVERGED
12	1.60727E+03	1.60727E+03	2.12563E-03	8	NOT CONVERGED
13	1.60325E+03	1.60325E+03	2.21156E-03	8	NOT CONVERGED
14	1.59982E+03	1.59982E+03	1.93156E-03	8	NOT CONVERGED
15	1.59712E+03	1.59712E+03	1.71320E-03	8	NOT CONVERGED
16	1.59493E+03	1.59493E+03	1.59055E-03	8	NOT CONVERGED
17	1.59350E+03	1.59350E+03	4.90625E-04	8	CONVERGED

the previous iteration's approximation to the objective function. It also identifies for each design cycle, the **MAXIMUM CONSTRAINT VALUE** and the **MAXIMUM CONSTRAINT IDENTIFICATION**, which specifies a reference number of the constraint having this value. Finally, it includes the status of the **APPROXIMATE CONVERGENCE**.

25.5.2.7 ARCHIVE Database

As indicated in 25.4.2.4, the MDO process generates large volumes of output data which include: all of the analysis solution results and design data for every iteration; and all sensitivity data for every iteration. If you have used one or more **ARCHIVE** commands in your design job, then a number of relations will be placed on your ARCHIVE database. The schemata of these data are documented in the ***eBase Archive Schemata Description Manual***. You may then query the results and create plots using the ***eShell*** Interactive Interface.

25.6 EXAMPLE PROBLEMS

This section provides you with nine examples of design problems which may be solved with the MDO capability. These include design problems which encompass statics, normal modes and frequency response analyses using both property and geometric design variables. The specific design features illustrated in each of the example problems is summarized in the following table.

FEATURE CLASS	FEATURE	EXAMPLE PROBLEM								
		1a	1b	2	3a	3b	4	5	6	7
DISCIPLINE	Statics	✔	✔	✔			✔	✔		
	Normal Modes				✔	✔			✔	✔
	Direct Frequency Response									✔
MODEL TYPE	Discrete	✔	✔				✔			✔
	Continuum			✔	✔	✔		✔	✔	
DESIGN VARIABLES	Element Properties	✔	✔	✔	✔	✔	✔		✔	
	GRID Point Locations						✔	✔		✔
VARIABLE LINKING	Shared Property	✔	✔		✔	✔	✔		✔	
	Property Linking				✔	✔	✔	✔		✔
	Reduced Basis Linking			✔						
DESIGN CONSTRAINTS	Displacement		✔	✔						
	Element Stress	✔	✔	✔			✔			
	Element Strain			✔						
	Element Force							✔		
	Frequency				✔	✔			✔	✔
	Eigenvector								✔	
OBJECTIVE FUNCTION	Weight Minimization	✔		✔	✔	✔	✔			
	Volume Minimization							✔		
	Eigenvector Difference Minimization								✔	
	Frequency Response Minmization									✔

☞ The design problems solved in this section were executed on an IBM RS/6000 computer. The nature of design is such that answers will differ when run on other computers.

25

25.6.1 Ten Bar Truss

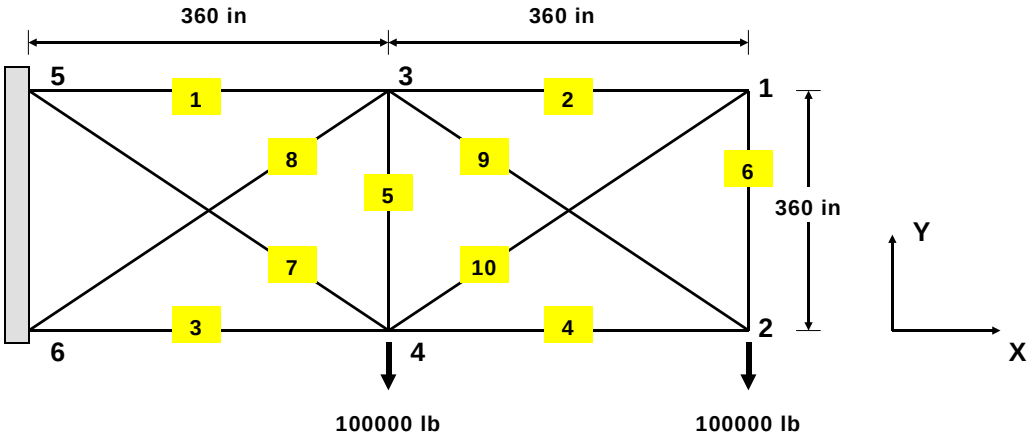
This section provides you with two examples of design optimization of a structure subject to static response constraints. The first example illustrates the use of stress constraints and the second the use of both displacement and stress constraints.

The first two example problems are based on the classic Ten Bar Truss problem [Haftka90] used in optimization texts and as a validation of optimization software. They illustrate the following MDO features:

- ROD elements
- Point loads
- Stress constraints and combined stress and displacement constraints
- Weight minimization

Example Problem 25-1a
A ten bar truss, fixed at one end, is subjected to two point loads of 100,000 lb in the downward y-direction. You wish to determine the cross sectional areas of the members that will minimize the truss weight while insuring that the no member stress exceeds 25,000 psi.

Figure 25-4. 10-BAR TRUSS DESIGN MODEL



LOADING CONDITION	
GRID	F_y (lb)
2	-100000
4	-100000

PROPERTIES

$E = 1.0 \times 10^7$ psi
$\rho = 0.1$ lb/in^3

CONSTRAINTS

Stress: $\pm 2.5 \times 10^4$ psi

Problems 25-1a and 25-1b

Displacement: ±2.0 in

At GRID points 1,2,3 and 4: Problem 25-1b only

The finite element model and initial physical characteristics are shown in Figure 25-4. You may find this model in file **MDO1A**. The model is comprised of 10 **ROD** elements. Each element references a separate property so that ten design variables, each representing the cross sectional area of a ROD element, may be designed. The objective is to minimize the weight of the structure while satisfying the specified stress constraints. The Case Control commands are straight-forward:

```
OBJECTIVE MINIMIZE WEIGHT
   MAXITER  = 20
   CONVERGE = 0.25
CASE 1 STATICS
   SPC    = 1
   LOAD   = 1
   SET 1  = 100
   DESCON = 1
```

The **OBJECTIVE** command specifies the type of optimization to be performed. The two subcommands, **MAXITER** and **CONVERGE**, are used to force the solution through many iterations. This is done because of the academic nature of this problem. The analysis **CASE** is then given. For this problem, there is a single static analysis with one loading condition.

To solve this problem, you must specify ten design variables which represent each of the cross sectional areas. The Bulk Data entries required to do this are:

DVPROP	**ROD001**	**PROD**	**1**	**0.1**	**10000.**	**A**			
DVPROP	**ROD002**	**PROD**	**2**	**0.1**	**10000.**	**A**			
...									
DVPROP	**ROD009**	**PROD**	**9**	**0.1**	**10000.**	**A**			
DVPROP	**ROD010**	**PROD**	**10**	**0.1**	**10000.**	**A**			

In order to constrain each element so that its stress is between -25000 psi and 25000 psi, you must use the **DCELEM** Bulk Data entry. For example, to constrain the first element you could use:

DCELEM	**100**	**SIGX1T**	**STRESS**	**SIGX**		**25000.**		**1**	
DCELEM	**100**	**SIGX1C**	**STRESS**	**SIGX**	**-25000.**			**1**	

which defines two constraints, one on the tensile stress, **SIGX1T**, and one on the compressive stress, **SIGX1C**. This results in 20 lines of input data and 20 constraints during optimization. Alternately, because the **DCELEM** entry allows both an upper and lower bound, a single entry:

DCELEM	**100**	**SIGX1T**	**STRESS**	**SIGX**	**-25000.**	**25000.**		**1**	

may be used. In this case, the constraint values are more difficult to identify in the printed output. This reduces the number of lines of input data to 10, but still results in 20 constraints during optimization.

Because the Von Mises stress for a ROD element is simply the unsigned value of the axial stress, a single **VONMISES** constraint can be used instead of the two axial stress constraints:

DCELEM	**100**	**VONM1**	**STRESS**	**VONMISES**		**25000.**		**1**	

This both reduces the lines of input and the number of design constraints to 10. The final option is to specify all of the stress constraints by using a range of element identification numbers on the **DCELEM** entry:

DCELEM	100	VONM	STRESS	VONMISES		25000.		1	+A
+A	THRU	10							

While still resulting in 10 design constraints, the input has been reduced to only two lines!

The example problem is then executed in **UAI/*NASTRAN***. The optimization took 17 iterations. The final values for the cross sectional areas are shown in Table 25-15a along with results from other sources.

Example Problem 25-1b

The same ten bar truss is designed to determine the cross sectional areas of the members that will minimize the truss weight while insuring that the no member stress exceeds 25,000 psi and that the displacement in the y-direction at GRID points 1, 2, 3, and 4 does not exceed 2.0 in.

The finite element model and initial physical characteristics are as before. This model is found in file **MDO1B**. The data are identical to example 25-1a except that new constraints are defined for the displacements at the specified GRID points. This is done with the following **DCGRID** Bulk Data entries.

DCGRID	100	NODE1	DISP	T2	-2.0	2.0		1	
DCGRID	100	NODE2	DISP	T2	-2.0	2.0		2	
DCGRID	100	NODE3	DISP	T2	-2.0	2.0		3	
DCGRID	100	NODE4	DISP	T2	-2.0	2.0		4	

Note that the **DCGRID** data, like the **DCELEM** data, has many options for reducing the time for preparing input data. The method that you use for data entry depends on the type of solution output that you wish to see. In particular, the constraint values are labelled based on the names assigned on the Bulk Data entries.

The job is then executed. The results from **UAI/*NASTRAN*** and other references are shown in Table 25-15b.

Table 25-15a. TEN BAR TRUSS WITH STRESS CONSTRAINTS

DESIGN VARIABLE OBJECTIVE	UAI/*NASTRAN*	[Haftka90]	[Johnson88]
1	7.94	7.94	7.89
2	0.10	0.10	0.15
3	8.06	8.06	8.10
4	3.94	3.94	3.90
5	0.10	0.10	0.10
6	0.10	0.10	0.15
7	5.74	5.74	5.80
8	5.57	5.57	5.50
9	5.57	5.57	5.50
10	0.11	0.10	0.21
WEIGHT	1593.50	1593.20	1598.50

Table 25-15b. TEN BAR TRUSS — STRESS AND DISPLACEMENT CONSTRAINTS

DESIGN VARIABLE OBJECTIVE	UAI/*NASTRAN*	[Haftka90]	[Johnson88]
1	30.28	30.52	30.54
2	0.38	0.10	0.37
3	24.01	23.20	23.96
4	14.74	15.22	14.73
5	0.10	0.10	0.10
6	0.38	0.55	0.38
7	8.54	7.46	8.54
8	21.10	21.04	21.01
9	20.98	21.53	20.88
10	0.32	0.10	0.35
WEIGHT	5108.55	5060.85	5109.15

25.6.2 Intermediate Complexity Wing

This example problem illustrates the design of structures using reduced basis design variable linking. It features:

- ❒ Reduced Basis Linking of Physical Properties
- ❒ A Built-up Structure
- ❒ Strain Response Constraints
- ❒ Stress Response Constraints
- ❒ Tip Deflection Constraints
- ❒ Weight Minimization

Example Problem 25-2

This example problem is based on a model used frequently in the aerospace design optimization literature [Johnson88]. This particular version minimizes the weight of an aluminum wing box modelled with bending plate elements. The substructure is composed of shear panels and rod elements. Reduced basis linking is used to create design variables that represent tapers. For the wing skin, these are: constant; linear in the x-coordinate of the element centroid; linear in the y-coordinate of the element centroid; and quadratic in the y-coordinate of the element centroid. For the spars, they are constant and linear-y tapers. Both the strains in the wing skin and the spar web stresses are constrained. A tip deflection limit is also imposed for two load cases which simulate a subsonic and supersonic air load.

The geometry of this model is shown in Figure 25-5. This is a cut-away view showing the bottom wing skin, which is modelled with 32 QUAD4 elements, and the substructure, which includes 23 spar webs. The top wing skin has the same model as the bottom. You may find this model in file **MDO2**. The Case Control commands are similar to those already used:

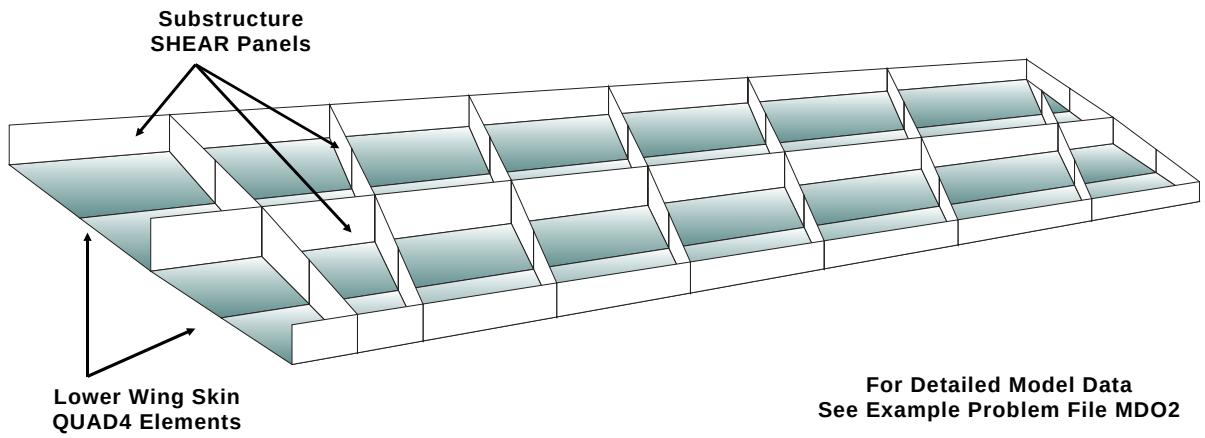
```
TITLE = INTERMEDIATE COMPLEXITY WING
SUBTITLE = QUAD4 ELEMENTS/REDUCED BASIS LINKING
SPC = 1
AUTOSPC ( SPC ) = YES
OBJECTIVE MINIMIZE WEIGHT
  MAXITER=15
$
MAXRETAIN = 50
SET 100 = 100
DISP(PRINT) = ALL
DESCON    = 100
CASE 1 STATICS
   LABEL = SUBSONIC LOAD CONDITION
   LOAD = 1
CASE 2 STATICS
   LABEL = SUPERSONIC LOAD CONDITION
   LOAD = 2
```

The **DESCON** command option appears above the **CASE** level and therefore applies to both static load conditions.

To solve this problem, you could independently design each of the element thickness or area properties. That would result in 158 physical design variables. This model, however, is not refined enough to warrant that level of detail. Instead, as a preliminary or conceptual designer, one chooses to ignore the caps and posts and ignore the ribs altogether and "design" only the wing skins and the spar webs. These are the design variables in the literature for this test case.

After this simplification, there are 87 properties that are design variables: 64 wing skins (32 on each surface) and 23 spar webs. Now suppose that the upper and lower surfaces are linked together so that the upper and lower skins are identical. That reduces the problem to 55 variables. In this case, however, we know that the final design will be a smoothly varying thickness over the skins and spar webs. To reduce computational effort and to eliminate design degrees of freedom that are not really present (since you will not be building the

Figure 25-5. INTERMEDIATE COMPLEXITY WING MODEL



structure in "elements."), reduced basis design variables will be used.

The variables selected depend on how an acceptable design is defined. In this case, smooth tapers over the geometry with enough different tapers to allow the optimizer to place the minimum and maximum thicknesses anywhere on the skin will be allowed. Using experience, however, we limit the spar webs to be linearly varying — it can have one minimum and one maximum and those will be at the extremes of the geometry. Mathematically, we will allow the following:

$$t_{skin} = t_c + t_x^1 + t_x^2 + t_x^3$$

$$t_{spar} = t_c + t_x^1$$

and

$$t_x^n = (x_{centroid})^n$$

These vectors of "shapes" are normalized such that the largest value is unity and the data are entered using **DVLINK** Bulk Data entries as the coefficients relating the mathematical and physical variables. First, the 87 physical variables are defined: 23 SHEAR panels for the spar webs and 64 QUAD4 elements for the (linked) skins:

DVPROP	**SH402**	**PSHEAR**	**40002**	**0.040**		**T**			
DVPROP	**SH403**	**PSHEAR**	**40003**	**0.040**		**T**			
...									
DVPROP	**SH423**	**PSHEAR**	**40023**	**0.040**		**T**			
DVPROP	**SH424**	**PSHEAR**	**40024**	**0.040**		**T**			
DVPROP	**TR301**	**PSHELL**	**10001**	**0.0525**		**T**			
DVPROP	**TR302**	**PSHELL**	**12001**	**0.0525**		**T**			
...									
DVPROP	**QD463**	**PSHELL**	**30031**	**0.0525**		**T**			
DVPROP	**QD464**	**PSHELL**	**32031**	**0.0525**		**T**			

Now, having the physical design variables, the shape coefficients are defined using **DVLINK** entries. Beginning with the uniform thicknesses for each spar web and for the skins, define the **SPRFUNI**,**SPRMUNI**,**SPRAUNI** (**SP**a**R**,**F**orward/**M**id/**A**ft,**UNI**form Taper) variables and the **M11** (Main Box, constant-x, constant-y) variable.

DVLINK	**SPRFUNI**	**0.15**	**0.0**	**1.0000**	**SH402**	**0.0**	**1.0000**	**SH403**	
		0.0	**1.0000**	**SH404**	**0.0**	**1.0000**	**SH405**		
		0.0	**1.0000**	**SH406**	**0.0**	**1.0000**	**SH407**		
		0.0	**1.0000**	**SH408**					
DVLINK	**SPRMUNI**	**0.15**	**0.0**	**1.0000**	**SH409**	**0.0**	**1.0000**	**SH410**	
		0.0	**1.0000**	**SH411**	**0.0**	**1.0000**	**SH412**		
		0.0	**1.0000**	**SH413**	**0.0**	**1.0000**	**SH414**		
		0.0	**1.0000**	**SH415**	**0.0**	**1.0000**	**SH416**		
DVLINK	**SPRAUNI**	**0.15**	**0.0**	**1.0000**	**SH417**	**0.0**	**1.0000**	**SH418**	
		0.0	**1.0000**	**SH419**	**0.0**	**1.0000**	**SH420**		
		0.0	**1.0000**	**SH421**	**0.0**	**1.0000**	**SH422**		

		0.0	1.0000	SH423	0.0	1.0000	SH424		
DVLINK	M11	0.15	0.0	1.0000	TR301	0.0	1.0000	TR302	
		0.0	1.0000	QD403	0.0	1.0000	QD404		
		0.0	1.0000	QD405	0.0	1.0000	QD406		
		0.0	1.0000	QD407	0.0	1.0000	QD408		
...									
		0.0	1.0000	QD459	0.0	1.0000	QD460		
		0.0	1.0000	QD461	0.0	1.0000	QD462		
		0.0	1.0000	QD463	0.0	1.0000	QD464		

The linear taper in the y-coordinate direction is then defined. Again there are four mathematical variables; three for the spar webs and one for the skins: **SPRFLINY**,**SPRMLINY**,**SPRALINY** (**SP**a**R**,**F**orward/**M**id/**A**ft,**LIN**ear-in-**Y** Taper) variables and the **M14** (Main Box, constant-x, linear-y) variable.

DVLINK	SPRFLINY	0.0	0.0	1.0000	SH402	0.0	0.8529	SH403	
		0.0	0.7058	SH404	0.0	0.5588	SH405		
		0.0	0.4117	SH406	0.0	0.2536	SH407		
		0.0	0.0845	SH408					
DVLINK	SPRMLINY	0.0	0.0	1.0000	SH409	0.0	0.8975	SH410	
		0.0	0.7513	SH411	0.0	0.6051	SH412		
		0.0	0.4589	SH413	0.0	0.3128	SH414		
		0.0	0.1793	SH415	0.0	0.0595	SH416		
DVLINK	SPRALINY	0.0	0.0	1.0000	SH417	0.0	0.8586	SH418	
		0.0	0.7019	SH419	0.0	0.5452	SH420		
		0.0	0.3886	SH421	0.0	0.2319	SH422		
		0.0	0.1152	SH423	0.0	0.0384	SH424		
DVLINK	M14	0.00	0.0	1.01222	TR301	0.0	1.01222	TR302	
		0.0	1.00000	QD403	0.0	1.00000	QD404		
		0.0	0.98484	QD405	0.0	0.98484	QD406		
		0.0	0.96912	QD407	0.0	0.96912	QD408		
...									
		0.0	0.07515	QD457	0.0	0.07515	QD458		
		0.0	0.06445	QD459	0.0	0.06445	QD460		
		0.0	0.05356	QD461	0.0	0.05356	QD462		
		0.0	0.04248	QD463	0.0	0.04248	QD464		

Note that the upper and lower surfaces share the same coefficients. Finally, the linear taper in the x-coordinate direction is defined along with the y^2 shape. These variables apply only to the elements in the wing box: **M12** (Main Box, linear-x, constant-y) and **M17** (Main Box, constant-x, quadratic-y) variable.

DVLINK	M12	0.00	0.0	0.69354	TR301	0.0	0.69354	TR302	
		0.0	0.79929	QD403	0.0	0.79929	QD404		
		0.0	0.89877	QD405	0.0	0.89877	QD406		
		0.0	1.00000	QD407	0.0	1.00000	QD408		
		0.0	0.65456	QD409	0.0	0.65456	QD410		
		0.0	0.74889	QD411	0.0	0.74889	QD412		
...									
		0.0	0.45616	QD461	0.0	0.45616	QD462		

		0.0	0.62033	QD463	0.0	0.62033	QD464		
DVLINK	M17	0.00	0.0	1.024575	TR301	0.0	1.024575	TR302	
		0.0	1.00000	QD403	0.0	1.00000	QD404		
		0.0	0.96992	QD405	0.0	0.96992	QD406		
		0.0	0.93919	QD407	0.0	0.93919	QD408		
...									
		0.0	0.00565	QD457	0.0	0.00565	QD458		
		0.0	0.00415	QD459	0.0	0.00415	QD460		
		0.0	0.00287	QD461	0.0	0.00287	QD462		
		0.0	0.00180	QD463	0.0	0.00180	QD464		

The stress, strain and displacement constraints are applied using:

DCELEM	100		STRESS	MAXSHEAR		4.5+4		65	THRU
DCELEM	100		STRAIN	EPSX		4.5-3		1	THRU
DCELEM	100		STRAIN	EPSY		4.5-3		1	THRU
DCGRID	100		DISP	T3		10.0		1	
	2	THRU	10						

The job is then run in **UAI/*NASTRAN*** and convergence is achieved in five design cycles (six analysis iterations). The design iteration history is shown in Table 25-16a and the final mathematical design variables are shown in Table 25-16b. Notice that the initial design used only the uniform thicknesses (initially, the taper design variables are 0.0). Notice also that the final design is almost the same weight as the initial design but the maximum constraint value is now 0.0 instead of 1.3. Since these data are normalized, this means that the initial design violated constraint 193 by 130%!

The final physical thicknesses are derived from the linking relationship. Since each **DVLINK** contains the non-zero terms of a column of the linking matrix $\boldsymbol{T}$ in:

$$\boldsymbol{v}_p = \boldsymbol{v}_p^{inv} + \boldsymbol{T}\,\boldsymbol{v}_m \tag{25-11}$$

You may manually compute the final thickness using the final mathematical variables and the input coefficients. For example, the QUAD4 405 element thickness appears in each of the four **DVLINK** entries associated with the skins:

$$t_{405} = 1.0 \cdot \mathbf{M11} + 0.89877 \cdot \mathbf{M12} + 0.98484 \cdot \mathbf{M14} + 0.96992 \cdot \mathbf{M17}$$

For the final values of M_{ij}, we obtain a thickness of 0.08777. Table 25-16c shows some of the physical design variable values. **UAI/*NASTRAN*** automatically computes these data at each iteration. Figures 25-6 and 25-7 present these data graphically. You can see that the thicknesses vary smoothly over the region.

For complex models where cutouts, load introduction points and other features can cause local concentrations of load, **DVLINK** can still be used to help assure a manufacturable design. Simply add more smooth shapes around the region of the concentration and let the optimizer pick how much of the global and local shapes to use. In general, the optimizer will use the local shape to address the local problem since it has the least severe consequences for other constraints and the objective function.

Table 25-16. INTERMEDIATE COMPLEXITY WING — DESIGN RESULTS

a. Design History

O P T I M I Z A T I O N H I S T O R Y

CONVERGED SOLUTION REACHED AFTER 6 ANALYSES

ANALYSIS	OBJECTIVE VALUE	PREDICTED VALUE	MAXIMUM CONSTRAINT VALUE	MAXIMUM CONSTRAINT IDENTIFICATION	APPROXIMATE CONVERGENCE
1	1.51258E+02	N/A	1.29957E+00	193	NOT CONVERGED
2	1.37359E+02	1.37357E+02	4.10402E-01	66	NOT CONVERGED
3	1.81147E+02	1.81147E+02	9.98190E-02	66	NOT CONVERGED
4	1.45810E+02	1.45808E+02	1.61430E-01	122	NOT CONVERGED
5	1.50613E+02	1.50613E+02	2.29753E-02	122	NOT CONVERGED
6	1.51645E+02	1.51645E+02	7.08530E-04	122	CONVERGED

b. Summary of Mathematical Design Variables

M A T H E M A T I C A L D E S I G N V A R I A B L E V A L U E S
D U R I N G A N A L Y S I S 6

ASSIGNED ID	NAME	CURRENT VALUE	INITIAL VALUE	MINIMUM VALUE	MAXIMUM VALUE
1	M11	2.86813E-01	1.50000E-01	-1.00000E+20	1.00000E+20
2	M12	1.50777E-01	0.00000E+00	-1.00000E+20	1.00000E+20
3	M14	-6.62772E-01	0.00000E+00	-1.00000E+20	1.00000E+20
4	M17	3.28033E-01	0.00000E+00	-1.00000E+20	1.00000E+20
5	SPRALINY	-5.93500E-02	0.00000E+00	-1.00000E+20	1.00000E+20
6	SPRAUNI	9.93499E-02	1.50000E-01	-1.00000E+20	1.00000E+20
7	SPRFLINY	1.09323E-06	0.00000E+00	-1.00000E+20	1.00000E+20
8	SPRFUNI	4.16043E-02	1.50000E-01	-1.00000E+20	1.00000E+20
9	SPRMLINY	-1.97233E-05	0.00000E+00	-1.00000E+20	1.00000E+20
10	SPRMUNI	9.10384E-02	1.50000E-01	-1.00000E+20	1.00000E+20

c. Summary of Physical Design Variables

P H Y S I C A L D E S I G N V A R I A B L E V A L U E S
D U R I N G A N A L Y S I S 6

ASSIGNED ID	NAME	TYPE	ID	SYMBOL	CURRENT VALUE	INITIAL VALUE	LOWER BOUND	UPPER BOUND
1	QD403	PSHELL	30001	T	7.25879E-02	1.50000E-01	5.25000E-02	1.00000E+09
2	QD404	PSHELL	32001	T	7.25879E-02	1.50000E-01	5.25000E-02	1.00000E+09
3	QD405	PSHELL	30002	T	8.77676E-02	1.50000E-01	5.25000E-02	1.00000E+09
4	QD406	PSHELL	32002	T	8.77676E-02	1.50000E-01	5.25000E-02	1.00000E+09
5	QD407	PSHELL	30003	T	1.03369E-01	1.50000E-01	5.25000E-02	1.00000E+09
6	QD408	PSHELL	32003	T	1.03369E-01	1.50000E-01	5.25000E-02	1.00000E+09
7	QD409	PSHELL	30004	T	5.24986E-02	1.50000E-01	5.25000E-02	1.00000E+09
8	QD410	PSHELL	32004	T	5.24986E-02	1.50000E-01	5.25000E-02	1.00000E+09
9	QD411	PSHELL	30005	T	6.84973E-02	1.50000E-01	5.25000E-02	1.00000E+09
10	QD412	PSHELL	32005	T	6.84973E-02	1.50000E-01	5.25000E-02	1.00000E+09
...								
54	QD456	PSHELL	32027	T	3.06336E-01	1.50000E-01	5.25000E-02	1.00000E+09
55	QD457	PSHELL	30028	T	2.58737E-01	1.50000E-01	5.25000E-02	1.00000E+09
56	QD458	PSHELL	32028	T	2.58737E-01	1.50000E-01	5.25000E-02	1.00000E+09
57	QD459	PSHELL	30029	T	2.89689E-01	1.50000E-01	5.25000E-02	1.00000E+09
58	QD460	PSHELL	32029	T	2.89689E-01	1.50000E-01	5.25000E-02	1.00000E+09
59	QD461	PSHELL	30030	T	3.21035E-01	1.50000E-01	5.25000E-02	1.00000E+09
60	QD462	PSHELL	32030	T	3.21035E-01	1.50000E-01	5.25000E-02	1.00000E+09
61	QD463	PSHELL	30031	T	3.52780E-01	1.50000E-01	5.25000E-02	1.00000E+09
62	QD464	PSHELL	32031	T	3.52780E-01	1.50000E-01	5.25000E-02	1.00000E+09

Figure 25-6. WING SKIN — FINAL DESIGN

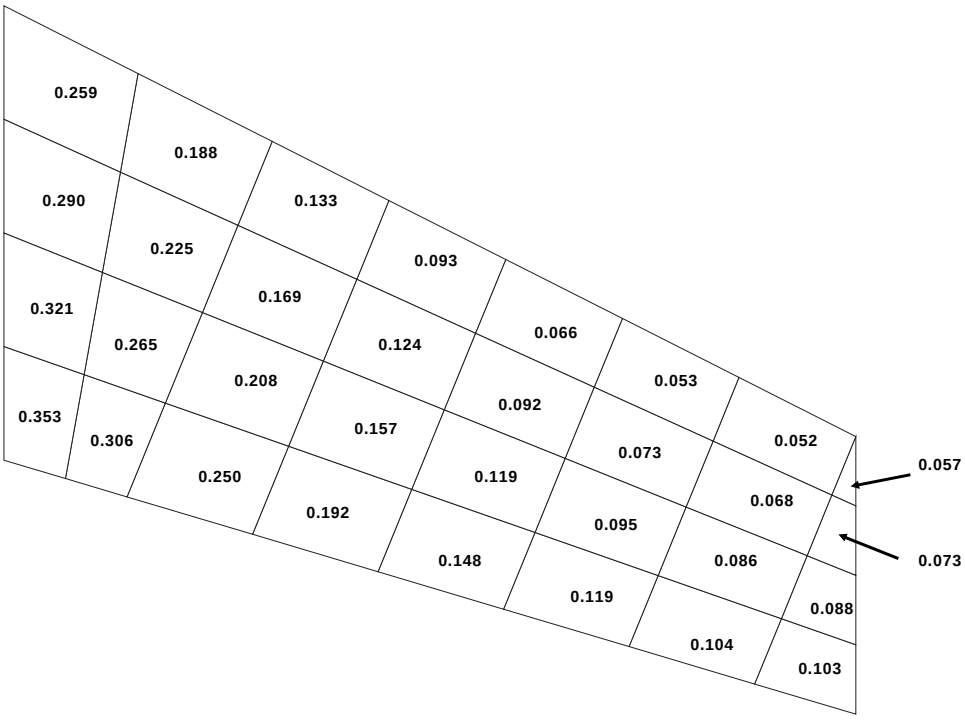
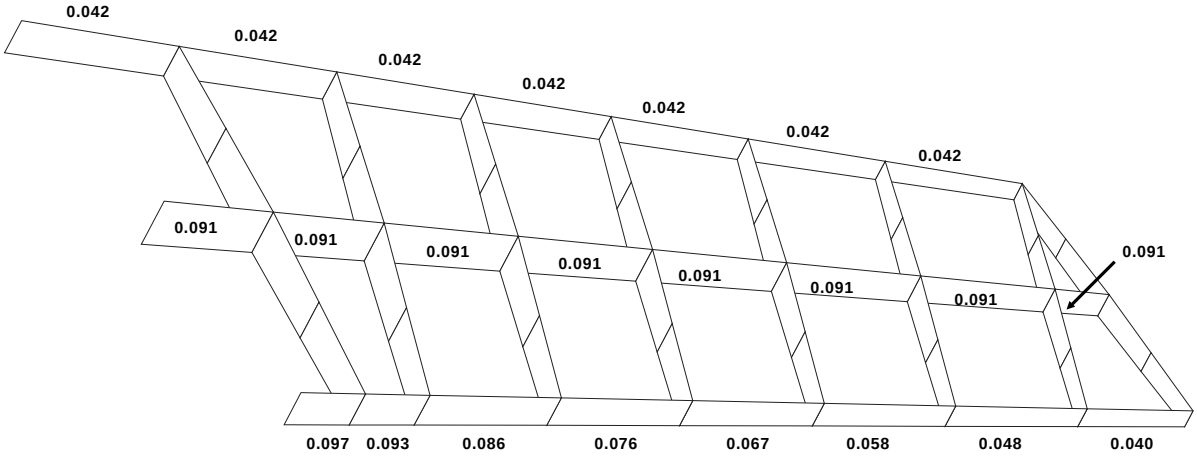


Figure 25-7. SUBSTRUCTURE — FINAL DESIGN



25.6.3 Turner's Problem

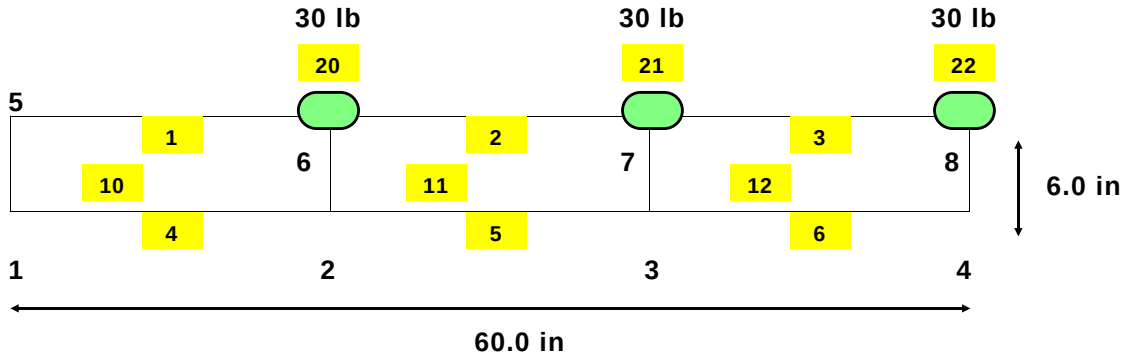
This section presents a simple, but useful, example, and a variant, which show how structures may be designed to satisfy constraints placed on their natural frequencies. They illustrate the following MDO features:

- ❒ Tapered BEAM elements
- ❒ Modal analysis with frequency constraints
- ❒ Nonstructural (undesigned) mass
- ❒ Weight minimization

Example Problem 25-3
An aluminum beam of constant depth supports three nonstructural masses as shown in Figure 25-8. You wish to determine the web thickness and chord member areas to obtain a natural frequency of 20 Hz in the fundamental mode while minimizing the weight of the structure.

The geometry and finite element model are shown in Figure 25-8. You may find this model in file **MDO3A**. The web is modeled with 3 QUAD4 elements and the chord with 6 BEAM elements, three along the upper edge (elements **1**, **2**, and **3**), and three along the bottom edge (elements **4**, **5**, and **6**) . The nonstructural masses were placed on the upper edge of the beam and are defined using **CONM2** Bulk Data entries.

Figure 25-8. TURNER'S PROBLEM — DESIGN MODEL



CONSTRAINT	PROPERTIES
First Mode Frequency ≤ 20.0 Hz	$E = 1.03 \times 10^7$ psi $\rho = 0.1$ lb/in^3

Each **QUAD4** element references a separate **PSHELL** Bulk Data entry which specifies the original thickness value. Turner's solution [Turner67] assumes that the chord is tapered. To model this, BEAM elements with tapered cross sections were used. For example, the properties of the first element were defined with:

PBEAM1	1	101	ROD	0.7854					+PB1
+PB1			2	0.7854					

A **ROD** shaped cross-section is used, and a diameter of **0.7854** in. is used to correspond to an initial area of 1.0 in^2. This value is used at both End A and End B of the BEAM. Next, it is required to link the adjoining beam cross-sections to ensure geometric continuity. This is done using the following linking relationships (which also include the unlinked design variable definitions):

DVPROP	A1A	PBEAM1	1	0.01	100	D1A			
DVPROPS	A1B2A	PBEAM1	1	0.01	100	D1B			+A1B2A
+A1B2A		PBEAM1	2	0.01	100	D1A			
DVPROPS	A2B3A	PBEAM1	2	0.01	100	D1B			+A2B3A
+A2B3A		PBEAM1	3	0.01	100	D1A			
DVPROP	A3B	PBEAM1	3	0.01	100	D1B			

In this data, the variable **A1A** represents the area at End A of elements 1 and 4. Similarly, **A1B2A** represents the area of End B of elements 1 and 4 and End A of elements 2 and 5, and so on. These data insure the required design continuity and symmetry.

The design constraint is applied using:

DCFREQ	100	F1	LOWER	20.0	1				

which indicates that the first mode will have a lower bound of 20.0 Hz.

The job was then executed in **UAI/*NASTRAN***. The final results are shown in Table 25-17 along with those of [Turner67]. While the weight is actually lower for the MDO design, the design variables are substantially different.

Further analysis indicates that the FE model has more sophistication than Turner's formulation. In particular, he does not consider bending and shear in the beam chords. To better simulate his problem, the finite element model was modified. The QUAD4 elements were

Table 25-17. TURNER'S PROBLEM — FINAL DESIGN

DESIGN VARIABLE OBJECTIVE	[Turner67]	UAI/*NASTRAN* Model MDO3A	UAI/*NASTRAN* Model MDO3B
A1A	1.130	1.11	1.09
A1B2A	0.690	0.94	0.80
A2B3A	0.280	0.55	0.28
A3B	0.0	0.30	0.10
T100	0.037	0.049	0.039
T200	0.034	0.043	0.035
T300	0.023	0.025	0.022
WEIGHT	97.27	97.22	97.86

replaced by SHEAR panels and the BEAM bending and shear characteristics were removed. Additionally, massless CONROD elements were placed between the SHEAR panels. This model, found in file **MD03B**, was then executed. The results, also shown in Table 25-17, are now much closer to those of Turner.

25.6.4 Transmission Tower

This example problem illustrates the design of a structure which allows both GRID point motion and element property changes. It features:

- ❒ Multiple static loading conditions
- ❒ GRID point coordinate linking for geometric symmetry
- ❒ Element property linking for member symmetry
- ❒ Weight minimization

Example Problem 25-4

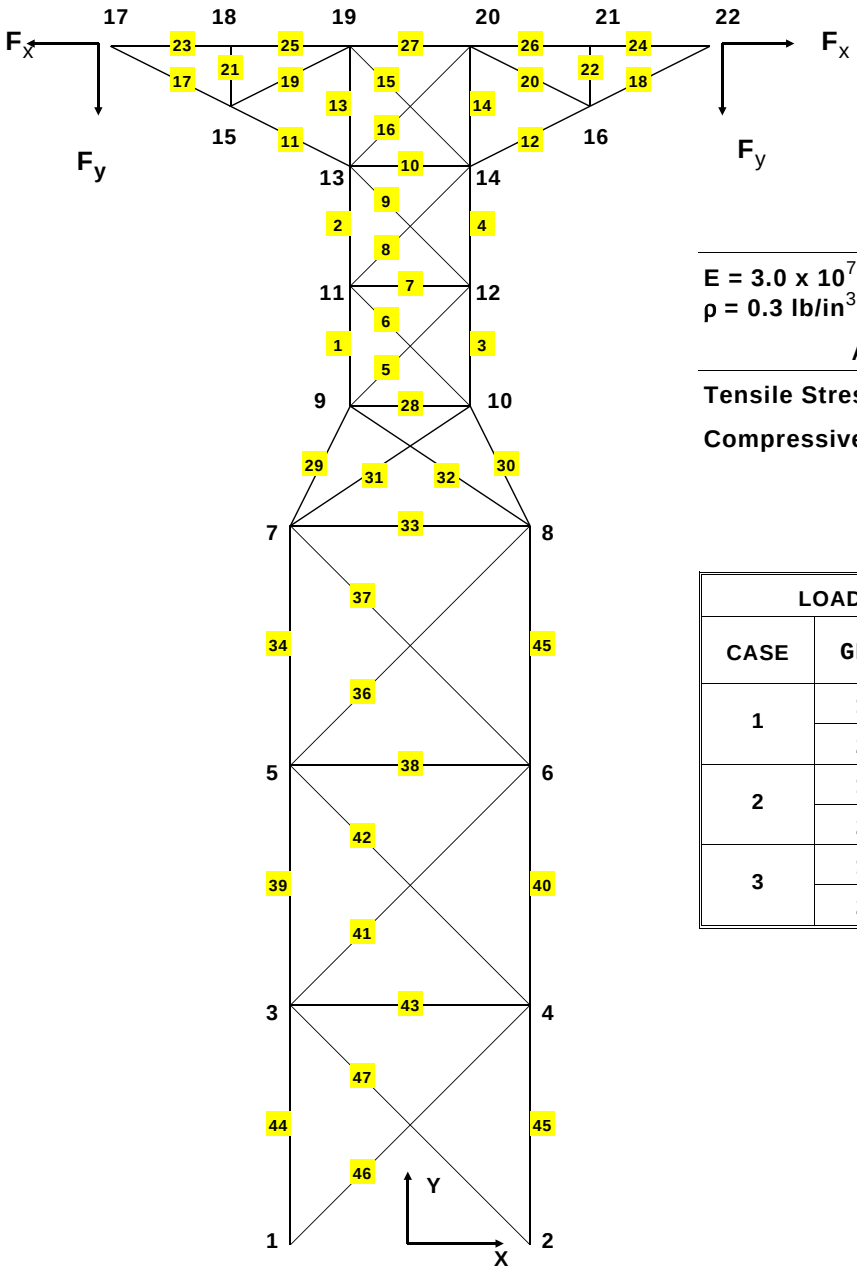
A planar transmission tower is to be designed subject to three separate loading conditions. The first condition simulates the expected load when the tower supports two power lines. The second two conditions model the behavior encountered when one of the power lines has snapped, first the left, and then the right. Natural stress constraints in the members are imposed and GRID points are allowed to move to new locations.

The finite element model, shown in Figure 25-9, is composed of 47 ROD elements. The data are found in file **MD04**. The three loading conditions are summarized in the figure. The design model includes 44 independent design variables. These include 27 ROD cross-sectional areas, and 17 coordinate values. In order to maintain symmetry, design variable linking is used for both the property and coordinate variables. The relationships imposed are also shown in Figure 25-9. The Case Control commands for this probelm are:

```
MINIMIZE WEIGHT
   MAXITER = 50
   CONVERGE = 0.5
   MAXRETAIN = 70
SPC = 1
STRESS = ALL
DISP = ALL
SET 100 = 100
DESCON = 100
CASE 1 STATICS
   LABEL = LOAD CASE 1
   LOAD = 1
CASE 2 STATICS
   LABEL = LOAD CASE 2
   LOAD = 2
CASE 3 STATICS
   LABEL = LOAD CASE 3
   LOAD = 3
```

The new addition here is the three CASEs. The physical design variables are defined in the manner illustrated in previous example problems, and symmetric linking is assured by simply referencing symmetric elements to the same property variable, i.e. shared property linking. The GRID point locations are specified as design variables by using the **DVGRID** Bulk Data

Figure 25-9. TRANSMISSION TOWER MODEL



PROPERTIES

$E = 3.0 \times 10^7$ psi
$\rho = 0.3$ lb/in^3

ALLOWABLES

Tensile Stress: 2.0×10^4 psi

Compressive Stress: -1.5×10^4 psi

LOADING CONDITIONS			
CASE	GRID	F_x (lb)	F_y (lb)
1	17	6000	-14000
	22	6000	-14000
2	17	6000	-14000
	22	0	0
3	17	0	0
	22	6000	-14000

25

entry. Examples showing this are:

DVGRID	X1	1		-600.0	600.0	X			
DVGRID	X2	2		-600.0	600.0	X			
DVGRID	X3	3		-600.0	600.0	X			
DVGRID	Y3	3		0.0	1000.	Y			
DVGRID	X4	4		-600.0	600.0	X			
DVGRID	Y4	4		0.0	1000.	Y			
...									
DVGRID	X20	20		-600.0	600.0	X			
DVGRID	X21	21		-600.0	600.0	X			
DVGRID	Y20	20		0.0	1000.	Y			
DVGRID	Y21	21		0.0	1000.	Y			

The linking relationships are defined using the following **DVLINK** Bulk Data entries:

DVLINK	X1-2	10.0	-50.0	-1.0	X1	50.0	1.0	X2	
DVLINK	X3-4	10.0	-50.0	-1.0	X3	50.0	1.0	X4	
...									
DVLINK	X18-21	10.0	-80.0	-1.0	X18	80.0	1.0	X21	
DVLINK	X19-20	10.0	-20.0	-1.0	X19	20.0	1.0	X20	
DVLINK	Y3-4	10.0	110.0	1.0	Y3	110.0	1.0	Y4	
DVLINK	Y5-6	10.0	230.0	1.0	Y5	230.0	1.0	Y6	
...									
DVLINK	Y18-21	10.0	590.0	1.0	Y18	590.0	1.0	Y21	
DVLINK	Y19-20	10.0	590.0	1.0	Y19	590.0	1.0	Y20	

Note that the coefficients (indicated by the shading) have all been set to unity (+1.0 or -1.0) in order to improve the accuracy of the sensitivities. This allows the finite difference step size to be a constant for each of the coordinates. The initial value of the mathematical variables (10.0) and the invariant portions of the physical design variables are then determined so that they correspond to the initial geometry. Note that the linking used here is not reduced basis linking because each **DVGRID** design variable appears on only one **DVLINK** entry.

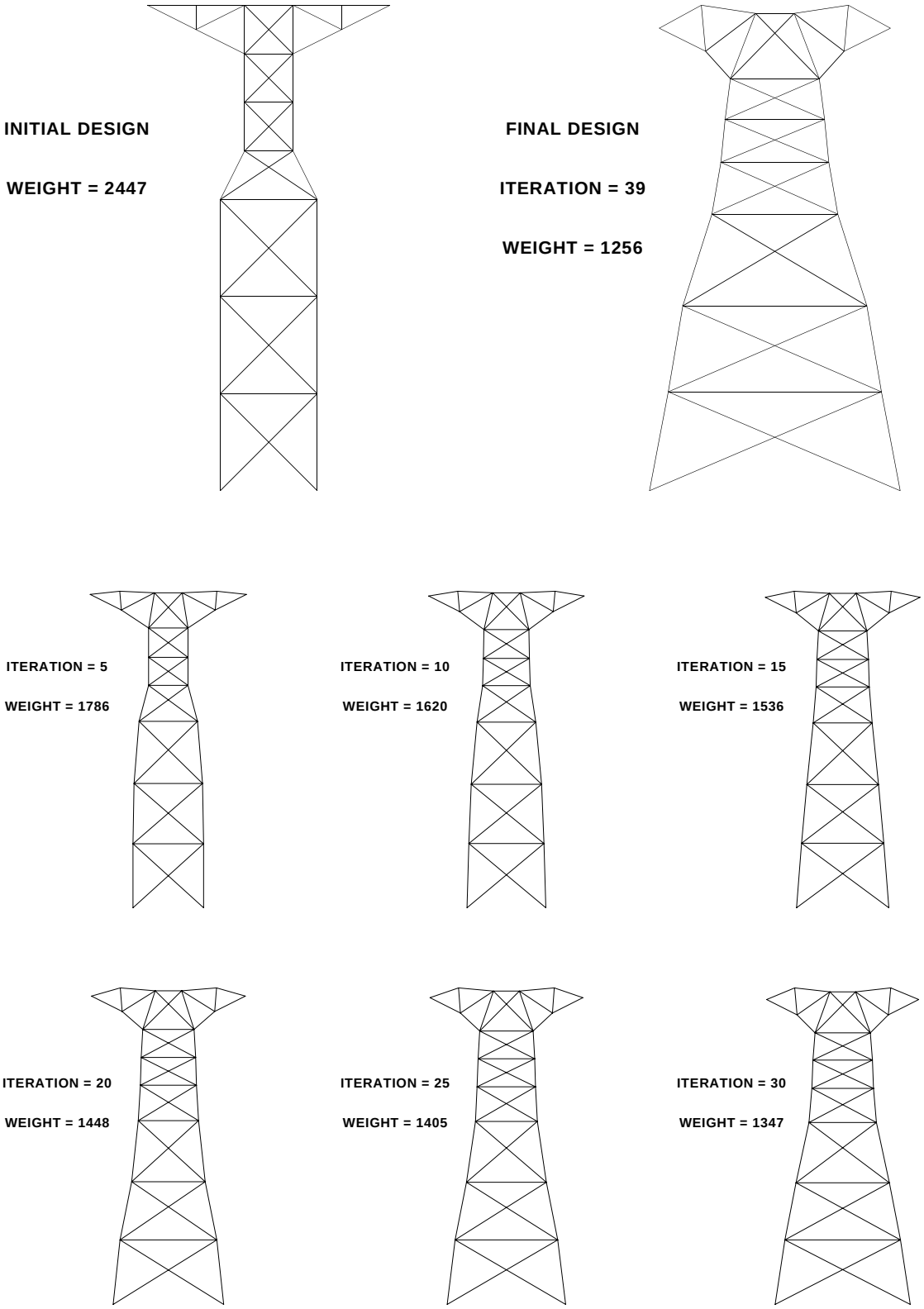
The stress constraints are applied as in earlier examples. The job was then executed in **UAI/*NASTRAN***. The final results, including all of the design variable values and the final weight, are shown in Table 25-18. The shaded blocks indicate coordinates that were fixed during the redesign. Plots of the initial and final geometric shape, as well six intermediate configurations are shown in Figure 25-10. These are similar to [Hansen90] which considered other design constraints.

Table 25-18. TRANSMISSION TOWER FINAL DESIGN

MEMBER AREAS	INITIAL VALUE	FINAL VALUE	GRID COORDINATES	INITIAL VALUE	FINAL VALUE
$A_1 = A_3$	3.8	1.96	$X_1 = -X_2$	60.0	162.8
$A_2 = A_4$	3.4	1.91	$Y_1 = Y_2$	0.0	
$A_6 = A_5$	0.8	0.29	$X_3 = -X_4$	60.0	138.3
A_7	0.9	0.10	$Y_3 = Y_4$	120.0	128.1
$A_9 = A_8$	0.9	0.45	$X_5 = -X_6$	60.0	119.4
A_{10}	1.8	1.08	$Y_5 = Y_6$	240.0	239.9
$A_{11} = A_{12}$	2.1	1.53	$X_7 = -X_8$	60.0	81.7
$A_{13} = A_{14}$	1.2	0.50	$Y_7 = Y_8$	360.0	358.9
$A_{16} = A_{15}$	1.6	0.63	$X_9 = -X_{10}$	30.0	70.0
$A_{17} = A_{18}$	2.1	1.23	$Y_9 = Y_{10}$	420.0	426.1
$A_{19} = A_{20}$	0.7	0.15	$X_{11} = -X_{12}$	30.0	64.5
$A_{21} = A_{22}$	0.9	0.77	$Y_{11} = Y_{12}$	480.0	481.4
$A_{23} = A_{24}$	1.7	0.93	$X_{13} = -X_{14}$	30.0	57.7
$A_{25} = A_{26}$	1.7	0.88	$Y_{13} = Y_{14}$	540.0	534.4
A_{27}	1.4	0.69	$X_{15} = -X_{16}$	90.0	
A_{28}	0.9	0.14	$Y_{15} = Y_{16}$	570.0	
$A_{29} = A_{30}$	3.7	2.01	$X_{17} = -X_{22}$	150.0	
$A_{32} = A_{31}$	1.5	0.35	$Y_{17} = Y_{22}$	600.0	
A_{33}	0.7	1.00	$X_{18} = -X_{21}$	90.0	95.4
$A_{34} = A_{35}$	2.9	2.23	$Y_{18} = Y_{21}$	600.0	629.7
$A_{37} = A_{36}$	0.7	0.28	$X_{19} = -X_{20}$	30.0	25.0
A_{38}	1.6	0.11	$Y_{19} = Y_{20}$	600.0	619.2
$A_{39} = A_{40}$	3.7	2.26			
$A_{42} = A_{41}$	1.6	0.18	WEIGHT	2447.0	1256.5
A_{43}	0.7	0.10			
$A_{44} = A_{45}$	4.5	2.39			
$A_{47} = A_{46}$	1.6	0.12			

Shading indicates coordinates fixed during design.

Figure 25-10. TRANSMISSION TOWER DESIGN CONFIGURATIONS



25.6.5 Bimetallic Strip

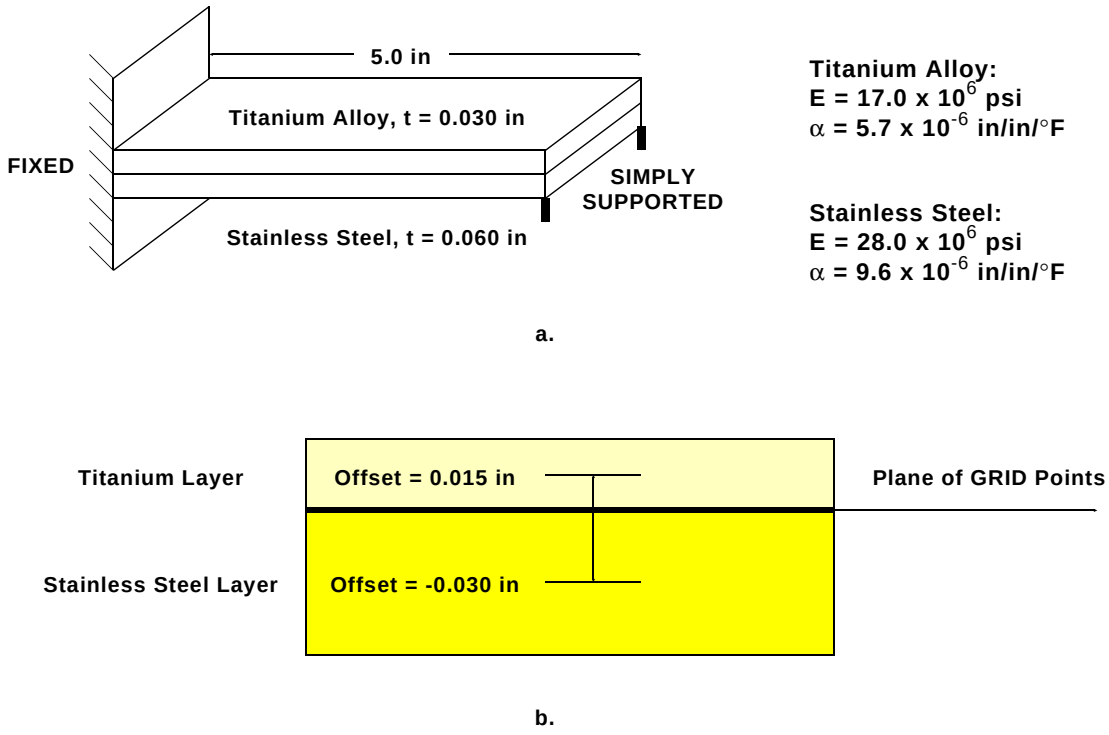
Bimetallic strips are often used in instruments to sense and control temperatures by acting as switches. The following example problem includes the following MDO features:

- Plate elements with offsets
- Thermal loads
- GRID point coordinate linking for mesh consistancy
- Volume minimization
- Technique for modeling a reaction force constraint

Example Problem 25-5
The bimetallic strip shown in Figure 25-11 is made by bonding a thin strip of titanium alloy to a thin strip of stainless steel. The strip is fixed at the left end and simply supported at the right. The design problem is to determine the minimum length of the strip which will develop a reaction force of 5 oz. at the simply supported end when its temperature is raised 50°F.

The structure and its physical properties are shown in Figure 25-11a. The model is found in file **MDO5**.

Figure 25-11. BIMETALLIC STRIP MODEL



It is composed of two distinct layers of QUAD4 elements which share one set of GRID points. The top layer represents the titanium alloy, and the bottom layer the stainless steel. The material properties are shown in the figure. This very simple type of composite construction is well modeled by using offsets from the plane of the GRID points to the neutral axes of each layer of elements as shown in Figure 25-11b. The finite element model is comprised of 20 elements of each type as shown in Figure 25-12a.

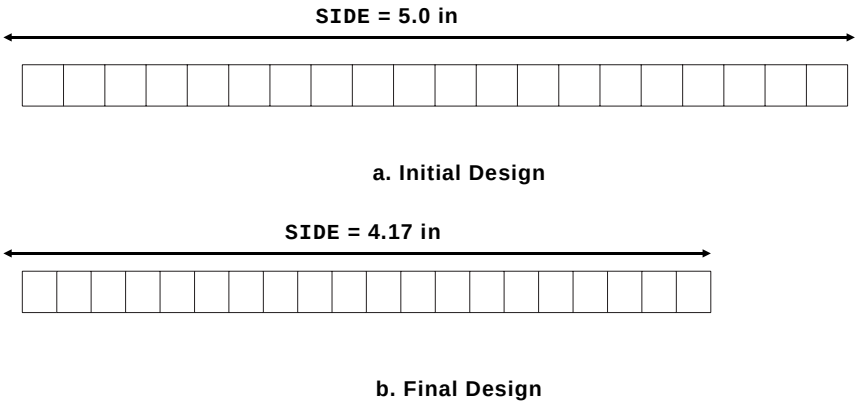
There are several approaches to the design model. First, one could assign each of the GRID point x-coordinates to a design variable using **DVGRID** Bulk Data entries:

DVGRID	**X2**		**2**	**0.0**	**10.0**	**X**			
DVGRID	**X3**		**3**	**0.0**	**10.0**	**X**			
...									
DVGRID	**X20**		**20**	**0.0**	**10.0**	**X**			
DVGRID	**X21**		**21**	**0.0**	**10.0**	**X**			
DVGRID	**X32**		**32**	**0.0**	**10.0**	**X**			
...									
DVGRID	**X51**		**51**	**0.0**	**10.0**	**X**			

Note that GRID points 1 and 31 are not defined as design variables because they are to remain fixed during the design process.

Because this is a continuum structure which will under a shape change, care must be taken to insure that the finite element mesh does not become overly distorted, and, more importantly, does not result in elements disappearing or penetrating into adjacent elements. Since this model is so simple, the mesh can actually be controlled by linking all of the GRID point x-coordinates. This may be done by defining a single mathematical variable, called **LEN**. **LEN** simply represents the x-coordinates of the two GRID points at the simply supported end of the strip, 21 and 51. Then, all of the grid points on each edge of the model are linked to insure that they maintain the same relative proportion. Mathematically, this is stated by:

Figure 25-12. BIMETALLIC STRIP — FINAL DESIGN



$$\begin{Bmatrix} X2 \\ X3 \\ X4 \\ \cdots \\ X20 \\ X21 \\ X31 \\ X32 \\ X33 \\ \cdots \\ X50 \\ X51 \end{Bmatrix} = \begin{Bmatrix} 0.05 \\ 0.10 \\ 0.15 \\ \cdots \\ 0.95 \\ 1.00 \\ 0.05 \\ 0.10 \\ 0.15 \\ \cdots \\ 0.95 \\ 1.00 \end{Bmatrix} \text{LEN}$$

Note that the coefficients in the ***T*** matrix are the ratios of the x-coordinates of each of the GRID points to the tip x-coordinate. This linking relationship would then be specified using **DVLINK** Bulk Data entries. This method results in 60 line of Bulk Data: 40 for the **DVGRID** data; and 20 for the **DVLINK** data.

Because the actual linking relationship is simply the original coordinate values, a simpler approach may be used — the **DVGRIDS** Bulk Data entry. The entire design variable specification may be made using:

DVGRIDS	SIDE	1		0.0	10.0	X			+A
+A		2		0.0	10.0	X			+B
...		...		...	...	...			...
+S1		50		0.0	10.0	X			+T1
+T1		51		0.0	10.0	X			

This results in 40 lines of input (the same as required for the **DVGRID** entries previously shown) and eliminates the need for any additional linking data because the **DVGRIDS** entry defines a simple linking in which the physical design variables maintain the ratios given by their initial values.

The next design modeling technique is used to define a constraint on the reaction force at the simply supported end of the strip. Since the **SPCFORCE** itself cannot be constrained a simple artifice is used — stiff springs to ground. These are modeled by ELAS2 elements which have spring constants large enough to result in displacements that are four orders of magnitude less than the real model displacements. These data are:

CELAS2	200	1.+8	21	3					
CELAS2	300	1.+8	51	3					

The axial force generated in these springs will now simulate the reaction force along the edge. The design constraints are now defined in terms of the ELAS2 element forces with the entries:

DCELEM	100	TIPF	FORCE	AXIAL		.15625		200	
DCELEM	100	TIPF	FORCE	AXIAL		.15625		300	

One half of the specified force (5 oz = 0.3125 lb) is constrained at each side of the strip. The job is then executed in **UAI/*NASTRAN*** . Convergence is obtained after four iterations. The final value of the design variable **SIDE** is 0.834 which results in **LEN** = 4.17 in. This is in excellent agreement with the solution of 4.11 in. found in [Young87]. The final finite element model is shown in Figure 25-12b for easy comparison with the initial shape.

25.6.6 System Identification

An important area of research is the tuning of finite element models to experimental test results. This is often called system identification. This example problem illustrates how optimization may be used to address these requirements. It features:

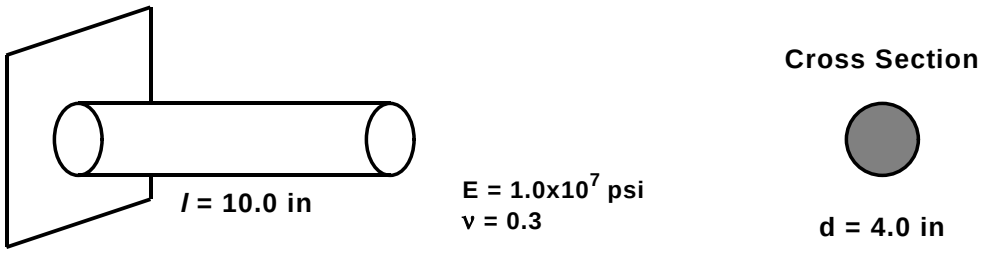
- ❒ Normal modes optimization
- ❒ Constraints on RMS error in mode shapes
- ❒ Frequency constraints
- ❒ Using an analytical response as the objective

Example Problem 25-6
Consider the model shown in Figure 25-13. It is a simple cantilever beam which is composed of 10 BAR elements with circular cross sections having a constant diameter of 4.0 in. The cross-sectional area of the three BAR elements at the root of the beam may be varied. You wish to design this area such that it matches known test results.

The finite element model, which is found in file **MDO6**, is shown in Figure 25-13. Table 25-19 presents the test results and a definition of the design constraints for the problem. These data reflect that two natural modes, mode 1 (first bending) and mode 3 (first extensional) have been measured experimentally at three locations which correspond to GRID points **3**, **6** and **9** in the finite element model. The frequency of the first mode, **61.912**, has also been measured.

The design model is simple having a single design variable which represents the root cross-sectional area. The Case Control commands are:

Figure 25-13. SYSTEM ID — SIMPLE BEAM MODEL



BAR 1 A	BAR 2 A	BAR 3 A	BAR 4	BAR 5	BAR 6	BAR 7	BAR 8	BAR 9	BAR 10

```
SET 1000 = 1000
SET 3000 = 1000,2000
OBJECTIVE MINIMIZE 1000
   MAXITER = 3
CASE 1001 MODES
   DISP = ALL
   METHOD = 1
   DESCON = 3000
```

You will notice that the **OBJECTIVE** command is different than before. Rather than specifying **WEIGHT** or **VOLUME**, a set identification number is given. This set defines a single response constraint that is to be minimized. There are three constraints in the design model. These are:

DCMODR	**1000**	**MODE1**	**1**	**RMS**	**0.002**	**MAX**			**+A**
+A		**3**	**T3**	**1.431-2**	**6**	**T3**	**1.741-1**		**+B**
+B		**9**	**T3**	**6.381-1**					
DCMODR	**2000**	**MODE3**	**1**	**RMS**	**0.002**	**MAX**			**+A**
+A		**3**	**T1**	**1.204-1**	**6**	**T1**	**5.431-1**		**+B**
+B		**9**	**T1**	**9.216-1**					
DCFREQ	**2000**	**MODE1**	**LOWER**	**61.911**	**1**				
DCFREQ	**2000**	**MODE1**	**UPPER**	**61.913**	**1**				

The first constraint is applied to the first mode shape. The **DCMODR** entry is essentially a table which defines specific measured points on the eigenvector. As you can see, not all points are required. The parent line also specifies that the **RMS** error between the first computed shape and the input shape must be less than **0.002**. It also indicates that the normalization method used for the experimental mode was **MAX**. Note that the other normalizations may also be used. The second constraint is applied to the third mode in a similar fashion. Finally, an equality constraint is simulated by using two **DCFREQ** entries, one an upper bound and one a lower bound, which constrain the first eigenvalue to its measured value of **61.912**.

The job is then executed in **UAI/*NASTRAN***. Convergence is achieved in five iterations. The objective function value is nearly zero (**4.44×10^{-5}**). The final solution results in a frequency of **61.911** for the first mode and the root area has been raised to **7.84** in^2. Figures 25-14 and 25-15 show the initial and final mode shapes for the first and third modes along with the experimental data. The design modes match the experiment exactly in both cases. Note that the bend in the third mode shape at GRID point four is caused by an area discontinuity effect as the area of the bar drops rapidly.

Table 25-19. SYSTEM ID — EXPERIMENTAL DATA

GRID	MODE 1		MODE3	
	COORDINATE	VALUE	COORDINATE	VALUE
3	T3	0.1431	T1	0.1204
6	T3	0.1741	T1	0.5431
9	T3	0.6381	T1	0.9216
Frequency (Hz)	61.912			

Figure 25-14. FIRST BENDING MODE — FINAL DESIGN

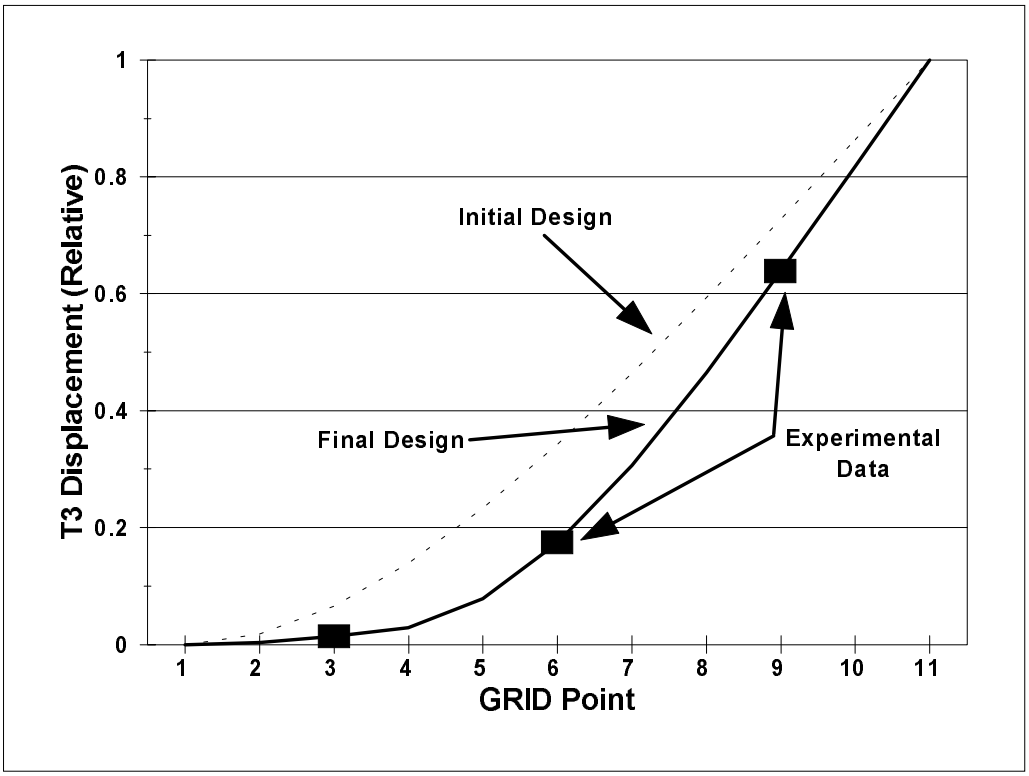
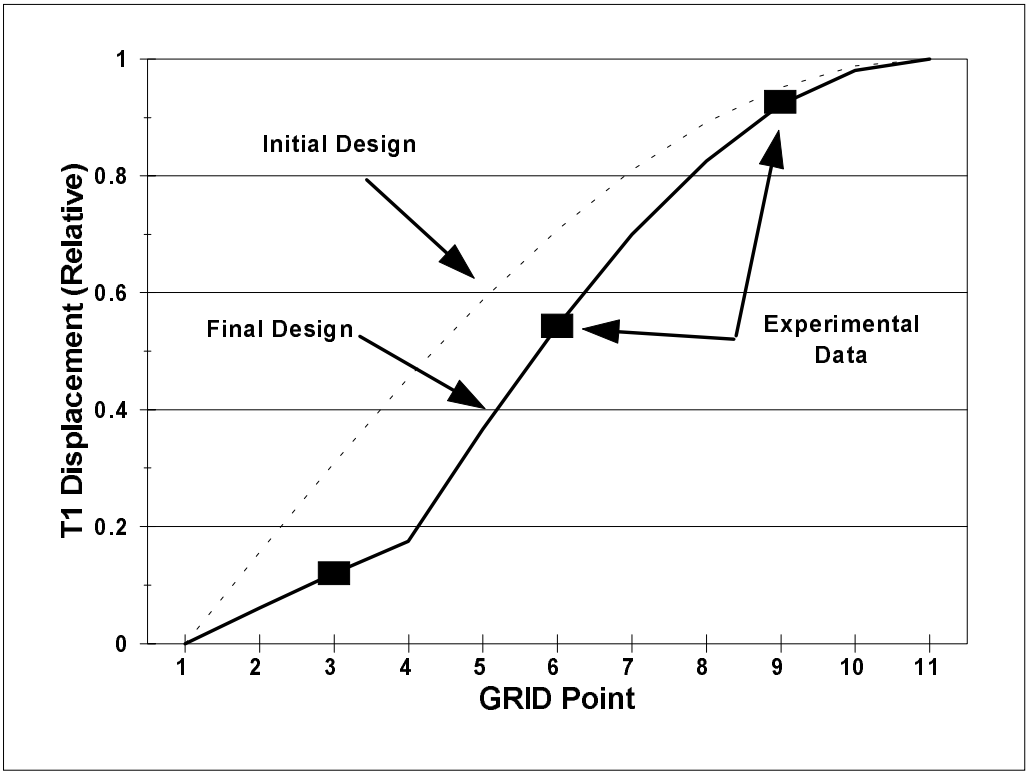


Figure 25-15. FIRST EXTENSIONAL MODE — FINAL DESIGN



25

25.6.7 Engine Mount Design

It is often desirable that the frequency response of a structure be controlled to avoid large amplitude excitation. This example problem illustrates the design of the mount points of an engine to minimize the engine accelerations when the vehicle is subjected to loads simulating a rough road. Further, lower bound modal frequency constraints are imposed to ensure that the optimizer will not attempt to address the engine accelerations by making the chassis/mount too flexible (otherwise, decoupling the engine from the chassis is a good way to limit accelerations). It features:

- Multidisciplinary Optimization
- Dynamic Response Constraints
- GRID point coordinate design variables attached to Rigid Elements
- Using an analytical response as the objective

Example Problem 25-7

A simple dynamics model of a vehicle chassis, engine and engine mount is modelled as a series of beams, rigid elements, concentrated masses and scalar springs. The vehicle is grounded using simulated tire flexibilities. In one CASE, a frequency dependent load roughly equal in magnitude to the total weight is applied equally over all frequencies to the two front tires. The loads are 90 degress out-of-phase. In another CASE, the normal modes are computed.
The design variables are the x- and y-coordinates of the engine mount points. Constraints are place on the engine accelerations due to the dynamic load and on the natural frequencies of the engine-dominated normal modes. The average vertical accelerations of the engine are minimized.

The initial geometry and mount point locations are shown in Figure 25-16. You may find this model in file **MD07**. The forward engine mounts are located at GRID points 111 and 112, the rear mount is located at **GRID** 113.

Figure 25-16. ENGINE MOUNT MODEL

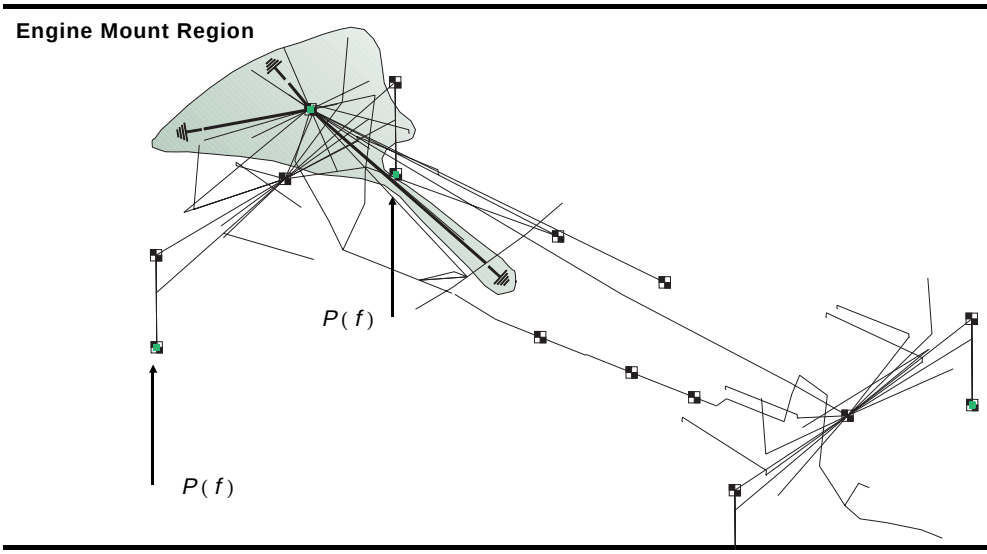
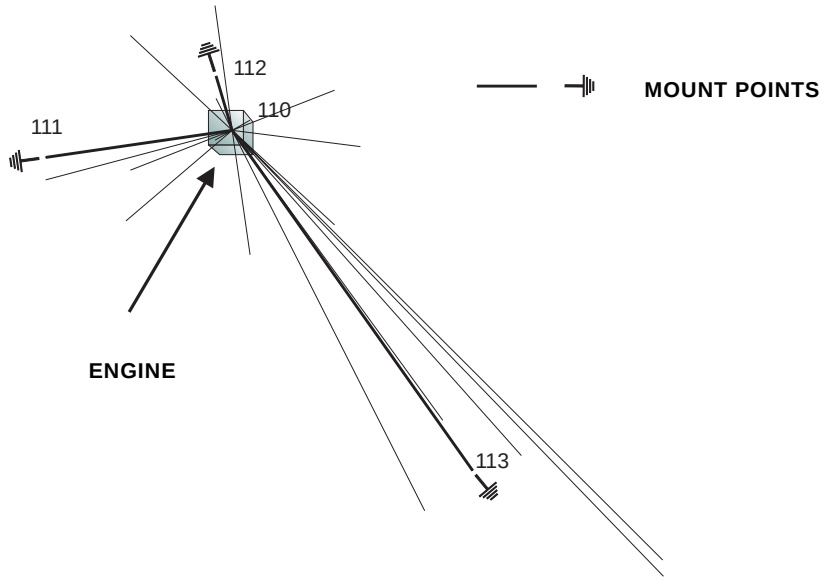


Figure 25-17. BLOW-UP OF ENGINE MOUNTS



The engine is located at GRID point 110 and is modeled using **CONM2** 30 as shown in the blow-up in Figure 25-17.

The Case Control commands are important since they demonstrate an invocation of the multidisciplinary analyses that are integral to MDO:

```
OBJECTIVE MINIMIZE 1000
   MAXITER  = 20
$
MAXRETAIN = 50
$
CASE 900 MODES
   LABEL = NORMAL MODES ANALYSIS
   SUMMARY = NONE
$
   SET 900=900
$
   SPC    = 1
   METHOD = 1
   DESCON = 900
$
$   DISP=ALL
CASE 1000 DFREQUENCY
   LABEL = DIRECT FREQUENCY RESPONSE
$
   SET 1000=1000,1001
$
   SPC    = 1
   FREQ   = 1001
   DLOAD  = 1002
   DESCON = 1000
```

The **OBJECTIVE** command refers to a constraint set-id of a single-valued response that is the objective. Notice that, in **CASE 1000**, that constraint set-id appears in the **SET** reference by the **DESCON** command. This chaining in necessary to associate the set-id in the **OBJECTIVE** command with a particular **CASE**. Notice also, that there are additional responses in the direct

frequency analysis that are constrained (**SET 1001**). The **MODES** analysis is also constrained with a different set (**SET 900**) of constraints.

To solve this problem, you must specify six design variables which represent the x- and y-coordinates of each engine mount point. The Bulk Data entries required to do this are:

DVGRID	X111	111		1000.	2000.	X			
DVGRID	Y111	111		-700.	-300.	Y			
DVGRID	X112	112		1000.	2000.	X			
DVGRID	Y112	112		-300.	700.	Y			
DVGRID	X113	113		2000.	3000.	X			
DVGRID	Y113	113		-500.	500.	Y			

Notice that the model units are millimeters, Megagrams, and Newtons for length, mass, and force, respectively.

To define the modal frequency constraints, an initial analysis was performed and the modes in which the engine participated were identified. Lower bounds on these modes were applied to force the optimizer to find the minimum accelerations using a mount that is at least as stiff as the original. This choice is merely representative of one choice a designer might make, it is not the only way to pose the optimization problem. The Bulk Data entries required to do this are:

DCFREQ	900	MODE10	LOWER	3.5	10				
DCFREQ	900	MODE12	LOWER	7.5	12				
DCFREQ	900	MODE14	LOWER	10.5	14				
DCFREQ	900	MODE15	LOWER	13.5	15				

Finally, that same initial (multidisciplinary) analysis was used to identify the critical engine accelerations and the range of frequencies in which they occurred. These data are shown along with the final design responses in Figures 25-18. Since the engine mount points are the only design variables, one can safely assume that there will only be a small shift in the frequency band of the response. Therefore, dynamic response constraints are applied over only these initial ranges, with a little padding to account for the small potential shift. The objective function is to minimize the **AVERAGE** vertical acceleration in the range where the current peak exists. In addition, the **PEAK** lateral and fore-aft accelerations will be limited to the values obtained for the original mount. Thus we are minimizing the vertical accelerations while demanding that the lateral accelerations get no larger. The Bulk Data entries required to do this are:

DCDYNRG	1000	A110Z	ACCEL	AVG		1.0	5.	15.	+DA110Z
+DA110Z	110	T3	1.0						
DCDYNRG	1001	A110Y	ACCEL	PEAK		4.125+3	5.	15.	+DA110Y
+DA110Y	110	T2	1.0						
DCDYNRG	1001	A110X	ACCEL	PEAK		3.790+3	5.	15.	+DA110X
+DA110X	110	T1	1.0						

Notice that constraint **SET 1000** is our objective function while **SET 1001** are the lateral acceleration constraints that are also applied.

The job is then executed. Convergence is achieved in seven design cycles. Table 25-20 shows the changes to the design variables and the critical responses. In addition to those data, one of the lower bound frequencies is just satisfied at the optimum. Figures 25-19 show the frequency responses of the engine block (GRID point 110) to the applied load before and after the optimi-

Table 25-20. ENGINE MOUNT FINAL DESIGN

DESIGN VARIABLE RESPONSE	INITIAL DESIGN	FINAL DESIGN
111 x	1480.0	1305.0
111 y	-215.0	-121.0
112 x	1420.0	1789.5
112 y	230.0	368.2
113 x	2575.0	2716.6
113 y	20.0	36.8
PEAK $\ddot{u}$	2933.8 @ 10.4 Hz	3789.6 @ 8.5 Hz
PEAK $\ddot{v}$	4124.0 @ 7.3 Hz	4125.0 @ 7.0 Hz
PEAK $\ddot{w}$	8971.0 @ 10.1 Hz	7658.3 @ 10.2 Hz
AVG $\ddot{w}$	3781.8 [5,15] Hz	2980.8 [5,15] Hz

zation. The shaded portions of Figures 25-19a and 25-19b denote the applied constraints on the response, while the shading in Figure 25-19c shows the frequency range over which the objective function was computed. Note that the average vertical accelerations are reduced by 21% in the frequency range of the objective function and the peak acceleration is reduced by 23% and the lateral and fore/aft accelerations are still bounded by the original peaks. Figure 25-19 shows a close-up view of the engine mount points along with the original points.

The results presented in this example problem are indicative of the first steps one might take in performing optimization of a real physical system to tune its frequency response performance. At this point, one would need to examine the full range of behaviors over other loading conditions, all frequencies and for other critical constraints. For example, the new mount points might have unacceptable internal loads under static load conditions, or the high or low frequency behavior might be degraded. If so, the next steps would be to apply the requisite **CASE**s with their attendant design constraints and obtain the optimum that satisfies the constraints of a more complete design specification.

Figure 25-18. FINAL DESIGN OF MOUNT POINTS

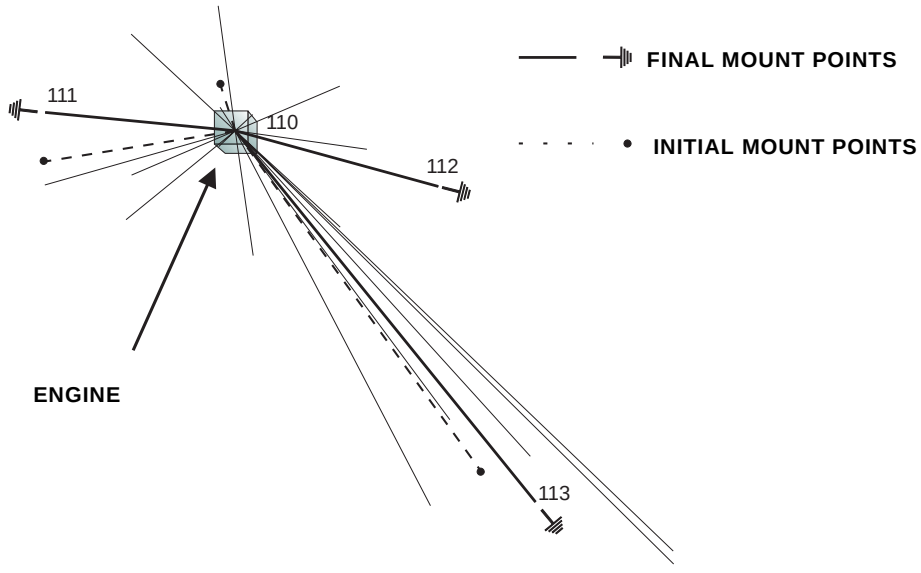
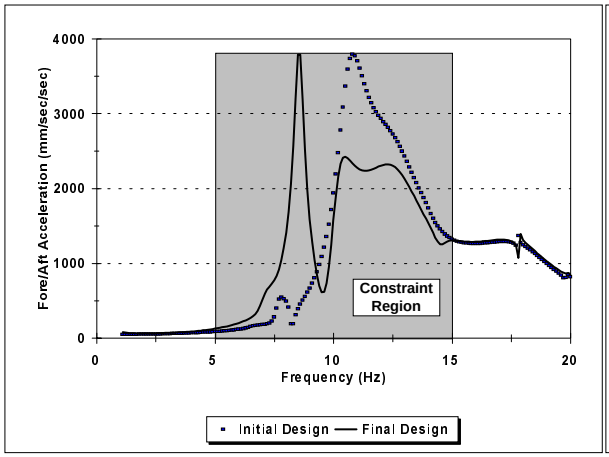
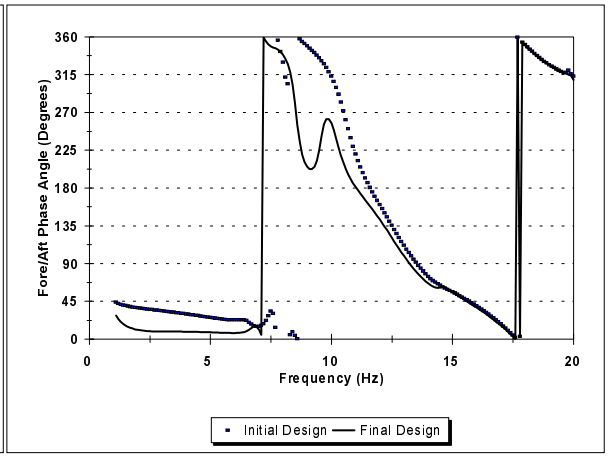


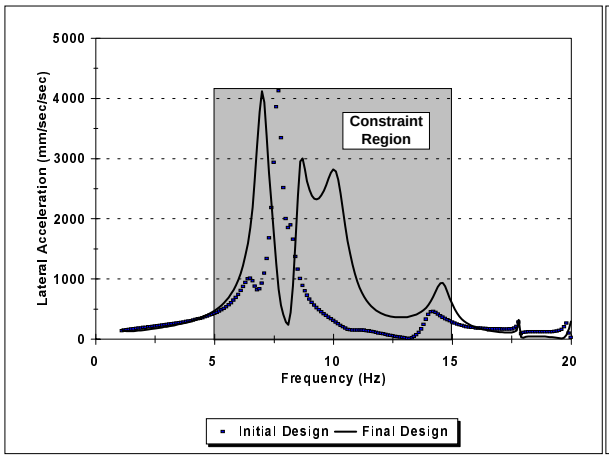
Figure 25-19. FREQUENCY RESPONSE RESULTS



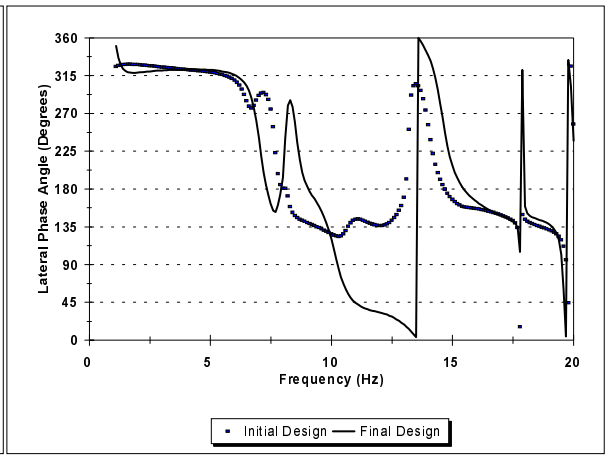
a1. Fore/Aft Acceleration of the Engine (Magnitude)



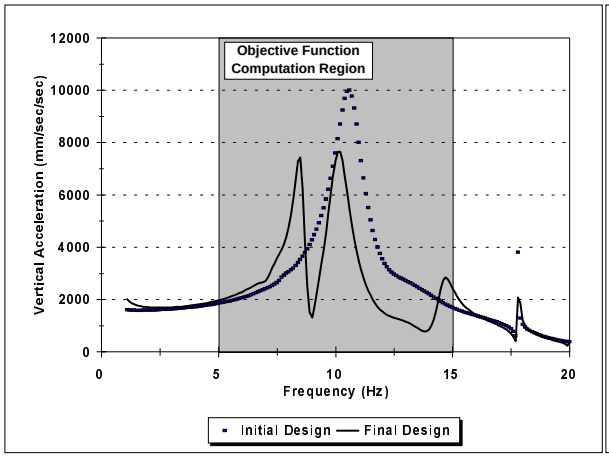
a2. Fore/Aft Acceleration of the Engine (Phase)



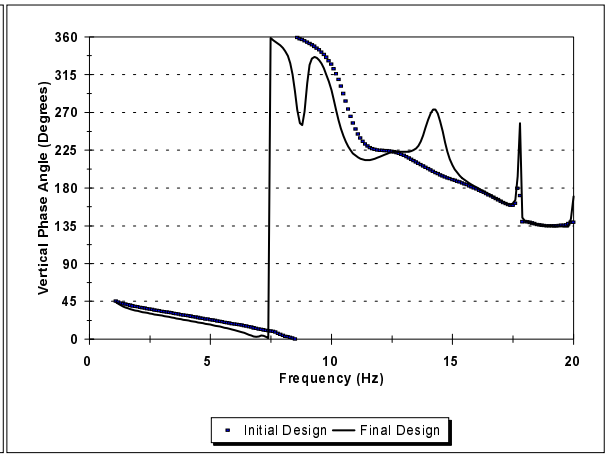
b1. Lateral Acceleration of the Engine (Magnitude)



b2. Lateral Acceleration of the Engine (Phase)



c1. Vertical Acceleration of the Engine (Magnitude)



c2. Vertical Acceleration of the Engine (Phase)

25.7 REFERENCES

[Canfield90]: Canfield, R.A., "High-Quality Approximation of Eigenvalues in Structural Optimization," *AIAA Journal*, Vol. 28, No. 6, June 1990.

[Haftka90]: Haftka, R.T., Gurdal, Z., and Kamat, M.P., Elements of Structural Optimization, Kluwer Academic Press, 1990.

[Hansen90]: Hansen, S.R. and Vanderplaats, G.N., "Approximation Method for Configuration Optimization of Trusses," *AIAA Journal*, Vol. 28, No. 1, January, 1990.

[Herendeen86]: Herendeen, D.L., Hoesly, R.L., Johnson, E.H., and Venkayya, V.B., "ASTROS — An Advanced SOftware Environment for Automated Design," *Proceeding of the AIAA/ASME/ASCE/AHS 27th Structures, Structural Dynamics and Materials Conference*, San Antonio, TX, 1986.

[Kamat93]: *Structural Optimization: Status and Promise*, M.P. Kamat (ed.), American Institute of Aeronautics and Astronautics, 1993.

[Johnson88]: Johnson, E.H. and Neill, D.J., "Automated Structural Optimization System (ASTROS), Volume III — Applications Manual," AFWAL-TR-88-3028, Flight Dynamics Laboratory, Air Force Wright Aeronautical Laboratories, December, 1988.

[Neill90] Neill, D.J., Johnson, E.H., and Canfield, R.A., "ASTROS — A Multidisciplinary Automated Structural Design Tool," *Journal of Aircraft*, Vol. 27, No. 12, December, 1990.

[Schmidt76]: Schmidt, L.A. Jr. and Miura, H., *Approximation Concepts for Efficient Structural Synthesis*, NASA CR-2552, 1976.

[Sobieski89]: Sobieski, J., "Multidisciplinary Optimization for Engineering Systems: Achievement and Potential," *Lecture Notes in Engineering*, C. Brebbia and S.A. Orszag (Eds.), June, 1989.

[Turner67]: Turner, M.J., "Design of Minimum Mass Structures with Specified Natural Frequencies," *AIAA Journal*, Vol. 5, No. 3, March, 1967.

[Vander84]: Vanderplaats, G.N., *Numerical Optimization Techniques for Engineering Design*, McGraw-Hill, Inc., 1984.

[Young89]: Young, W.C., *Roark's Formulas for Stress and Strain*, McGraw-Hill, Inc., 1989.

Chapter 26

SENSITIVITY ANALYSIS

☞ Please read Chapter 25 before reading this Chapter.

Sensitivity analysis is a fundamental building block of Multidisciplinary Design Optimization (MDO). As such, all of the concepts of design variables and linking and design constraints are described in Chapter 25.

This chapter describes how Sensitivity Analysis is performed using DMAP Rigid Formats and how its results may be used to assist you in redesigning your structure if you do not use the **UAI/*NASTRAN*** MDO capability, or if you are using optimization software that is external to the program. This Chapter presents the differences between the Sensitivity Rigid Formats and the MDO APEX solution sequence which is described in Chapter 25.

26.1 CONSTRAINT DERIVATIVES

The constraint derivatives are computed within **UAI/*NASTRAN*** by using first order finite differences. First, each response of the model for design variable v_j , called $R(v_j)$, is computed. Then v_j is perturbed by a small percentage, δB, which results in an incremental perturbation of $\delta v_j = \delta B \cdot v_j$. A new response solution is then performed. This is followed by the computation of $R(v_i + \delta v_i)$. In each case, the nondimensional constraint value is evaluated:

$$g_i(v_j) = \frac{R(v_j) - R_{lim}}{\left|R_{lim}\right|} \tag{26-1}$$

$$g(v_j + \delta v_j) = \frac{R(v_j + \delta v_j) - R_{lim}}{\left|R_{lim}\right|} \tag{26-2}$$

The constraint sensitivity value for the j^{th} design variable is then computed directly using:

$$\frac{\partial g_i}{\partial v_j} \approx \frac{g_i(v_j + \delta v_j) - g_i(v_j)}{(v_i + \delta v_j) - v_j} = \frac{g_i(v_j + \delta v_j) - g_i(v_j)}{\delta v_j} = \frac{g_i(v_j + \delta v_j) - g_i(v_j)}{v_j\,\delta\beta} \tag{26-3}$$

In the unique case where $v_j = 0.0$, **UAI/*NASTRAN*** sets the increment $\delta v_j = \delta\beta$. In this case, (26-3) becomes:

$$\left.\frac{\partial g_i}{\partial \beta}\right|_j = \frac{g_i(\delta\beta) - g_i(0.0)}{(0.0 + \delta\beta) - 0.0} = \frac{g_i(\delta\beta) - g_i(0.0)}{\delta\beta} \tag{26-4}$$

From (26-3) and (26-4) it is apparent that $\frac{\partial g_i}{\partial v_i}$ and $\left.\frac{\partial g_i}{\partial \beta}\right|_j$ differ only by the scale factor v_j. By default, **UAI/*NASTRAN*** computes, and prints $\left.\frac{\partial g_i}{\partial \beta}\right|_j$, but you may request the scaled value, $\frac{\partial g_i}{\partial v_i}$, by using the Bulk Data entry:

PARAM	**'SENSVAL'**	**'SCALED'**							

As you will see in the next section, both sensitivity values may be used to compute a new value of the mathematical design variable, v_j. Notice that the scaled and unscaled values of the constraint sensitivity, when $v_j = 0.0$, are identical.

Finally, the incremental perturbation, δv_j, is taken as 0.1% of the current value of the design variable v_j, that is, $\delta\beta$ is 0.001 by default. You may change this value by using the Bulk Data entry:

PARAM	**'DELTAB'**	δ**B**							

26.2 OPTIMIZATION: DESIGN VARIABLE SCALING

You may use the constraint sensitivities to determine the new design variable values that will result in the satisfaction of violated constraints. The constraint sensitivities allow the constraints themselves to be approximated by a first-order Taylor series expansion:

$$g_j(\boldsymbol{v}) = g_j(\boldsymbol{v}_0) + \sum_{k=1}^{m} \frac{\partial g_j}{\partial v_k}\left(v_k - v_k^0\right) = g_j(\boldsymbol{v}_0) + \nabla \boldsymbol{g}^T \cdot \Delta \boldsymbol{v} \tag{26-5}$$

To see how this information may be used, consider a single constraint, g, and its sensitivity to a single design variable, v, from (26-5):

$$g(v) = g(v_0) + \frac{\partial g}{\partial v}\Delta v \tag{26-6}$$

Assume that you wish to determine a new value for design variable v of the form:

$$v_{new} = v_{old} + \Delta v \tag{26-7}$$

You will recall that the value of the current constraint represents the amount by which it has been violated. Therefore, to make the new constraint value approach zero, you set $g(v)$ to zero and from (26-6) you resize v by the relationship:

$$\Delta v = -g(v_0)\left[\frac{\partial g}{\partial v}\right]^{-1} \tag{26-8}$$

This algorithm works perfectly well for the **SCALED** sensitivities $\frac{\partial g}{\partial v}$. For unscaled sensitivities, $\frac{\partial g}{\partial \beta}$, it is seen from (26-3) and (26-4) that:

$$\frac{\partial g}{\partial \beta} = v_j \frac{\partial g}{\partial v} \tag{26-9}$$

in which case, (26-8) becomes:

$$\Delta v = -g(v_0)\, v_{old}\left[\frac{\partial g}{\partial \beta}\right]^{-1} \tag{26-10}$$

in which case (26-7) becomes:

$$v_{new} = v_{old} - g(v_o)\, v_{old}\left[\frac{\partial g}{\partial \beta}\right]^{-1} = v_{old}\left(1 - g(v_o)\left[\frac{\partial g}{\partial \beta}\right]^{-1}\right) \tag{26-11}$$

The physical interpretation of the unscaled sensitivities, then, is that they represent the effectiveness of a fractional change in v_j on the constraint g_i. Scaled sensitivities, on the other hand, represent absolute changes in v.

Notice that in the case $v_{old} = 0.0$, that (26-11) does not work. This is because the fractional change is undefined for the variable. In **UAI/*NASTRAN***, both scaled and unscaled sensitivities for this case must use (26-7) and (26-8) to compute v_{new}.

In the general case, where more than one constraint exists, it is not possible to identically satisfy all of the constraints. Usually, only the violated constraints are considered during each design iteration. As you redesign your structure, however, other constraints which are close to violation, sometimes called ***active constraints***, may become violated. When this happens, you generally must rely on an external linear programming algorithm to compute new design variables based on the minimization of a specified objective function.

26.3 INPUT DATA REQUIREMENTS

This section provides you with a description of the **UAI/*NASTRAN*** input data required for performing Sensitivity Analysis.

26.3.1 Executive Control Commands

Sensitivity Analysis in **UAI/*NASTRAN*** is performed by specifying a **SOL** command which selects the appropriate Rigid Format. The available Rigid Formats are:

```
SOL 51    (Static Analysis)
SOL 52    (Normal Modes Analysis)
```

26.3.2 Case Control Commands

There are two Case Control commands used in Sensitivity Analysis. The first of these is the **DESCON** command which is used to select response constraint Bulk Data entries. The format of this command is:

```
[SET rcsid = set_specifier]
...
DESCON = { ALL   }
         { rcsid }
```

Note that an optional **SET** command is also shown. The principal use of the **DESCON** command is to select some, or all, of the constraint data from your Bulk Data packet. If only some of the data is to be selected, then the set *rcsid* contains one or more identification numbers of constraint Bulk Data entries as described in the next section. Note that if you specify an *rcsid*, then a corresponding **SET** with the same identification number must appear in your Case Control command packet before it is referenced. When the *rcsid* is replaced by the keyword **ALL**, then all of the design constraints which appear in your Bulk Data packet will be used in the analysis.

The second Case Control command allows you to control the sensitivity results which are recovered. This command is:

```
       ( { DISP   }                                            )
       ( { FREQ   }                                            )     { ALL  }
SENS   ( { STRESS } [ , { PRINT } ] [ , PUNCH ] [ , FILE ]     )  =  {  n   }
       ( { STRAIN }     { POST  }                              )     { NONE }
       ( { FORCE  }                                            )
       ( { VECTOR }                                            )
```

You will note that this command is similar to the respective output selection commands. The **FILE** option allows you to write the sensitivity matrices to a **UAI/*NASTRAN*** file so that you may use them in an external post-processor. If you select **FILE**, then you must use an Executive Control command **ASSIGN** to define the **UAI/*NASTRAN*** file **DESSEN**. The **USE** parameter on the **ASSIGN** command must be **OUTPUT4**.

☞ Note that the MDO Case Control commands (**OBJECTIVE**, **MODDESCON** and **MAXRETAIN**) are not used when performing Sensitivity Analysis.

26.3.3 Bulk Data Entries

All Bulk Data entries used in Sensitivity Analysis to define design variables, design variable linking, and design constraints are the same as those used for MDO as described in Chapter 25 with one exception, which is described below.

26.3.3.1 Modal Response Sensitivity

The Bulk Data entry **DCMODE** is available in Sensitivity Analysis. This is not, in fact, a constraint, rather it allows you to determine the ***response sensitivity*** of one or more eigenvectors in a normal modes analysis by specifying:

DCMODE	DCSID	DCNAME	MODE1	MODE2	MODE3	MODE4	MODE5	MODE6	-cont-
-cont-	MODE7	MODE8	*CONTINUES WITH LIST OF MODES*						

For example, to request eigenvector sensitivities for the first ten modes of your model, assuming that the constraint set identification number is 1 and the **DCNAME** is **SHAPES**, is specified using:

DCMODE	1	SHAPES	1	THRU	10				

The output, shown in Table 26-1, gives the sensitivity of each eigenvector component to each design variable you have defined.

Table 26-1. EIGENVECTOR SENSITIVITY RESULTS

S E N S I T I V I T I E S O F R E A L E I G E N V E C T O R N O . 1

TO DESIGN VARIABLE ATIP

POINT-ID	TYPE	T1	T2	T3	R1	R2	R3
1	GRID	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
2	GRID	0.000000E+00	0.000000E+00	5.175089E-03	0.000000E+00	-3.070642E-03	0.000000E+00
3	GRID	0.000000E+00	0.000000E+00	1.885824E-02	0.000000E+00	-5.672144E-03	0.000000E+00
4	GRID	0.000000E+00	0.000000E+00	3.964208E-02	0.000000E+00	-7.804788E-03	0.000000E+00
5	GRID	0.000000E+00	0.000000E+00	5.586611E-02	0.000000E+00	-3.375410E-03	0.000000E+00
6	GRID	0.000000E+00	0.000000E+00	6.035002E-02	0.000000E+00	3.197582E-05	0.000000E+00
7	GRID	0.000000E+00	0.000000E+00	5.607883E-02	0.000000E+00	2.483164E-03	0.000000E+00
8	GRID	0.000000E+00	0.000000E+00	4.579291E-02	0.000000E+00	4.079713E-03	0.000000E+00
9	GRID	0.000000E+00	0.000000E+00	3.186971E-02	0.000000E+00	4.964957E-03	0.000000E+00
10	GRID	0.000000E+00	0.000000E+00	1.619073E-02	0.000000E+00	5.327908E-03	0.000000E+00
11	GRID	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	5.405381E-03	0.000000E+00

26.4 SOLUTION RESULTS

There are two basic modes of output for Sensitivity Analysis results. The first is tabular output, and the second is matrix output which is suitable for post-processing. You may select one or both of these modes in a given Sensitivity Analysis. As you saw in Section 26.3.2, Sensitivity Analysis output is selected with the command:

```
SENS ( { DISP   }                                                    )
     ( { FREQ   }                                                    )
     ( { STRESS }  [ , { PRINT } ]  [ , PUNCH ]  [ , FILE ]          )  = { ALL  }
     ( { STRAIN }      { POST  }                                     )    { n    }
     ( { FORCE  }                                                    )    { NONE }
     ( { VECTOR }                                                    )
```

The sections below describe the use of this command for both output modes.

26.4.1 Tabular Output

When you select the **PRINT**, **PUNCH** or **POST** option of the **SENS** command, the following output tables are created:

- Current Constraint Values (Freeform entity **OCONS1**)
- Sensitivities of Constraints to Design Variables (Freeform entities **OSENS1** and **OSENS2** — **SORT1** and **SORT2**, respectively
- Sensitivities of Selected Eigenvector Components to Design Variables (Freeform entity **ODPHI1 – SORT1** only

As usual, the default output media is **PRINT**. You may also use these entities directly in a post-processor by using the **OUTPUT2** interface module.

26.4.2 Matrix Output

In addition to, or in place of, the tabular output, you may specify that any of the available output is to be processed in matrix form. This is done by selecting the **FILE** option of the **SENS** command. If you select this option, the following matrices are created:

- Current Constraint Values (Matrix entity **GCONTU**)
- Sensitivities of Constaints to Design Variables (Matrix entity **SENSTU**)
- Sensitivities of Selected Eigenvector Components to Design Variables (Matrix entity **DPHIGU**)

These matrices are then automatically processed by the **OUTPUT4** module and written to the logical file **DESSEN**. You must **ASSIGN** a file with the logical name of **DESSEN** and specify **USE=OUTPUT4** if you use this option. The following sections show the format of these three matrices.

26.4.2.1 Constraint Values

The Matrix entity **GCONTU** contains the values of the design constraints. It contains one row which includes all constraint types. The organization of **GCONTU** is shown schematically as:

SUBCASE 1	Displacements sorted by assigned identification number.
	Stresses sorted by assigned identification number.
	Forces sorted by assigned identification number.
	Strains sorted by assigned identification number.
	Frequencies sorted by assigned identification number.
SUBCASE 2	Displacements sorted by assigned identification number.
	Stresses sorted by assigned identification number.
	Forces sorted by assigned identification number.
	Strains sorted by assigned identification number.
	Frequencies sorted by assigned identification number.
...	...
SUBCASE n	Displacements sorted by assigned identification number.
	Stresses sorted by assigned identification number.
	Forces sorted by assigned identification number.
	Strains sorted by assigned identification number.
	Frequencies sorted by assigned identification number.

26.4.2.2 Constraint Sensitivities

The Matrix entity **SENSTU** contains the design constraint sensitivities. It contains one row for each design variable that has been defined, and one column for each constraint which appears in the **GCONTU** entity. The order of the columns is the same as that shown in the previous section.

26.4.2.3 Eigenvector Sensitivities

A final Matrix entity, **DPHIGU**, is created only if you are performing a normal modes analysis and are computing eigenvector sensitivities. It contains one column for each design variable for each mode that you have constrained with **DCMODE** Bulk Data entries. Each row of this matrix is a degree of freedom in the set that you selected with the Case Control command:

```
SENS(VECTOR,FILE) = n
```

The format of **DPHIGU** is:

		MODE 1				MODE 2				...	MODE m			
		v_1	v_2	...	v_n	v_1	v_2	...	v_n	...	v_1	v_2	...	v_n
GRID 1	T1	...	...	...	...	...	...	...	...	...	...	...	...	...
	T2	...	...	...	...	...	...	...	...	...	...	...	...	...
	T3	...	...	...	...	...	...	...	...	...	...	...	...	...
	R1	...	...	...	...	...	...	...	...	...	...	...	...	...
	R2	...	...	...	...	...	...	...	...	...	...	...	...	...
	R3	...	...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
GRID k	T1	...	...	...	...	...	...	...	...	...	...	...	...	...
	...	...	...	...	...	...	...	...	...	...	...	...	...	...
	R3	...	...	...	...	...	...	...	...	...	...	...	...	...

The following examples illustrate the flexibility of this command in selecting output.

1. Compute displacement constraint sensitivities for all GRID points and stress constraint sensitivities for elements whose identification numbers are between **12000** and **12300**. Do not print the displacement sensitivities.

```
SET 1 = 12000 THRU 12300
SENS(DISP,POST) = ALL
SENS(STRESS) = 1
```

2. Compute and print all element force sensitivities. Additionally, request that the **GCONTU** and **SENSTU** Matrix entities be written on the file named **MYFILE** using the **OUTPUT4** module.

```
ASSIGN DESSEN='MYFILE',...,USE=OUTPUT4
...
SENS(FORCE,PRINT,FILE) = ALL
```

3. Define a set with an identification number of **999**, which selects GRID points in the ranges **1** through **100** and **1001** through **1100**. Compute the displacement constraint sensitivities for these points and request that the **GCONTU** and **SENSTU** Matrix entities be written on the file named **SAVFILE** using the **OUTPUT4** module. Additionally, write the displacement sensitivities to the **PUNCH** file. Do not **PRINT** any of these results.

```
ASSIGN DESSEN='SAVFILE',...,USE=OUTPUT4
...
SET 999 = 1 THRU 100, 1001 THRU 1100
SENS(DISP,FILE)   = 999
SENS(DISP,PUNCH)  = 999
```

4. Compute eigenvector sensitivites for all GRID points in the ranges **34** through **77** and **990** through **1008**. Write them to a file named **MYFILE** using the **OUTPUT4** module.

```
ASSIGN DESSEN='MYFILE',...,USE=OUTPUT4
...
SET 100 = 34 THRU 77, 990 THRU 1008
SENS(VECTOR,FILE) = 100
```

26.5 EXAMPLE PROBLEMS

This section provides you with four examples of how design sensitivities may be used to assist you in refining your design. The first example illustrates the resizing of plate thickness to satisfy constraints on displacement and stress, the second and third show how design variable linking can be used in a static analysis, the fourth illustrates how you may *tune* a natural frequency of a structure, and the final example shows how an eigenvector may be *shaped*. In all of these example problems scaled sensitivities have been computed by using the Bulk Data entry:

PARAM	**SENSVAL**	**SCALED**							

26.5.1 Static Analysis

This section provides you with two examples of how sensitivities may be used to assist you is designing a structure subject to static response constraints. The first example illustrates the use of displacement and stress constraints and the second the use of linked design variables.

26.5.1.1 Displacement and Stress Contraints

For this example problem, a model will be designed to identically satisfy a displacement constraint while insuring that stress constraints are not violated.

Example Problem 26-1

A rectangular flat plate with fixed edges is subjected to a uniform pressure of 2000 psi is the downward z-direction. You wish to determine the minimum thickness that will result in a deformation that does not exceed 0.1 inches in the negative z direction while insuring that the maximum normal-y stress does not exceed 500 ksi. Because of symmetry considerations, only one-quarter of the plate is modeled.

The finite element model and initial physical characteristics are shown in Figure 26-1. You may find this model in file **SA1BAS**. The model is comprised of 20 QUAD8 elements. GRID point **405** is highlighted in the figure because this is the point of maximum deformation and GRID point **5** is highlighted because it is the point of maximum stress. Note that because the maximum stress occurs near the boundary that a local mesh refinement has been made to achieve reasonable results.

To solve this problem, you must specify a single design variable, the thickness of the plate, and the two response constraints. The Bulk Data entries required to do this are:

DVPROP	**THICK**	**PSHELL**	**1**			**T**			
DCGRID	**1**	**U405**	**DISP**	**T3**	**-0.1**			**405**	
DCELEM	**1**	**SIG5**	**STRESS**	**SIGY**	**-5.0+5**			**2002**	

The example problem is then executed in **UAI/*NASTRAN***. The displacement results, the constraint evaluation and the sensitivity computations are shown in Table 26-2. You will notice in Table 26-2b that both of the constraints are marked ***violated***. Then, reviewing Table 26-2c, you note that the sensitivity of the displacement to the thickness is twice that of the stress to thickness, and that they both decrease with increasing thickness. From this, you realize that resizing using the displacement sensitivity will yield an improved design faster than using the stress sensitivity.

Figure 26-1. STATIC ANALYSIS DESIGN MODEL

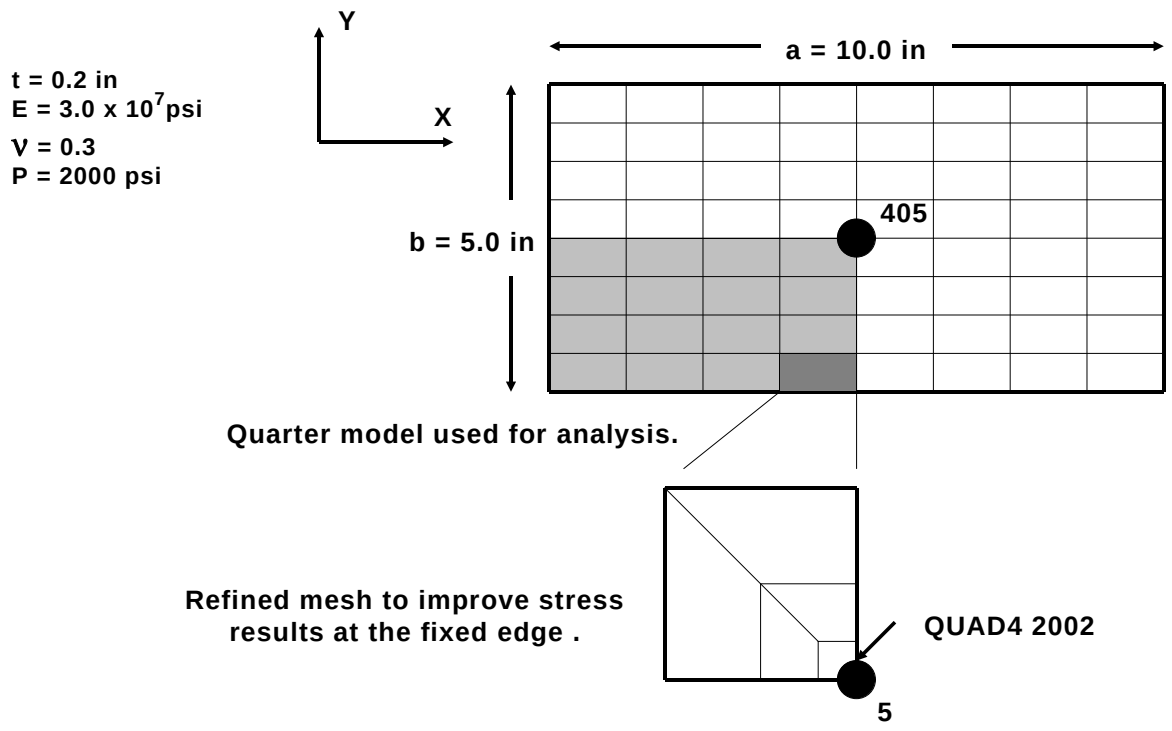


Table 26-2. STATIC ANALYSIS SENSITIVITY RESULTS

a. Displacement at GRID Point 405

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
405	GRID	0.0	0.0	-1.438776E-01	0.0	0.0	0.0

b. Constraint Values

D I S P L A C E M E N T C O N S T R A I N T V A L U E S

SET-ID	LABEL	GRID-ID	COMPONENT	BOUND	CONSTRAINT VALUE	
1	U405	405	T3	LOWER	4.38775E-01	VIOLATED

S T R E S S C O N S T R A I N T V A L U E S

SET-ID	LABEL	ELEM-ID	QUANTITY	BOUND	CONSTRAINT VALUE	
1	SIG5	2002	SIGY	LOWER	8.04494E-03	VIOLATED

c. Sensitivities of Constraints to Design Variable U405

D E S I G N C O N S T R A I N T S E N S I T I V I T I E S

CONSTRAINT INFORMATION					VARIABLE	SENSITIVITY	VARIABLE	SENSITIVITY
SET-ID	LABEL	TYPE	ID	QUANTITY	BOUND	NAME	VALUE	NAME	VALUE
1	U405	DISP	405	T3	LOWER	THICK	-2.16939E+01		

CONSTRAINT INFORMATION					VARIABLE	SENSITIVITY	VARIABLE	SENSITIVITY
SET-ID	LABEL	TYPE	ID	QUANTITY	BOUND	NAME	VALUE	NAME	VALUE
1	SIG5	STRESS	2002	SIGY	LOWER	THICK	-1.03625E+01		

26

Therefore, you compute a new thickness value from the displacement constraint using the resizing relations:

$$\Delta t = -g_{U405}\left[\frac{\partial g_{U405}}{\partial t}\right]^{-1} = \frac{-0.4388}{-21.694} = 0.0202$$

$$t_{new} = t_{old} + \Delta t = 0.2202 \text{ in}$$

The **PSHELL** Bulk Data entry which defines the thickness of the plate is then modified replacing t_{old} with t_{new}. This process is repeated until you achieve the desired displacement value within an acceptable tolerance. The example problem data for the design iteration jobs is found in files **SA1IT1**, and **SA1IT2**. It takes a total of three executions to arrive at a constraint value that is within 0.3% of the target. A summary of these runs is presented in the table below.

RUN	t_{old}	u_{U405}	σ^{y}_{SIG5}	g_{U405}	$\frac{\partial g_{U405}}{\partial t}$	Δt	t_{new}
1	0.2000	-0.1439	-504 KSI	0.4388	-21.694	0.0202	0.2202
2	0.2202	-0.1077	-413 KSI	0.0767	-14.750	0.0052	0.2254
3	0.2254	-0.1003	-394 KSI	–	–	–	–

The converged solution occurs at a thickness value of 0.2254 in. You will also note that the required stress constraint is satisfied for this thickness.

An analytical solution to this problem is presented in [Young89]. The maximum deformation at GRID point **405** and the maximum normal-y stress at GRID point **5**, both as a function of the plate thickness, are:

$$u_{U405} = \frac{-0.0277 P b^4}{E t^3} = \frac{-1.1542 \times 10^{-3}}{t^3}$$

$$\sigma^{y}_{SIG5} = \frac{-0.4974 P b^2}{t^2} = \frac{24870.0}{t^2}$$

from these equations, the thickness needed to result in a displacement of 0.1 is simply:

$$t_u = \left[\frac{-5.7708 \times 10^{-4}}{-0.1}\right]^{1/3} = 0.2254 \text{ in}$$

and that the maximum stress at that thickness is:

$$\sigma^{y}_{SIG5} = \frac{-24870.0}{0.225^2} = -491.3 \text{ ksi}$$

The **UAI/*NASTRAN*** displacement result is within 0.3% of the exact solution, and the maximum stress is in reasonable agreement for the level of mesh refinement in the model.

26.5.1.2 Linking of Design Variables

This section presents two examples which provide simple, but useful, illustrations of how you may define dependency linking of physical design variables. In the first example, the design variables are totally free to assume the best values that satisfy the specified constraints. The second example imposes a fixed value to one of the physical design variables. Together, these problems show the **UAI/*NASTRAN*** linking capabilities.

Example Problem 26-2

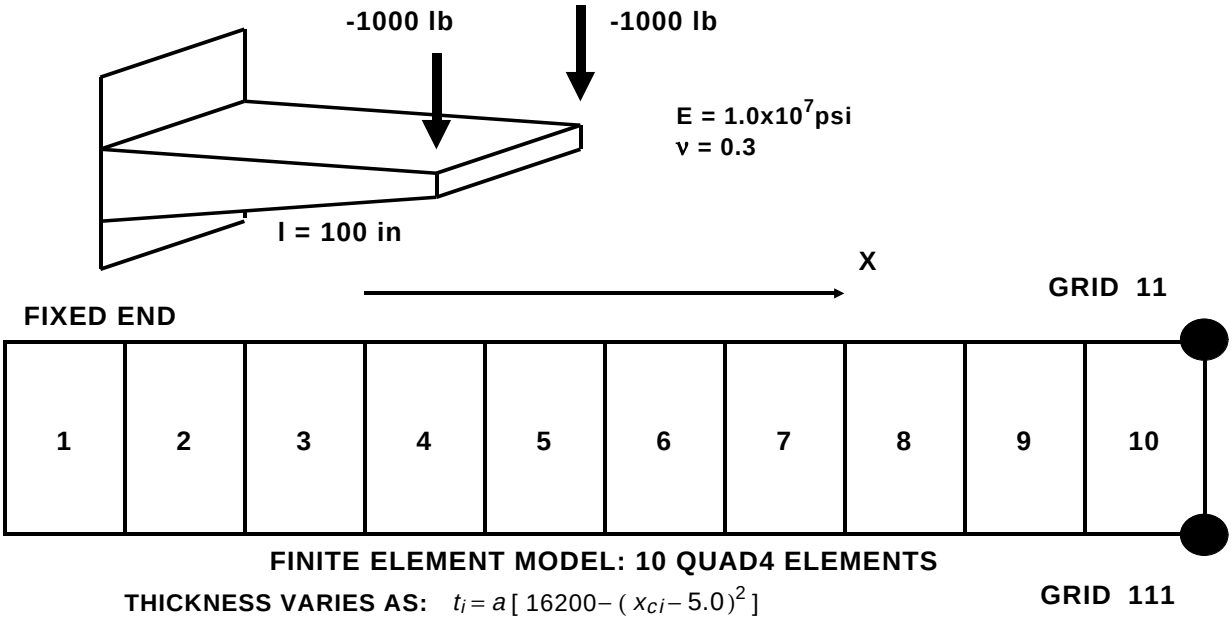
A plate which is 100 in long and 10 in wide is fixed to a wall and subjected to two point loads as shown in Figure 26-2. The thickness of the plate varies quadratically in the x-direction as:

$$t_i = a\left[\, 16200 - (\, x_{ci} - 5.0\,)^{\,2}\,\right]$$

where x_{ci} represents the x coordinate of the centroid of the i^{th} element. The mathematical design variable a is given an initial value of 2.469×10^{-4} which makes the initial thickness at the root of the plate, Element 1, 4.0 in and that at the tip, Element 10, 2.0 in. The design problem is: determine the minimum thickness of the beam such that the maximum tip deflection does not exceed -2.0 in, within ±1%, in the z-direction.

The finite element model is composed of 10 QUAD4 elements as shown in Figure 26-2.

Figure 26-2. LINKED VARIABLE DESIGN MODEL



Each **QUAD4** element references a separate **PSHELL** Bulk Data entry which specifies the original thickness value. This corresponds to the initial design. You may then design such a plate by defining the single mathematical design variable, *a*.

The thicknesses for each element can be obtained through the dependency linking relationship, ***T***, which is defined as:

$$\boldsymbol{t} = \boldsymbol{T}\boldsymbol{a}$$

which, for this case is:

$$\begin{Bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \\ t_8 \\ t_9 \\ t_{10} \end{Bmatrix} = \begin{Bmatrix} 16200 \\ 16100 \\ 15800 \\ 15300 \\ 14600 \\ 13700 \\ 12600 \\ 11300 \\ 9800 \\ 8100 \end{Bmatrix} a$$

The linked design variable is defined using a **DVLINK** Bulk Data entry:

DVLINK	**A**	**2.469-4**	**0.0**	**16200.**	**T1**	**0.0**	**16100.**	**T2**	
			0.0	**15800.**	**T3**	**0.0**	**15300.**	**T4**	
			0.0	**14600.**	**T5**	**0.0**	**13700.**	**T6**	
			0.0	**12600.**	**T7**	**0.0**	**11300.**	**T8**	
			0.0	**9800.**	**T9**	**0.0**	**8100.**	**T10**	

This entry indicates that the design variable named **A** links the physical properties defined by **DVPROP** entries. These entries, shown below, then reference the **PSHELL** Bulk Data entries for each of the plates and define the dependency linking coefficients.

DVPROP	**T1**	**PSHELL**	**1**			**T**			
DVPROP	**T2**	**PSHELL**	**2**			**T**			
DVPROP	**T3**	**PSHELL**	**3**			**T**			
DVPROP	**T4**	**PSHELL**	**4**			**T**			
DVPROP	**T5**	**PSHELL**	**5**			**T**			
DVPROP	**T6**	**PSHELL**	**6**			**T**			
DVPROP	**T7**	**PSHELL**	**7**			**T**			
DVPROP	**T8**	**PSHELL**	**8**			**T**			
DVPROP	**T9**	**PSHELL**	**9**			**T**			
DVPROP	**T10**	**PSHELL**	**10**			**T**			

Finally, you must specify the design constraint on the tip deformation using a **DCGRID** Bulk Data entry.

DCGRID	1	TIPD	DISP	T3	-2.0			11	
	111								

The example file, named **SA2BAS**, is then executed using **UAI/*NASTRAN***.

The important results of this run are:

$$u_{TIPD} = -1.611$$

$$g_{TIPD} = -0.1944$$

$$\frac{\partial g_{TIPD}}{\partial a} = -9.799 \text{x} 10^{3}$$

Resizing of the design variable a is then performed using:

$$\Delta a = -g_{TIPD}\left[\frac{\partial g_{TIPD}}{\partial a}\right]^{-1} = \frac{0.1944}{-9.799 \text{x}\, 10^{3}} = -1.984 \text{x}\, 10^{-5}$$

$$\boldsymbol{t} = \boldsymbol{T}(a + \Delta a)$$

The **DVLINK** Bulk Data entry is then modified to reflect the new thicknesses and the model is executed again. This process is repeated a second time at which point the tip displacement has converged within the required tolerance. The data for these two may be found in the files **SA2IT1** and **SA2IT2**.

RUN	a_{old}	u_{TIPD}	g_{TIPD}	$\frac{\partial g_{TIPD}}{\partial a}$	Δa	a_{new}
1	2.469x10^{-4}	-1.611	-0.1944	-9.799x10^{3}	-1.984x10^{-5}	2.271x10^{-4}
2	2.271x10^{-4}	-2.071	0.0353	-1.369x10^{4}	2.576x10^{-6}	2.297x10^{-4}
3	2.297x10^{-4}	-2.001	–	–	–	–

The quadratic shape is maintained through the design and, when converged, results in a root thickness of 3.721 in and a tip thickness of 1.861 in. The element thicknesses are shown in the next table.

RUN	ELEMENT NUMBER									
	1	2	3	4	5	6	7	8	9	10
1	4.000	3.975	3.901	3.778	3.605	3.383	3.111	2.790	2.419	2.000
2	3.679	3.656	3.588	3.475	3.316	3.111	2.861	2.566	2.226	1.840
3	3.721	3.698	3.629	3.514	3.354	3.147	2.894	2.596	2.251	1.861

Note that the volume of the initial design is 3296 in^3 whereas the redesigned plate has a volume of only 3065 in^3. Your redesign is able to reduce the weight by 7%.

The next example is similar in nature, but it illustrates the use of a fixed contribution to the linked design variable.

Example Problem 26-3

A flat plate is fixed to a wall and subjected to two point loads as shown in Figure 26-3. For the initial design, the plate has a rectangular cross section as illustrated. The design problem is: determine the amount of uniform, linear taper, if any, that the plate may have so that the maximum tip deflection does not exceed -1.0, within ±1%, in the z-direction. The cross section at the root must remain the same. This will naturally reduce the weight and material requirements for the design.

As in the previous example, the finite element model is composed of 10 QUAD4 elements as shown in Figure 26-3. Each element references a separate **PSHELL** Bulk Data entry which specifies a thickness value of 3.175 corresponding to the initial design. Since you wish to design a linearly tapering plate, a single design variable, *m*, which represents the slope of the taper of the thickness, is all that is required. The thicknesses for each element are obtained through the linking relationship.

$$\boldsymbol{t} = \boldsymbol{h}_{init}^{inv} + \boldsymbol{T}m$$

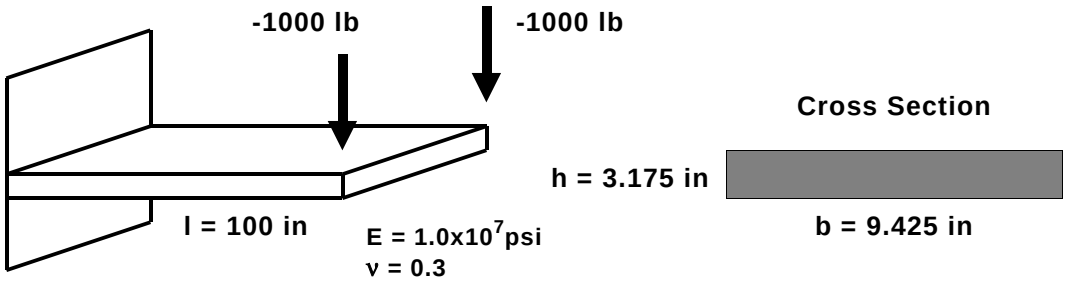
These thicknesses will override any values that you have entered on the **PSHELL** Bulk Data entries. In this case, the thicknesses are:

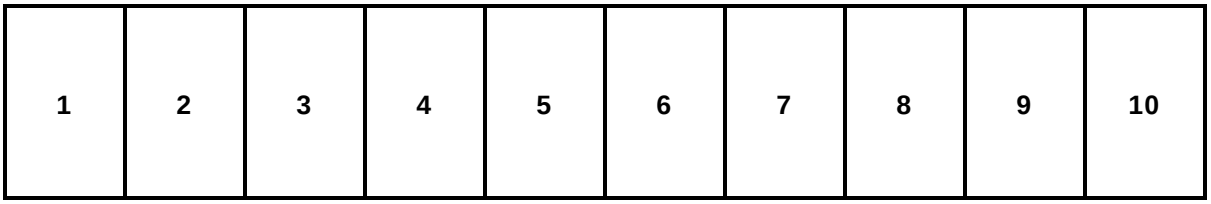
$$\begin{Bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \\ t_8 \\ t_9 \\ t_{10} \end{Bmatrix} = \begin{Bmatrix} 3.175 \\ 3.175 \\ 3.175 \\ 3.175 \\ 3.175 \\ 3.175 \\ 3.175 \\ 3.175 \\ 3.175 \\ 3.175 \end{Bmatrix} + \begin{Bmatrix} 0.0 \\ 10.0 \\ 20.0 \\ 30.0 \\ 40.0 \\ 50.0 \\ 60.0 \\ 70.0 \\ 80.0 \\ 90.0 \end{Bmatrix} m$$

The linked design variable is defined using a **DVLINK** Bulk Data entry:

DVLINK	**M**	**-.001**	**3.175**	**0.0**	**T1**	**3.175**	**10.**	**T2**	
			3.175	**20.**	**T3**	**3.175**	**30.**	**T4**	
			3.175	**40.**	**T5**	**3.175**	**50.**	**T6**	
			3.175	**60.**	**T7**	**3.175**	**70.**	**T8**	
			3.175	**80.**	**T9**	**3.175**	**90.**	**T10**	

Figure 26-3. LINKED VARIABLE DESIGN MODEL





FINITE ELEMENT MODEL: 10 QUAD4 ELEMENTS

This entry indicates that the design variable named **M** links the physical properties defined by **DVPROP** entries and that its initial value is -0.001. These entries, shown below, then reference the **PSHELL** Bulk Data entries for each of the plates and define the linking coefficients.

DVPROP	**T1**	**PSHELL**	**1**			**T**			
DVPROP	**T2**	**PSHELL**	**2**			**T**			
DVPROP	**T3**	**PSHELL**	**3**			**T**			
DVPROP	**T4**	**PSHELL**	**4**			**T**			
DVPROP	**T5**	**PSHELL**	**5**			**T**			
DVPROP	**T6**	**PSHELL**	**6**			**T**			
DVPROP	**T7**	**PSHELL**	**7**			**T**			
DVPROP	**T8**	**PSHELL**	**8**			**T**			
DVPROP	**T9**	**PSHELL**	**9**			**T**			
DVPROP	**T10**	**PSHELL**	**10**			**T**			

Finally, you must specify the design constraint on the tip deformation using a **DESCON** Bulk Data entry.

DCGRID	**1**	**TIPD**	**DISP**	**T3**	**-1.0**			**11**	
	111								

26

The example file, named **SA3BAS**, is then executed using **UAI/*NASTRAN***. Once again, the results needed to modify the design are:

$$u_{TIPD} = -0.894 \text{ and } g_{TIPD} = -0.1239 \text{ and } \frac{\partial g_{TIPD}}{\partial m} = -17.41$$

Resizing of the design variable h is then performed using:

$$\Delta m = -g_{TIPD}\left[\frac{\partial g_{TIPD}}{\partial m}\right]^{-1} = \frac{0.1239}{-17.42} = -0.0071$$

from which:

$$m_{new} = m_{old} + \Delta m = (-0.001) + (-0.0071) = -0.0081$$

The linking relationship is then used to compute the new physical design variables:

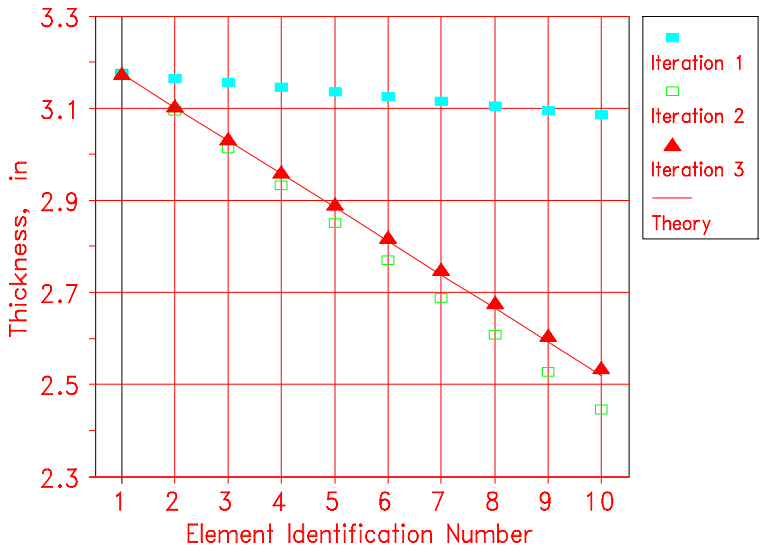
$$\boldsymbol{t}_{new} = \boldsymbol{h}_{init}^{inv} + \boldsymbol{T} m_{new}$$

The **DVLINK** Bulk Data entry is then modified to reflect the new slope and the model is executed again. This process is repeated a second time at which point the tip displacement has converged within the required tolerance. The exact value of the thickness at the free end, derived from [Young89] is given by:

$$h = \left[0.579\frac{4Pl^3}{Ebu}\right]^{1/3} = 2.52\,\text{in}$$

Figure 26-4 presents the resulting element thicknesses of the basic run and the two design iterations which are found in the files **SA3IT1** and **SA3IT2**. Again, the **UAI/*NASTRAN*** results are in excellent agreement with the reference theoretical solution.

Figure 26-4. DESIGN ITERATIONS FOR TAPERED PLATE



26.5.2 Normal Modes Analysis

As you have seen, you may specify constraints on modal quantities including the eigenvalue (frequency) and eigenvector. You use constraint sensitivities from a Normal Modes analysis in the same manner as those for constraints on static response variables. The two example problems described in the following sections illustrate how a structure may be tuned to a frequency constraint, and how a mode shape may be altered.

26.5.2.1 Frequency Constraints

It is often desirable that the natural modes of a structure be within certain frequency ranges to avoid being excited by other behaviors within a product. This example illustrates how sensitivity analysis can be used for this purpose.

Example Problem 26-4
Redesign a simply supported, square plate of uniform thickness such that its first cyclic frequency is 5.0 Hz.

The finite element model and initial physical characteristics are shown in Figure 26-5. The model in found in file **SA4BAS**. The model is built from 64 QUAD4 elements, each of which is square. The design variable, which is once again the plate thickness, is defined with a **DVPROP** Bulk Data entry:

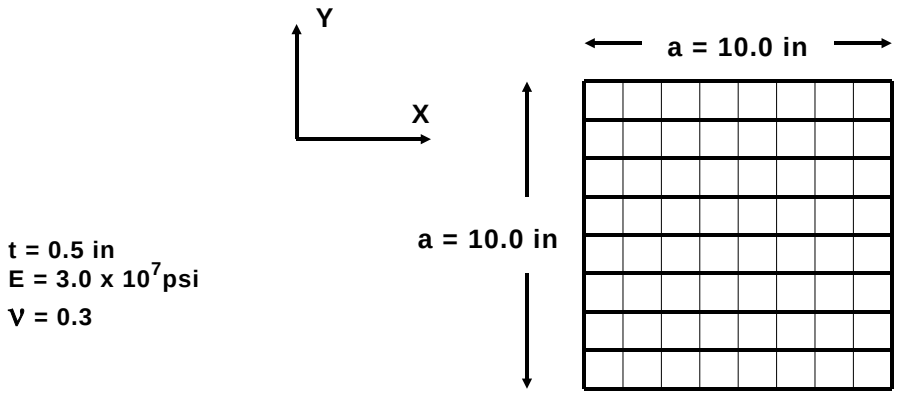
DVPROP	**THICK**	**PSHELL**	**1**			**T**			

You then must specify the frequency constraint:

DCFREQ	**1**	**F1**	**LOWER**	**5.0**	**1**				

Note that the **DCFREQ** entry for a frequency constraint differs from static response constraints in that no component of response is specified — simply the frequency value in Hz.

Figure 26-5. NORMAL MODES ANALYSIS DESIGN MODEL



After the example problem is executed in **UAI/*NASTRAN***, the following results, shown in Table 26-3, are obtained:

$$f_1 = 3.268\,\text{Hz}$$

$$g_{F1} = 0.5728$$

$$\frac{\partial g_{F1}}{\partial t} = -1.712$$

Note that Table 26-3b indicates that in this case the frequency constraint is *violated*. The usual resizing relationships are used to determine the new value of thickness needed to move closer to the required frequency:

$$\Delta t = -g\left[\frac{\partial g_{F1}}{\partial t}\right]^{-1} = \frac{-0.5728}{-1.712} = 0.335$$

and then:

$$t_{new} = t_{old} + \Delta t = 0.835\,\text{in}$$

Table 26-3. SOLUTION RESULTS FOR MODAL SA4BAS

a. First Eigenvector

R E A L E I G E N V A L U E S

MODE NO.	EXTRACTION ORDER	EIGENVALUE	RADIAN FREQUENCY	CYCLIC FREQUENCY	GENERALIZED MASS	GENERALIZED STIFFNESS
1	145	4.215976E+02	2.053284E+01	3.267903E+00	1.932779E+02	8.148549E+04
2	147	2.623604E+03	5.122112E+01	8.152095E+00	0.000000E+00	0.000000E+00
3	146	2.623604E+03	5.122112E+01	8.152095E+00	0.000000E+00	0.000000E+00
4	144	6.401722E+03	8.001077E+01	1.273411E+01	0.000000E+00	0.000000E+00

b. Frequency Constraint for Design Variable F1

F R E Q U E N C Y C O N S T R A I N T V A L U E S

SET-ID	LABEL	MODE-NO	BOUND	CONSTRAINT VALUE	
1	FREQ	1	LOWER	5.72832E-01	VIOLATED

c. Constraint Sensitivity to Design Variable THICK

D E S I G N C O N S T R A I N T S E N S I T I V I T I E S

CONSTRAINT SET-ID	CONSTRAINT LABEL	CONSTRAINT TYPE	INFORMATION ID	INFORMATION QUANTITY	INFORMATION BOUND	VARIABLE NAME	SENSITIVITY VALUE	VARIABLE NAME	SENSITIVITY VALUE
1	FREQ	FREQ	1			THICK	-1.71199E+00		

You then replace the value of the design variable, *t*, with the new value and rerun **UAI/*NASTRAN***. The table below shows the design history which converged in two iterations. The data for these iterations are found in the files **SA4IT1** and **SA4IT2**.

RUN	t in	f_1 Hz	g_{F1}	$\frac{\partial g_{F1}}{\partial t}$	Δt	t_{new}
1	0.5	3.268	0.5728	-1.712	0.335	0.835
2	0.835	5.458	-0.1916	-2.859	-0.067	0.768
3	0.768	5.020	–	–	–	–

The first mode of vibration for this model is found in [Young89]. As a function of the thickness, it is:

$$f_1 = \pi\left[\frac{gEt^2}{12(1-\nu^2)\,a^6\rho}\right]^{1/2} = 6.50892t$$

Since the frequency of the first mode is simply a linear function of thickness, you will note that the thickness required to satisfy the frequency constraint is:

$$t = \frac{f_1}{6.50892} = \frac{5.0}{6.50892} = 0.768\text{in}$$

You will note the identical agreement between this solution and the **UAI/*NASTRAN*** result.

26.5.2.2 Changing Eigenvector Response

UAI/*NASTRAN* also provides the capability to determine the sensitivity of one or more eigenvectors to your design variables. As stated earlier, in this case it is the actual response sensitivities that are provided — not constraint sensitivities. The first demonstration problem is found in file **SA5BAS**. It contains the baseline model as it would be executed for this simple modal analysis.

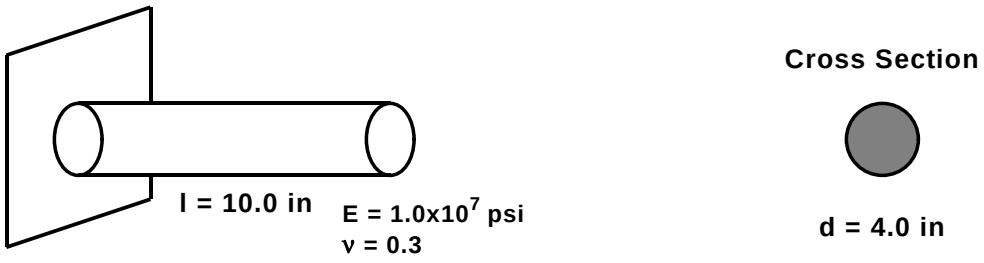
Example Problem 26-5

Consider the model shown in Figure 26-6. It is a simple cantilever beam which is composed of 10 **BAR** elements with circular cross sections having a constant diameter of 4.0 in. The design problem to be solved is: determine **BAR** cross sectional areas, subject to the rules set forth below, that make the first mode shape *flatter* in the sense that the relative deformation in the z-direction at GRID point **6** is 50% of its baseline value. Iterate to within a tolerance of 5% of the exact result.

26

By using your engineering judgement, you expect that the mode shape will be flatter if the stiffness near the fixed end, or root, of the beam is increased. You choose to do this by defining two design variables. The first design variable, **AROOT**, will represent the area of the first three beam elements by reference to the diameter parameter of the **PBAR1** Bulk Data entry defining the element properties, while the second design variable, **ATIP**, will represent the area of the other seven elements in the same manner. Although you could use design variable linking relationships such as those which were used in Examples 26-2 or 26-3, this is

Figure 26-6. PROBLEM FOR EIGENVECTOR SENSITIVITY



not necessary because the three root elements are defined to have the same area as are the seven tip elements. Therefore, they may be designed by simply using the same **PBAR1** data for each of the element groups. The finite element model is shown in Figure 26-7. The **DVPROP** data used to define the two physical design variables is:

DVPROP	**AROOT**	**PBAR1**	**1**			**D1**			
DVPROP	**ATIP**	**PBAR1**	**2**			**D1**			

Note that the design variable in each case is **D1** — which is the symbol for the diameter of the **BAR** cross section. The eigenvector sensitivity request is specified using a **DCMODE** Bulk Data entry:

DCMODE	**1**	**MODE1**	**1**						

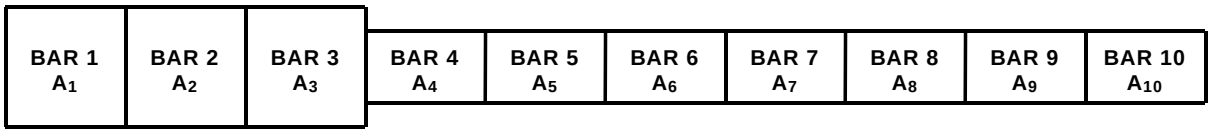
The only other Sensitivity Analysis input required is the following Case Control commands:

```
DESCON = ALL
SENS(VECTOR) = ALL
```

The first command requests that sensitivities be computed for all of the constraints that you have specified with respect to each of the defined design variables. The second command requests that the eigenvector sensitivity be printed for all of the GRID points in the model.

If you then execute the data file **SA5BAS**, you will get the results shown in Table 26-4. Looking at the eigenvector sensitivity of the first mode with respect to design variable **AROOT** you will note that all values are negative. This means that if **AROOT** is increases, that the relative

Figure 26-7. EIGENVECTOR SENSITIVITY FE MODEL



FINITE ELEMENT MODEL
10 BAR ELEMENTS

Table 26-4. SOLUTION RESULTS FOR EIGENVECTOR SA

a. First Eigenvector

MODE NUMBER = 1 FREQUENCY = 3.365206E+01 HZ. EIGENVALUE = 4.470778E+04

REAL EIGENVECTOR NO. 1

POINT ID.	TYPE	T1	T2	T3	R1	R2	R3
1	GRID	0.0	0.0	0.0	0.0	0.0	0.0
2	GRID	0.0	0.0	1.808139E-02	0.0	-1.076998E-02	0.0
3	GRID	0.0	0.0	6.608705E-02	0.0	-1.995540E-02	0.0
4	GRID	0.0	0.0	1.392653E-01	0.0	-2.757320E-02	0.0
5	GRID	0.0	0.0	2.329395E-01	0.0	-3.366470E-02	0.0
6	GRID	0.0	0.0	3.425974E-01	0.0	-3.830476E-02	0.0
7	GRID	0.0	0.0	4.640048E-01	0.0	-4.160919E-02	0.0
8	GRID	0.0	0.0	5.933390E-01	0.0	-4.374026E-02	0.0
9	GRID	0.0	0.0	7.273358E-01	0.0	-4.491069E-02	0.0
10	GRID	0.0	0.0	8.634465E-01	0.0	-4.538618E-02	0.0
11	GRID	0.0	0.0	1.000000E+00	0.0	-4.548677E-02	0.0

b. Eigenvector Sensitivites for Design Variable AROOT

SENSITIVITIES OF REAL EIGENVECTOR NO. 1

TO DESIGN VARIABLE AROOT $\frac{\partial \varphi_1}{\partial v_1}$

POINT-ID	TYPE	T1	T2	T3	R1	R2	R3
1	GRID	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
2	GRID	0.000000E+00	0.000000E+00	-4.501475E-03	0.000000E+00	3.127410E-03	0.000000E+00
3	GRID	0.000000E+00	0.000000E+00	-1.767484E-02	0.000000E+00	5.778840E-03	0.000000E+00
4	GRID	0.000000E+00	0.000000E+00	-3.810122E-02	0.000000E+00	7.954323E-03	0.000000E+00
5	GRID	0.000000E+00	0.000000E+00	-5.411261E-02	0.000000E+00	3.560449E-03	0.000000E+00
6	GRID	0.000000E+00	0.000000E+00	-5.852452E-02	0.000000E+00	1.811185E-04	0.000000E+00
7	GRID	0.000000E+00	0.000000E+00	-5.432074E-02	0.000000E+00	-2.249337E-03	0.000000E+00
8	GRID	0.000000E+00	0.000000E+00	-4.424351E-02	0.000000E+00	-3.832008E-03	0.000000E+00
9	GRID	0.000000E+00	0.000000E+00	-3.067560E-02	0.000000E+00	-4.709339E-03	0.000000E+00
10	GRID	0.000000E+00	0.000000E+00	-1.550810E-02	0.000000E+00	-5.068947E-03	0.000000E+00
11	GRID	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	-5.145683E-03	0.000000E+00

c. Eigenvector Sensitivites for Design Variable ATIP

SENSITIVITIES OF REAL EIGENVECTOR NO. 1

TO DESIGN VARIABLE ATIP

POINT-ID	TYPE	T1	T2	T3	R1	R2	R3
1	GRID	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
2	GRID	0.000000E+00	0.000000E+00	5.175089E-03	0.000000E+00	-3.070642E-03	0.000000E+00
3	GRID	0.000000E+00	0.000000E+00	1.885824E-02	0.000000E+00	-5.672144E-03	0.000000E+00
4	GRID	0.000000E+00	0.000000E+00	3.964208E-02	0.000000E+00	-7.804788E-03	0.000000E+00
5	GRID	0.000000E+00	0.000000E+00	5.586611E-02	0.000000E+00	-3.375410E-03	0.000000E+00
6	GRID	0.000000E+00	0.000000E+00	6.035002E-02	0.000000E+00	3.197582E-05	0.000000E+00
7	GRID	0.000000E+00	0.000000E+00	5.607883E-02	0.000000E+00	2.483164E-03	0.000000E+00
8	GRID	0.000000E+00	0.000000E+00	4.579291E-02	0.000000E+00	4.079713E-03	0.000000E+00
9	GRID	0.000000E+00	0.000000E+00	3.186971E-02	0.000000E+00	4.964957E-03	0.000000E+00
10	GRID	0.000000E+00	0.000000E+00	1.619073E-02	0.000000E+00	5.327908E-03	0.000000E+00
11	GRID	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	5.405381E-03	0.000000E+00

26

motion will be decreased — which is the target for this example. On the other hand, the sensitivities with respect to **ATIP** are positive, indicating that an increase in area will actually increase the relative motion. Therefore, you must increase the cross sectional area for **AROOT** or decrease the value of **ATIP**. Resizing a design variable based on a target response is simpler even than constraint satisfaction. The following general relationship is used:

$$\Delta R = \frac{\partial R}{\partial v}\Delta v \text{ or } \Delta v = \Delta R\left[\frac{\partial R}{\partial v}\right]^{-1}$$

From which, the new value of the design variable is simply:

$$v_{new} = v_{old} + \Delta v$$

For this example problem, the root area is resized using:

$$\Delta A_{ROOT} = [\, \varphi_1^{6-target} - \varphi_1^6 \,] \left[\frac{\partial \varphi_1^6}{\partial A_{ROOT}} \right]^{-1}$$

When you study Table 26.4a, you see that the initial relative z-displacement for GRID point **6** is 0.3426. You then set the target deformation to be 0.1713, 50% of the original value. You must resize the **D1** parameter of the **AROOT** using the corresponding sensitivity from Table 26-4b, which is -0.1. This results in:

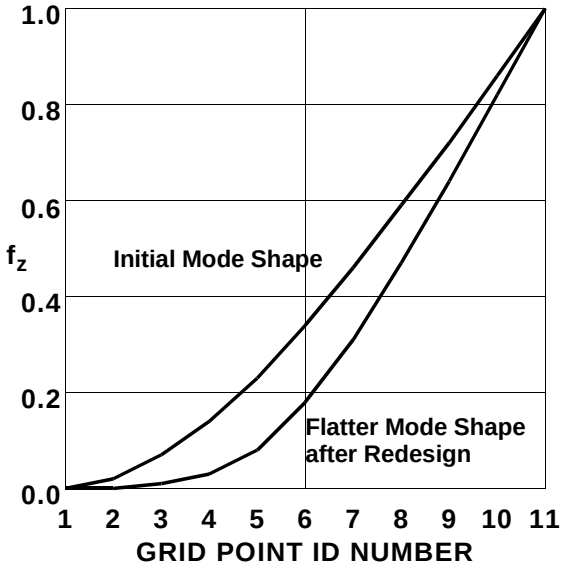
$$\Delta A_{ROOT} = \frac{(0.1713 - 0.3426)}{-0.0585} = 2.928$$

From which:

$$A_{ROOT}^{new} = 4.000\text{in} + 2.928\text{in} = 6.928\text{in}$$

You then modify the **PBAR1** Bulk Data entry used for elements **1**, **2** and **3** with the new value of **D1** and rerun the **UAI/*NASTRAN*** job. This process continues until you have satisfied your target deformation. Figure 26-8 shows the eigenvector obtained by performing four redesigns. The modal displacement at GRID point **6** is 0.1715, which is within 0.12% of the target solution. The iteration history is also shown in the figure. The data for each of these iterations are found in files **SA5IT1**, **SA5IT2**, and **SA5IT3**.

Figure 26-8. EIGENVECTOR SHAPING RESULTS



RUN	A_{ROOT}	φ_1^6	$\frac{\partial \varphi_1^6}{\partial A_{ROOT}}$	ΔA_{ROOT}	$A_{ROOT}^{new} = A_{ROOT}^{old} + \Delta A_{ROOT}$
1	4.000	0.3426	-0.0585	2.928	6.928
2	6.928	0.1957	-0.0294	0.830	7.758
3	7.758	0.1757	-0.0196	0.224	7.982
4	7.982	0.1715	–	–	–

26

Chapter 27

MESH ERROR ESTIMATES

The finite element method is a numerical method for approximating a continuous structure with a collection of discrete elements. This collection of elements is called the ***finite element model***. Because this model is only an approximation of the real structure, the solution results will differ from the exact solution. This difference is called the ***discretization*** or ***mesh error***. Large errors indicate that your model is inadequate for representing the behavior of your structure, while small errors provide evidence that your model is yielding reasonable results. **UAI/*NASTRAN*** provides you with a capability to estimate the mesh error for your model.

This chapter describes how Mesh Error Analysis is selected and how the results may be used for improving your model.

27.1 TERMINOLOGY

This section describes the terminology used in the computation of Mesh Error Estimates.

Mesh Error Field. Is the set of elements that you select for which Mesh Error Estimates will be computed. Note that mesh errors may be computed only for plate elements (TRIA3, TRIA6, TRIAR, QUAD4, QUAD8, and QUADR) and solid elements (HEXA, PENTA, and TETRA).

Global Error Estimate. Is a measure of the total error in the specified Mesh Error Field. There is a global estimate for each static loading condition or mode for which you obtain a solution.

Local Error Estimate. Is a measure of the error in an individual element within your Mesh Error Field for a specific loading condition or mode shape.

Stress Norm. Is a single stress value which is determined by combining all of the components of a three-dimensional stress tensor.

Exact Stress. Is an *a posteriori* estimate of the true stresses in the model.

The mathematical meaning of these terms and the method in which they are used are described in the following section.

27.2 MATHEMATICAL BACKGROUND

The error estimates are obtained by comparing the computed state of stress within an individual element with the estimate of the "exact" stress within the element. This "exact" stress is represented by smooth continuum stresses which are determined by computing the grid point stress values within the set of elements for which you request Mesh Error Estimates. In both cases, the stress tensor is combined, as described in the sections that follow, to give a single value called the ***stress norm***. The difference in these two stress norms is a measure of the discretization error in your model.

27.2.1 Computing the Stress Norm for an Element

During your analysis, the element stresses for each element, *e*, in the Mesh Error field, are computed at either the corner grid points of the element or at the element centroid. These stress tensors are called $\tilde{\sigma}_e^g$. Only the stress components that you select are included in these vectors. These stresses are then combined for all of the points, *np*, to yield a single stress norm for the element:

$$\sigma_e^{computed} = \left[\frac{1}{np} \sum_{g=1}^{np} \tilde{\sigma}_e^g \bullet \tilde{\sigma}_e^g \right]^{\frac{1}{2}} \tag{27-1}$$

27.2.2 Predicting the "Exact" Stresses

The fundamental requirement for computing Mesh Error Estimates is to obtain an estimate of the *exact* stress distribution within the model. In **UAI/*NASTRAN***, this is done by computing the GRID point stress values for all GRID points which are contained in the Mesh Error field. The *exact* stress tensor at each grid *g* is called $\tilde{\sigma}^g$.

27.2.3 The Estimated Element Stress

Next, the estimated smooth continuum stresses are computed for each element by using the stress vector, $\tilde{\sigma}^g$, at each corner grid point of the element. These are combined using the same relationship as above:

$$\sigma_e^{predicted} = \left[\frac{1}{np} \sum_{g=1}^{np} \tilde{\sigma}^g \bullet \tilde{\sigma}^g \right]^{\frac{1}{2}} \tag{27-2}$$

27.2.4 The Element Stress Error

The stress error percentage in element *e* can then be approximated by:

$$\eta_e = 100 \frac{\sigma_e^{computed} - \sigma_e^{predicted}}{\sigma_e^{predicted}} \tag{27-3}$$

Note that this error estimate has a signed value which depends on whether the computed stress is greater than or less than the predicted value. Additionally, the local element error does not reflect the overall importance of an element. For example, although the maximum stress in your model may be 40 Ksi, an element with an insignificant stress may show a very large error. To correct for this behavior, a normalized element error, described in the next section, is also computed.

27.2.5 The Normalized Element Error Measures

As you have seen in the previous sections, the absolute element error estimates do not reflect the importance of the stresses in an element within the context of the entire model. To correct this, first, the difference $\tilde{\delta}_e^{\,g}$ between the *exact* and the computed element stress at each grid point is obtained from:

$$\tilde{\delta}_e^{\,g} = \tilde{\sigma}^{\,g} - \tilde{\sigma}_e^{\,g} \tag{27-4}$$

Then, the estimated error norm for element *e*, $||\varepsilon||_e$, is obtained from:

$$||\varepsilon||_e = \left[\frac{1}{np} \sum_{g=1}^{np} \tilde{\delta}_e^{\,g} \bullet \tilde{\delta}_e^{\,g} \right]^{\frac{1}{2}} \tag{27-5}$$

where *np* is the number of stress points in element *e* which are also connected to other elements. By definition, the error from those stress points which are only connected to element *e* is zero. You will note that the estimated error norm for each element represents the RMS value of the components of all the estimated error vectors in that element.

27.2.5.1 Local Estimated Error Percentages

The estimated percentage error for element *e*, $\hat{\eta}_e$, is obtained using the following formulae:

$$\hat{\eta}_e = 100 \frac{||\varepsilon||_e}{\max\limits_e (\sigma_e^{predicted})} \tag{27-6}$$

As you will see later, in addition to the error estimates, the volume or area of each element is provided. This information may be useful if you wish to perform adaptive modeling using the error estimates.

27.2.5.2 Global Estimated Error Percentage

A global estimate of percentage error may be defined as the RMS value of the local estimated error percentages for all elements, *nel*, in the following manner:

$$\eta_{global} = \left[\frac{1}{nel} \sum_{e=1}^{nel} \hat{\eta}_e^2 \right]^{\frac{1}{2}} \tag{27-7}$$

nel is the number of elements for which a local estimated error percentage has been calculated.

27.2.6 Statistical Distribution

The error percentages derived herein do not possess a physical significance, and hence, may not be viewed in absolute terms without prior empirical knowledge. Therefore, statistical comparison values are necessary to establish a measure of relative importance for each individual error percentage.

27.2.6.1 Mean Error Percentage

A mean error percentage is obtained using:

$$\bar{\eta} = \frac{1}{nel} \sum_{e=1}^{nel} \hat{\eta}_e \tag{27-8}$$

nel is the number of elements for which a local estimated error percentage has been calculated.

27.2.6.2 Standard Deviation of Error Percentages

The standard deviation of all estimated local error percentages is defined by:

$$sd = \left[\frac{1}{nel} \sum_{e=1}^{nel} (\hat{\eta}_e - \bar{\eta})^2 \right]^{\frac{1}{2}} \tag{27-9}$$

27.3 INPUT DATA REQUIREMENTS

This section provides you with a description of the **UAI/*NASTRAN*** input data required when you select the Mesh Error Estimation feature.

27.3.1 Executive Control Commands

While there are no specific Executive Control commands for Mesh Error Estimates, the feature is available only when you perform Static or Normal Modes analyses. Thus, you must be using one of these disciplines individually, or as part of a Multidisciplinary Design Optimization. When performing Fluid-Structure Interaction analyses, you may request the error estimates for the structural part of a model, but the estimates are not computed (or defined) for the fluid.

27.3.2 Case Control Commands

There is a single Case Control command used for requesting the Mesh Error Estimate feature. This command, **MESHERR**, is used to select a set of elements for which error estimates will be computed and output and to perform output filtering operations on these results. You may also limit the error estimates to selected stress components, although this option is not recommended for general use. The syntax of the command is:

```
MESHERR ( [ { PRINT   } , ] [ PUNCH , ] [ [ { COMBINED    } ] [ { BOTH  } ] ]
          [ { NOPRINT } , ]                [ { stress_comp } ] [ { UPPER } ]
                                                               [ { LOWER } ]

          [ [ { PERCENTILE = pc  } ] ] ) = { ALL  }
          [ [ { EXCEED     = nsd } ] ]     { sid  }
          [ [ { THRESHOLD  = thr } ] ]     { NONE }
```

You will note that this command is similar to other output selection commands with regard to the **PRINT**, **NOPRINT** and **PUNCH** options. The remaining command parameters allow you to limit the number of stress components used for the error calculation. When you specify **COMBINED**, then the complete stress norm, as defined in (27-1), is used. If you specify a single stress component *stress_comp*, then only that component will be used in the stress norm computation. Note that this should be used with caution in that it may provide misleading results if you have a multi-dimensional stress field. Similarly, the **UPPER** and **LOWER** allows you to select the stresses in plate elements at either the upper or lower fiber location. The final parameter allows you to define a filter which restricts the amount of output that you receive. This is usually desirable in that many of the elements in a large model are in low stress regions which are not critical to the analysis and design process. Three filtering options are provided. The first is the **PERCENTILE**. When using this option, the local error estimates are printed only for those elements whose errors are in the highest *pc* percentile. You may also specify the **EXCEED** option which allows you to print only those errors which exceed *nsd* standard deviations from the mean error. Finally, you may simply specify a **THRESHOLD** value, *thr* which defines a print cut-off. Finally, you select the Mesh Error Field by either specifying **ALL** of the elements in your model or by defining a SET of elements for which the errors will be calculated. Note that in the latter case, the Global Error Estimates will be global only with respect to the element set.

When performing Multidisciplinary Design Optimization, you may specify a **MESHERR** command for each analysis case. The selected output set will be printed for every design iteration.

27.3.3 Bulk Data Entries

There are no special Bulk Data entries required for Mesh Error Estimation.

27.4 SOLUTION RESULTS

There are two sets of results from the error estimate feature. The first is the Global Error Estimate for each SUBCASE, when you perform a Statics analysis, or for each eigenvector that you compute when performing Normal Modes analysis. An example of these results is shown in Table 27-1. The second is a table of the local error estimates for each of the elements selected with the **MESHERR** command. This table is sorted and filtered using the selections that you specified in the **MESHERR** command. Each of these is described in the following sections.

27.4.1 Global Error Estimate

The global error estimates are computed using (27-7) for each of the elements defined by the SET selected with the **MESHERR** Case Control command. One estimate is computed for each static loading condition and one for each mode shape. The resulting output is shown in Table 27-1a.

27.4.2 Local Error Estimates

The local error estimates are computed for each of the elements defined by the SET selected. The resulting values are then filtered using the rule that you select. The resulting output, shown in Table 27-1b, includes the element identification number and type, its area or volume depending on the type, the normalized element error from (27-6), the element error from (27-3), and the predicted element stress norm from (27-2).

Table 27-1. MESH ERROR SOLUTION RESULTS

a. Global Error Estimates

G L O B A L M E S H E R R O R E S T I M A T E S

MODE NUMBER	GLOBAL ERROR PERCENTAGE
1	2.7
2	16.2
3	10.6
4	31.4
5	2.7
6	35.3
7	17.1
8	29.7
9	28.5
10	24.1

b. Local Error Estimates

L O C A L M E S H E R R O R E S T I M A T E S

ELEMENT NUMBER	ELEMENT TYPE	ELEMENT AREA OR VOLUME	NORMALIZED ELEMENT ERROR	ELEMENT ERROR	PREDICTED ELEMENT STRESS NORM
3	QUAD8	1.38886E+00	7.5	-30.1	1.95738E+05
2	QUAD8	1.38878E+00	5.2	-17.5	1.81010E+05
202	QUAD8	1.38878E+00	4.7	7.5	3.51297E+05
102	QUAD8	1.38894E+00	4.4	7.5	2.80102E+05
1	QUAD8	1.38886E+00	4.4	5.4	1.56725E+05
103	QUAD8	1.38903E+00	3.6	3.2	3.32517E+05
203	QUAD8	1.38886E+00	3.3	3.9	4.28057E+05
101	QUAD8	1.38903E+00	2.7	3.6	1.82378E+05
201	QUAD8	1.38886E+00	2.3	-3.2	2.04680E+05

27.5 EXAMPLE PROBLEMS

This section provides you with two examples of how you can use Mesh Error Estimates to create more accurate models of your structure. The first example illustrates the convergence characteristics of the error computation for a static analysis. The second shows how such estimates may be used to improve eigenvector accuracy when you perform normal modes analysis.

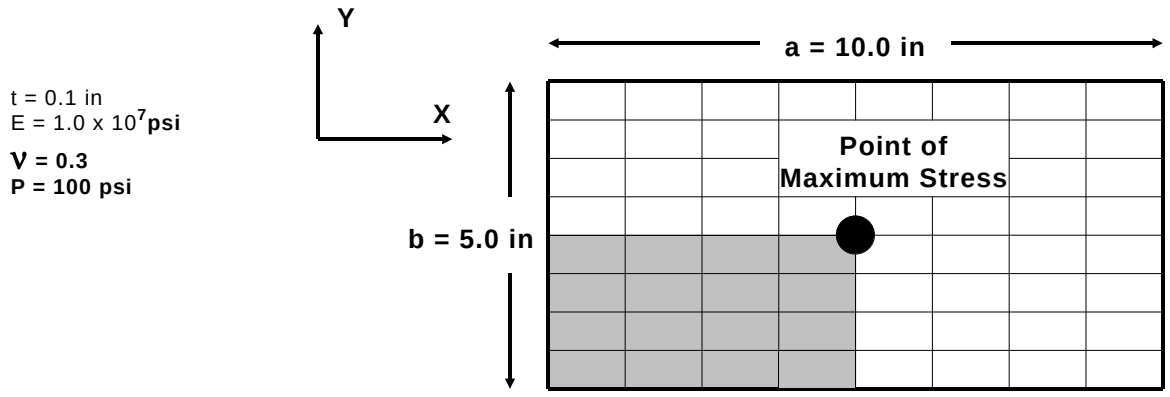
27.5.1 Convergence Characteristics

For this example problem, a model is successively refined in a uniform manner to compare the computed mesh error with the theoretical results.

Example Problem 27-1
A rectangular flat plate with simply-supported edges is subjected to a uniform pressure of 100 psi is the downward z-direction. You wish to evaluate the convergence characteristics of models having successively refined meshes, and to compare these results with the Mesh Error Estimate feature.

The first finite element model and physical characteristics are shown in Figure 27-1. You may find this model in file **MEST1A**. The model is comprised of 16 QUAD8 elements in a **4x4** mesh. The GRID point at the center of the plate is highlighted in the figure because this is the point of maximum stress. The actual maximum stress component is in the y-direction. Subsequent models were created using rectangular meshes with equal refinements in both coordinate directions. These models, found in files **MEST1B** through **MEST1G**, have models with meshes of **6x6**, **8x8**, **10x10**, **20x20**, **30x30**, **40x40**, respectively. All of the data streams were then executed in **UAI/*NASTRAN*** selecting the Mesh Error Estimate feature.

Figure 27-1. MESH ERROR MODEL



Quarter model used for analysis.

The solution results are presented in Figure 27-2. The analytical solution from [Young89] for the normal-y stress at the center GRID point is 152.55 KSI. The Figure shows the element centroidal stress closest to the center, the GRID point stress at the center, and the theoretical solution for each of the models. As anticipated, convergence is uniform.

Figure 27-2. STRESS RESULTS

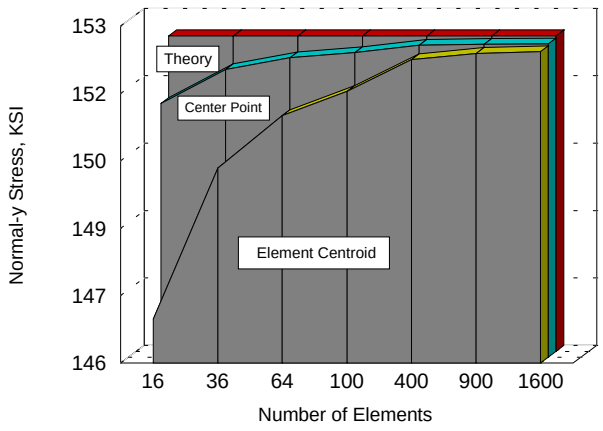
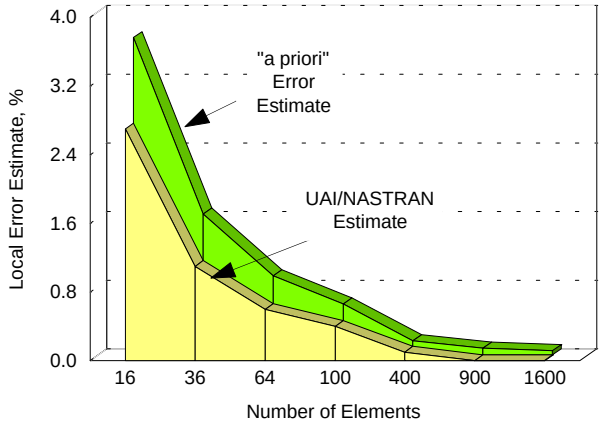


Figure 27-3 shows the resulting normalized local error estimate in the element adjacent to the center of the plate for each model. Also shown is the error computed by comparing the **UAI/*NASTRAN*** results to the theoretical solution. This is an example of what is called an a priori estimate. Both the convergence trends and error measures are in quite good agreement, indicating that the Mesh Error Estimates computed by **UAI/*NASTRAN*** reflect the actual validity of the model.

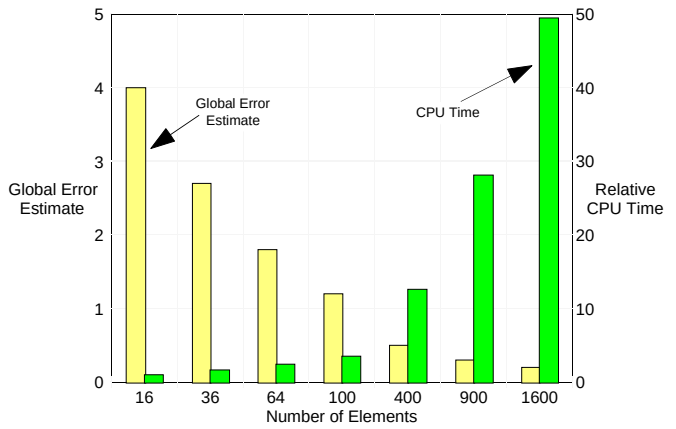
Figure 27-3. LOCAL ERROR ESTIMATES



27

Finally, Figure 27-4 shows the global error estimate for each model along with the relative cost of performing these analyses. Again, the convergence trend of the error estimate is excellent. The Figure illustrates how there is a point of diminishing returns for refinement as the cost of the analysis increases dramatically.

Figure 27-4. GLOBAL ERROR ESTIMATE



27.5.2 Normal Modes Analysis

This section provides an example of how Mesh Error Estimates may be used when performing Normal Modes analyses.

Example Problem 27-2

A thin plate 20.0 in long and 2.0 in wide is fixed at one end. Determine the first three out-of-plane bending modes of each model, compute the mesh error estimates, and analyze the results.

The four finite element models used are shown in Figure 27-5. These models are found in files **MENM3A**, **MENM3B**, **MENM3C**, and **MENM3D**. The first model is composed of 10 QUAD4 elements. The other models are created by simply subdividing the previous model in the x-direction. The four models are then executed in **UAI/*NASTRAN*** using the Mesh Error Estimate feature. The results are discussed in the remainder of this section.

Figure 27-5. FINITE ELEMENT MODELS FOR MODES

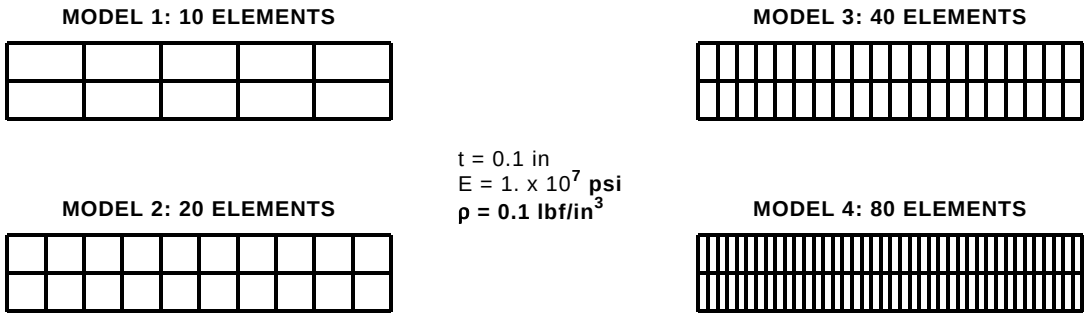


Figure 27-6. MODAL DEFORMATION

The first three bending modes of the plate were determined for each model. The error in these frequencies with respect to the beam theory solution are shown in Figure 27-6.

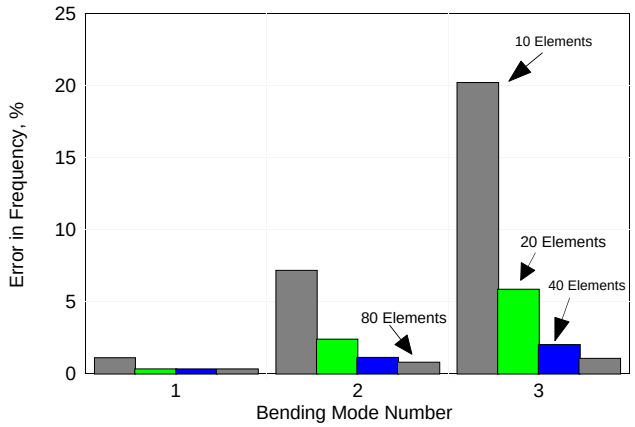
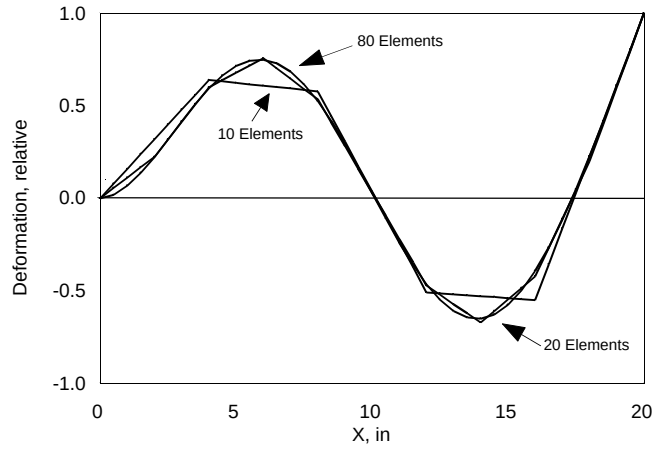


Figure 27-7. ERROR IN NATURAL FREQUENCIES

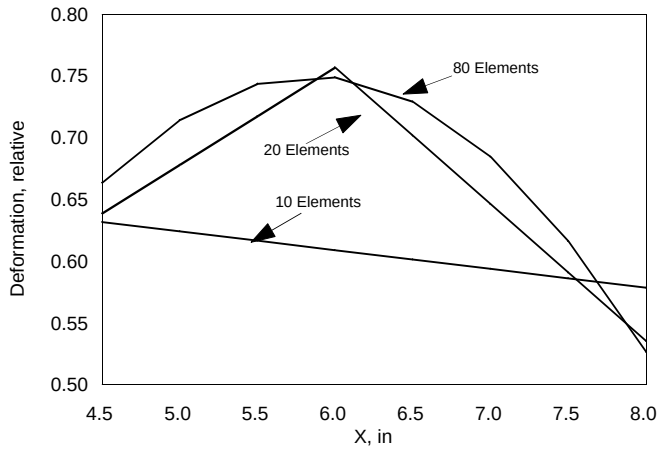
Next, the third mode is studied more carefully by plotting the modal deformation along the centerline of the model, as shown in Figure 27-7. Here, the forty element model has been omitted for clarity. It is now apparent that there is a significant error in the actual mode shape of the coarse model, whereas the two more refined models appear to be in reasonable agreement. This is, however, illusory as shown in Figure 27-8.



27

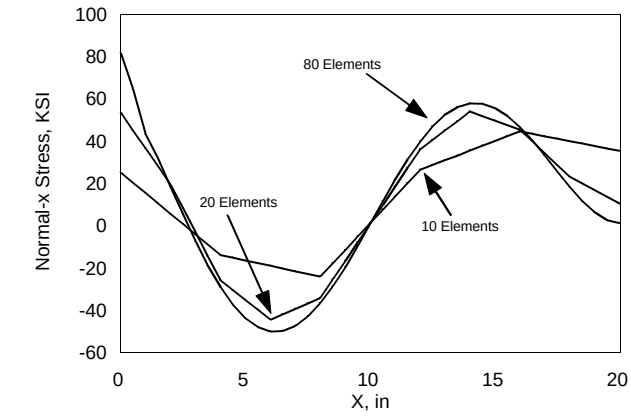
The Figure shows that the 20 and 80 element models have a significant variation in the modal deformation. It is expected that the sharp peaks occurring at element boundaries will be reflected in both the stresses and the mesh error estimates for the models.

Figure 27-8. BLOW-UP OF HIGH STRESS REGION



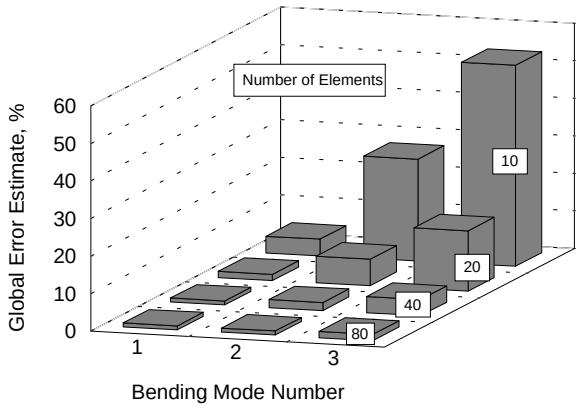
This is verified by considering a plot of the normal-x stress along the centerline of the beam as shown in Figure 27-9. There is a considerable difference in both the magnitude and location of the high stress values. This should be seen in the Mesh Error Estimates.

Figure 27-9. MAXIMUM NORMAL-X STRESS



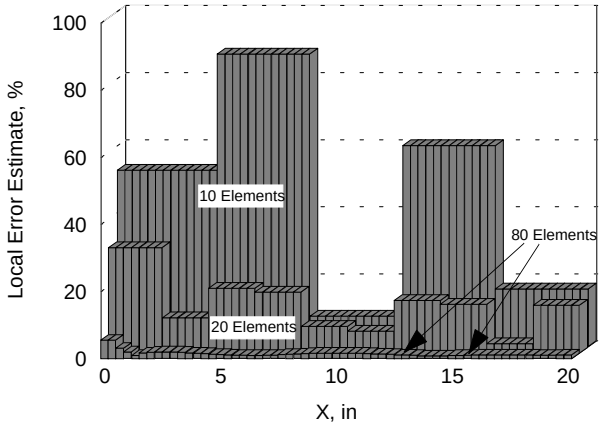
The global error estimates for the first three bending modes, shown in Figure 27-10, show that the most refined model provides very accurate results in all cases.

Figure 27-10. GLOBAL ERROR ESTIMATE



The local error estimates for the third bending mode, plotted along the length of the plate, are consistent with the modal deformation of the structure as seen in Figure 27-11. Where bending is high in the coarse models, for example at **X=6** and **X=14**, these errors are high. As the model is refined, the local errors become smaller, but retain the same characteristics. Finally, with the good 80 element model, the local errors have been reduced to very small values.

Figure 27-11. LOCAL ERROR ESTIMATES



27

Chapter 28

AEROELASTIC FLUTTER

Aeroelasticity is the study of the interaction effects of inertial, elastic and aerodynamic forces on the stability or response of flexible structures in aerodynamic or hydrodynamic flowfields.

UAI/*NASTRAN* can perform a Dynamic Aeroelastic Stability or Flutter analysis. The following chapter describes how stability can be assessed by performing a Flutter analysis in **UAI/*NASTRAN***.

28

28.1 BACKGROUND

In many simulation problems, the external forces acting on a body are assumed to be independent of the deformation of that body. This assumption is consistent with most analysis disciplines treated in **UAI/*NASTRAN***. However, there are some exceptions to this such as follower forces in Nonlinear Statics, and the **NOLIN** element forces in Transient Response. In aeroelastic problems the external forces are highly dependent on the structural deflections.

Aeroelasticity in general, and flutter in particular, have been important to aircraft design from the earliest days of flight. Because aircraft structures must be light they tend to deform appreciably under load. The aerodynamic loads on the vehicle are, in turn, changed by these deformations. This interaction is dynamic in nature and can lead to flutter, a self-excited vibration with sustained or divergent amplitudes, at sufficiently high flow velocities. Though predominantly of concern to aircraft designers, aeroelasticity can also affect windmills, bridges, stop signs or telephone lines in high winds. Aeroelastic effects can sometimes be treated as quasi-static for the analysis of flying qualities and maneuvering loads.

UAI***/NASTRAN*** computes unsteady aerodynamics and performs aeroelastic flutter solutions. While there is no specific support for panel flutter, an example problem demonstrates using the finite surface spline and the CPM supersonic unsteady aerodynamics method to predict this type of flutter.

28.2 THE AEROELASTIC MODEL AND ANALYSIS

An Aeroelastic analysis requires the assembly of a matrix equation of motion that relates the structural deformations and the resulting inertial, elastic, viscous and aerodynamic forces.

The inertial, damping and elastic forces are represented by the structural mass, damping and stiffness matrices in the same manner as other static and dynamic analyses in **UAI/*NASTRAN***. The aerodynamic forces are introduced from a separate aerodynamic model that are mapped onto the structural model through a transformation matrix that is based on several available splining methods.

Before the flutter analysis is performed, the reduced dynamic modal equations of motion are combined with an aerodynamic force matrix after it has been transformed to the structural grid. Then, a series of complex eigensolutions is performed with varying aerodynamic parameters. The aerodynamic parameters that may be varied are Mach number, air density, dynamic pressure and reduced frequency. The flutter speed is found by determining the combinations of aerodynamic conditions where the damping in a structural mode vanishes or passes from a stable to unstable condition.

The specific procedure for perturbing aerodynamic parameters depends on the choice of flutter method. There are three methods available, a *k* method and two variations on the *p-k* method, each with options. All three methods include the automatic recovery of flutter crossings, and the flutter mode shape output, if requested. The *p-k* methods include an iterative method which has generally excellent convergence. The other *p-k* algorithm is a sweep method that is used for the rare cases that experience convergence difficulties. These methods are described in Section 28.4.4.

For all solution methods, there are root tracking and interpolation utilities that process the series of eigensolutions to identify and present the flutter boundaries.

28.2.1 The Unsteady Aerodynamic Model

The unsteady aerodynamic model is used to predict aerodynamic pressure due to motion at the aerodynamic grid. The aerodynamic model geometry is defined by one or more **CAERO1** Bulk Data entries. In addition to the effects of planform geometry, the aerodynamic forces are also a function of the ***aerodynamic parameters***, which are Mach number, reduced frequency and aerodynamic symmetry condition. The complete unsteady aerodynamic model consists of the geometric definition and all of the aerodynamic force matrices evaluated at the aerodynamic parameters that you select. The term ***hard points*** is used to denote the specified number of values at which the Aerodynamic Force Matrix data was computed.

The term ***Aerodynamic Influence Coefficient (AIC)*** is sometimes used interchangeably with the ***Downwash Matrix***. In this Chapter, the ***AIC*** matrix always refers to the inverse of the Downwash Matrix. Unsteady Aerodynamic methods are described more fully in Section 28-3.

Two unsteady lifting surface theories are available, the ***Doublet Lattice Method*** for subsonic flows, and the ***Constant Pressure Panel Method*** for supersonic flows. An ***aerodynamic model*** may consist of ***AIC*** matrix data from a combination of both theories. The generation of the aerodynamic model is controlled by subcommands of the aerodynamics case, while subcommands of the flutter case can control the selection of which aerodynamic model and what portion of it will be used in a particular flutter solution.

☞ An Aerodynamic model may contain data for multiple symmetry conditions, while any single flutter solution case will use the aerodynamic data from a single symmetry condition.

28.2.2 The Structural Model

Flutter analysis in **UAI/*NASTRAN*** is performed using the modal method. The structural stiffness, mass and damping matrices are transformed from the physical degrees of freedom to a set of generalized coordinates which are defined by the normal modes of the structural model. This transformation is performed for all modal methods in **UAI/*NASTRAN***.

The Normal Modes analysis provides you with control over any structural boundary conditions and allows you to request output for the normal modes analysis separately from the flutter analysis. More details on the input and output options of Normal Modes analyses are discussed in Chapter 11 of this manual.

Later, the aerodynamic force matrix is transformed into the same modal, or generalized coordinate, set.

28.2.3 The Aeroelastic Model

An aeroelastic model is defined by the combination of the aerodynamic force matrices and the structural modes. To assemble the equations of motion, the aerodynamic force matrix is mapped to the structural modes with user-defined spline relations. The aeroelastic solution may then be performed.

Since there are no restrictions on the aerodynamic and structural meshes, a mapping between the aerodynamic and structural degrees of freedom is required. Spline methods are used to determine a transformation matrix between the aerodynamic grid and structural grid. To solve this problem, **UAI/*NASTRAN*** has adopted the approach of a pre-computed transformation matrix based on the geometric relationships of the aerodynamic and structural grids. This has the advantage of reusability when multiple aeroelastic models are being assembled from the same aerodynamic and structural grids. The specific spline methods are discussed in section 28.4.2. Note that the aeroelastic model is assembled with only one set of structural modes and a single aerodynamic symmetry condition. Although the aerodynamic model can be generated for multiple symmetry conditions, an aeroelastic model will only use a portion of the complete aerodynamic model. The relationship between multiple symmetry conditions is further illustrated in the example problems.

28.2.4 The Flutter Analysis

Once the aeroelastic model is defined, a flutter analysis can proceed. The critical roots of the system are extracted at the specified flight conditions. The flutter boundary is then identified as the Mach / speed points where the critical modes are neutrally stable.

The relative stability in a flutter analysis is measured by the *required damping*. This term denotes the additional amount of damping that would be required to produce a neutrally stable system. This definition results in sign convention where a negative *required damping* indicates a stable system.

Since the reduced frequency at flutter is usually unknown, and the unsteady aerodynamic force matrix is computed at a limited number of reduced frequency values, interpolation methods are used to determine the aerodynamic force matrix at any required value of reduced frequency.

As indicated earlier, the term hard points refers to the limited number of values at which the aerodynamic matrix data is computed. Similarly, the term ***soft points*** denotes any reduced frequency values used by the flutter solution. The aerodynamic matrix data for soft points is determined by interpolation of the hard point data.

This relationship between hard points and soft points could also be applied to the Mach Number dependence of the aerodynamic matrix data. At the present time, the interpolation procedure does not perform interpolation over Mach number. This means that a flutter analysis may only be performed at the hard point Mach Number values.

28.2.5 The Aeroelastic Analysis Solution Procedure

Aeroelastic analysis by nature is a multidisciplinary problem, and it has been implemented within **UAI/*NASTRAN*** as such. The process of generating and solving the aeroelastic model is organized in the steps or CASEs. The flutter analysis is solved in the third CASE using the aeroelastic model assembled from the results of two previously executed CASEs: a Normal Modes analysis and an Aerodynamics analysis. The Normal Modes analysis provides the structural stiffness and mass model in modal coordinates, and the results of at least one Aerodynamics analysis provides the unsteady aerodynamics model.

The relative order of the Normal Modes and Aerodynamics analyses is irrelevant except that these must have occurred prior to the Flutter analysis.

When performing multidisciplinary analyses, i.e. **SOL MULTI**, certain analyses may use results of previous disciplines. This is done with the **USING** Case Control command. This is always the case when performing modal method analyses such as Flutter or Modal Frequency Response. In both cases, the solution uses the results of a Normal Modes analysis as the generalized coordinates. This approach reduces ambiguity, providing you with explicit control over the generation and output requests of the Normal Modes analysis. You may control the number of eigenvectors to be produced, and can request any special output from the Normal Modes analysis.

The symmetry, Mach number and reduced frequency parameters for which the Aerodynamic model will be generated are requested in the Aerodynamics analysis CASE. These specific parameter values define the hard point matrix data. During the flutter solution, interpolation techniques are used to produce intermediate aerodynamic parameter data (i.e. soft points) from these hard points.

28

28.3 MATHEMATICAL BACKGROUND

This section presents a brief theoretical background for the Aeroelastic analysis capabilities in **UAI/*NASTRAN***.

There are a number of specialized analysis modules required to solve the class of Aeroelastic problems addressed by **UAI/*NASTRAN***. These include: unsteady aerodynamic theories that solve the linearized potential flow equation; splining methods to provide a interface between the structural and aerodynamic grids; and automated complex eigensolvers to perform a flutter analysis. The subsonic problem is solved using the Doublet Lattice Method [Geising72], and the supersonic flow problem is solved using the Constant Pressure Method [Appa87]. The Aerodynamic to Structural grid interconnection is solved with two Surface Spline methods; that are presented in [Harder72] and [Appa89] and a Linear Spline method from [NASTRAN79]. The following sections will deal with each specialized area individually.

28.3.1 Unsteady Aerodynamics

The Doublet Lattice Method (DLM) and Constant Pressure Panel Method (CPM) procedures calculate matrices that produce forces on the aerodynamic panels as a function of the deflections of the panels. This requires the discretization of the aerodynamic configuration into a number of trapezoidal panels with zero incidence and side edges aligned with the airstream. The capability of these codes include:

- Symmetric, antisymmetric and asymmetric analyses with respect to the XZ plane of symmetry in the aerodynamic coordinate system. (usually the aircraft centerline plane of symmetry)
- Symmetric, antisymmetric and asymmetric analysis with respect to the XY plane (sometimes referred to as Ground Effect and Biplane symmetry).
- Both CPM and DLM solve the Multiple Interfering (aerodynamically coupled) Lifting Surface problem.
- Consistent with linearized potential flow (small disturbance) theory, thickness and camber effects are treated as additional perturbations on the flowfield.

28.3.1.1 Unsteady and Steady Aerodynamics Methodology

The aerodynamics methods are based on three basic matrix equations:

$$\boldsymbol{w}_j = \boldsymbol{A}_{jj}\,\boldsymbol{p}_j \tag{28-1}$$

$$\boldsymbol{w}_j = \boldsymbol{D}_{jk}\,\boldsymbol{u}_k \tag{28-2}$$

$$\boldsymbol{F}_k = \boldsymbol{S}_{kj}\,\boldsymbol{p}_j \tag{28-3}$$

where:

$\boldsymbol{w}_j$	=	Downwash at the aerodynamic control points
$\boldsymbol{A}_{jj}$	=	Aerodynamic Downwash influence matrix, Inverse of Aerodynamic Influence Coefficient matrix

$\boldsymbol{p}_j$	=	Pressure on the aerodynamic panel at the force point
$\boldsymbol{D}_{jk}$	=	"Substantial differentiation" matrix
$\boldsymbol{u}_k$	=	Displacements at the aerodynamic grid points
$\boldsymbol{F}_k$	=	Forces and moments at the aerodynamic grid points
$\boldsymbol{S}_{kj}$	=	Integration matrix

The details of the development of the $\boldsymbol{A}_{jj}$ matrix for the Doublet Lattice and Constant Pressure Panel Methods is well-documented in [Geising72], [Appa87] and [Appa88]. For the purpose of this document, it is adequate to state that the $\boldsymbol{A}_{jj}$ matrix is a function of the discretized wing geometry, Mach number and the reduced frequency:

$$k = \frac{\omega \bar{c}}{2V} \tag{28-4}$$

where ω is the circular frequency, $\bar{c}$ is the wing reference chord, and V is the freestream velocity. $\boldsymbol{D}_{jk}$ is a complex function of geometry that relates translation and rotation at the 0.5 chord position of each aerodynamic element(sometimes referred as a 'box') to the control point (0.75 chord for DLM, and a variable position beginning at 0.75 chord and moving aft to the 1.0 chord as Mach varies from 1. to infinity for the CPM). The terms in the $\boldsymbol{D}$ matrix appear as:

$$D_{jk} = 1 + i\,\omega\,\delta x_k \tag{28-5}$$

where δx_k is the chordwise distance from the 0.5 chord position to the control point of the k^{th} aerodynamic element. The $\boldsymbol{S}_{kj}$ matrix is an integration matrix of element areas and moment arms that relate the pressure distribution at the load point (0.25 chord for DLM and 0.5 chord for CPM) to force and moment at the 0.5 chord position.

The complete procedure occurs in two distinct stages. The first stage is the generation of geometry, or meshing of the discretized wing geometry, and the generation of the $\boldsymbol{A}_{jj}$, $\boldsymbol{D}_{jk}$ and $\boldsymbol{S}_{kj}$ matrices is accomplished in the initial stages not unlike the generation of the structural mass and stiffness matrices. The $\boldsymbol{A}_{jj}$ matrix is inverted to form $\boldsymbol{Q}_{jj}$ which is stored on the database for later use. Before generating $\boldsymbol{A}_{jj}$, substantial cross checking of the input data is accomplished before any relatively expensive matrix generation, decomposition or solution is made.

The second stage occurs when the modal aeroelastic equation of motion is formed. The $\boldsymbol{Q}_{jj}$, $\boldsymbol{D}_{jk}$ and $\boldsymbol{S}_{kj}$ matrices are transformed to generalized aerodynamic forces. This procedure is described in the following Section.

28.3.2 Spline methods and Aerodynamic to Structural Model Connectivity

The aerodynamic matrices are computed at an aerodynamic mesh that is rarely coincident with any structural mesh. The transfer of forces and displacements from one mesh to the other is accomplished with transformation matrices based on the theory of splines.

There are two surface spline methods in **UAI/*NASTRAN***, the infinite plate spline [Harder72] and a finite surface spline [Appa89]. It is important to note the finite spline surface has a free

edge boundary condition. Because the edge boundary is free from both bending moment and shear force, it has a distinct advantage over the infinite surface spline (with its simple supports at infinity) when extrapolating to the edges of a lifting surface from centrally located structural grid points. An example of this is a complete wing aerodynamic model coupled to a structural model that is limited to only the primary wing box structure. The wing structure might extend from 30% to 60% of wing chord, thus requiring a chordwise extrapolation of the wing box motion to the leading and trailing edge areas of the wing. This situation is illustrated with the BAH wing example problems in the Section 28-6. This example compares the performance of the finite surface spline to the infinite surface spline.

The finite surface spline method is well suited to both displacement and force transformations and is less sensitive to grid selection than the infinite surface spline. The finite surface spline is capable of treating higher order deflection patterns, without the undesirable artifacts that can occur with high order polynomial fits. The infinite surface method behaves reasonably well for displacement field transformations as required by the flutter discipline, as long as sufficient care is taken in choosing the structural grid connectivity of the spline, and only lower order motions are involved.

Splines are used to generate a transformation matrix $\boldsymbol{G}_{kg}$ that defines the slope and displacement at the aerodynamic k-set degrees of freedom in terms of the displacements of the structural g-set:

$$\boldsymbol{u}_k = \boldsymbol{G}_{kg}\boldsymbol{u}_g \tag{28-6a}$$

Note that only the displacement normal to the plane of the surface spline is used. The local g-set rotations and motion parallel to the surface plane are ignored. Any transformations required between the g-set global coordinates and the surface spline coordinate system is computed automatically within **UAI/*NASTRAN***. It is important to remember that while the transformation is based on the equivalent displacements, it also results in the force transformation:

$$\boldsymbol{P}_g = \boldsymbol{G}_{kg}^T\boldsymbol{P}_k \tag{28-6b}$$

28.3.2.1 Finite Surface Spline

The Finite Surface Spline method implemented in **UAI/*NASTRAN*** is based on the spline proposed in [Appa85]. As the name implies this method employs a finite uniform elastic plate as the spline surface. This elastic plate, is modeled by a mesh of elastic plate elements, is called the ***virtual mesh***.

The Finite Element Method is used to solve the plate bending problem of the spline surface. As discussed in [Appa89], **UAI/*NASTRAN*** uses an improved method which employs $\boldsymbol{C}^0$ continuous, rectangular quadrilateral plate elements with coupled interpolation functions for deflection and rotation. Testing has shown this implementation strikes a good balance between spline robustness, CPU efficiency, and user convenience.

Three matrices are generated for the finite-surface spline method:

- An interpolation matrix ψ_{vg} that is based on the shape functions of the elements comprising the virtual mesh. This matrix defines the mapping from q_v, the virtual grid displacement vector, to u_g, the structural grid displacement vector:

$$u_g = \psi_{gv} q_v \tag{28-7}$$

- A similar interpolation matrix maps the displacement vector at the aerodynamic grid u_k to the virtual mesh points. This transformation is written as:

$$u_k = \psi_{kv} q_v \tag{28-8}$$

- The stiffness matrix, K_{vv}, at the virtual grid is produced. The equations of motion are assembled and solved using the penalty method to impose a displacement constraint at the structural grid points. The equilibrium equation is:

$$\left[K_{vv} + \alpha \psi_{gv}^T \psi_{gv} \right] q_v = \alpha \psi_{gv}^T u_g \tag{28-9}$$

where α is the penalty constant selected such that:

$$\alpha \psi_{gv}^T \psi_{gv} >> K_{vv}$$

The resulting mapping matrix G_{kg} which relates the structural displacement vector u_g to the aerodynamic displacement vector u_k is obtained by solving (28-9) for q_v and substituting into (28-8) yielding:

$$u_k = \Psi_{kv} \left[\alpha^{-1} K_{vv} + \psi_{gv}^T \psi_{gv} \right]^{-1} \psi_{gv}^T u_g \tag{28-10}$$

Thus, the transformation of (28-5) is:

$$G_{kg} = \psi_{kv} \left[\alpha^{-1} K_{vv} + \psi_{gv}^T \psi_{gv} \right]^{-1} \psi_{gv}^T \tag{28-11}$$

The Finite Surface Spline method in **UAI/*NASTRAN*** uses the rectangular plate element proposed in [Zienkiewicz77]. The shape function for this element uses a single two-dimensional polynomial to interpolate both deflections and slopes in terms of twelve nodal degrees of freedom — one deflection and two slopes for each of the four elemental grid points. This element does not guarantee slope compatibility across the elemental boundary of adjoining elements, the slopes are only continuous at nodal points, however extensive testing has demonstrated satisfactory interpolations and extrapolations. Testing has included deflections and slopes over a variety of two-dimensional functions that model rigid-body motions, elastic bending and twisting as well as higher order deformations such as constrained panels may undergo due to panel flutter.

The rectangular virtual spline surface geometry is automatically sized to contain both the structural and aerodynamic grid points by some margin in both the spanwise and chordwise directions. The user can adjust the virtual surface on the **SPLINE1** Bulk Data entry by specifying the margins and the number of elements in the chordwise and spanwise directions of the virtual mesh. The current defaults are 10 spanwise and 10 chordwise elements with 1.0%

margins in the chord and span directions. These defaults have produced good to excellent results on a range of test problems.

28.3.2.2 Infinite Surface Spline

A surface spline is used to find a function $w(x, y)$ for all points (x, y) when w is known for a discrete set of points, $w_i = w(x_i, y_i)$. An infinite plate is introduced to solve for the total deflection pattern given deflections at a discrete set of points. This surface spline is a smooth continuous function which is nearly linear in x and y at large distances from the points (x_i, y_i). Furthermore, the problem can be solved in closed form.

The deflection of the plate is synthesized as the response due to a set of point loads on the infinite plate. The response due to a single load is called a fundamental solution. The fundamental solutions have polar symmetry. If the load is taken at $x_i = y_i = 0$, and polar coordinates are used $x = r\cos\theta$, $y = r\sin\theta$ the governing differential equation is

$$D\ \nabla^4 w = D\frac{1}{r}\frac{d}{dr}\left\{ r\ \frac{d}{dr}\left[\frac{1}{r}\ \frac{d}{dr}\ r\frac{dw}{dr} \right] \right\} = q \tag{28-12}$$

The load q vanishes except near r=0. A solution to the general spline problem, formed by superimposing solutions of (28-12) is given by

$$w(x, y) = a_0 + a_1 x + a_2 y + \sum_{i+1}^{N} K_i(x,y)\, P_i \tag{28-13}$$

where

$$K_i(x,y) = \left(\frac{1}{16\,\pi\, D}\right) r_i^2\ \ln\ r_i^2 \quad \text{and} \quad r_i^2 = (x - x_i)^2 + (y - y_i)^2$$

and P_i is the concentrated load at (x_i, y_i).

The $N + 3$ unknowns $(a_0, a_1, a_2, P_i,\ i=1, N)$ are determined from the $N + 3$ equations

$$\sum P_i = \sum x_i P_i = \sum y_i P_i = 0$$

and

$$w_j = a_0 + a_1 x_j + a_2 y_j + \sum_{i=1}^{N} K_{ij} P_i \quad (j=1, N) \tag{28-14}$$

where

$$K_{ij} = K_i\left(x_j, y_j\right)$$

Note that $K_{ij} = K_{ji}$, and that $K_{ij} = 0$ when $i=j$. These equations can then be summarized in matrix form

$$w(x,y) = \left[1, x, y, \ \boldsymbol{K}_1(x,y), \boldsymbol{K}_2(x,y), ..., \boldsymbol{K}_N(x,y) \right] \begin{Bmatrix} \boldsymbol{a}_0 \\ \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \\ \hline \boldsymbol{P}_1 \\ \boldsymbol{P}_2 \\ \cdot \\ \cdot \\ \boldsymbol{P}_N \end{Bmatrix} \tag{28-15}$$

The transformation matrix is then produced by evaluating (28-15) twice. First, to relate the structural grid point deflections to loads at the structural points:

$$\boldsymbol{u}_g = \boldsymbol{K}_1 \boldsymbol{P}_g \tag{28-16}$$

and second, to relate deformations at the aerodynamic grid to loads at the structural points:

$$\boldsymbol{u}_k = \boldsymbol{K}_2 \boldsymbol{P}_g \tag{28-17}$$

Solving (28-16) for the loads, and substituting in (28-17) results in:

$$\boldsymbol{u}_k = \boldsymbol{K}_2 \boldsymbol{K}_1^{-1} \boldsymbol{u}_g \tag{28-18}$$

or, written in simpler form:

$$\boldsymbol{u}_k = \boldsymbol{G}_{kg} \boldsymbol{u}_g \tag{28-19}$$

28.3.2.3 Linear Spline

The linear spline satisfies the equations:

$$EI \frac{d^4 w}{dx^4} = q - \frac{dM}{dx} \tag{28-20}$$

$$GJ \frac{d^2 \theta}{dx^2} = -T \tag{28-21}$$

where q is the applied load, M is the applied moment, and T is the applied torque. A symmetric fundamental solution for $x \neq 0$ is used for loads $q = P\,\delta(x)$, and an antisymmetric fundamental solution is used for moments. The solution for the general case is found by superimposing these fundamental solutions:

$$w(x) = a_o + a_1 x + \sum_{i=1}^{N} \left(\frac{M_i (x - x_i) \mid x - x_i \mid}{4EI} + \frac{P_i \mid x - x_i \mid^3}{12EI} \right) \tag{28-22}$$

$$w'(x) = \frac{dw}{dx} = a_1 + \sum_{i=1}^{N} \left(\frac{M_i \mid x - x_i \mid}{2EI} + \frac{P_i (x - x_i) \mid x - x_i \mid}{4EI} \right) \tag{28-23}$$

$$\theta(x) = b_0 + \sum_{i=1}^{N} \left(\frac{- \mid x - x_i \mid}{2GJ} T_i \right) \tag{28-24}$$

These equations may be written in matrix form as:

$$\begin{Bmatrix} w(x) \\ w'(x) \end{Bmatrix} = \begin{bmatrix} 1 & x & \frac{\mid x - x_1 \mid^3}{12EI} & \cdots & -\frac{(x - x_1) \mid x - x_1 \mid}{4EI} & \cdots \\ 0 & 1 & \frac{(x - x_1) \mid x - x_1 \mid}{4EI} & \cdots & -\frac{\mid x - x_1 \mid}{2EI} & \cdots \end{bmatrix} \begin{Bmatrix} a_o \\ a_1 \\ -- \\ P_1 \\ \cdots \\ P_N \\ -- \\ M_1 \\ \cdots \\ M_N \end{Bmatrix} \tag{28-25}$$

$$\begin{Bmatrix} 0 \\ -- \\ \theta_1 \\ \cdots \\ \theta_N \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & 0 & \frac{-\left| x_2 - x_1 \right|}{2GJ} & \cdots & \frac{-\left| x_N - x_1 \right|}{2GJ} \\ 1 & \frac{-\left| x_1 - x_2 \right|}{2GJ} & 0 & \cdots & \frac{-\left| x_N - x_2 \right|}{2GJ} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & \frac{-\left| x_1 - x_N \right|}{2GJ} & \frac{-\left| x_2 - x_N \right|}{2GJ} & \cdots & 0 \end{bmatrix} = \begin{Bmatrix} b_o \\ -- \\ T_1 \\ \cdots \\ T_N \end{Bmatrix} \tag{28-26}$$

The transformation matrix $\boldsymbol{G}_{kg}$ is again developed in the same manner as for surface splines, the equations (28-22) thru (28-24) are evaluated twice to produce the relationship between the loads at the structural grids and resulting deformations at both the structural and aerodynamic grids. These matrices are used in the same manner as shown in (28-18).

28.3.3 Aeroelastic Model — Matrix Assembly and Reduction

The process to produce the aeroelastic equations of motion requires the transformation of the aerodynamic force matrices to generalized coordinates for each Mach number and reduced frequency of the form:

$$\boldsymbol{Q}_{hh} = \boldsymbol{\Phi}_{dh}^{T}\boldsymbol{G}_{kd}^{T}\,\boldsymbol{W}_{kk}\boldsymbol{S}_{kj}\,\boldsymbol{Q}_{jj}\,\boldsymbol{D}_{jk}\,\boldsymbol{G}_{kd}\,\boldsymbol{\Phi}_{dh} \tag{28-27}$$

where:

$\boldsymbol{Q}_{hh}$	=	Generalized Aerodynamic Force matrix,
$\boldsymbol{\Phi}_{dh}$	=	eigenvector matrix of retained normal modes for the modal solution
$\boldsymbol{G}_{kd}$	=	Aerodynamic to Structural Grid transformation matrix from the structural d-set to the aerodynamic k-set degrees of freedom
$\boldsymbol{W}_{kk}$	=	User supplied correction factor matrix (optional)

The spline methods used to produce the $\boldsymbol{G}_{kd}$ matrix are discussed in the previous Section. The aerodynamic matrix data contained in $\boldsymbol{S}_{kj}$, $\boldsymbol{D}_{jk}$ and $\boldsymbol{Q}_{jj}$, were defined in the earlier section on Unsteady Aerodynamics methods.

28.3.4 Flutter Analysis

The flutter analysis in **UAI/*NASTRAN*** can be accomplished by two distinct methods of solution; a *k* method and two variations of the *p-k* method.

28.3.4.1 k Method of Flutter Solution

The basic equation of motion for modal flutter analysis by the *k* method is:

$$\left[-\boldsymbol{M}_{hh}\omega^{2} + (1+ig)\,\boldsymbol{K}_{hh} - \frac{1}{2}\rho\,V^{2}\boldsymbol{Q}_{hh}(k,M)\right]\boldsymbol{u}_{h} = 0 \tag{28-28}$$

where

$\boldsymbol{M}_{hh}$	=	generalized mass matrix
$\boldsymbol{K}_{hh}$	=	generalized stiffness matrix
$\boldsymbol{Q}_{hh}$	=	generalized aerodynamic force matrix.
ω	=	circular frequency
g	=	required damping (this artificial damping required is a product of the *k* flutter method.)
ρ	=	freestream fluid density
V	=	freestream velocity (a product of the *k* method)
$\boldsymbol{u}_{h}$	=	generalized coordinates

The traditional approach in the *k* method is to convert the generalized aerodynamic force matrix $\boldsymbol{Q}_{hh}$ from a complex spring to a complex mass and divide the equation through by either ω^2 or by $(1+ig)$. The second approach defines an inverse complex eigenvalue problem which requires that rigid body modes be swept out for the generalized stiffness matrix to be positive definite. The first approach is followed in **UAI/*NASTRAN***. This yields a complex eigenvalue problem of the form:

$$\left[-\left[\boldsymbol{M}_{hh} + \frac{\rho}{2}\left[\frac{\bar{c}}{2k}\right]^2 \boldsymbol{Q}_{hh}(k, M) \right] \frac{\omega^2}{(1+ig)} + \boldsymbol{K}_{hh} \right] \boldsymbol{u}_{hh} = 0 \tag{28-29}$$

The complex eigenvalue in this problem is then:

$$\lambda = \frac{\omega^2}{(1+ig)} \tag{28-30}$$

This form only requires the resulting complex mass matrix (mass plus aerodynamics) to be positive definite. The solution proceeds by setting the Mach number and density and then extracting the eigenvalues of the system for a series of values of reduced frequency.

The frequency and required damping ω, g are recovered directly from the eigenvalue, while the velocity V is derived from the frequency and the fixed value of reduced frequency by:

$$g = \frac{-\lambda^I}{\lambda^R}. \tag{28-31a}$$

$$\omega = \sqrt{\lambda^R (1+g^2)}. \tag{28-31b}$$

$$V = \frac{\bar{c}\,\omega}{2k}. \tag{28-31c}$$

28.3.4.2 p-k Method of Flutter Solution

The basic equation of motion for modal flutter analysis by the *p-k* method is:

$$\left[\boldsymbol{M}_{hh}\lambda^2 + \left[\boldsymbol{B}_{hh} - \frac{1}{4}\frac{\rho \bar{c} V}{k}\boldsymbol{Q}^I_{hh}(k, M) \right]\lambda + \left[\boldsymbol{K}_{hh} - \frac{1}{2}\rho V^2 \boldsymbol{Q}^R_{hh}(k, M) \right] \right] \boldsymbol{u}_h = \boldsymbol{0} \tag{28-32}$$

where

$\boldsymbol{M}_{hh}$ = generalized mass matrix

$\boldsymbol{K}_{hh}$ = generalized stiffness matrix

$\boldsymbol{Q}^R_{hh}$ = real part of generalized aerodynamic force matrix.

$\boldsymbol{Q}_{hh}^{I}$ = imaginary part of generalized aerodynamic force matrix, note that in the equation this imaginary stiffness has been converted to a real damping matrix

λ = eigenvalue (complex)

ρ = freestream fluid density

$k = \frac{\bar{c}\omega}{2V}$ = the reduced frequency.

V = freestream velocity

$\boldsymbol{u}_h$ = generalized coordinates

The eigenvalues extracted from (28-27) are interpreted for frequency and damping ω, g by:

$$\omega = = \lambda^I . \tag{28-33a}$$

$$g = 2\zeta = \frac{2\lambda^R}{|\lambda|} . \tag{28-33b}$$

In order to solve for the eigenvalues, (28-32) is transformed to canonical form from N second order equations, where N is the number of retained modes plus any EPOINT degrees of freedom, to 2N first order equations and the eigenvalue problem proceeds with a fixed set of values for Mach number, density, velocity and reduced frequency. The exact procedure for setting the values for the reduced frequency depends on the choice of *p-k* methods, either PKITER or PKSWEEP, which are described below.

The Iterative ***p-k* method.** The reduced frequency variation for the PKITER method is based on successive updates to the reduced frequency until the condition,

$$k = \frac{\bar{c}\,\lambda^I}{2V} \tag{28-34}$$

is satisfied. In (28-34), λ^I is the imaginary part of the eigenvalue of the current mode.

The procedure begins with a near zero value for k where the resulting real roots are assumed to satisfy the condition (28-34) immediately. The first iteration begins with a reduced frequency that conforms to the lowest non-zero frequency root. The iteration is repeated with succesive updates to the assumed reduced frequency. The iteration is assumed to be converged when (28-34) is satisfied to within an acceptable tolerance. The tolerance has a default value of 0.01, and can be reset by the user (on the **FLSOLVE** Bulk Data entry). When the iteration has converged on the first nonzero frequency root, the iteration begins on the next lowest frequency at the present step.

The Frequency Sweep *p-k* **method.** There are cases where the *p-k iter* method's convergence is slow or impossible; in these cases the *p-k sweep* method may provide the required results in a straightforward manner. This method does not rely on an iterative technique to find 'converged' roots but instead performs a set of eigensolutions at user selected values of reduced frequency. At each reduced frequency step through the range of interest, the algorithm checks the current eigenvalues for frequencies which lie between the reduced frequency of the last step and the current step. When the algorithm identifies this condition, then it has 'crossed' a valid root and a simple interpolation of the eigenvalues between the current and last step

yields the 'converged' root result. The *p-k sweep* method can be described as an incremental search followed by linear interpolation to find the desired 'converged' roots. It is important to note that this method makes no assumptions regarding the number of possible converged roots that may be extracted.

It is desirable to begin the sweep at a low enough reduced frequency so that all nonzero frequency eigenvalues are higher than the reduced frequency. If the initial reduced frequency value fails to capture the first root, then the algorithm will back up to a smaller starting frequency until the first nonzero frequency root is above the starting aerodynamic frequency. At this first step all real roots are also assumed to be 'converged' roots and the nonzero roots are then searched for at the next frequency step. This method has been found to have good success when the iterative technique cannot converge.

Alternate Rigid Body Coordinates with *p-k* **methods.** **UAI/*NASTRAN*** provides an alternate set of rigid body mode coordinates. The rigid body modes commonly used are based on inertial axes. These coordinates represent the modes using 2nd order equations. However, the displacement degrees of freedom merely add roots at the origin, which migrate slightly from the origin due to numerical problems. An obvious example would be the change in altitude of the air vehicle, which should not result in a net force on the vehicle.

When requested the rigid body modes in inertial axes are transformed to body axes coordinates. This is accomplished using the general transformation,

$$\begin{Bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{Bmatrix} = \begin{bmatrix} -s & 0 & 0 & 0 & 0 & 0 \\ 0 & s & 0 & 0 & 0 & U_0 \\ 0 & 0 & -s & 0 & U_0 & 0 \\ 0 & 0 & 0 & -s & 0 & 0 \\ 0 & 0 & 0 & 0 & s & 0 \\ 0 & 0 & 0 & 0 & 0 & -s \end{bmatrix} = \begin{Bmatrix} x \\ y \\ z \\ \varphi \\ \theta \\ \omega \end{Bmatrix} \tag{28-35}$$

where s is the complex operator. This transformation results in a set of 1st order rigid body mode equations with no zero roots at the origin. The possible exception is the u equation because the current unsteady aerodynamics methods simply do not produce drag or chordwise forces. The transformation matrix in (28-35) assumes the reference axes are oriented with X in the positive flow direction, the Y axis to the right, and Z axis up. This results in the body axis coordinates of X forward, the Y to the right, and the Z axis down.

28.4 INPUT DATA REQUIREMENTS

This section provides a description of the **UAI/*NASTRAN*** input data required to perform Aeroelastic analysis.

28.4.1 Executive Control Commands

Aeroelasticity is available in the multidisciplinary design and analysis Solution **MULTI**. This requires the following Executive Control Command:

```
SOL MULTI
```

28.4.2 Aeroelastic Analysis Case Control

A flutter analysis is performed by calculating the Complex Eigenvalues of the aeroelastic equations for a series of aerodynamic parameter perturbations. Before this can occur the aeroelastic matrix equations must be formulated by coupling the results of a Normal Modes solution (for the structural model mass and stiffness contribution) and the Unsteady Aero solution. The methods of coupling between the structural and aerodynamic grids which are rarely if ever common, are discussed in the section on Splines and Modal Interpolation. The present section will focus on the Case Control Commands available to control the analysis procedure.

As discussed in Section 28.2.5, a **MODES** case and at least one **AERO** case must precede a **FLUTTER** case. A **MODES** case performs a normal modes analysis of the structural model, where the user controls the number of eigenvectors to be extracted and thus available as generalized coordinates for later use in the **FLUTTER** case. The **AERO** case generates the hard point Aerodynamic Influence Coefficient matrix data. The hard point data is later transformed to modal coordinates and then used as a table within which the various flutter solvers will interpolate to the intermediate reduced frequencies or soft points.

28.4.3 Aerodynamics Case Control Commands

The primary purpose of the Aerodynamics case is to generate a set of aerodynamic matrix data for later use in one or more Aeroelastic analyses. If requested, these data will be saved on a permanent database for later use. The aerodynamic pressures from rigid body motion about the Aerodynamic Reference Coordinate system is also available.

28.4.3.1 Generating the Aerodynamic Force Matrix Data

The Aerodynamics Case has three subcommands required to produce the aerodynamic force matrix data, **MACH**, **KFREQ**, and **SYMMETRY**.

```
SYMMETRY = [ NOSYMM ]
           [  SYMM  ]
           [  ANTI  ]
```

SYMMETRY is required to specify the Aerodynamic Boundary Condition for the current Case. If the user has input an asymmetric or *full span* aerodynamic model, **SYMMETRY=NOSYMM** must still be specified since blank does not default to an unsymmetric or no symmetry condition.

```
MACH  = { ( Mach1 , Mach2 ,... ) }
        {          mid         }

KFREQ = { ( k1 , k2 ,... ) }
        {        kid       }
```

The **MACH** and **KFREQ** Commands each refer to a list of real values or to a **SET** command. The **SET** Case Control command is used to input a list of real values. Its format is:

```
SET = val1 , val2 ,...
```

Every combination of the input set of Mach number and reduced frequency are used to produce an unsteady Aerodynamic Influence Coefficient matrix for the aerodynamic symmetry condition specified. If the user prefers a unique set of reduced frequency values for a Mach number, then an individual Aerodynamics Case could be specified for each Mach number.

```
ARCHIVE AERO [ TO logicaldb_name:/dir_name ]
```

The **ARCHIVE AERO** command directs the current Aerodynamics Case results to be located on a permanent ARCHIVE database. This group of data is specifically designed for use by subsequent **UAI/*NASTRAN*** executions. The results from multiple **AERO** Cases may be stored at the same location in the ARCHIVE database, or in separate databases. An aerodynamic model is defined by or limited to a fixed set of geometry. Multiple symmetry conditions of the same Aerodynamic geometry definition can be stored together as one aerodynamic model for later use. Multiple models can be saved in the same database, so long as each model is placed in a unique subdirectory.

```
PRESSURE = { ALL  }
           { NONE }
```

As a model validation and diagnostic aid, the Pressure distributions on the unsteady aerodynamic model due to rigid body modes can be requested with the Case Control Command **PRESSURE**. If requested, the pressures will be generated for all Mach number, reduced frequency pairs of matrices produced in the current case. The rigid body modes are generated about a geometric reference. The **RCID** field on the **AEREFS** Bulk Data Entry selects a rectangular coordinate system to define the aerodynamic reference axes about which the rigid body motion is defined.

The resulting pressures are also integrated and printed. These may be interpreted as rigid aerodynamic stability derivatives.

28.4.4 FLUTTER Case Control Commands

A flutter analysis is controlled by a minimum set of commands that specify the structural modes to be used, the method of flutter solution, the aerodynamic symmetry condition and the density-speed conditions to be solved.

However, based on the selected method of flutter solution, there are additional Case commands that are either required, recommended, optional or not applicable. Table 28-1 presents a summary of the Case Control commands as they relate to the three flutter solution methods in **UAI/*NASTRAN***. The table indicates **R** for required, **r** for recommended, **O** for optional and **NA** for not applicable.

Table 28-1. FLUTTER CASE CONTROL COMMANDS

	Case Control Command	To Select or Define	Flutter Method		
			K	PKIter	PKSweep
AEROELASTIC MODEL DEFINITION	SYMMETRY	Aerodynamic Symmetry Condition	R	R	R
	USING AERO	Archived Aerodynamic Model	O	O	O
	USING MODES	Structural Modes	R	R	R
SOLUTION CONDITIONS	DENS	Atmosphere relation and Density-Altitude conditions	R	R	R
	VLIST	Speeds or Velocities for Non-Match Point analyses	NA	O	O
	KLIST	Soft Point Reduced Frequencies	r	NA	r
SOLVER CONTROLS	FMETHOD	Method of Flutter Analysis	R	R	R
	FLSOLVE	Flutter solver controls	O	O	O
HARD POINT SELECTION	KSELECT	Aerodynamic Hard Point data selection	O	O	O
	MSELECT	Aerodynamic Hard Point data selection	O	O	O
	HSELECT	Modal coordinate omission selection	O	O	O
OUTPUT	FLPRINT	Level of Diagnostic Output	O	O	O

28.4.4.1 Assembling the Aeroelastic Model

The aeroelastic model is assembled from the components defined by the three Case control commands, **SYMMETRY**, **USING AERO** and **USING MODES**.

The aerodynamic force matrices are limited to the current symmetry condition, **SYMMETRY** and mapped to the modal structural model. The structural model component is defined by a set of Normal Modes located from the **USING MODES** command. The **USING MODES** command must reference the **caseid** that produced the Normal Modes that are to be used. This is shown in the following example:

```
SYMMETRY = SYMM
CASE 10 AERO
   MACH = ( 0.1, 0.6 )
   KFREQ = ( 0.01, 0.1, 0.5 )
CASE 11 AERO
   MACHGEN = ( 0.6 )
   KFREQGEN = ( 1. )
CASE 20 MODES
   METHOD = 101
CASE 30 FLUTTER
   USING MODES 20
```

This sequence of commands executes a flutter analysis using the normal modes computed in Case 11 with the aerodynamics computed in the current execution. This example happened to produce aerodynamic data with two Cases. The first Case outputs aerodynamics at three reduced frequencies, 0.01, 0.1 and 0.5; and two Mach numbers, 0.1 and 0.6. The second Case produces aerodynamics for an additional reduced frequency at 1.0 and Mach 0.6.

When the aerodynamic model is created during the current run and is not being saved by the **ARCHIVE** command, then the command **USING AERO** is not required. The **USING AERO** Case Control command is only required to reference an Aerodynamic Model that was previously saved on a permanent database with the **ARCHIVE AERO** command. In the example above, if

the aerodynamic model was to be saved in a specific directory using the **ARCHIVE** command, the example would appear as:

```
SYMMETRY = SYMM
ARCHIVE AERO TO myaerodb:/symdir
CASE 10 AERO
   MACH = (0.1)
   KFREQ = (0.01, 0.05)
CASE 11 MODES
   METHOD = 101
CASE 12 FLUTTER
   USING AERO FROM myaerodb:/symdir
   USING MODES 11
```

Both **USING** commands may select previously generated data which reside on **ARCHIVE** databases. If both the modes and aerodynamics were saved in a run for use in a subsequent analysis, the first run would appear as:

```
SYMMETRY = SYMM
ARCHIVE AERO TO mydb
ARCHIVE MODES TO mydb
CASE 10 AERO
   MACH = (0.1)
   KFREQ = (0.01, 0.05)
CASE 11 MODES
   METHOD = 101
```

and the second run would be:

```
SYMMETRY = SYMM
CASE 12 FLUTTER
   USING AERO FROM mydb
   USING MODES 11 FROM mydb
```

28.4.4.2 Specifying the Flutter Analysis Conditions

The previous section described the input to assemble the required equations for a flutter analysis. The Case Control commands that specify the conditions to be solved are discussed here. The **FLUTTER** Case conditions and what solution method is used are selected using four Case Control commands.

FMETHOD = { K / PKITER / PKSWEEP }

The **FMETHOD** command is required to select the flutter analysis method.

The command **DENS** is always required to define the Density-altitudes to be processed.

DENS = { (ρ_1 , ρ_2 , . . .) / *aid* }

This command specifies a list of Air Densities or the identification number, *aid*, of an **ATMOS** Bulk Data entry that defines an atmospheric relationship including a list of density ratios. This relation is used to define the Mach required for the Flutter Crossing output or to define the match point analysis conditions.

The *p-k* flutter methods perform either a 'constant density' or 'match point' flutter analysis. If an atmosphere relation is defined by the **DENS** command, then match point velocities can be produced internally and processed.

If you want to perform a constant density analysis, then the **VLIST** command is used to specify the velocity set to be solved at the density points.

$$\textbf{VLIST} = \left\{ \begin{matrix} (v_1 , v_2 , \ldots) \\ vid \end{matrix} \right\}$$

By omitting the **VLIST** Command, the user chooses to run only those velocities corresponding to the match point conditions of each Mach number, Density-Altitude combination.

If **VLIST** is omitted, then the **DENS** command must select an **ATMOS** entry. The **ATMOS** Bulk Data Entry defines both the density list and the atmospheric relationship that allows the velocity to be computed internally for each Mach number, Density combination.

$$\textbf{KLIST} = \left\{ \begin{matrix} (k_1 , k_2 , \ldots) \\ kid \end{matrix} \right\}$$

KLIST is used by the *p-k sweep* and *k* methods to select a set of soft point reduced frequencies for the flutter analysis. This is generally required for effective use of the *k* and *p-k sweep* methods.

KLIST is not strictly required because the solver will use the available reduced frequency data from the Aerodynamic model. However, it is highly desirable to specify a more detailed set. This is because the 'hard point' reduced frequency set should be chosen based on the reduced frequency dependence of the unsteady AIC matrix, while the **KLIST** defined soft point solution set should be chosen based on the reduced frequency dependence of the aeroelastic system roots (the shape of the V-g curves). This Command is ignored by the *p-k iter* method which has an internal procedure which picks the solution reduced frequencies.

28.4.4.3 Controlling the Flutter Solvers

A number of commands are used for additional problem control in a **FLUTTER** Case. If desired, **NROOT** is used to specify a limited number of modes to be extracted and printed. For the *p-k* methods this reduces the number of converged roots that are searched for, and for the *k* method, **NROOT** reduces the number of roots that are printed. The default action is to extract and to print all roots.

$$\textbf{NROOT} = \left\{ \begin{matrix} \text{ALL} \\ n \end{matrix} \right\}$$

The **FLPRINT** subcommand is used to control the amount of intermediate and diagnostic output during the flutter solution.

$$\textbf{FLPRINT} = \left\{ \begin{matrix} \text{NONE} \\ \text{SUMMARY} \\ \text{DETAIL} \end{matrix} \right\}$$

The **FLSOLVE** selects an identification number of a **FLSOLVE** Bulk Data entry. The **FLSOLVE** entry can be used to adjust default settings to flutter solution parameters.

$$\textbf{FLSOLVE} = kid$$

28

The specific options on the **FLSOLVE** Bulk Data Entry are described in the following Section on Bulk Data.

28.4.4.4 Requesting Output and Saving Results from a Flutter Analysis

The command **ARCHIVE SOLUTION** will cause the results of a flutter analysis to be stored in relations on the ARCHIVE database. The schemas of these relations and methods of interfacing with them are are discussed in Sections 28.6 and 28.7.

```
ARCHIVE SOLUTION [ TO logical_db_name:/dir_name ]
```

28.4.4.5 Selecting Portions of the Aeroelastic Model

The optional subcommands, **MSELECT** and **KSELECT** can select a subset of available Mach number and reduced frequencies data within the aerodynamic model to be used for flutter analysis. The default action is to use the aerodynamic matrices at all available Mach number, reduced frequency combinations within the referenced Aerodynamic model. These commands can be useful to limit the scope of an analysis, or to study convergence behavior.

MSELECT = { (M_1 , M_2 ,...) / *setid* / ALL }

For example, you may want to perform a quick check analysis at only one Mach number while using a stored aerodynamic model that contains several Mach Numbers.

KSELECT = { (K_1 , K_2 ,...) / *setid* / ALL }

You might also wish to perform a convergence study to determine the required number of reduced frequencies for reliable interpolations. Several **FLUTTER** Cases could be solved, each with a different set of hard point reduced frequency sets using the **KSELECT** Command.

There are occasions where the you may wish to examine the system stability with and without key modes. For this purpose, **MODESEXCLUDE** can be used to specify a subset of the structural modes produced in the referenced MODES case.

MODESEXCLUDE = ($mode_1$, $mode_2$,...

This subset will be removed from the system to determine the effect of their non-participation. The default action is to use all modes.

28.4.5 Bulk Data Entries - Unsteady Aerodynamic Modeling

Table 28-2 presents a summary of the Aeroelastic Bulk Data entries sorted by function. The fundamental geometric reference parameters are defined by the **AEREFS** Bulk Data Entry, the format of which is:

AEREFS	ACID	RCID	CBAR						

ACID defines the Aerodynamic Coordinate System which locates the aerodynamic planes of symmetry and the mean flow direction. **RCID** locates the Reference Axis Coordinate System.

Table 28-2. AERO BULK DATA ENTRIES

FUNCTIONAL GROUP	BULK DATA ENTRIES
Fundamental Aerodynamic parameters	AEREFS AEUNITS ATMOS
Aerodynamic Geometry and Matrix generation	AEFACT CAERO1
Aerodynamic to Structure interconnection	SPLINE1 SPLINE2 SET1
Flutter Analysis Control	FLSOLVE

This system is used to define a set of Rigid Body Modes which are used in the Aerodynamic analysis to produce Pressure output if requested. The Reference Chord, **CBAR**, is used in the definition of reduced frequency $k = \frac{\omega \bar{c}}{2V}$.

The **CAERO1** Bulk Data entry, shown below, is used to create the aerodynamic paneling.

CAERO1	EID	IGID	CID	EQUAL SET	NSPAN SSID	EQUAL SET	NCHORD CSID	IGID	-cont-
-cont-	X1	Y1	Z1	CROOT	X2	Y2	Z2	CTIP	

The aerodynamic paneling is specified with either **EQUAL** divisions in the spanwise or chordwise directions or **SET** will indicate arbitrary divisions defined by the **AEFACT** Bulk Data Entry. **AEFACT** specifies a list of ascending real numbers between zero and one, that define any arbitrary division set.

AEFACT	SID	D1	D2	D3	D4	D5	D6	D7	-cont-
-cont-	D8	*CONTINUES WITH LIST OF DIVISION POINTS.*							-cont-

28.4.6 Bulk Data Entries - Aerodynamic to Structural Model Connections

Spline methods are used to provide the interconnection between the Aerodynamic and Structural grids. The **SPLINE1** Bulk Data Entry is used to define a surface spline where plate type deformations may occur.

SPLINE1	EID	CAERO	BOX1	BOX2	SETG	DZ	METHOD	NC	-cont-
-cont-	NS	FC	FS						

The surface spline orientation is defined by the referenced aerodynamic macro-element.

The **SPLINE2** Bulk Data Entry defines a linear spline which can be used when beam type deformations can occur. This is generally applicable where chordwise bending is deemed negligible or ignored. The coordinate system of the elastic beam is a required input on the **SPLINE2**.

SPLINE2	EID	CAERO	BOX1	BOX2	SETG	DZ	DTOR	CID	-cont-
-cont-	DTHX	DTHY							

In both spline inputs a unique **EID** is used for reference in error messages. The fields **BOX1** and **BOX2** are used to limit the spline to a portion of the **CAEROi** macroelement, otherwise the default values are **FIRST** and **LAST** indicating that all boxes are to be connected the spline. The structural points used to define the spline motion are input on the **SET1** Bulk Data entry. Each **SET1** entry is referenced on the **SETG** field on the **SPLINEi**.

SET1	LABEL	G1	G2	G3	G4	G5	G6	G7	-cont-
-cont-	G8	*CONTINUES WITH LIST OF GRID ID'S*							-cont-

It is important to note that surface splines only use structural motions normal to the spline surface, this leads to a minimum requirement of three structural grid points to define the most basic motion of the surface spline. The linear spline uses structural rotations in addition to translations, so at least a minimum of two structural grids are required.

28.4.7 Bulk Data Entries - Aeroelastic Solution Conditions

There are many cases where the air velocity in solution units is not convenient. The **AEUNITS** entry allows you to define a factor to relate the solution units to the user input and output units. A common factor in english units is 20.24 to relate the velocity unit of Knots to a solution unit of inches/sec. Note that a separate input on the **ATMOS** entry is used to define the relationship between true airspeed and equivalent airspeed. Both relationships are used together in **UAI/*NASTRAN*** to relate user input units of equivalent speed to the solution units of true velocity.

AEUNITS	VFAC	VUNITS							

The **ATMOS** Bulk Data Entry allows you to define a complete atmosphere relation. Its general format is:

ATMOS	AID	RHO0							-cont-
-cont-	LABEL1	DENS1	VELM11						-cont-
-cont-	LABEL2	DENS2	VELM12						-cont-
-cont-	*CONTINUES WITH GROUPS OF THREE*								-cont-

The **RHO0** field defines the reference air density to relate the input and output in equivalent speed units to true velocity units. The atmosphere definition assumes a linear relationship between Mach and velocity at a constant density-altitude. This is consistent with the "Standard" atmospheric definitions. If the flutter analysis was to be run with a "Standard Cold Day" and a "Standard Hot Day," then two **ATMOS** Entries are input and solved in separate cases.

For the case of either *p-k* methods, if an **ATMOS** Entry is referenced by the Case Control Command **DENS**, and no velocity set specified by a **VLIST** Command, the flutter analysis is performed at the *Match Point* conditions. For each density-altitude, the required velocity is determined by the current Mach number and the **VELM1i** specified. The **VELM1i** is assumed to be in user input units, and in equivalent speed units if **RHO0** is defined. The example below is appropriate for a aeroelastic model in english units of lb-in-sec system, assuming that 20.24 is specified for **VFAC** on the **AEUNITS** Entry. The reference density **RHO0** is in units of lb(mass)/in^3, and the **VELM1i** are in units of Knots Equivalent Airspeed.

ATMOS	12	1.147-7							
	H30K	0.3747	360.6						
	H20K	0.5332	448.3						
	H10K	0.7386	548.2						
	SL	1.0	661.						
	HN8K	1.256	760.9						
	HN16K	1.5598	869.8						

The **FLSOLVE** entry selects several optional controls of the flutter solver procedures.

FLSOLVE	FID	IMETHOD	VCUT	GCUT	GOFF	EPS	OMRX	RBDOF	

IMETHOD provides a choice of Generalized Aerodynamic Force interpolation methods, **LAGRANGE** selects a Lagrangian polynomial interpolation and **CUBIC** selects a cubic spline interpolation method.

VCUT specifies a maximum cutoff speed for the flutter speed crossing search. This is generally not required when the velocities are input, however for the *k* method there may be many nuisance crossings generated by the higher frequency modes that are far outside the envelope of the vehicle. For cases where there are numerous lightly damped modes, **GCUT** may be used to specify a minimum damping required to identify only the critical crossings. If a lightly damped mode never generates damping greater than **GCUT**, that mode will not be reported in the flutter crossing summary. **GOFF** is used for cases where the flutter speed is defined by a damping value other than zero. This can be used to track lightly damped modes with flutter speeds that may be low with zero damping, but have acceptable flutter speeds assuming one or two percent damping.

EPS can be used to control convergence for the *p-k iter* method. The convergence criteria for this method is based on an increment of reduced frequency. The default value of 0.001 yields adequate results unless you have selected a non-standard definition for the reference length **CBAR** (such as unity) or there are roots that are unusually sensitive to reduced frequency. In these cases, **EPS** can be used to set a more appropriate convergence criteria.

RBDOF is an optional list of the rigid body degrees of freedom. These are defined in terms of the **RCID** coordinate system. In the *p-k* flutter methods, the **RBDOF** degrees of freedom are transformed from second order inertial axis coordinates to 1st order body axis coordinates. This has the benefit of cleaning up numerical artifacts in the rigid body modes results from the *p-k* flutter methods. In most cases where rigid body modes are present, small positive real roots will occur in the flutter solution that the analyst must reconcile. For small textbook problems these real roots are usually very small (e.g. less than 0.01) and easily understood to be numerical noise. However, for larger models with greater numbers of elastic modes, these positive real roots can become larger than acceptable. This transformation serves to eliminate these spurious roots.

28

28.5 SOLUTION RESULTS

The solution results from a **UAI/*NASTRAN*** aeroelastic analysis include both intermediate diagnostic and documentation results and the flutter analysis final results.

The Flutter case produces an error check of the interpolation of generalized aerodynamic force matrices at each reduced frequency (hard points), and the final solutions of flutter analysis. The user may also optionally request further details of the flutter solution process.

28.5.1 Unsteady Aerodynamics Analysis Output

The aerodynamic matrix data produced by the Aerodynamic Case are automatically saved on the ARCHIVE database to be used later in the Flutter Analysis Case . The Aerodynamic Case also provides panel geometry information and the pressures on the aerodynamic elements due to rigid body displacements about the selected reference coordinate system.

The ***Aerodynamic Panel Geometry Summary***, shown in Table 28-3a, displays the geometry in the aerodynamic coordinate system for each panel specified by **CAERO*i*** Bulk Data entries. Each **CAERO*i*** entry defines an aerodynamic macroelement with a fixed number of spanwise strips each having a fixed number of chordwise boxes. The width, chord length and the Y and Z coordinates along the center line are given for each resulting strip. The leading edge coordinates at each strip edge boundary are also provided.

The ***Aerodynamics Table of Contents***, shown in Table 28-3b, lists the aerodynamic data available on the **UAI/*NASTRAN eBase*** database. The attribute **METHOD** indicates which aerodynamic method was used to produce the aerodynamic matrix data. The subscripted integer indices for symmetry type, Mach number and reduced frequency, respectively, are used to identify the combinations of these parameters for which the aerodynamic hard point matrix

Table 28-3. AERO CASE RESULTS

a. Aerodynamic Panel Summary

U N S T E A D Y A E R O D Y N A M I C S P A N E L G E O M E T R Y S U M M A R Y

(I N A E R O D Y N A M I C C O O R D I N A T E S Y S T E M)

CAERO PANEL ID 1001

		STRIP EDGE GEOMETRY				STRIP CENTER GEOMETRY	
STRIP	STRIP WIDTH	X_LE	Y_LE	Z_LE	CHORD_LENGTH	YS	ZS
		-7.87500E+01	0.00000E+00	0.00000E+00	2.25000E+02		
1	4.50000E+01					2.25000E+01	0.00000E+00
		-7.48125E+01	4.50000E+01	0.00000E+00	2.13750E+02		
2	9.30000E+01					9.15000E+01	0.00000E+00
		-6.66750E+01	1.38000E+02	0.00000E+00	1.90500E+02		
3	8.90000E+01					1.82500E+02	0.00000E+00
		-5.88875E+01	2.27000E+02	0.00000E+00	1.68250E+02		

b. Aerodynamics Table of Contents

A E R O D Y N A M I C S T A B L E O F C O N T E N T S

METHOD	CONFIGURATION	CONFIG_SUB	SYMM TYPE SYM_SUB	MACH NUMBER MACH_SUB	REDUCED FREQ RED_SUB
DOUBLET LATTICE	AFLUT		SYMM (+XZ)	0.00000E+00	1.00000E-03
		1	2	1	1
DOUBLET LATTICE	AFLUT		SYMM (+XZ)	0.00000E+00	5.00000E-02
		1	2	1	2

data exist. The Aerodynamics Table of Contents summarizes all of the aerodynamic matrix data which reside on the current aerodynamic model database.

As a diagnostic aid for validating the Aerodynamic modeling, you may request the pressure distributions on the aerodynamic elements due to the Rigid Body modes. The computation of aerodynamic rigid body pressures are requested by the **PRESSURE** command.

```
PRESSURE = ALL
```

The rigid body modes are written about the Reference Coordinate System defined by the **RCID** field of the **AEREFS** Bulk Data Entry. The pressures are computed at each hard point condition for the rigid body modes. The pressures are then integrated to produce a rigid body force matrix. A set of forces and pressures corresponding to each hard point condition are computed in the current Case.

Six column vectors are output, each representing a rigid body mode in the reference coordinate system. The force matrix is also represented by six pairs of rows. These correspond to the same six rigid body modes. This output is shown in Table 28-4a. The pressure data is given in two rows for each aerodynamic box. For each component, the first row represents the real, or in-phase, part of the component of pressure and the second row contains the imaginary, or out-of-phase component, as shown in Table 28-4b. These data can be saved on the ARCHIVE database for the later postprocessing, and are output in the print file.

Table 28-4. OUTPUT FROM PRESSURE REQUEST

a. Forces

A E R O D Y N A M I C R I D I D B O D Y P R E S S U R E

(UK(TRAN)*QKK*UK)

SYMMETRY TYPE	MACH NUMBER	REDUCED FREQUENCY
SYMM (+XZ)	0.0000E+00	1.0000E-03

	T1	T2	T3	R1	R2	R3
T1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
T2	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
T3	0.00000E+00	0.00000E+00	-4.20249E-03	-1.05192E+00	-3.70511E+05	0.00000E+00
	0.00000E+00	0.00000E+00	-5.64666E+00	-1.26140E+03	-1.12062E+02	0.00000E+00
R3	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

b. Rigid Body Pressure

A E R O D Y N A M I C R I D I D B O D Y P R E S S U R E

(QJK*UK)

SYMMETRY TYPE	MACH NUMBER	REDUCED FREQUENCY
SYMM (+XZ)	0.0000E+00	1.0000E-03

BOXID	T1	T2	T3	R1	R2	R3
1001	0.000000E+00	0.000000E+00	1.849117E-07	4.563545E-05	9.435530E+00	0.000000E+00
	0.000000E+00	0.000000E+00	1.437990E-04	1.594926E-02	-8.403040E-03	0.000000E+00
1002	0.000000E+00	0.000000E+00	2.053413E-08	1.591682E-05	3.959681E+00	0.000000E+00
	0.000000E+00	0.000000E+00	6.034619E-05	8.214865E-03	4.683951E-03	0.000000E+00

28

28.5.2 Flutter Analysis Output

The quantity of results produced from a Flutter Analysis can vary greatly depending on the demands of the current problem. In **UAI/*NASTRAN*** the amount of printed output to be produced is controlled by the **FLPRINT** subcommand.

$$\textbf{FLPRINT} = \left\{ \begin{array}{c} \text{NONE} \\ \text{SUMMARY} \\ \text{DETAIL} \end{array} \right\}$$

There are three levels of output print available, each of which is described in the following sections.

28.5.2.1 The **FLPRINT=NONE** *Option*

When this option is selected, only a minimum of printed output will appear. Both the flutter speed crossings summary and an error check of the aerodynamic force interpolation are output.

In a flutter analysis the *hard point* aero matrix data is generally interpolated to some unique set of reduced frequency values. In order to give the analyst some measure of the goodness of fit of the available interpolation methods, **UAI/*NASTRAN*** produces an evaluation of the hard point selected for the flutter analyses. This output is produced by interpolation at each hard point.

Two interpolation methods are available, a cubic spline and a Lagrangian interpolation method. For each computed generalized aerodynamic force matrix (**QHH**) at the selected hard points, the matrix is recalculated by interpolation of the remaining hard points. A comparison is made between the calculated and interpolated values, and the norm of the error matrix is output. From this table, you can judge the relative importance of selected hard points for the flutter analysis, or more precisely measure potential interpolation error if a specific hard point was omitted. This output is shown in Table 28-5a. In this example, the usual behavior is seen, where the end points display the greatest error due to extrapolation.

Table 28-5. FLUTTER SOLUTION DEFAULT RESULTS

a. Aerodynamic Force Interpolation Summary

G E N E R A L I Z E D A E R O D Y N A M I C F O R C E M A T R I C E S (Q H H)

I N T E R P O L A T I O N S U M M A R Y

MACH NUMBER	REDUCED FREQ	NORM(QHH[COMPT-INTER])/NORM(QHH[COMPT]) HELGA	CUBIC
0.00000E+00	1.000000E-03	1.1260032E-02	1.7199812E-02
	5.000000E-02	3.3194597E-03	3.3194597E-03
	1.000000E-01	5.2483841E-03	4.0090764E-03
	1.000000E+00	1.2575443E-01	1.3588943E-01

b. Flutter Crossings Summary

FLUTTER CROSSINGS SUMMARY

SYMMETRY TYPE	MACH NUMBER	DENSITY	MODE	KFREQ	SPEED	FREQUENCY	VELOCITY
SYMM (+XZ)	0.00000E+00	1.14800E-07	2	1.00552E-01	1.05555E+03	3.08930E+00	1.26666E+04
SYMM (+XZ)	0.00000E+00	1.14800E-07	4	2.40370E-01	1.67891E+03	1.17463E+01	2.01470E+04

The Flutter Speed Crossings Summary is also printed for each case. This is a formatted print of the **FLUT_CROSSINGS** Relation that is stored on the ARCHIVE database. An example is illustrated in Table 28-5b.

28.5.2.2 The **FLPRINT=SUMMARY** *Option*

When this option is selected, in addition to the Flutter Speed Crossings Summary, you obtain the V-g curve data and the flutter mode shapes. For each mode, the V-g data lists frequency and damping as a function of velocity. This is generally output in the complex form, as shown in Table 28-6a. When Real roots are found, they are output in a separate real form as shown in Table 28-6b.

Table 28-6. FLUTTER SOLUTION SUMMARY RESULTS

a. Complex Form

F L U T T E R S O L U T I O N S U M M A R Y

FLUTTER VG

MODE	MACH NUMBER	DENSITY	SYMMETRY TYPE	FLUTTER METHOD
1	0.0000E+00	1.1480E-07	SYMM (+XZ)	PKITER

KFREQ	SPEED	DAMPING	FREQUENCY	VELOCITY	NATURAL	LAMAR	LAMAI
1.73683E-01	4.00000E+02	-1.71733E-01	2.02213E+00	4.80000E+03	1.27525E+01	-1.09502E+00	1.27054E+01
1.38704E-01	5.00000E+02	-2.26901E-01	2.01860E+00	6.00000E+03	1.27656E+01	-1.44827E+00	1.26832E+01
1.15180E-01	6.00000E+02	-2.89570E-01	2.01149E+00	7.20000E+03	1.27732E+01	-1.84936E+00	1.26386E+01
9.80967E-02	7.00000E+02	-3.62735E-01	1.99868E+00	8.40000E+03	1.27699E+01	-2.31605E+00	1.25581E+01
8.48666E-02	8.00000E+02	-4.50474E-01	1.97614E+00	9.60000E+03	1.27439E+01	-2.87041E+00	1.24165E+01
7.39314E-02	9.00000E+02	-5.58012E-01	1.93670E+00	1.08000E+04	1.26719E+01	-3.53552E+00	1.21687E+01

b. Real Form

FLUTTER VG(REAL)

MODE	MACH NUMBER	DENSITY	SYMMETRY TYPE	FLUTTER METHOD
1	0.0000E+00	1.1480E-07	SYMM (+XZ)	PKITER

SPEED	VELOCITY	LAMAR
1.50000E+03	1.80000E+04	-1.69109E+01
1.60000E+03	1.92000E+04	-2.31075E+01
1.70000E+03	2.04000E+04	-2.92660E+01
1.80000E+03	2.16000E+04	-3.59728E+01
1.90000E+03	2.28000E+04	-4.34192E+01
2.00000E+03	2.40000E+04	-5.15611E+01
2.10000E+03	2.52000E+04	-6.01971E+01

c. Flutter Crossings with Eigenvector

FLUTTER CROSSINGS

SYMMETRY TYPE	MACH NUMBER	DENSITY	MODE	KFREQ	SPEED	FREQUENCY	VELOCITY
SYMM (+XZ)	0.00000E+00	1.14800E-07	2	1.00552E-01	1.05555E+03	3.08930E+00	1.26666E+04

EIGENVECTOR

REAL	IMAGINARY
1.00000E+00	0.00000E+00
6.38800E-01	-4.44772E-01
-6.40615E-03	-1.91617E-02
-8.31185E-03	-6.17724E-04
9.29832E-03	-1.66288E-03
-2.58086E-03	-4.25505E-04
5.55043E-04	-6.51302E-04
2.97828E-04	3.03594E-04
5.08891E-04	-9.80546E-06
-3.95483E-04	9.74880E-05

28

This output is a formatted print of the **FLUT_VG** and **FLUT_VG_REAL** relational entities that can also be saved on the ARCHIVE database for later plotting and cross-correlation. In addition, for each flutter crossing the flutter mode shape is also recovered and saved. The complex generalized eigenvector is printed and the flutter mode shape on the structural grid is also recovered and saved on the database as relation **DISP_FLUT** for post-processing. This is illustrated in Table 28-6c. The options for graphical processing of animated mode shapes are discussed in the Section entitled Results Processing.

Note that the mode numbering for the real root V-g curves are independent of the mode numbering in complex root V-g curves. The mode numbering system for the real roots is self-consistent, but independent. This means that there will be a Mode number 1 in real root output, which should only be interpreted as the first real root, while it may or may not be related to the complex mode number 1 in the **FLUT_VG** output. **UAI/*NASTRAN*** provides a root tracking algorithm for the complex modes. The real roots are ordered from minimum to maximum value, and the maximum valued set are used to *fill in* the otherwise blank fields in the complex valued V-g curves when the frequencies have dropped to 0.0. This behavior is illustrated in the example problems shown later in this Chapter.

28.5.2.3 The **FLPRINT=DETAIL** *Option*

This option allows additional output for the PKSWEEP and PKITER methods. It allows you to trace the iteration or sweep process during the flutter solution. The roots from the intermediate solution steps are printed, as shown in Table 28-7a, and saved. Plotted forms of this data can be very helpful in understanding difficult convergence problems. Plotting of this data is also described in the example problems. When a root is found, the message shown in Table 28-7b is issued and the aligned root is listed.

Table 28-7. FLUTTER SOLUTION DETAILED RESULTS

a. Detailed Results of PKITER Solution

P K I T E R D E T A I L S

SOFT_POINT	CURRENT AERODYNAMIC FREQUENCY	RED_FREQ	VELOCITY
1	7.31529E-02	1.00000E-03	4.80000E+03

NUMBER	CURRENT ROOT		LAST ROOT	
1	-1.18630E+00	1.27248E+01	0.00000E+00	1.27975E+01
2	-4.17254E-01	2.18879E+01	0.00000E+00	2.23214E+01
3	-6.95281E-01	4.57421E+01	0.00000E+00	4.57444E+01
4	-4.80272E-01	7.35365E+01	0.00000E+00	7.35042E+01
5	-2.63411E+00	9.22851E+01	0.00000E+00	9.34991E+01
6	-8.25797E-01	1.32917E+02	0.00000E+00	1.32891E+02
7	-1.58572E+00	1.54648E+02	0.00000E+00	1.54870E+02
8	-1.88899E+00	2.04809E+02	0.00000E+00	2.05228E+02
9	-3.24097E+00	2.44530E+02	0.00000E+00	2.45373E+02
10	-5.13388E-01	3.03056E+02	0.00000E+00	3.03038E+02

MODE NUMBER	ITER NUMBER	SOFT POINT	INCREMENTAL KFREQ	CURRENT AERODYNAMIC FREQUENCY	CURRENT STRUCTURAL FREQUENCY
1	0	1	1.72949E-01	7.31529E-02	1.27248E+01

b. Message for Root Extraction

```
*** USER INFORMATION MESSAGE: A LINED UP ROOT HAS BEEN FOUND.
```

MODE NUMBER	ROOTS FOUND	LINED UP ROOT REAL	LINED UP ROOT IMAGINARY	CURRENT AERODYNAMIC FREQUENCY	CURRENT STRUCTURAL FREQUENCY
1	1	-1.09502E+00	1.27054E+01	1.27248E+01	1.27054E+01

28.6 EXAMPLE PROBLEMS

This section discusses examples of aeroelastic modeling and flutter analysis. These include a well-known textbook example flutter analysis, a correlation with a wind tunnel flutter test, and a panel flutter case. A larger than usual example is included to illustrate common modeling practices with a more practical sized example of aeroelastic modeling. There is also a small discussion of some practical aspects when performing flutter analysis of complete flight envelopes.

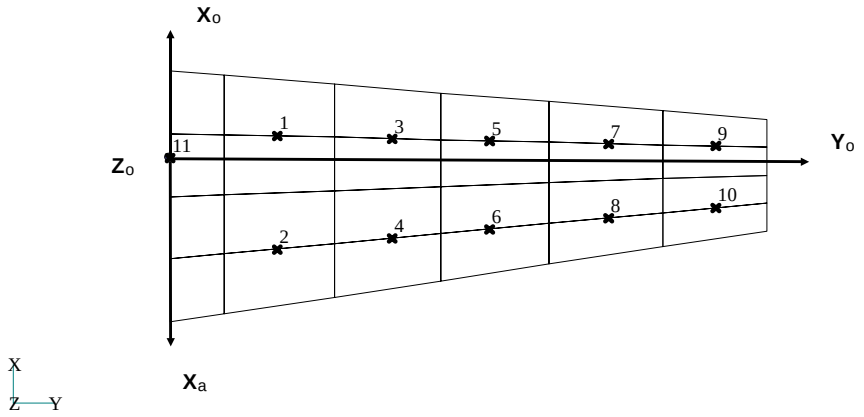
- ❒ BAH Jet Transport Wing Flutter
- ❒ 15-Degree Cantilevered Wing
- ❒ Simply Supported Panel
- ❒ Horizontal Stabilator

28.6.1 BAH Jet Transport Wing Flutter

A generic transport wing presented in the literature [Bisplinghoff55] is used for many example problems. This problem has become known as the BAH wing [Rodden59]. It is used here to demonstrate a *p-k iter* flutter analysis. In addition to a baseline analysis, a study of the various methods for splining the Aerodynamic and Structural models is presented.

Figure 28-1 illustrates the aerodynamic mesh and the idealized structural model used in this example. The input data are found in file **FLUT1**. The stiffness and mass definitions, given in [Rodden59], are used here for comparison. The original layout of geometry is retained: the origin of the Basic coordinate system (X_o,Y_o,Z_o) is located at the root of the wing on the elastic axis; the X-axis is forward and the Y-axis outboard along the elastic axis; thus the Z-axis is downward. For the aerodynamic coordinate system, the X-axis points aft in the direction of positive flow, the Y-axis is outboard, and the Z-axis upward. This choice of coordinate systems emphasizes the **CAERO1** geometry definitions. Note that although the leading edge points are defined in the basic coordinate system, which results in a positive X value as forward; the root and tip chords are defined in the X-axis of the aerodynamic coordinate system, which is the positive flow direction.

Figure 28-1. AERODYNAMIC AND STRUCTURAL MODELS



The wing stiffness is defined by a flexibility matrix, which is input as a GENEL element. The matrix coefficients define the normal displacements at five spanwise positions along the 25% and 75% chord lines. These control points are shown as GRID points one through ten in Figure 28-1. There is an additional point, **GRID 11**, on the centerline which represents the fuselage.

This arrangement illustrates a common problem for the aerodynamic to structure interface. For cases where the analyst has modeled only the wing primary structure, the spline equations must extrapolate from the interior structural points to the edges of the aerodynamic geometry. The linear beam **SPLINE2** is used in this case, however the effect of other spline options are discussed later in this Section.

28.6.1.1 Case Control Packet

The aerodynamic matrix data is computed at the Mach number, reduced frequency and symmetry conditions specified in the aerodynamic (**AERO**) Case. The **ARCHIVE** command is included to save the aerodynamic data for subsequent **UAI*/NASTRAN*** executions (for example, to evaluate alternative SPLINE inputs), thus a **USING AERO** command is also required. The *p-k iter* method is selected, and a symmetric aerodynamic condition is applied. All Mach and reduced frequency data are used in the flutter analysis. This is the default for these analyses. A constant density analysis is performed where both the altitude-density and a set of velocities are defined at each Mach number. **DENS** refers to an **ATMOS** Bulk Data entry which defines altitude-densities, while **VLIST** selects the velocities defined by a **SET** in Case Control. The Case Control packet defining this analysis is:

```
$       MACH NUMBERS
SET 1 = 0.0
$       REDUCED FREQUENCIES
SET 2 = 0.001, 0.05, 0.1, 0.2, 0.5, 1.0
$       VELOCITIES
SET 5 =  400.0,  500.0,  600.0,  700.0,  800.0,  900.0, 1000.0,
         1100.0, 1200.0, 1300.0, 1400.0, 1500.0, 1600.0, 1700.0,
         1800.0, 1900.0, 2000.0, 2100.0
SPC     = 1
CASE 11 AERO
   ARCHIVE MODEL
   SYMMETRY = SYMM
   MACH     = 1
   KFREQ    = 2
CASE 12 MODES
   METHOD  = 15
CASE 13 FLUTTER
   SYMMETRY = SYMM
   USING AERO FROM ARCHIVE
   USING MODES 12
   FMETHOD  = PKITER
   DENS     =  11
   VLIST    = 5
   FLPRINT = SUMMARY
```

If the aerodynamic model is not to be saved, then neither the **ARCHIVE AERO** nor **USING AERO** commands appear.

The following Control commands request the use of the previously saved Aerodynamic model:

```
ASSIGN ARCHIVE=lastrun,OLD,USE=ARCHIVE
SET 5 =  400.0,  500.0,  600.0,  700.0,  800.0,  900.0, 1000.0,
         1100.0, 1200.0, 1300.0, 1400.0, 1500.0, 1600.0, 1700.0,
         1800.0, 1900.0, 2000.0, 2100.0
SPC   = 1
CASE 12 MODES
   METHOD  = 15
CASE 13 FLUTTER
   SYMMETRY = SYMM
   USING AERO FROM ARCHIVE
   USING MODES 12
   FMETHOD  = PKITER
   DENS     =  11
   VLIST    = 5
   FLPRINT = SUMMARY
```

28.6.1.2 Bulk Data Definitions:

The aerodynamic reference geometry is defined with **AEREFS** Bulk Data. The aerodynamic coordinate system **ACID** is defined with coordinate system **1**, and the reference chord **CBAR** is input as 131.232 inches.

An **AEFACT** entry defines unequal divisions for the spanwise direction. The **CAERO1** entry specifies the **AEFACT** for the spanwise direction and 4 **EQUAL** divisions in the chordwise direction. The coordinate data on the continuation line locates the wing leading edge and chord length at the root and tip.

AEREFS	**1**	**0**	**131.232**						
AEFACT	**77**	**0.0**	**0.09**	**0.276**	**0.454**	**0.636**	**0.826**	**1.0**	
CAERO1	**1001**	**1**		**SET**	**77**	**EQUAL**	**4**	**1**	**+CA1**
+CA1	**78.75**	**0.0**	**0.0**	**225.0**	**35.0**	**500.0**	**0.0**	**100.0**	

For the baseline case, a linear **BEAM** spline is used. The rotation stiffness factors are set to -1.0. This is for the special case where only normal translation degrees of freedom are available from the structural grid points. If a structural beam model were used, the beam rotations would be meaningful and these factors would be set to 0.0.

SPLINE2	**100**	**1001**	**1005**	**1024**	**14**	**0.0**	**1.0**		**+S1**
+S1	**-1.0**	**-1.0**							
SET1	**14**	**1**	**THRU**	**11**					

The flutter analysis condition velocities were input in units of feet/sec. The solution units of this model are inches/sec. Thus, an **AEUNITS** Bulk Data entry is used to define a **VREF** of 12.0 to relate input and output units to the solution units.

AEUNITS	**12.0**								
ATMOS	**12**	**1.148-7**							**+A1**
+A1	**SEALEVEL**	**1.0**	**1114.9**						

Note that since this analysis is to be run at a single density, then **VREF** could be set to allow input and output units of Equivalent Velocity, without the use of **ATMOS**. The alternate input below uses input and output velocities in Knots Equivalent Airspeed (KEAS) for all density altitudes. In this case, **VREF** would be set to 20.24 to relate in/sec to Knots True Airspeed. The

reference density definition on the **ATMOS** Bulk Data entry provides a relationship between the input and output equivalent speed units and the solution's true velocity units.

$ Alternate input for full envelope in KEAS									
AEUNITS	20.24								
ATMOS	12	1.148-7							+A1
+A1	ALT30K	0.3747	360.6						+A2
+A2	ALT20K	0.5332	448.3						+A3
+A3	ALT10K	0.7386	548.2						+A4
+A4	SEALEVEL	1.0	661.						

Other alternate inputs include the Surface Spline definitions below. The impact of these are discussed in the next section.

$ Infinite and Finite Surface Spline inputs									
SPLINE1	100	1001	1005	1024	14	0.0			
SPLINE1	100	1001	1005	1024	14	0.0	FINITE		

28.6.1.3 Discussion of Results

There were no output requests specified for the **AERO** and **MODES** cases. The minimum output from these cases includes a summary of the aerodynamic geometry, a listing of the aerodynamic conditions that were generated and the normal mode frequencies extracted in the **MODES** case.

The **FLPRINT=SUMMARY** requests a print of the V-g curve data. XY plots of V-g and V-f, made with ***eShell***, are shown in Figure 28-2. The remainder of printed output is given automatically, including the Flutter Speed Crossings, shown in Table 28-8 below, and an error check on the generalized force interpolation methods. The Flutter Crossings indicate the critical point at 1055. ft/sec, 3.1 Hz flutter mode and a static (zero frequency) divergence root at 1650.1 ft/sec. There is also a second flutter crossing by a higher order mode at 1679 ft/sec with a frequency of 11.7 Hz.

The geometry of the structural model is limited chordwise to between 25% and 75% chord. This is a common situation where the aerodynamic planform geometry requires the spline methods to extrapolate beyond the structural grid. The Beam Spline can reliably extrapolate in the chordwise direction because it assumes rigid chordwise deformations. However, the Surface Splines are designed to represent plate deformations, and can not make this assumption. This same flutter analysis was performed again using both surface splines to quantify the effect of extrapolation errors.

Table 28-8. FLUTTER CROSSING SUMMARY

FLUTTER CROSSINGS SUMMARY

SYMMETRY TYPE	MACH NUMBER	ALTITUDE	DENSITY	MODE	KFREQ	SPEED	FREQUENCY	VELOCITY
SYMM (+XZ)	0.00000E+00	SL	1.14800E-07	1	0.00000E+00	1.65013E+03	0.00000E+00	1.98015E+04
SYMM (+XZ)	0.00000E+00	SL	1.14800E-07	2	1.00552E-01	1.05555E+03	3.08930E+00	1.26666E+04
SYMM (+XZ)	0.00000E+00	SL	1.14800E-07	4	2.40347E-01	1.67911E+03	1.17465E+01	2.01493E+04

Figure 28-2. V-g AND V-f PLOTS

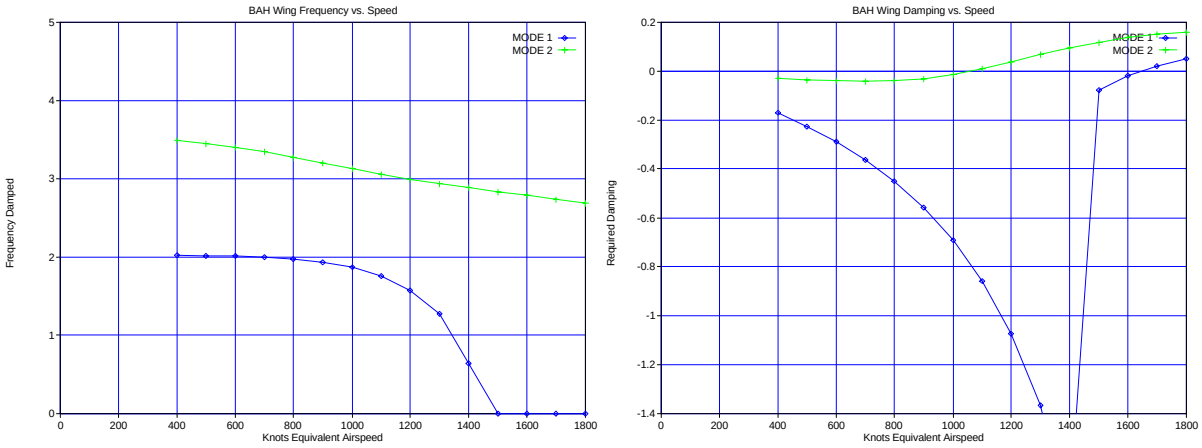


Table 28-9 illustrates the difference in results for each of the spline methods. The infinite surface spline misses the flutter crossing completely, and produces a low estimate of the divergence speed at 1354 ft/sec. The finite surface spline, which assumes a free edge on a localized rectangular surface, yields results that are much closer to the baseline beam spline results with a 3.1 Hz flutter mode at 955 ft/sec and divergence at 1500 ft/sec.

Table 28-9. COMPARISON OF SPLINING METHODS

FLUTTER CROSSINGS SUMMARY

Results from Beam Spline

SYMMETRY TYPE	MACH NUMBER	ALTITUDE	DENSITY	MODE	KFREQ	SPEED	FREQUENCY	VELOCITY
SYMM (+XZ)	0.00000E+00	SEALEVEL	1.14800E-07	1	0.00000E+00	1.65013E+03	0.00000E+00	1.98015E+04
SYMM (+XZ)	0.00000E+00	SEALEVEL	1.14800E-07	2	1.00552E-01	1.05555E+03	3.08930E+00	1.26666E+04
SYMM (+XZ)	0.00000E+00	SEALEVEL	1.14800E-07	4	2.40347E-01	1.67911E+03	1.17465E+01	2.01493E+04

Results from Infinite Surface Spline

SYMMETRY TYPE	MACH NUMBER	ALTITUDE	DENSITY	MODE	KFREQ	SPEED	FREQUENCY	VELOCITY
SYMM (+XZ)	0.00000E+00	SEALEVEL	1.14800E-07	1	0.00000E+00	1.35402E+03	0.00000E+00	1.62482E+04

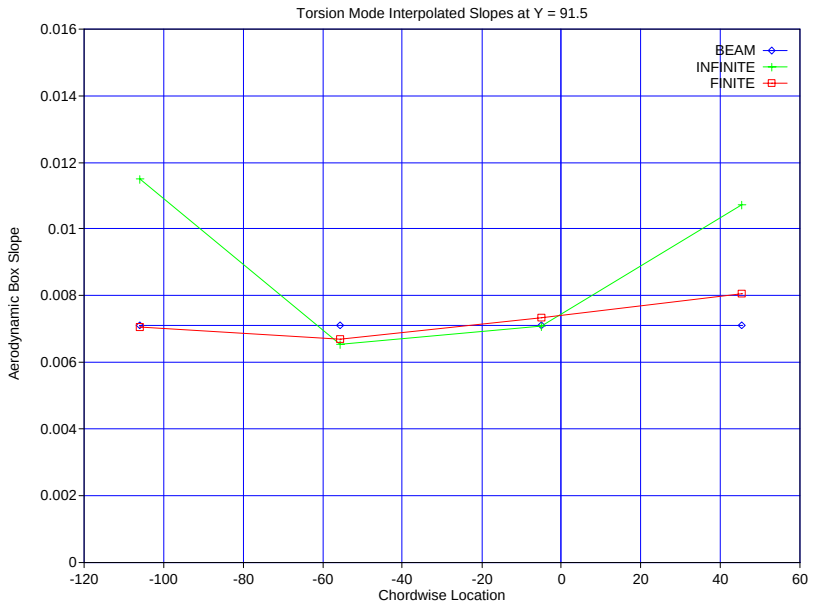
Results from Finite Surface Spline

SYMMETRY TYPE	MACH NUMBER	ALTITUDE	DENSITY	MODE	KFREQ	SPEED	FREQUENCY	VELOCITY
SYMM (+XZ)	0.00000E+00	SEALEVEL	1.14800E-07	2	1.11696E-01	9.55346E+02	3.10592E+00	1.14641E+04
SYMM (+XZ)	0.00000E+00	SEALEVEL	1.14800E-07	4	2.72895E-01	1.49996E+03	1.19142E+01	1.79995E+04

28

The physical effect of these spline methods is better illustrated by plotting the mode shape on the aerodynamic grid. Figure 28-3 is an ***eShell*** plot that compares the torsion mode slopes on the aerodynamic grid at the Y=91.5 in spanwise location. Each curve represents the slopes along the chord produced by each of the spline methods. Although the finite surface spline is not an exact reproduction of the beam spline result, it is a much closer match than the infinite surface spline result. The infinite surface spline assumes a pinned restraint on the spline surface at infinity, resulting in a "potato chip" effect in the extrapolated region. The finite surface spline, by assuming a free edge on a localized rectangular surface, results in much less extrapolation error.

Figure 28-3. INTERPOLATED MODE SHAPES



28.6.2 15° Cantilevered Wings

Tuovila and McCarty [Tuovila55] reported on a series of wind tunnel flutter model tests that evaluated a variety of wing planforms at subsonic and supersonic speeds. The planforms tested included delta and rectangular shapes. Two test cases of a rectangular wing with 15° sweep are analyzed here at Mach 0.45 and 1.30. The geometry of the test model was determined by the material stock used. The 15° sweep on 2.0 inch wide plate stock produces a wing chord of 2.07055, and the semispan is 5.525. The airfoil section is a constant thickness except for a 0.25 inch wide bevel along the leading and trailing edges. The subsonic and supersonic models are constructed from 0.041 inch thickness aluminum.

The structural properties are represented by a rather coarse mesh of QUAD4 elements shown in Figure 28-4. The root boundary conditions as given in [Tuovila55] were stated to be 'clamped' with no further details on the design of the clamping mechanism. It is assumed that there was clamping contact at the front and rear edges of the plate, where the beveled area begins. This produces reasonable agreement with the documented frequencies and mode shapes. The input data are found in file **FLUT2**.

The aerodynamic model, illustrated in Figure 28-5, is rather coarse, like the structural model. In supersonic cases more spanwise divisions are often required, particularly with increasing sweep angle. The need for an increased number of spanwise divisions is due to the theoretical assumption of constant spanwise pressure over the aerodynamic element. The spanwise variation in pressure can be quite abrupt for supersonic flows, so additional spanwise segments are needed to describe these pressure variations. In this particular case, calibration runs using additional spanwise aerodynamic divisions did not have a significant effect.

Figure 28-4. FINITE ELEMENT MODEL

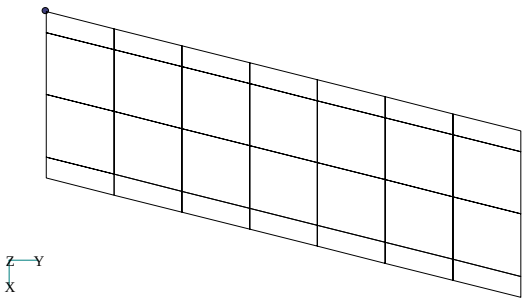
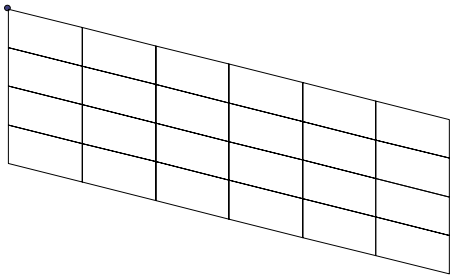


Figure 28-5. AERODYNAMIC MODEL



28

28.6.2.1 Executive and Case Control Packets

The Executive and Case Control packets for the analysis are:

```
SOL MULTI
CEND
TITLE = EXAMPLE FLUT2: THE 15-DEG SWEPTBACK WING FLUTTER ANALYSIS
SET 4   = 0.2, 0.16667, 0.14286, 0.12500, 0.11111, 0.10000
SET 12  = 1.30
SET 13  = 0.01, 0.03, 0.04, 0.05
SET 14  = 1200.0, 1300.0, 1400.0, 1500.0, 1600.0, 1700.0
SPC      = 1
FLPRINT  =SUMMARY
FLSOLVE  = 1
SYMMETRY = SYMM
USING MODES 12
$
SUBCASE 10 AERO
LABEL = MACH 0.45 AERODYNAMICS WITH DYNAMIC SETS
  MACH     = ( 0.45 )
  KFREQ    = ( 0.001, 0.10, 0.12, 0.14, 0.16, 0.2 )
SUBCASE 11 AERO
LABEL = MACH 1.30 AERODYNAMICS USING CASE SETS
  MACH     = 12
  KFREQ    = 13
SUBCASE 12 MODES
  METHOD  = 15
SUBCASE 13 FLUTTER
  LABEL    = MACH 0.45 AERODYNAMICS, K METHOD
  MSELECT  = ( 0.45 )
  KLIST    = ( 0.2, 0.16667, 0.14286, 0.12500, 0.11111, 0.10000 )
  FMETHOD  = K
  DENS     = ( 1.056-7 )
SUBCASE 14 FLUTTER
  LABEL    = MACH 1.30 AERODYNAMICS, PKITER METHOD
  MSELECT  = 12
  VLIST    = 14
  FMETHOD  = PKITER
  DENS     = ( 0.23594-7 )
```

Note there are two AERO cases, to allow the specification of a unique set of reduced frequencies at each Mach number to be analyzed. This is not generally required, however you may sometimes wish to tailor the hard points at a specific Mach number. In general practice, a single comprehensive set could be used. In this case, the flutter point at Mach=0.45 was at a reduced frequency of 0.13 while the Mach 1.3 point was at a reduced frequency of 0.043. This example used a reduced frequency range of 0.001 to 0.2 for Mach=0.45 and 0.01 to 0.05 for the Mach=1.3 data. A common set of 0.001, 0.04, 0.08, 0.15, 0.2 would produce equivalent results.

The **AERO CASE 10** uses the dynamic set feature while **AERO CASE 11** references SET's defined above the CASE level. This option allows the user to produce CASE control commands that are more readable. The **MACH** Command in Case 10 specifies 0.45, while **MACH** Command in CASE 11 reference the Case **SET 12**.

Further, there are two **FLUTTER** cases, in order to specify unique air density and flutter solver methods at each Mach number. The **MSELECT** command is used to restrict the case to a specific Mach number. The default action would be to perform the flutter analysis at all Mach numbers available on the database, in this example at Mach 0.45 and 1.30. The *k* method is selected for the Mach 0.45 Case and the *p-k iter* method for Mach 1.30. The **DENS** command defines a single fluid density for each of the flutter test points.

The multiple cases for multiple Mach numbers approach used in this example is discussed to illustrate the flexibility of **UAI/*NASTRAN***, and is not required as a common practice. However, it is common practice and may be desirable to perform flutter analyses at multiple aerodynamic **SYMMETRY** conditions. To perform a symmetric and antisymmetric analysis would require multiple **AERO** and **FLUTTER** cases.

28.6.2.2 Bulk Data Definitions

The Aerodynamic Reference Geometry is defined with **AEREFS**, which defaults the Aerodynamic Coordinate System and the Reference Coordinate System to the **BASIC** system. The reference chord **CBAR** is 2.07. The single **CAERO1** entry specifies 6 **EQUAL** spanwise divisions and 4 **EQUAL** divisions in the chordwise direction. The **CAERO1** continuation line locates the wing leading edge and chord length at the root and tip.

AEREFS	**0**	**0**	**2.07**						
CAERO1	**101**			**EQUAL**	**6**	**EQUAL**	**4**	**1**	
+	**0.0**	**0.0**	**0.0**	**2.07055**	**1.48044**	**5.52510**	**0.0**	**2.07055**	

An Infinite Surface Spline is used, the first and last box fields are left blank, indicating that all boxes are to be splined. The structural grids to be used are defined by the **SET1** entry.

SPLINE1	**200**	**101**			**100**	**0.0**	**INFINITE**		
SET1	**100**	**2**	**4**	**6**	**8**	**9**	**11**	**13**	**+SET1**
+SET1	**15**	**18**	**20**	**22**	**24**	**25**	**27**	**29**	**+SET2**
+SET2	**31**	**34**	**36**	**38**	**40**				

The flutter analysis condition velocities were input in units of feet/sec. The solution units of this model are inches/sec so **AEUNITS** will be input to define a **VREF** of 12.0 to relate input and output units to the solution units.

AEUNITS	**12.0**								

28.6.2.3 Discussion of Results

The Case Control command **FLPRINT=SUMMARY** results in a print of the V-g and V-f curve data which is not reproduced here. The remainder of printed output is automatic, including the Flutter Speed Crossings shown in Table 28-10 below.

The subsonic test point was reported to be V_f = 495 ft/sec and F_f = 120 Hz compared with the **UAI/*NASTRAN*** result of 500 ft/sec and 114 Hz respectively. The supersonic point was overpredicted, with the test point at was V_f = 1280 ft/sec and F_f = 102 Hz compared with the predicted speed of 1620 ft/sec and a 139 Hz flutter frequency. This result is consistent with other efforts to correlate the Mach 1.3 point.

Table 28-10. FLUTTER SPEED CROSSINGS

FLUTTER CROSSINGS SUMMARY

SYMMETRY TYPE	MACH NUMBER	ALTITUDE	DENSITY	MODE	KFREQ	SPEED	FREQUENCY	VELOCITY
SYMM (+XZ)	4.50000E-01	..N/A..	1.05600E-07	2	1.23448E-01	5.00004E+02	1.13866E+02	6.00004E+03
SYMM (+XZ)	1.30000E+00	..N/A..	2.35940E-08	2	4.66629E-02	1.61909E+03	1.39373E+02	1.94291E+04

28

28.6.3 Simply Supported Panel

Panel flutter is the flutter of flat or curved panels in supersonic flow. Such flutter is a local design requirement on fuselage skins and other similiar structures. The most common examples of aeroelastic flutter result from a combination of bending and twisting motions such as those that occur for a cantilevered wing. If it were possible, an additional restraint at the free end of the wing could substantially increase the flutter speed. However, panel flutter is a special case in supersonic flow where the panel with all edges fully restrained, can still flutter. Instead of the bending and torsion motions of the wing tip, the instability arises from coupling between the first and second streamwise bending modes of the panel.

This example illustrates the solution of a flat rectangular simply supported panel. Since the intention is to simulate flow on only the outer surface, the fluid density is halved. The solution to the following example was given by [Hedgepeth57].

The structural model is a 10 by 10 element mesh representing a 10 inch by 10 inch square panel. The aerodynamic model consists of three panels, each with 10 equal spanwise and chordwise divisions. The central panel is similiarly sized and splined to the structural grid, while the outer panels on either side are sized to keep the Mach lines from the free edge away from the central panel. As shown in Figure 28-6, this arrangement simulates the panel as part of a larger surface. The central panel is highlighted in the figure to distinguish it from the adjoining interference elements. Recall that the Mach lines form an angle to the flow by:

$$\theta = \sin^{-1}\left(\frac{1}{M}\right)$$

This relationship requires that as the Mach number is reduced, the side panels must become wider to maintain 2-dimensional supersonic flow over the central panel. The aerodynamic and structural grids are connected by a finite surface spline. The input data are found in file **FLUT3**.

Figure 28-6. AERODYNAMIC MODEL

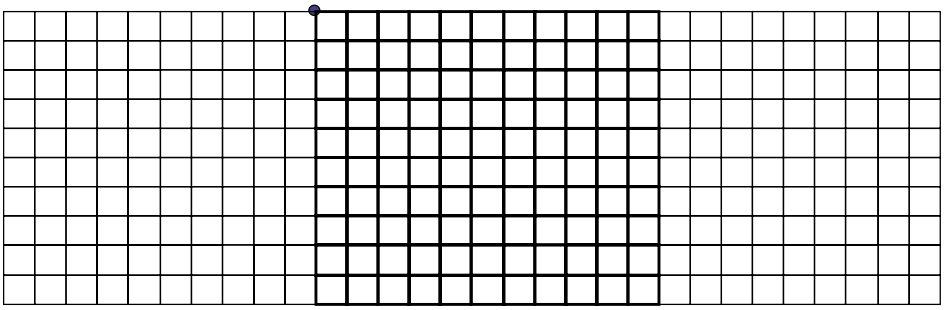


Table 28-11. FLUTTER CROSSINGS

FLUTTER CROSSINGS SUMMARY

SYMMETRY TYPE	MACH NUMBER	ALTITUDE	DENSITY	MODE	KFREQ	SPEED	FREQUENCY	VELOCITY
NOSYMM	2.00000E+00	..N/A..	5.73400E-08	3	2.47741E-01	1.86206E+03	1.76207E+02	2.23447E+04
NOSYMM	3.00000E+00	..N/A..	5.73400E-08	3	2.02488E-01	2.42112E+03	1.87261E+02	2.90535E+04

The analysis is performed, and the resulting Flutter Crossing is shown in Table 28-11. These results compare favorably with the results in [Hedgepeth57]. The stability parameter λ_{cr} is defined by,

$$\lambda_{cr} = \frac{2qL^3}{\beta D}$$

where:

$$\beta^2 = M^2 - 1 \text{ and } D = \frac{Et^3}{12(1-g\nu^2)}$$

and L is the streamwise panel length, and q is the critical dynamic pressure. For the two cases analyzed at Mach numbers of 2.0 and 3.0, the **UAI/*NASTRAN*** results are 1859 ft/sec and 2407 ft/sec respectively. These translate to stability parameters λ_{cr} of 500 and 513, compared with 492 and 499 from [Hedgepeth57].

Another important influence on panel flutter can be the effect of preload. This is similar to panel buckling where the effect of a inplane tension or compression load can stabilize or destabilize the system. This effect can be included in your analysis by performing the flutter analysis using structural modes that were computed by the Normal Modes with Differential Stiffness method as described in Chapter 16.

28.6.4 Horizontal Stabilator

This example problem illustrates the use of **UAI/*NASTRAN*** in a production environment. The analyst wishes to define a flutter boundary. This boundary may be defined using either a constant density criteria, or a match point boundary criteria. The flutter analysis may be performed with only match point conditions, or with off-matchpoint conditions.

Because there are many ways to perform a flutter analysis, this example covers a wide range of approaches by discussing three different combinations of analysis and results processing approaches:

- A match point analysis which automatically defines a match point boundary
- A constant density analysis with a constant density boundary criteria applied
- A constant density analysis with a match point boundary criteria applied

The example problem is the left-half symmetric model of an all moving horizontal stabilator. The structural model is in Figure 28-7, and the aerodynamic model in Figure 28-8. The exam-

Figure 28-7. STRUCTURAL MODEL

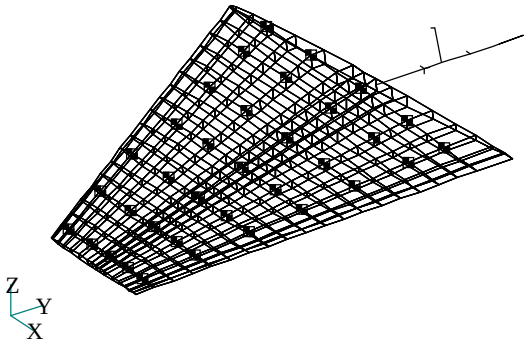
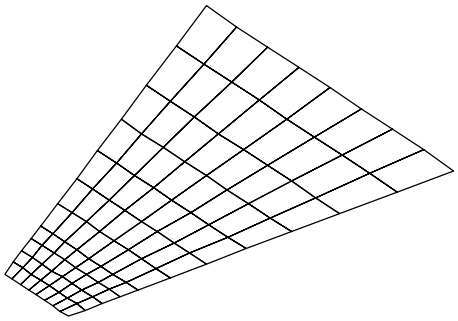


Figure 28-8. AERODYNAMIC MODEL



ple exhibits boundary condition-specific flutter behavior, and results in multiple modes of instability. The input data for this problem are found in file **FLUT4**.

28.6.4.1 Case Control Data Packet

To analyze the symmetric and anti-symmetric flutter boundaries for a range of Mach and altitude combinations, the following Case Control commands are used:

```
TITLE = HORIZONTAL STABILATOR FLUTTER
SUBTITLE = MACH 0.4 to 1.6, ALTITUDE  -16K to 30K FT
ARCHIVE
MPC = 220
METHOD=10
$ AERO CASE DEFAULTS
 MACH  = ( 0.4,0.6,0.85, 1.2, 1.40, 1.6 )
 KFREQ = ( 0.001,0.1,0.2,0.4,0.8,1.4,2.0,3.0 )
$ FLUTTER CASE DEFAULTS
 DENS = 12
 NROOT=10
 VLIST=  (  300., 350., 400.,450., 500., 550., 600.,650., 700., 750.,
         800.,850., 900., 950.,1000.,1050.,1100.,1150.,1200.,1250.)
 FMETHOD = PKITER
 FLPRINT=SUMMARY
 FLSOLVE = 1
$CASE 100 AERO
 SYMMETRY=SYMM
CASE 200 AERO
 SYMMETRY=ANTI
```

Continued on next page

```
CASE 1001 MODES
 LABEL = SYMMETRIC MODES
 SPC = 1000
CASE 1002 MODES
 LABEL = ANTI-SYM MODES
 SPC = 2000
CASE 2001 FLUTTER
 SYMMETRY=SYMM
 LABEL = SYMMETRIC FLUTTER, PKITER
 SPC = 1000
 USING MODES 1001
 DISP=ALL
CASE 2002 FLUTTER
 SYMMETRY=ANTI
 LABEL = ANTI-SYM FLUTTER, PKITER
 SPC = 2000
 USING MODES 1001
 DISP=ALL
```

The aerodynamic model is generated for a range of Mach numbers and reduced frequencies for symmetric and anti-symmetric boundary conditions. Similarly, two **MODES** Cases are required to impose the symmetric **SPC** set, 1000, and the anti-symmetric **SPC** set, 2000. In the **FLUTTER** Cases the relevant SPC sets are called out, remembering that the **FLUTTER** Case requires the same structural boundary condition that was used in the **MODES** case that it references with the **USING MODES** command.

The *p-k iter* method is selected and a SUMMARY print is requested to produce a V-g data printout along with the Flutter Speed Crossing output. The flutter analysis solution conditions are defined by the combination of **DENS** and **VLIST** commands. The **VLIST** command is included to define a Constant Density Analysis.

This example uses several common options for both the symmetric and anti-symmetric cases, such as the Mach and reduced frequency hard points for the aerodynamic model. These common Case Control commands are entered once before any analysis Cases.

28.6.4.2 Bulk Data Packet

The structural model uses a common modeling practice for full depth honeycomb sandwich structure. The cover skins are QUAD4 elements while the honeycomb core is a combination of SHEAR and ROD elements. **PARAM,WTMASS** is used to convert the input weight units to consistent lb-in-sec units, and **PARAM,GRDPNT** requests the total weight matrix integrated about the root hinge position.

The Aerodynamic reference geometry is input on the **AEREFS** defines the reference length as 48.0inches. As noted above, Figure 28-8 illustrates the aerodynamic paneling, which is defined by a single **CAERO1** panel extending from the vehicle centerline to the surface tip. This aerodynamic panel defines both the moving surface and a non-moving fuselage section. The spanwise divisions are input with an **AEFACT** entry to precisely locate the spanwise break between the moving and non-moving surfaces. The break line is 27.5 inches out from the centerline, giving the division at 0.296. An additional inboard division is included to produce two strips on the fuselage and there are nine strips used on the movable surface.

AEREFS	0	0	48.						
AEFACT	101	0.0	0.148	0.296	0.423	0.539	0.643	0.736	
	0.821	0.866	0.913	0.956	1.000				
CAERO1	1001	1		SET	101	EQUAL	8	1	
	481.881	0.0	-7.393	88.463	541.67	-93.04	-16.14	22.8	

The **FLUTTER** Case specific inputs define the splines that relate structural grid motion to motion of the aerodynamic grid and the flutter conditions to be solved. The **SPLINE1** requests a Finite Surface Spline attached to aero boxes 1017 to 1088, leaving the fuselage section motionless. The **SET1** entry lists a spread of 70 structural grids to be splined, 7 chordwise points at 10 spanwise locations.

SPLINE1	1000	1001	1017	1088	101	0.	FINITE		
SET1	101	31203	31205	31209	31213	31217	31221	85203	
	31225	36225	42225	48225	54225	60225	85205	85209	
	36203	36205	36209	36213	36217	36221	85213	85217	
	42203	42205	42209	42213	42217	42221	85221	79203	
	48203	48205	48209	48213	48217	48221	79205	79209	
	54203	54205	54209	54213	54217	54221	79213	79217	
	60203	60205	60209	60213	60217	60221	79221	73203	
	66203	66205	66209	66213	66217	66221	73205	73209	
	66225	73225	79225	85225	73213	73217	73221		

The **AEUNITS** entry relates the user input velocities in Knots to the solution units of in/sec. The **FLSOLVE** entry requests the use of **CUBIC** spline method of generalized aerodynamic force interpolation, and sets limits on the flutter crossings to include only modes that produce a peak damping required below -0.01 and to locate the crossing by the 0.0001 damping required position. This was used to avoid tracking lightly damped hump modes that never generate significant damping levels.

AEUNITS	20.	24	KNOTS						
FLSOLVE	1	CUBIC			0.01	0.0001			

The **ATMOS** entry defines the altitude/densities that will be solved. When no **VLIST** Case Control command was used, only match point velocities were solved using the current solution density's **VELM1** field from **ATMOS** scaled by the current Mach Number. The **VELM1** fields are input in units of Knots Equivalent, due to the use of **VFAC=20.24** on **AEUNITS** and the Reference Density on the first line of the **ATMOS** Bulk Data Entry. Some intermediate altitudes were added for the match point velocities at 5000. ft. altitude increments above Sea Level and 4000. ft. increments below. These increments are somewhat arbitrary, however the 8000. feet below sea level density was chosen to coincide with a 1.15 speed margin requirement.

ATMOS	**12**	**1.147-7**							
	H30K	**0.3747**	**360.6**						
	H20K	**0.5332**	**448.3**						
	H10K	**0.7386**	**548.2**						
	SL	**1.0**	**661.**						
	HN8K	**1.256**	**760.9**						
	HN16K	**1.5598**	**869.8**						

28.6.4.3 Discussion of Results

This model was run twice, the first time without the **VLIST** request so that only the Match Point velocities would be solved, then with a user specified list of velocities that were run at each Mach number, density combination.

Match Point Flutter Results

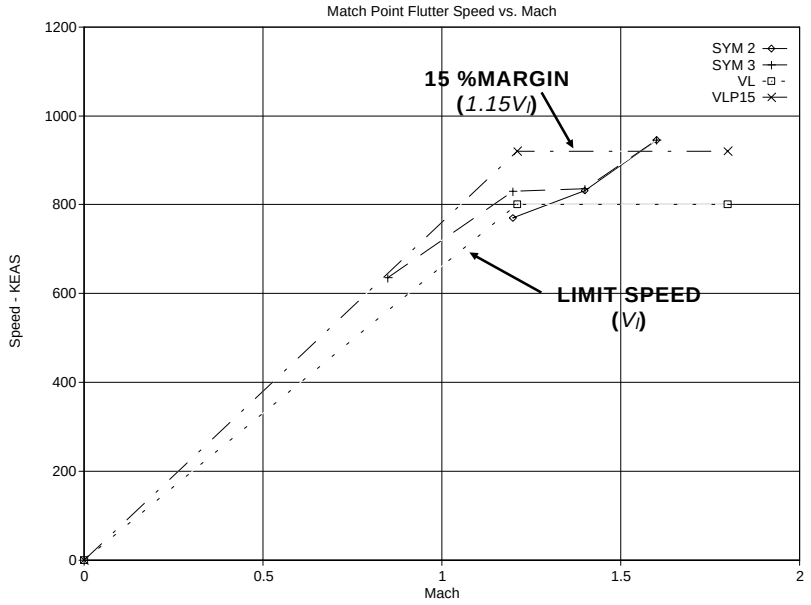
The critical flutter mechanism is a primary bending torsion coupling. As expected for a high speed vehicle, the analysis results show no flutter below Mach 0.85, primarily due to the stabilizing effect of increased damping from the higher air density. An excerpt of the printed V-g output for Mach 1.20, Antisymmetric and the resulting summary print of the flutter crossings are shown in Table 28-12.

Table 28-12. V-g RESULTS AND FLUTTERS CROSSINGS

```
                              MODE   MACH NUMBER     SYMMETRY TYPE      FLUTTER METHOD
                              ----   -----------     -------------      --------------
                                 1    1.2000E+00       ANTI (-XZ)           PKITER

     ALTITUDE      DENSITY       KFREQ        SPEED       DAMPING     FREQUENCY    VELOCITY       LAMAR        LAMAI

     H30K       4.29781E-08  2.26773E-01  4.32720E+02 -2.20755E-01  2.15167E+01  1.43079E+04 -1.50141E+01  1.35194E+02
  ... lines removed
     HN4K       1.40393E-07  2.67782E-01  8.51880E+02 -2.27691E-02  2.76751E+01  1.55847E+04 -1.97976E+00  1.73888E+02
     HN8K       1.44063E-07  2.51688E-01  9.13080E+02  3.34277E-02  2.75230E+01  1.64901E+04  2.89076E+00  1.72932E+02
     HN12K      1.60695E-07  2.45013E-01  9.77040E+02  4.04885E-02  2.71457E+01  1.67072E+04  3.45360E+00  1.70562E+02
     HN16K      1.78909E-07  2.35778E-01  1.04376E+03  3.22796E-02  2.64477E+01  1.69152E+04  2.68239E+00  1.66176E+02

                              MODE   MACH NUMBER     SYMMETRY TYPE      FLUTTER METHOD
                              ----   -----------     -------------      --------------
                                 2    1.2000E+00       ANTI (-XZ)           PKITER

     ALTITUDE      DENSITY       KFREQ        SPEED       DAMPING     FREQUENCY    VELOCITY       LAMAR        LAMAI

     H30K       4.29781E-08  2.75191E-01  4.32720E+02 -3.48943E-05  2.61107E+01  1.43079E+04 -2.86236E-03  1.64059E+02
     H25K       5.14544E-08  2.69491E-01  4.83480E+02 -4.70156E-05  2.61104E+01  1.46103E+04 -3.85660E-03  1.64056E+02
     H20K       6.11580E-08  2.64045E-01  5.37960E+02 -6.00249E-05  2.61097E+01  1.49113E+04 -4.92360E-03  1.64052E+02
     H15K       7.22036E-08  2.58931E-01  5.96040E+02 -6.02066E-05  2.61084E+01  1.52051E+04 -4.93826E-03  1.64044E+02
     H10K       8.47174E-08  2.54103E-01  6.57840E+02  1.09993E-05  2.61063E+01  1.54927E+04  9.02110E-04  1.64031E+02
  ... lines removed
     HN16K      1.78909E-07  2.32813E-01  1.04376E+03  2.38344E-03  2.61152E+01  1.69152E+04  1.95545E-01  1.64086E+02


MACH 0.4 TO 1.6, ALTITUDE  -16K TO 30K FT.
SYMMETRIC FLUTTER, PKITER                                                                          CASE 2001 FLUTTER

                                        FLUTTER CROSSINGS SUMMARY

        SYMMETRY TYPE    MACH NUMBER ALTITUDE    DENSITY    MODE     KFREQ        SPEED      FREQUENCY    VELOCITY
      ------------------ ------------ -------- ------------ ---- ------------ ------------ ------------ ------------
          SYMM (+XZ)      1.20000E+00 ..N/A..   1.30902E-07    1  2.72281E-01  8.21755E+02  2.81117E+01  1.55690E+04
          SYMM (+XZ)      1.40000E+00 ..N/A..   1.10416E-07    1  2.50371E-01  8.78588E+02  3.00923E+01  1.81243E+04
          SYMM (+XZ)      1.60000E+00 ..N/A..   1.07898E-07    1  2.27695E-01  9.92323E+02  3.12680E+01  2.07080E+04

ANTI-SYM FLUTTER, PKITER                                                                           CASE 2002 FLUTTER

                                        FLUTTER CROSSINGS SUMMARY

        SYMMETRY TYPE    MACH NUMBER ALTITUDE    DENSITY    MODE     KFREQ        SPEED      FREQUENCY    VELOCITY
      ------------------ ------------ -------- ------------ ---- ------------ ------------ ------------ ------------
          ANTI (-XZ)      8.50000E-01 ..N/A..   1.40713E-07    1  3.59931E-01  6.37273E+02  2.77958E+01  1.16453E+04
          ANTI (-XZ)      1.20000E+00 ..N/A..   1.25947E-07    1  2.56834E-01  8.39823E+02  2.76278E+01  1.62213E+04
          ANTI (-XZ)      1.40000E+00 ..N/A..   1.12947E-07    1  2.47117E-01  8.88839E+02  2.97092E+01  1.81292E+04
          ANTI (-XZ)      1.60000E+00 ..N/A..   1.08045E-07    1  2.26849E-01  9.93014E+02  3.11523E+01  2.07083E+04
```

28

Figure 28-9. MATCH POINT FLUTTER ENVELOPE



Using ***eShell***, the above results can be presented in a number of ways. One example is an envelope plot of the flutter speed versus Mach Number, with the vehicle flight envelope and required margin included. These results are shown in Figure 28-9. Note these results represent a preliminary design case whose flutter speed is just outside the envelope but with insufficient margin.

This plot was created by the following ***eShell*** script. This script is delivered in the file **vm_mpa** with additional ***eShell*** scripts which are discussed in the following section.

```
/* Vf vs Mach for match point flutter analysis     */
/* usage  < vm_mpa ;>                              */
/*                                                 */

set xmin to 0.0;
set xmax to 2.0;
set ymin to 0.0;
set ymax to 1200.0;

set ftitle to 'Match Point Flutter Speed vs. Mach';
set xtitle to 'Mach ';
set ytitle to 'Speed - KEAS ';

mxyplot sym,mach,keas from flut_crossings
       order by sym,mach;

addcurves mach,vl,vlp15 from flt_env where rho_label='SL';
replot;
```

The vehicle flight envelope lines were produced from the relation **flt_env** which was created using data from the **ATMOS** Bulk Data entry which was also stored on the ARCHIVE database. The script used to create the relation **flt_env** is shown below and delivered in the file **envdata**.

```
/*
envdata,    to create envelope lines
usage < envdata atmos_id ;>
the margin, max mach and max speed are set in this script for
 a) 15% speed margin
 b) 1.8 max mach
 c) 800 keas limit speed
*/
purge  rel flt_env;
create rel flt_env(group int,inum int,rho_label char(8), mach rsp,VL rsp,VLp15
rsp );
undefine *;
sel distinct group from atmos;
/* Select ATMOS Group ID:                          */
insert into flt_env sel group,1,label,0.0,0.0,0.0 from atmos
      where group=&aid;
/* points at mach for max q  */
insert into flt_env sel group,5,label,800./velm1,800.,920. from atmos
      where group=&aid;
/* points at q for max mach   */
insert into flt_env sel group,3,label,1.8,  1.8*velm1,2.07*velm1 from atmos
      where group=&aid;
/* end points at max mach and max q  */
insert into flt_env sel group,6,label,1.8,800.,920. from atmos
      where group=&aid;
delete from flt_env where mach>1.81;
/* delete from flt_env where VL >801. ; */
delete from flt_env where VLp15 >925.;
insert envdata script, slide 15 from ppt
```

The flight envelope data was produced with a script however the data could be just as easily inserted by hand. In this case, a script was used so that constant density envelope lines could be created for every **ATMOS** density. In an ongoing project, this envelope relation could be created once and stored in a separate database. This small database would be opened and used with a variety of results during the course of the project.

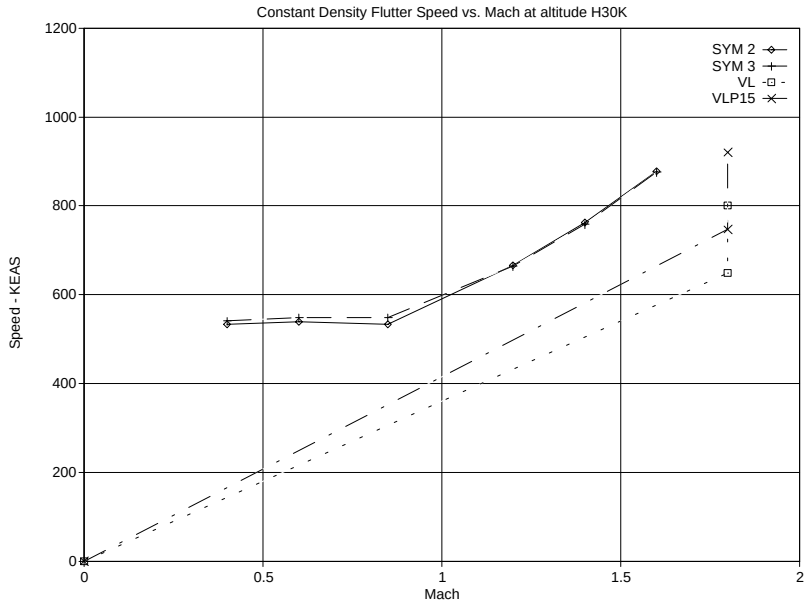
Constant Density Flutter Results

The same model was run a second time with a set of velocities specified with the **VLIST** Command. This produces a Constant density analysis, meaning that the flutter speed is solved for at each Mach Number, Density combination. The resulting flutter speed margin is then defined as the ratio of the flutter speed to the required speed at a specified Mach/Density combination. The plot below shows the flutter speed as a function of Mach number at the density corresponding to Sea Level altitude. A separate plot is produced at each altitude, and a flutter speed margin is defined at any Mach number at each altitude.

This can be rather conservative since the flutter speed can be moderately sensitive to density and this criteria does not account for potential increase in flutter speed with increased density. Figure 28-10 is an ***eShell*** plot of the flutter speed using constant density criteria.

28

Figure 28-10. CONSTANT DENSITY FLUTTER SPEED



This plot was produced using the following ***eShell*** script.

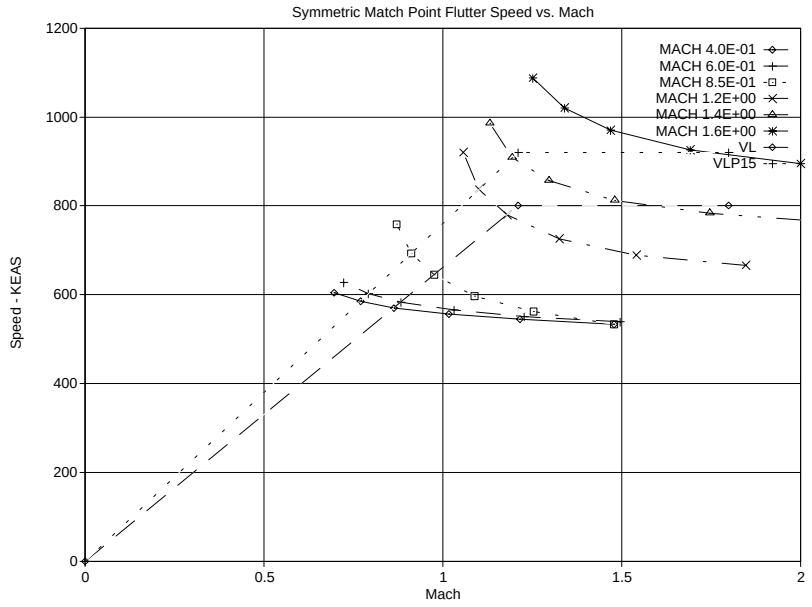
```
/*
Vf vs Mach for constant density analysis
   usage    vm_cd < rho_label >
   example: vm_cd h30k
*/
set xmin to 0.0;
set xmax to 2.0;
set ymin to 0.0;
set ymax to 1200.0;

set ftitle to 'Constant Density Flutter Speed vs. Mach at altitude &1';
set xtitle to 'Mach ';
set ytitle to 'Speed - KEAS ';

select  mode,freq,sym,mach,keas from flut_crossings
       where rho_label='&1'
       order by sym,mach;

mxyplot sym,mach,keas from flut_crossings
       where rho_label='&1'
       and freq>2. and freq<50.
       order by sym,mach;

  addcurves mach,vl,vlp15 from flt_env where rho_label='&1';
replot;
```

Match Point Boundaries using Constant Density Flutter Results

In order to remove potential conservatism due to the constant density criteria, this section uses the same results as the previous section but applies a match point criteria.

The second analysis specified a list of velocities with **VLIST** Command. This can produce a flutter crossing at every Mach Density combination, however the velocity may not occur in the

Figure 28-11. MATCH POINT FLUTTER SPEED CRITERIA



atmosphere. To illustrate the flutter stability behavior with changing density, the flutter crossings can be cross plotted against the Mach number required. The Mach number required is defined as the Mach number in the atmosphere corresponding to the density and speed of the flutter crossing. Each curve is plotted as a line of flutter speed at a constant Mach Number for various densities. The *match point* flutter speed is the point of intersection between a curve's Mach number required and the corresponding analysis Mach. Figure 28-11 is an ***eShell*** plot using the match point criteria applied to the results of a constant density analysis. This plot was produced using the ***eShell*** script **vm_mp**.

```
/*
Vf vs Mach for match point plots of constant density analysis
   usage  < vm_mp >
*/

set xmin to 0.0;
set xmax to 2.0;
set ymin to 0.0;
set ymax to 1200.0;

set ftitle to 'Symmetric Match Point Flutter Speed vs. Mach';
set xtitle to 'Mach ';
set ytitle to 'Speed - KEAS ';

select sym, mode,freq,mach,mach_req,keas from flut_crossings
       where  sym=2 and freq<50. and freq> 1.
       order by mach,keas;

mxyplot mach,mach_req,keas from flut_crossings
       where  sym=2 and freq<50. and freq> 1.
       order by mach,keas;

addcurves mach,vl,vlp15 from flt_env where rho_label='SL' ;

replot;
```

28

28.7 REFERENCES

[Appa85]: Appa, K., Cowan, D.L., "The Determination of Load and Slope Transformation Matrices for Aeroelastic Analyses" *Journal of Aircraft*, Vol. 22, August 1985, pp. 734-736.

[Appa87]: Appa, K., "Constant Pressure Panel Method for Supersonic Unsteady Airloads Analysis", *Journal of Aircraft*, Volume 24, October 1987, pp 696-702.

[Appa88]: Appa, K. and Smith M.J.C., "Evaluation of the Constant Pressure Panel Method (CPM) for Supersonic Unsteady Air Loads Prediction", presented as paper AIAA-88-2282 at AIAA/ASME/ASCE/AHS 29th Structures, Structural Dynamics and Materials Conference, Williamsburg, Virginia, April 1988.

[Appa89]: Appa, K., " Finite-Surface Spline ", *Journal of Aircraft*, Vol. 26, No. 5, 1989, pp. 495 - 496.

[Bisplinghoff55]: Bisplinghoff, R.L., Ashley, H. and Halfman, R.L. (1955). *Aeroelasticity*, Addison-Wesley Publishing Company.

[Geising72]: Giesing, J.P., Kalman, T.P., and Rodden, W.P., "Subsonic Steady and Oscillatory Aerodynamics for Multiple Interfering Wings and Bodies," *Journal of Aircraft*, Volume 9, No. 10, October 1972, pp. 693-702.

[Harder72]: Harder, R.L. and Desmarais, R.N., "Interpolation Using Surface Splines", *Journal of Aircraft*, Volume 9, No. 2, February 1972, pp. 189 - 191.

[Hedgepeth57]: Hedgepeth, J. M. (1957). Flutter of Rectangular Simply Supported Panels at High Supersonic Speeds. *J. Aero. Sci.*, Vol. 24, pp. 563-573, 586.

[Rodden59]: Rodden,W. P. (1959) A Matrix Approach to Flutter Analysis. Institute of the Aeronautical Sciences, Fairchild Fund Paper No. FF-23.

[Tuovila55]: Tuovila, W.J., and McCarty, J. L., (1955) Experimental Flutter Results for Cantilever Wing Models at Mach Numbers Up to 3.0, NACA RM L55E11.

[Zienkiewicz77]: Zienkiewicz, O. C., *The Finite Element Method*, McGraw-Hill Book Company (UK) Limited, 1977.

Chapter 29

STRUCTURAL PLOTTING

Structural Plotter commands are used to define plots of the finite element model and its GRID point-based structural responses. The actual Structural Plots are not created during your **UAI/*NASTRAN*** execution. Rather, the plot information is placed on a file that is post-processed.

UAI provides four utilities for this purpose. The first two programs, available on most computers, are ***uaiplotps*** and ***uaiplotgl***. These utilities allows you to send your plots to a PostScript laser printer and a Hewlett Packard graphics device, respectively. The third program, available on Unix host computers is ***uaiplot***. This is an interactive program, based on Motif and X-Windows, which allows you to selectively view your plots. Chapter 31 of this manual and Chapter 1 of the ***User's Reference Manual*** provides instructions on how to use these programs for most systems. There is also a program, provided as source code, illustrating how your facility may create their own plotter software, if necessary. In some cases, you must contact your **UAI/*NASTRAN*** Support Specialist for details describing how your site interfaces with these post-processor.

This Chapter describes the Structural Plotter terminology, the input data requirements for creating such plots, and provides examples which illustrate the use of many of the plotter commands.

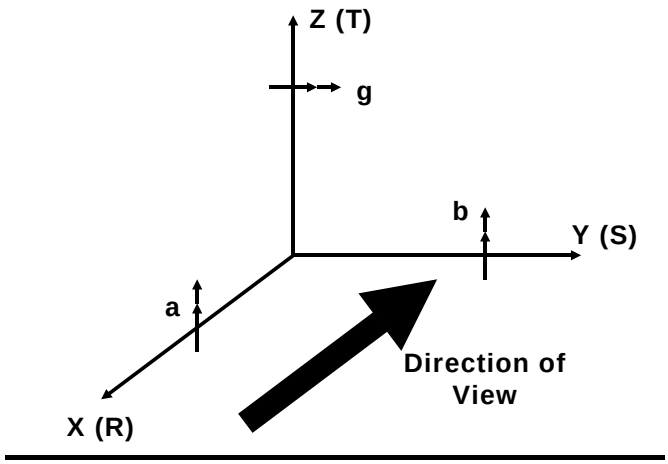
29.1 STRUCTURAL PLOTTING TERMINOLOGY

This section reviews important plotting concepts that will assist you in creating plots of your model and its solution results.

29

Plotter Coordinate System. When performing graphics, there is an underlying coordinate system which is called the plotter coordinate system. The actual structural model coordinate system is mapped to the plotter system prior to plotting. The plotter coordinate system is called the ***RST-System***. This system is fixed with respect to the plot: you are generally looking down the R-axis toward the plotter system origin. Initially, the model coordinate system is aligned with the plotter coordinate system as shown in Figure 29-1. Rotations are always performed with respect to the RST-System. The angles of rotation are shown in the Figure.

Figure 29-1. PLOTTER COORDINATE SYSTEM



Graphic Projections. When three-dimensional objects are plotted, there are numerous ways in which their coordinates may be mapped to a two-dimensional plotting surface. **UAI/*NASTRAN*** supports two of these, the ***orthographic projection*** and the ***perspective projection.*** When an object is projected to the surface along parallel lines, it is called orthographic. If, on the other hand, the object is projected along lines that converge to a point, it is called perspective. These two cases are illustrated in Figure 29.2.

Vantage Point. As seen in Figure 29.2b, the point at which the lines of a perspective projection converge is called the ***Vantage Point***. The vantage point is used for two purposes. For both the orthographic and perspective projections, it allows you to define the RST-coordinates of your vantage point. This allows you to view your model from a number of different angles. This feature provides an alternative to defining the angles of rotation of the model. When you use the perspective projection, the vantage point also controls the amount of perspective that you see. The closer you are to your model, the more perspective distortion in the plot.

Plot Sets. A plot set defines a group of elements within your model. Plot sets are specified by using the **SET** command.

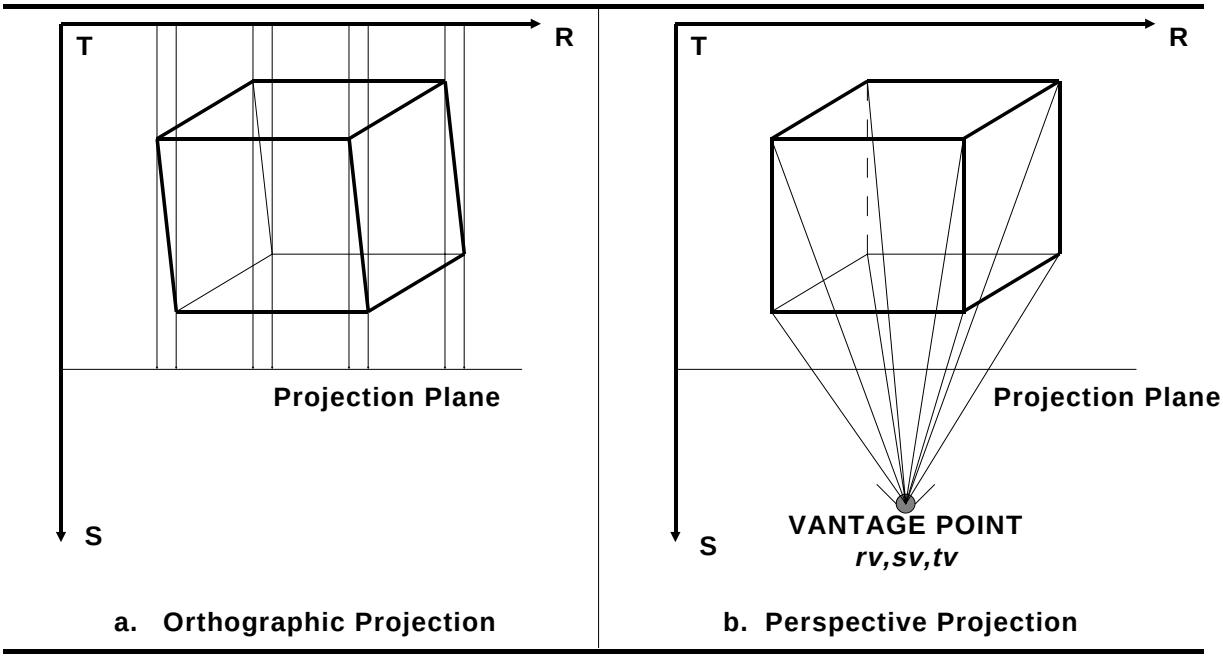
Plot. A plot is a single frame which displays one or more ***subplots*** and, optionally, their solution results. Each of the subplots is defined as a separate plot set and each may have their own graphic characteristics.

Deformed Plot. A deformed plot is one which is created by applying the static deformations to the GRID point locations prior to plotting. Because the deformations are often very small relative to the model, they are scaled in order to be visible. When performing dynamic response analyses, you may also represent the velocities and accelerations as deformations.

Vector Plot. Rather then creating a deformed plot, you may also request that the deformations be plotted in the form of vectors. The vectors originate at the GRID points and point in a direction which is determined by the displacement components that you select.

Labels. You may selectively label the GRID points and finite elements within your model. This may be done when plotting the structural model or its deformed shape.

Figure 29-2. GRAPHIC PROJECTIONS



Coordinate System Triad. Each plot includes a coordinate system triad which shows the orientation of your model coordinate system with respect to the plot surface.

29.2 STRUCTURAL PLOTTER DATA REQUIREMENTS

This section provides you with a description of the **UAI/*NASTRAN*** input data required for creating Structural Plots.

29.2.1 Executive Control Commands

Although there are no specific Executive Control commands which are required to perform Structural Plotting, you may **ASSIGN** a logical file on which the plot data will be written for use by the UAIPLOT plotting program. This is done with the command:

```
ASSIGN logical_name=physical_name,NEW,USE=PLOT
```

The assignment of the plot file is optional. If you do not assign the appropriate file, the default value defined in the **UAI/*NASTRAN*** section of the Preference File will be used. There are two formats available for the plot file, **BINARY** and **FORMATTED**. The one which you use depends on the implementation of the UAIPLOT utility at your site. Contact your Support Specialist for detailed information.

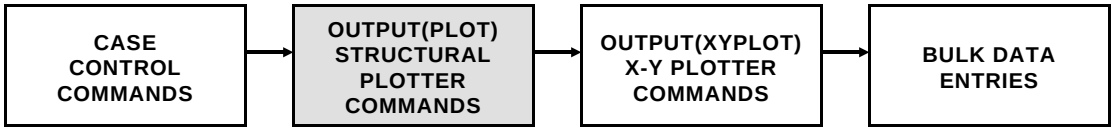
29.2.2 Substructure Control Commands

When you use the Substructuring capability, the **BASIC** command has an optional subcommand called **SAVEPLOT** which defines a set of elements which may be used for creating plots during a PHASE 2 operation. When using this subcommand, your PHASE 1 data stream must include a Structural Plotter subpacket which includes the definition of the referenced set.

29.2.3 Case Control Commands

The Structural Plotter commands form a subpacket within the Case Control command packet as shown in Figure 29-3.

Figure 29-3. THE STRUCTURAL PLOTTER SUBPACKET



Both the Structural Plotter subpacket and the X-Y Plotter subpacket, described in Chapter 30, may be interchanged in the data stream. It is not necessary for you to issue Case Control output request commands when you are requesting **DEFORM**ed or **VECTOR** Plots. **UAI/*NASTRAN*** automatically extracts the necessary data from the solution results.

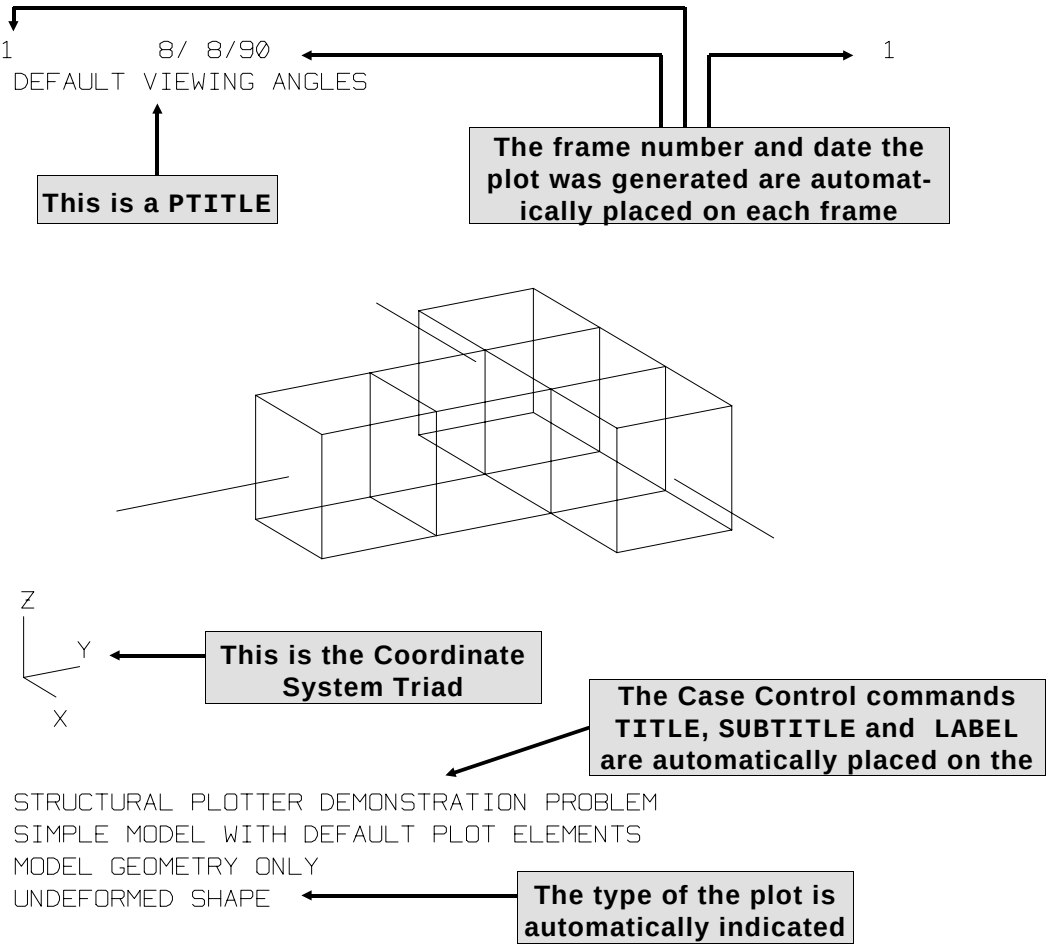
29.2.4 Bulk Data

There is a single Bulk Data entry related to Structural Plotting. The **PLOTEL** entry, which is not an actual modeling element, allows you to define straight lines which connect two GRID points. You may then select these lines when creating your plots. This feature is useful for providing customized capabilities not explicitly supported by the Structural Plotter, such as the outline of your structural model.

29.3 STRUCTURAL PLOT ELEMENTS

Figure 29-4 shows a typical structural plot. The plot is annotated to indicate the various plot elements. Some of these are controlled by commands that you may use, while others are automatically generated by **UAI/*NASTRAN***. The next section describes these commands in more detail.

Figure 29-4. BASIC STRUCTURAL PLOT ELEMENTS



29.4 STRUCTURAL PLOTTER COMMANDS

The general form of the Structural Plotter Command packet is:

```
OUTPUT(PLOT)
      option_command
      option_command
      ...
      PLOT_command
          option_command
          option_command
PLOT_command
          ...
OUTPUT(XYPLOT) or BEGIN BULK
```

The packet must begin with the **OUTPUT(PLOT)** command and it ends when either an **OUTPUT(XYPLOT)** or **BEGIN BULK** command is encountered. The other commands in the packet define the plot frames that you wish to create. Some of these are *option_command*s which define the characteristics of the plots to be made while others are the *PLOT_command*s which actually specify what is to be plotted.

☞ When performing Multidisciplinary Design Optimization, only a single Structural Plotter Command packet is used. The requested plots are generated for **each** appropriate discipline at **each** design iteration.

The available Structural Plotter Commands are summarized in Table 29.1 where they are grouped in functional categories. The remainder of this section will provide you with an overview of these commands.

Table 29-1. STRUCTURAL PLOTTER COMMANDS

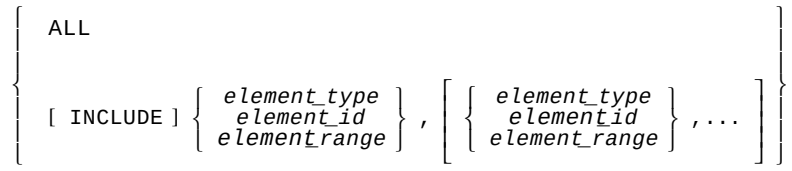
COMMANDS WHICH:	COMMAND NAME
DEFINE PLOT SETS	SET
CONTROL VIEWING OPTIONS	AXES
	PERSPECTIVE FACTOR
	PROJECTION
	VANTAGE POINT
	VIEW
	ZOOM
SOLUTION SCALING	DEFORMATION SCALE
CREATE PLOTS	PLOT
	subplot_list
DEFINE TITLES	PTITLE

29.4.1 Set Definition

You may plot all of your model or selected portions of the model which are defined using the **SET** command. The Structural Plotter **SET** command is similar to the Case Control command, but considerably more flexible. The general form of the command is:

```
SET set_id include_part [modifier_part]
```


The *set_id* is a unique integer identification number which is used to reference the set in subsequent Structural Plotter commands. The *include_part* defines the collection of elements that will be members of the set. Its general form is:

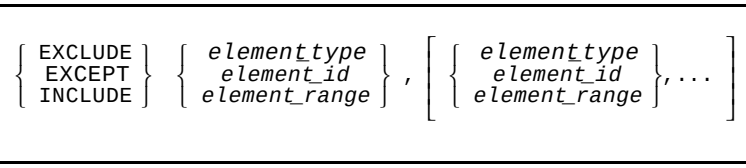


You may include **ALL** elements in the model, or you may select one or more *element_type*s. You will find a list of the allowable types in Chapter 6 of the ***User's Reference Manual***.

You may also select elements by their individual identification numbers, *element_id*, or by specifying an *element_range* of the form:

```
element_id_1 THRU element_id_2
```

The *modifier_part* may then be used to modify the i*nclude_part* by either adding new element types or identifiers, or by excluding selected elements from the previous *include_part*. The syntax of the *modifier_part* is:



Note that the **EXCLUDE** and **EXCEPT** options are synonymous and may be used interchangeably. The following examples illustrate these different forms. Examples of set definitions are given in Chapter 5 of the ***User's Reference Manual***.

You may select all of the elements in your model by defining:

```
SET n = ALL
```

However, as you will see later, this is not necessary because the Structural Plotter default is to plot the entire model.

29.4.2 Viewing Option Commands

The viewing option commands are used to specify the characteristics, or elements, of the plot that you wish to create. The three most important viewing options are described in this section.

29.4.2.1 Viewing Angles

Figure 29.4 shows a typical structural plot with its plot elements annotated. The basic viewing angles, which is shown in the figure, are:

$$(\gamma, \beta, \alpha) = (34.27°, 23.17°, 0.0°)$$

Remember that the order in which the rotations are performed: first γ, the rotation about T, then β, the rotation about S, and finally α, the rotation about R, is crucial. The rotations are specified by the **VIEW** command. The special command **AXES** allows you to perform 90° rotations quickly by simply specifying the correspondence between your models coordinate system and the plotter system. Section 29.6 shows how the three orthogonal views of a model may be obtained in this manner.

The viewing angles may also be changed by moving your position relative to the plotter coordinate system. This is done by using the **VANTAGE POINT** command. It is important that you remember the order in which these three commands are performed. The model coordinate system is first aligned with the plotter coordinate system with the **AXES** command. The **VIEW** command is then used to perform specified rotations of the model. Finally, your location is moved to the specified **VANTAGE POINT**.

29.4.2.2 The Graphics Projection

The Structural Plotter provides you with two graphics projections. You select the type of projection by specifying either:

```
ORTHOGRAPHIC PROJECTION or
PERSPECTIVE PROJECTION
```

When you request **PERSPECTIVE PROJECTION** plots, the amount of perspective distortion is determined by your distance from the model. This distance may be determined in one of three ways. Firstly, if you use the defaults, **UAI/*NASTRAN*** automatically determines this distance. Secondly, you may use the **PERSPECTIVE FACTOR** command to directly specify the amount of perspective distortion that you want. This is used to automatically compute the distance to the model. Finally, you may use the **VANTAGE POINT** command to move your location in space which determines the distance to the model. Note, that if you wish to move your vantage point while maintaining the same perspective distortion, then you should use both commands.

29.4.2.3 ZOOMing

The final viewing option command is **ZOOM**. This command allows you to enlarge or reduce the size of the structure about any point that you select.

29.4.3 Solution Scaling

When you create deformed plots, you may control the scale factor which is applied to the actual displacements by using the **DEFORMATION SCALE** command. You use this command to allow small deformations to become visible and to allow the deformed shape to be distinguished from the undeformed shape.

29.4.4 Defining a Plot Title

The command **PTITLE** allows you to define an additional plot title element, as shown in Figure 29-4.

29.4.5 Controlling the Graphics Device

There are four commands that you use to define the characteristics of the graphics device that will be used to create your plots using the UAIPLOT program or an alternate program developed at your site. Consult your **UAI/*NASTRAN*** Systems Support Specialist to get information on these parameters.

29.5 PLOTTING THE MODEL GEOMETRY

Once you have defined the general viewing options, or allowed **UAI/*NASTRAN*** to select defaults, and defined any **SET**s that you wish to use, you may create one or more plots of your model. This is done with the command:


```
PLOT [global_options] [subplot_list]
```

The command allows you to specify *global_options*, which control the plot *line_style*, select *label*ing and *symbol* options, and to define the subplots within your model that are to be plotted in the same frame. The *subplot_list* allows you to define any number of subsets of your model and define different characteristics for each of them.

You may use one or more of the **SET**s you have defined in your **PLOT** command. The syntax of the *subplot_list* is:

```
subplot_list ⇒ subplot_term,subplot_term,...
```

Each *subplot_term* in the *subplot_list* selects a **SET** and defines plotting options for it. The general form of the *subplot_term* is:

```
subplot_term ⇒ SET setid [set_options]
```

The *set_options* are similar to the *global_options*, but they only apply to the previously appearing **SET**.

Examples of **PLOT** commands using *subplot_lists* are given in Chapter 5 of the ***User's Reference Manual*** and in Section 29-7 of this Chapter.

29.6 PLOTTING SOLUTION RESULTS

You may also plot geometry-based solution results using the Structural Plotter. Such plots are called ***deformed shapes*** of the structural model. In this case, deformations include the grid point displacements and, for dynamic response analyses, the velocities and accelerations. The deformed plots may be made for selected subcases, time ranges, or frequency ranges, depending on the solution discipline that you are using. The general syntax of the **PLOT** command, when used for solution results, is:

```
PLOT results_type [UNDEFORMED][subcase_list][solution_range]
     [plot_type][global_options] [subplot_list]
```

This form of the **PLOT** command is quite different from that used to plot the structural model. First, you must select the *results_type* to be plotted. This may be displacements, velocities or accelerations depending on the solution discipline that you are using. You may choose to plot the **UNDEFORMED** shape as well as the deformed shape. Then, again depending on the analysis discipline, you may plot the solution results for selected subcases or a *solution_range* of times or frequencies. You also select a *plot_type*. The two types are **DEFORMED**, which draws the deformed structural model, and **VECTOR**, which represents the deformation as a vector. Examples of these two plots are shown in Figure 5-7. In both cases, the **UNDEFORMED** shape has also been plotted. The *subplot_list* differs from that used for model plotting in that each **SET** may specify a different *plot_type*.

Examples of the **PLOT** command for solution results are also given in Chapter 5 of the ***User's Reference Manual*** and in Section 29-7 of this Chapter.

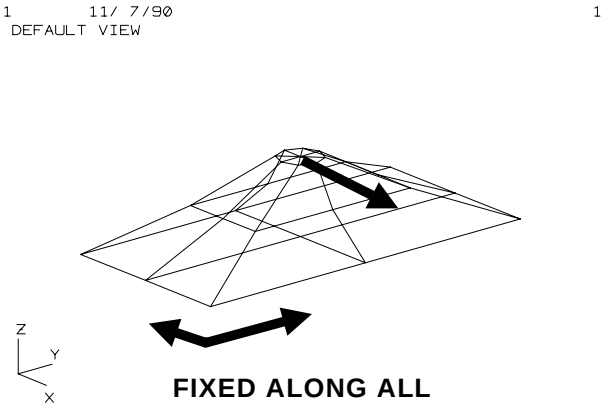
29.7 STRUCTURAL PLOTTER EXAMPLES

This section uses tutorial examples to illustrate the capabilities of the Structural Plotter. Included are examples of both model plots and plots of the solution responses. The examples use a simple model designed to illustrate the plotting concepts.

It is easy to create a model plot using all of the default values:

```
OUTPUT(PLOT)
PTITLE=DEFAULT VIEW
PLOT
```

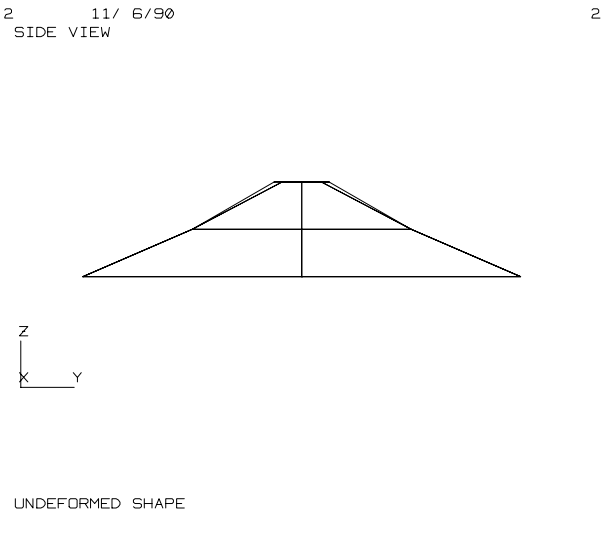
The simple model used for these examples is fixed along the edges of its base and a point load is applied to a single point on the upper ring as shown.



In many cases, model checkout includes plotting the three views of the structure. This can be done in several ways:

```
PTITLE=SIDE VIEW
VIEW = 0.,0.,0.
AXES = X,Y,Z
PLOT
```

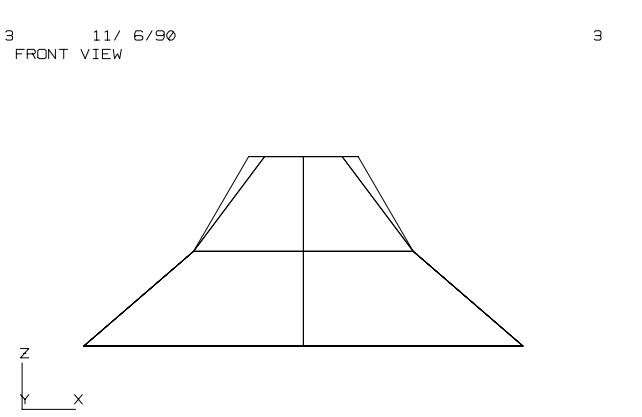
In this case, the **VIEW** command was used to clear the default viewing angles and the **AXES** command used to align the model X-Axis with the R-Axis of the plot. You could also accomplish this by using the **VIEW** command with 90 degree rotations.



A front view is created in a similar manner:

```
PTITLE=FRONT VIEW
AXES = Y,X,Z
PLOT
```

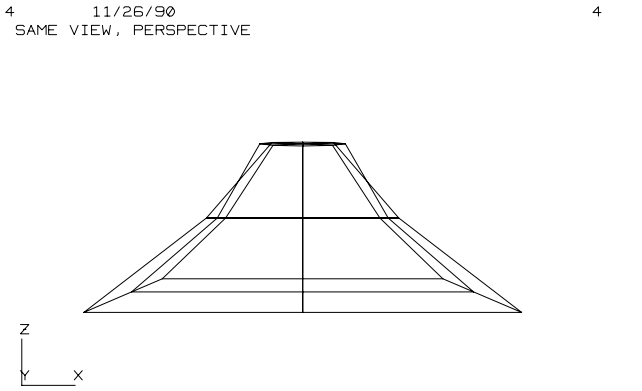
The **VIEW** command is no longer needed because the previous plot had reset the values. This time, the **AXES** command is used to align the model Y-Axis with the R-Axis of the plot.



Unless otherwise selected, all plot are created using an **ORTHOGRAPHIC PROJECTION**. In some cases, you may prefer the realism afforded by the **PERSPECTIVE PROJECTION**:

```
PTITLE=SAME VIEW WITH PERSPECTIVE
PERSPECTIVE PROJECTION
AXES = Y,X,Z
VIEW 0.,0.,0.
PLOT
```

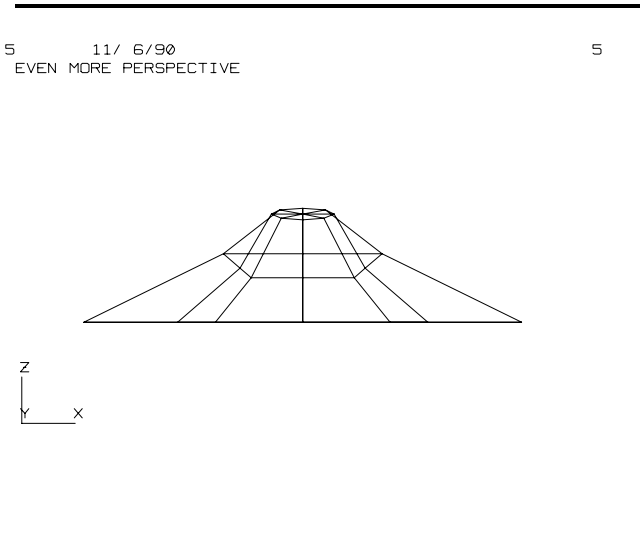
Note that the **AXES** and **VIEW** must be reset if the **PROJECTION** is changed.



You may control the level of perspective distortion. For example:

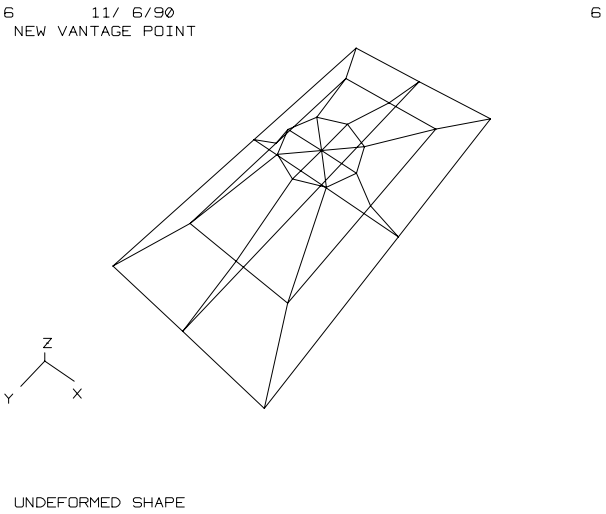
```
PTITLE=EVEN MORE PERSPECTIVE
PERSPECTIVE FACTOR 60.0
PLOT
```

The factor specified is an approximate ratio of the size of the background of the plot to the foreground. In this case, **60.0** indicates that the background is about 60% smaller than the foreground.



You may move to a different viewing point to better understand your model, for example:

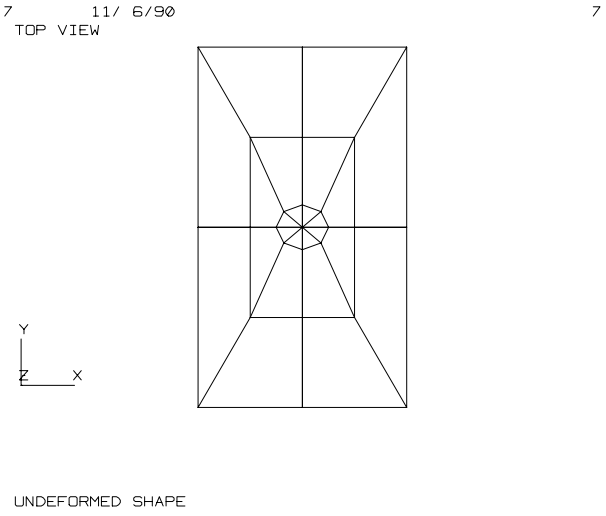
```
PTITLE=NEW VANTAGE POINT
VANTAGE POINT
PLOT
```



Finally, the third view is plotted:

```
PTITLE=TOP VIEW
ORTHOGRAPHIC PROJECTION
VIEW 0.,0.,0.
AXES = Z,X,Y
PLOT
```

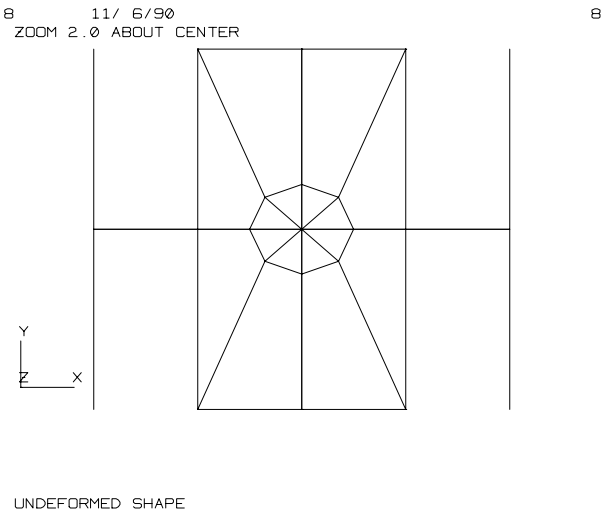
The **PROJECTION** is reset to **ORTHOGRAPHIC** and the **VIEW** command is used to reset the default viewing angles. Then, the **AXES** command is used to align the model Z-Axis with the R-Axis of the plot.



In many cases, you may wish to enlarge your plot about a particular point. For example,

```
PTITLE=ZOOM 2.0 ABOUT CENTER
ZOOM 2.0
PLOT
```

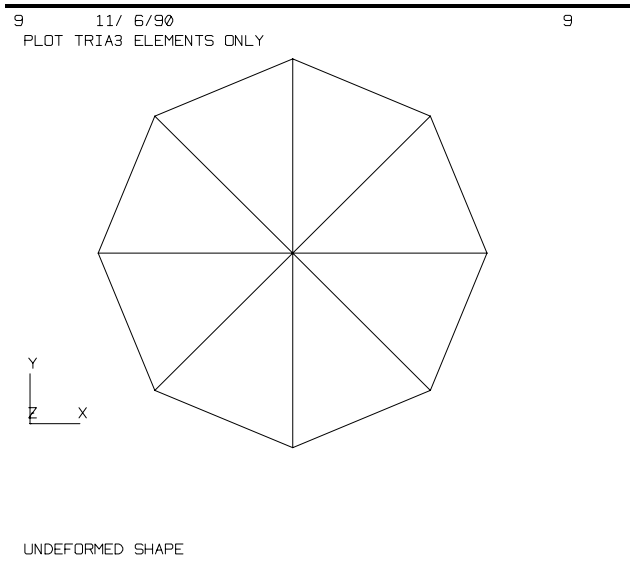
results in the plot shown. Note that the structure was automatically clipped at the boundaries. You may specify any center of **ZOOM** — you are not limited to the center of the plot. A factor of less than one will result in a reduction in the size of the model.



Now, you wish to create a plot that is just the TRIA3 elements on the top surface. This is easily accomplished with the commands:

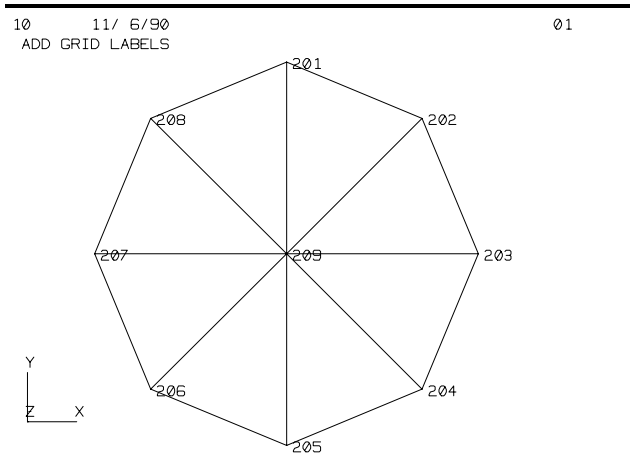
```
SET 2 = TRIA3
PTITLE=PLOT TRIA3 ELEMENTS ONLY
PLOT SET 2
```

Note that all other viewing options have remained in place, and that the selected **SET** is rescaled to fill the viewing area.



Model checking usually involves identifying GRID points and elements in the model. To add labels for the GRID points, you use:

```
PTITLE=ADD GRID LABELS
PLOT SET 2 LABEL GRIDS
```

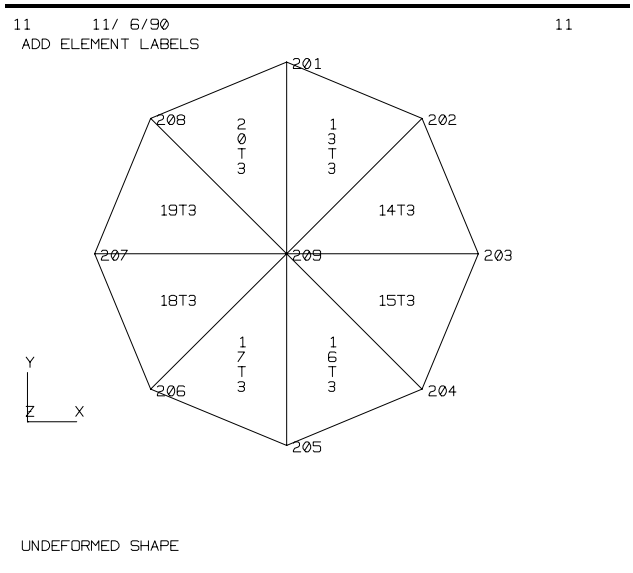


For simple plots like this one, adding the element identification numbers is also useful:

```
PTITLE=ADD ELEMENT LABELS
PLOT SET 2 LABEL BOTH
```

Note that the element type is appended to the identification number as:

```
16T3
```

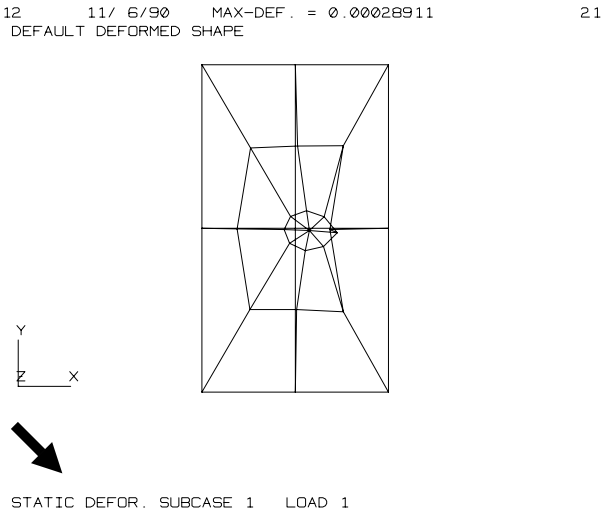


29

Plotting the deformed shape of the structure is equally simple. For example:

```
PTITLE=DEFAULT DEFORMED SHAPE
PLOT STATIC DISP DEFORMED
```

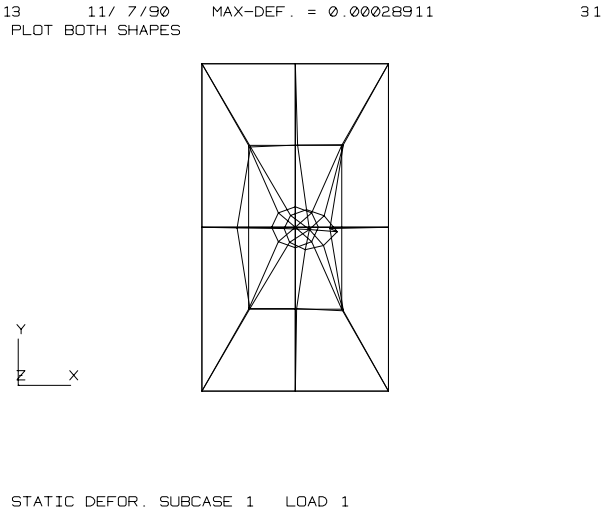
Note that the subcase is automatically noted.



You may also plot the original, or undeformed, shape by using:

```
PTITLE=PLOT BOTH SHAPES
PLOT STATIC DISP UNDEFORMED
     DEFORMED
```

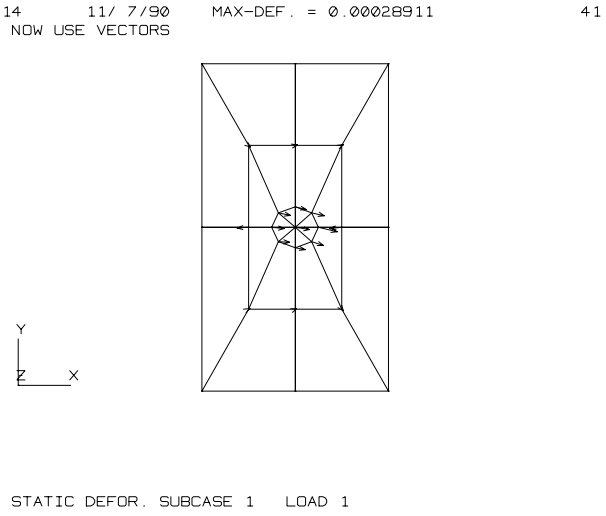
For other than small models, the number of lines in such plots are often excessive. The **VECTOR** option can help solve this problem.



You may plot vectors which represent the deformations of your model. You may select individual components of displacement or, as in this example, the resultant:

```
PTITLE=NOW USE VECTORS
PLOT STATIC DISP VECTOR R
```

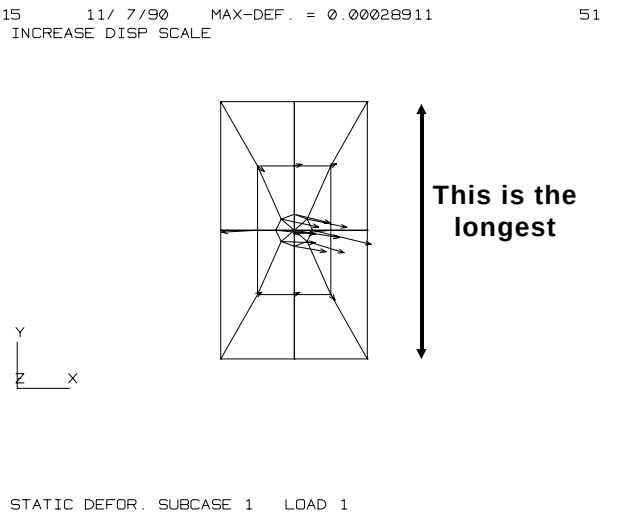
The length of the vectors drawn depends on the scale that you select in the same manner as the deformed shape.



The deformation scale is changed by a single command:

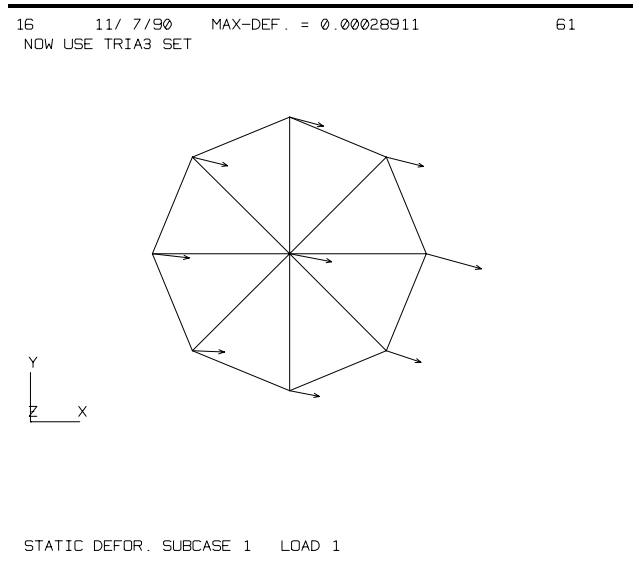
```
PTITLE=INCREASE DISP SCALE
DEFORMATION SCALE 20.0
PLOT STATIC DISP VECTOR R
```

The factor represents a percentage of the maximum dimension in the ***plotted model***. For example, 20.0 indicates that the largest deformation will be plotted such that it has a motion equal to 20% of the longest plot dimension, as noted.



Naturally, deformed plots may also be created for sets:

```
PTITLE=NOW USE TRIA3 SET
PLOT STATIC DISP SET 2 VECTOR R
```



The examples shown in this section illustrate the use of all of the Structural Plotter commands. Many more capabilities are available and you should develop a model which allows you to exercise them.

29

This page is intentionally blank.

Chapter 30

X-Y PLOTTING

When you use the dynamic response analysis disciplines within **UAI/*NASTRAN***, most often you are faced with reviewing and analyzing large volumes of solution results. To aid you in this task, **UAI/*NASTRAN*** provides an X-Y plotting capability for representing this data. The actual X-Y Plots, like the Structural Plots, are not created during your **UAI/*NASTRAN*** execution. Rather, the plot information is placed on a file that is post-processed.

UAI provides four utilities for this purpose. The first two programs, available on most computers, are ***uaiplotps*** and ***uaiplotgl***. These utilities allows you to send your plots to a PostScript laser printer and a Hewlett Packard graphics device, respectively. The third program, available on Unix host computers is ***uaiplot***. This is an interactive program, based on Motif and X-Windows, which allows you to selectively view your plots. Chapter 31 of this manual and Chapter 1 of the ***User's Reference Manual*** provides instructions on how to use these programs for most systems. There is also a program, provided as source code, illustrating how your facility may create their own plotter software, if necessary. In some cases, you must contact your **UAI/*NASTRAN*** Support Specialist for details describing how your site interfaces with these post-processor.

This Chapter describes the X-Y Plotter terminology, the input data requirements for creating such plots, and provides examples which illustrate the use of many of the plotter commands.

30.1 X-Y PLOTTING TERMINOLOGY

Most frequently, an X-Y Plot consists of one or more functions of the form $y_i = F_i(x)$. The independent variable, x, may be: time, if you are solving transient response problems; frequency, when you are solving frequency response problems; or subcase identification number when you are solving either linear or nonlinear statics problems. The dependent variables, y_i, may represent a response quantity that varies with the independent variable. Such responses include displacement, accelerations, and the many other input and solution quantities which are available. This type of X-Y Plot is called a ***history*** plot.

Correlation Plot. This type of plot allows you to plot one response quantity against another. For these plots, the set of X-Y pairs is extracted from the data, sorted by increasing value of the X coordinate, and then plotted.

Frame. Analogously to photography, each displayable plot is called a frame.

Whole Frame Plot. Each frame may contain one or two plots. When the frame contains a single plot it is called a whole frame plot.

Half Frame Plots. Alternately, a frame may contain two plots, a ***top half frame plot*** and a ***bottom half frame plot***.

You select between these two options based on the plot resolution that you require and the use that you intend for the graphic results. Obviously, the two half frame plots will not allow the accuracy or readability that a whole frame plot will. On the other hand, half frame plots are perfectly fine for a fast, qualitative analysis of the data.

Curves. Each of the plots you request may contain one or more curves. The curves may be drawn as a set of points which may optionally be connected by straight line segments. Additionally, graphic symbols may be placed at the locations of the actual data points. Symbols change from curve to curve so that you may differentiate them.

There are many optional commands that you may use to modify the format and labeling of your plots. These commands operate on ***plot elements***. **Most of these commands function as toggles**: **you turn them on or off and they remain in that state until you toggle them again.**

Command Verbs. The X-Y Plotter includes a number of different functions which are selected by specifying a command verb. Available commands not only result in graphic plots, but allow you to print the coordinates of the plotter points, obtain a summary of the minimum and maximum values for each curve, write the plotter coordinates on a file for use by other programs, and allow you to create plots on a standard line printer in the event that you do not have access to a graphics device.

All of these options and capabilities are discussed in the remainder of this Chapter.

30.2 X-Y PLOTTER DATA REQUIREMENTS

This section provides you with a description of the **UAI/*NASTRAN*** input data required for creating X-Y Plots.

30.2.1 Executive Control Commands

Although there are no specific Executive Control commands which are required to perform X-Y Plotting, you may **ASSIGN** a logical file on which the plot data will be written for use by the ***uaiplot*** plotting program. This is done with the command:

```
ASSIGN logical_name=physical_name,NEW,USE=PLOT
```


The assignment of the plot file is optional when you use the **XYPLOT** command. Note that there are two formats available for the plot file, **BINARY** and **FORMATTED**. The one which you use depends on the implementation of the ***uaiplot*** utility at your site. Contact your Support Specialist for detailed information.

Similarly, if you are using the **XYPUNCH** command, you may **ASSIGN** a file with **USE=PUNCH**. In both cases, if you do not assign the appropriate file, the default value defined in the **UAI/*NASTRAN*** section of the Preference File will be used. If you are using only the **XYPAPER**, **XYPRINT** or **XYPEAK** commands, it is not necessary to **ASSIGN** any files.

☞ Both X-Y Plots and Structural Plots (described in Chapter 29) are written to the same plot file.

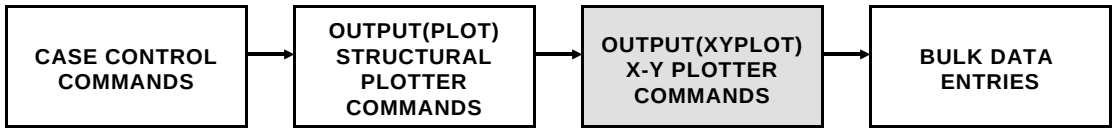
30.2.2 Substructure Control Commands

The substructuring capability has no specific control commands to support X-Y Plotting. You may create X-Y Plots during Phase 3 data recovery operations in the same manner as you create them for non-substructuring analyses.

30.2.3 Case Control Commands

As was the case for the Structural Plotter, the X-Y Plotter commands form a subpacket within the Case Control command packet as shown in Figure 30-1.

Figure 30-1. LOCATION OF THE X-Y PLOTTER SUBPACKET



Both the Structural Plotter subpacket, described in Chapter 29, and the X-Y Plotter subpacket may be interchanged in the data stream. It is not necessary for you to issue Case Control output request commands when you are requesting X-Y Plots. **UAI/*NASTRAN*** automatically extracts the necessary data from the solution results.

30.2.4 Bulk Data

There are no Bulk Data entries which control X-Y Plots in any manner. Naturally, many entries are used to control the domain of the solution results which are computed and available to create the plots.

30.2.5 Using eSHELL

In addition to the X-Y Plotter described in this Chapter, you may also create a wide variety of similar plots using the ***eBase*** Interactive Shell, ***eShell***. In order to use ***eShell***, you must create an ARCHIVE database during your job. This database may then be used for subsequent plotting operations. For details, please see the ***eShell User's Manual*** and the ***eBase ARCHIVE Schemata Description***.

30.3 THE X-Y PLOT ELEMENTS

The following sections describe the X-Y ***Plot elements***. Each of these elements has a name which corresponds to the X-Y Plotter command used to control it. These commands are different depending on whether you are creating whole frame or half frame plots. However, the similarity of these commands is such that the term command family is used to describe them. For example, a command called **COMM** when applied to whole frame plots is called **XCOMM**, if it acts only on an X-Axis plot element, and it is called **YCOMM** if it acts on the corresponding Y-Axis plot element. Similarly, if the command may also be applied to upper and lower half frame plots, the command names would be **XTCOMM**, **YTCOMM** and **XBCOMM**, **YBCOMM**, respectively. This will be made clearer in the descriptions in the following sections.

30.3.1 Whole Frame Plots

Figure 30-2 shows a typical whole frame plot. It illustrates the most often encountered plot elements. You will notice that both the X-Axis and the Y-Axis have text labels. These elements are called the **XTITLE** and **YTITLE**, respectively. Unless you request otherwise, solid lines are drawn representing the axes themselves. They are naturally called the **XAXIS** and **YAXIS**. The two axes intersect at the **XINTERCEPT** and **YINTERCEPT**. In the figure, the default intercept coordinates of **0.0** and **0.0** were used. You will notice that there are small cross-hairs, called ***tics***, at various positions along both axes. These are called **LEFT TICS** and **LOWER**, or **BOTTOM**, **TICS**. The tics may be toggled **ON** or **OFF** as you wish. The number of tics which appear on the axes are called **XDIVISIONS** and **YDIVISIONS**. Because **UAI/*NASTRAN*** creates axis labels which are rounded to avoid the use of irregular values, the number of divisions that are actually plotted may vary slightly from the number specified. For example, the default value of five divisions has been used for the plot in Figure 30-2, but only four divisions appear so that the axis labels are round.

The plot contains two curves, each of which is connected by a **LINE**. Optional **SYMBOLS** have been placed at actual plot point locations. Finally the frame itself has been given a title, called an **FTITLE**, which is included along with the standard Case Control titles.

30.3.2 Half Frame Plots

In addition to the whole frame plot elements discussed above, there is another group of elements which are found in half frame plots. Figure 30-3 shows such a plot. Again you will notice that the X-Axis and the Y-Axis have textual labels for both the upper and lower plots. While the label on the X-Axis is still the **XTITLE**, the Y-Axis label for the top plot, **YTTITLE** and bottom plot, **YBTITLE**, can be independently defined. As with whole frame plots, solid lines are drawn representing the axes themselves unless you request otherwise. The Y-Axis is still the **YAXIS**, but the X-Axes are now the **XTAXIS** and **XBAXIS**. Also separately controllable are: the Y-Axis intersection, **YTINTERCEPT** and **YBINTERCEPT**; the left tics, **TLEFT TICS** and **BLEFT TICS**; and many other plot elements. These are discussed in detail in the next section.

Figure 30-2. PLOT ELEMENTS FOR WHOLE FRAMES

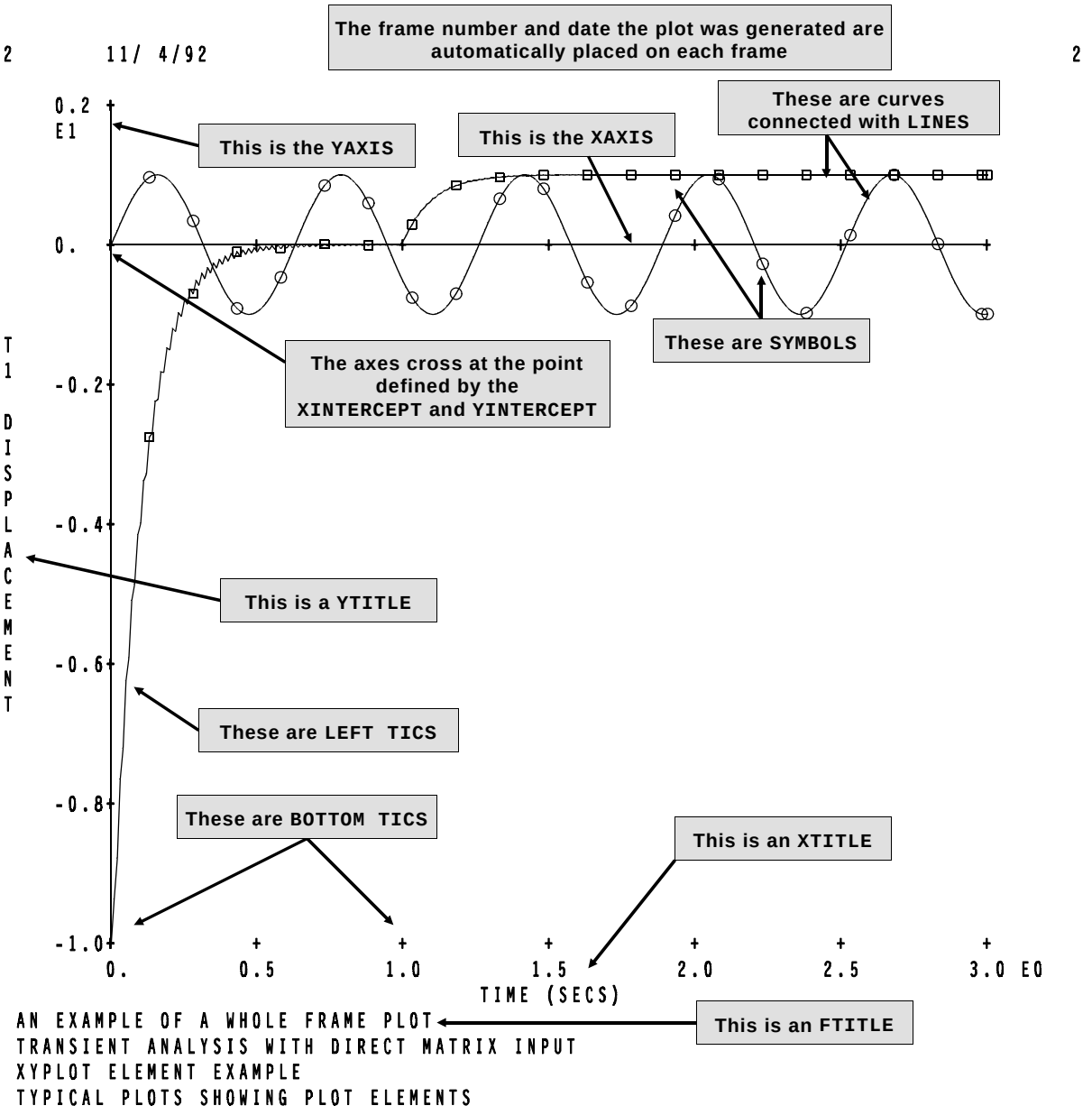
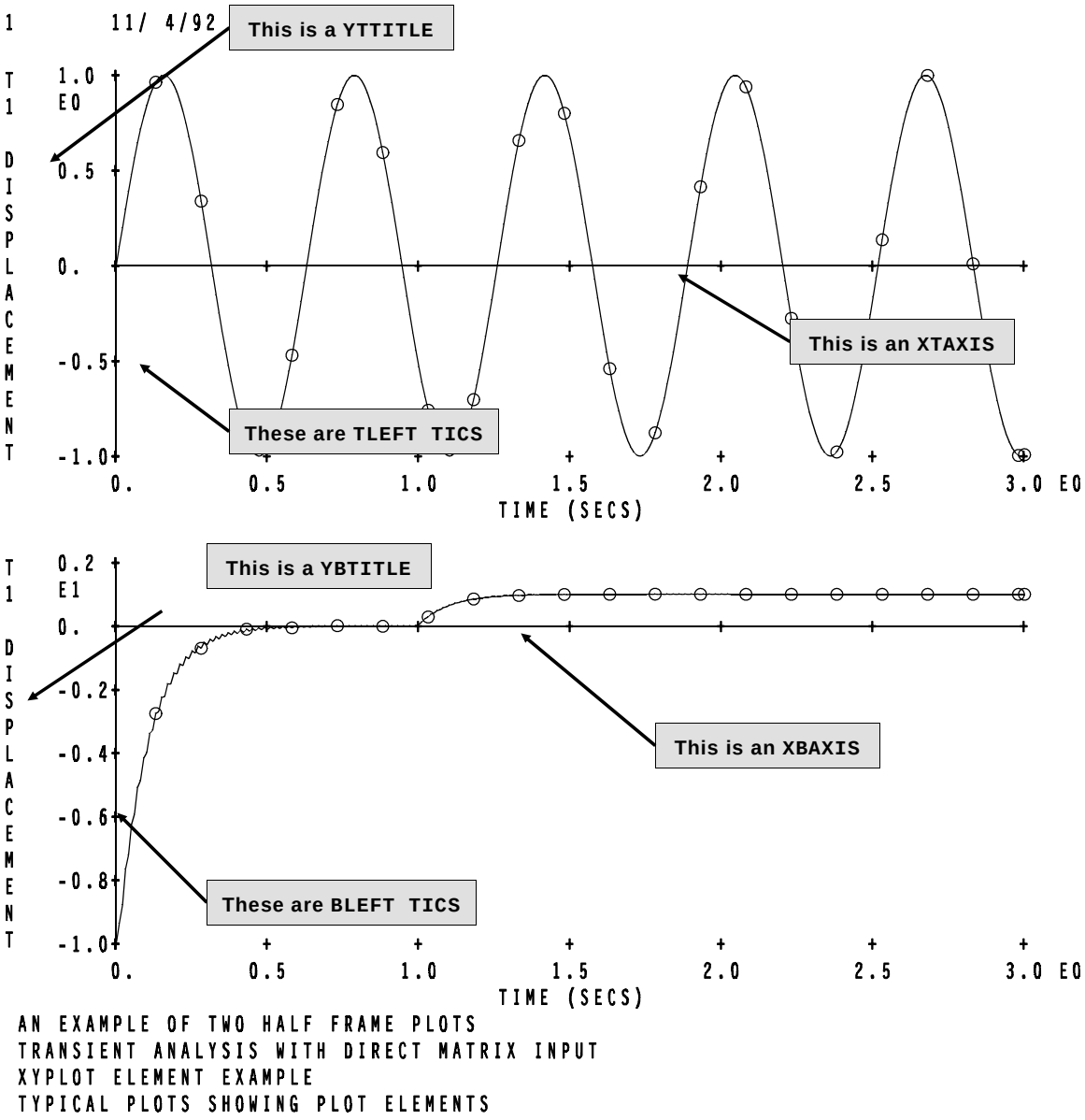


Figure 30-3. PLOT ELEMENTS FOR HALF FRAMES



30.4 X-Y PLOTTER COMMANDS

The general form of the X-Y Plotter Command packet is:

```
OUTPUT(XYPLOT)
      option_command
      option_command
      ...
      PLOT_command_verb
      option_command
      option_command
      PLOT_command_verb
      ...
OUTPUT(PLOT) or BEGIN BULK
```


The packet must begin with the **OUTPUT(XYPLOT)** command and it ends when either an **OUTPUT(PLOT)** or **BEGIN BULK** command is encountered. The other commands in the packet define the plot frames that you wish to create. Some of these are *option_command*s which define the characteristics of the plots to be made while others are the *PLOT_command_verb*s which actually specify what is to be plotted or printed.

☞ When performing Multidisciplinary Design Optimization, only a single X-Y Plotter Command packet is used. The requested plots are generated for **each** appropriate discipline at **each** design iteration.

The available X-Y Plotter Commands are summarized in Table 30.1 where they are grouped in functional categories. The remainder of this section will provide you with an overview of these commands.

Table 30-1. X-Y PLOTTER COMMANDS

COMMANDS WHICH CONTROL:	WHEN SELECTING PLOTS WHICH ARE:		
	WHOLE FRAME	UPPER HALF FRAME	LOWER HALF FRAME
DATA RANGE SELECTION	XMIN XMAX YMIN YMAX	YTMIN YTMAX	YBMIN YBMAX
AXIS SELECTION AND CONTROL	XAXIS YAXIS	XTAXIS	XBAXIS
	XDIVISIONS YDIVISIONS	YTDIVISIONS	YBDIVISIONS
	XINTERCEPT YINTERCEPT	YTINTERCEPT	YBINTERCEPT
GRID LINES	XGRID YGRID	XTGRID YTGRID	XBGRID YBGRID
TITLING	FTITLE		
	XTITLE YTITLE	YTTITLE	YBTITLE
SCALE MARKERS AND VALUES	UPPER SCALES LOWER SCALES LEFT SCALES RIGHT SCALES XVALUE SCALES YVALUE SCALES	TLEFT SCALES TRIGHT SCALES YTVALUE SCALES	BLEFT SCALES BRIGHT SCALES YBVALUE SCALES
CURVE APPEARENCE	DRAWLINE		
	SYMBOL		
	LINESTYLE		
LOGRITHMIC SCALES	XLOG YLOG	YTLOG	YBLOG
UTILITY	CLEAR		
GRAPHICS DEVICE	ASPECT RATIO		
	CHARACTER PRECISION		
	CHARACTER SCALE		

30.4.1 Data Range Selection

The command families **MIN** and **MAX** allow you to specify the minimum and maximum values of the data that you wish to plot. This allows you to isolate active response areas in the plot, thus making the data more readable. Note that you may only control the minimum and maximum value of the ordinate when using half frame plots, because both plots must share the same X-Axis data range.

30.4.2 Axis Selection and Control

You may select the drawing of the X- and Y-Axes on your plots by using the **AXIS** command family. The approximate number of divisions along each axis may be selected with one of the **DIVISIONS** family of commands. You may also control the point at which the axes intersect with the **INTERCEPT** commands.

30.4.3 Grid Lines

You may simulate graph paper by using the **GRID** command family to draw lines along the divisions of the coordinate axes. This option is best used if you plan to read values from the plot. Generally, plots are more legible with fewer lines.

30.4.4 Titling

You may add an additional title to your plot frame by using the **FTITLE** command. The other **TITLE** commands allow you to label the coordinate axes. Note that both X-Axes on half frame plots must have the same title.

30.4.5 Scale Markers and Values

You may also request that tic marks and scale values be drawn on the upper, lower, left and right borders of each plot frame by using the **SCALES** command family. These commands also allow you to control which tic marks will have scale values.

30.4.6 Curve Appearance

You may draw curves which place a symbol at each data point by using the **SYMBOL** command. The **DRAWLINE** command is used to connect the data points with straight line segments. Naturally, you may select both of these options. Depending on the capabilities of your plot post-processor, you may also use the **LINESTYLE** command to control the appearance of the straight line segments.

30.4.7 Logarithmic Scales

The default X-Y Plotter mode is to plot data using cartesian coordinates. You may use the **LOG** command family to use semi-log or log-log scales. If you select logarithmic scales, then the X-Y Plotter automatically determines the number of cycles to plot and control the labeling of the axes. Chapter 6 of the ***User's Reference Manual*** describes how this is done.

30.4.8 Utility Operation

The **CLEAR** command may be used selectively to enable or disable various plot options.

30.4.9 Controlling the Graphics Device

There are three commands that you use to define the characteristics of the graphics device that will be used to create your plots using the ***uaiplot*** program or an alternate program developed at your site. Consult your **UAI/*NASTRAN*** Systems Support Specialist to get information on these parameters.

30.5 THE XYPLOT COMMAND

You use the **XYPLOT** command to create histograms of input data and solution results as a function of time or frequency depending on the type of analysis you are performing. You may also use it to process random response results by creating power spectral density and autocorrelation plots. Finally, you may create correlation plots of one response quantity against another using time or frequency as a parameter. The **XYPLOT** command results in the creation of a plot file that is then post-processed to display the plot on your graphics device.

The general form of the X-Y Plot command family is:

```
{ XYPLOT  }
{ XYPRINT }
{ XYPUNCH } { HISTORY     } y_data VS x_data ptype subcase_list curve_list
{ XYPEAK  } { CORRELATION }
{ XYPAPER }
```


Any combination of the plot command verbs may be used in a single plot command. There are many options within each clause of these commands. All of these clauses are the same for each command, only the form of output differs, as you will see in subsequent sections. Note that when you create history plots, the **HISTORY** and **VS** keywords and the *x_data* selection are not required.

30.5.1 Selecting the *y_data* and *x_data*

The first data included on the command is a definition of the *y_data* and *x_data* that will form the plot. Various parameters and response quantities may be selected for plotting for each axis. These are described in Chapter 6 of the ***User's Reference Manual***.

30.5.2 The *plot_type*

There are three *plot_type*s from which you may choose:

```
{ RESPONSE      }
{ AUTOCORRELATION }
{ PSDF          }
```

The default *plot_type* is **RESPONSE**. This indicates that you are plotting one of the solution response quantities or input quantities. The other two options, **AUTOCORRELATION** and the power spectral density function, **PSDF**, may only be used if you are performing a random analysis. Details about these are found in Chapter 13 of this manual.

30.5.3 The *subcase_list*

The *subcase_list* is simply a list of the subcases for which you wish plots to be generated. The general form of this list is:

```
subid_1 [,subid_2,...]
```

where each *subid* is simply the identification number of a **SUBCASE** that you have defined in your Case Control command packet.

30.5.4 The *curve_list*

The *curve_list* defines the number of frames, whether plots will be whole frame or half frame format, and the output quantities that will be plotted. Each frame specification entry begin with a slash ('**/**') as shown below:

```
/ frame_1 [/ frame_2 ... ]
```

Note that any number of frames may be defined. Each of the frame descriptors is composed of a list of curve descriptors:

```
curve_1 [,curve_2, ... ]
```

Each of these descriptors requests a single curve and specifies whether the curve will be place on a whole frame or on a top or bottom half frame. A whole frame curve is defined by:

```
grid_id ( comp ) ,... or
elem_id ( comp ) ,...
```

First, you must specify a *grid_id* or *elem_id* whose response component is to be plotted. Then, enclosed in parentheses, you select the code for that response component. Again, all of the available codes are given in Chapter 6 of the ***User's Reference Manual***. To define half frame curves, the form is:

```
grid_id ( [comp_top] [ , comp_bot ] ) ,... or
elem_id ( [comp_top] [ , comp_bot  ]) ,...
```

The first component named, *comp_top*, will be plotted in the upper half frame. Similarly, the second quantity, *comp_bot*, will be plotted in the lower half frame. Either of these quantities may be omitted in which case there will be no curve on the corresponding half frame plot. The comma must always appear in the command to differentiate the two curves. Note that you may change response components, *grid_id*s, and *elem_id*s from curve to curve.

30.5.5 XYPLOT Output

In addition to creating the plot file used by your graphics post-processor, the **XYPLOT**, and in fact, all of the X-Y Plotter command verbs, generates a summary for each curve which includes the quantities plotted, the titling information and the ranges of the data values. An example is shown in Table 30-2.

Table 30-2. X-Y PLOTTER CURVE SUMMARY

```
                    X Y - O U T P U T   S U M M A R Y

SUBCASE         1
RESPONSE
HISTORY CURVE, DISPLACEMENT RESP 4(T3) VS TIME


XY-PAIRS WITHIN FRAME LIMITS WILL BE PLOTTED.
PLOTTER SPECIFIED IS CRT

THIS IS CURVE   1 OF WHOLE FRAME    1

CURVE  TITLE = WHOLE FRAME, SINGLE CURVE, ALL DEFAULTS

X-AXIS TITLE =

Y-AXIS TITLE =



THE FOLLOWING INFORMATION IS FOR THE ABOVE DEFINED CURVE ONLY.

WITHIN THE FRAME X-LIMITS       ( X =   0.000000E+00 TO X =   8.000001E-01 )

                        THE SMALLEST Y-VALUE =  -5.000000E-02 AT X =   0.000000E+00

                        THE LARGEST  Y-VALUE =   2.839540E-02 AT X =   1.600000E-02


WITHIN THE X-LIMITS OF ALL DATA ( X =   0.000000E+00 TO X =   8.000001E-01 )

                        THE SMALLEST Y-VALUE =  -5.000000E-02 AT X =   0.000000E+00

                        THE LARGEST  Y-VALUE =   2.839540E-02 AT X =   1.600000E-02


                                  E N D   O F   S U M M A R Y
```

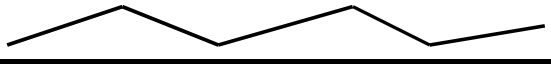
30.6 THE XYPRINT COMMAND

You use the **XYPRINT** command to create tables of the actual data values that are used, or would be used, to create a corresponding **XYPLOT**. This feature must be used when performing random response analysis because it is the only method available for printing solution data. Table 30-3 illustrates a sample output from the **XYPRINT** command.

Table 30-3. EXAMPLE OF XYPRINT OUTPUT

HISTORY CURVE, DISPLACEMENT RESP 2(T3) VS TIME - WHOLE FRAME

STEP	X-DATA	Y-DATA
1	0.000000E+00	0.000000E+00
2	2.000000E-03	-3.798743E-04
3	4.000000E-03	-4.968450E-03
4	6.000000E-03	-8.386047E-03
5	8.000000E-03	-4.311383E-03
6	1.000000E-02	4.382742E-03
7	1.200000E-02	6.099680E-03
8	1.400000E-02	6.316201E-04
9	1.600000E-02	-2.237488E-04
10	1.800000E-02	5.604031E-03
11	2.000000E-02	8.317089E-03
12	2.200000E-02	4.415992E-03
13	2.400000E-02	2.026191E-04
14	2.600000E-02	-2.814618E-04
15	2.800000E-02	5.873423E-04
16	3.000000E-02	-3.714329E-04
17	3.200000E-02	-3.283890E-03
18	3.400000E-02	-5.641322E-03
19	3.600000E-02	-5.226959E-03
20	3.800000E-02	-3.133581E-03



30.7 XYPUNCH — INTERFACING WITH POST-PROCESSORS

You may use the **XYPUNCH** command to write the X-Values and Y-Values of each curve to a file you have assigned with a **USE=PUNCH**. The data for each curve is preceded by a series of descriptor records each of which begin with a dollar sign, $. The actual curve data then follows, one record for each pair of X-Y values. Each record contains a sequence number in positions 72 through 80. This feature is useful for importing these data into another program. The Fortran format of these records is:

```
FORMAT(2E20.6,32X,I8)
```

Table 30-4 shows a sample of the contents of the result file.

Table 30-4. EXAMPLE OF XYPUNCH OUTPUT

```
$TITLE   = SIMPLE TRANSIENT RESPONSE ANALYSIS                              1
$SUBTITLE= USER'S GUIDE SAMPLE PROBLEM                                     2
$LABEL   = USING VARIOUS PLOT ELEMENTS AND OPTIONS                         3
$SUBCASE ID =          1                                                   4
$XY HISTORY CURVE                                                          5
$XY DISPLACEMENT RESP 2(T3) VS TIME                                        6
        0.000000E+00        0.000000E+00                                   7
        2.000000E-03       -3.798743E-04                                   8
        4.000000E-03       -4.968450E-03                                   9
        6.000000E-03       -8.386047E-03                                  10
        8.000000E-03       -4.311383E-03                                  11
        1.000000E-02        4.382742E-03                                  12
        1.200000E-02        6.099680E-03                                  13
        1.400000E-02        6.316201E-04                                  14
        1.600000E-02       -2.237488E-04                                  15
        1.800000E-02        5.604031E-03                                  16
        2.000000E-02        8.317089E-03                                  17
        2.200000E-02        4.415992E-03                                  18
        2.400000E-02        2.026191E-04                                  19
        2.600000E-02       -2.814618E-04                                  20
        2.800000E-02        5.873423E-04                                  21
        3.000000E-02       -3.714329E-04                                  22
```

30.8 XYPEAK — DATA RANGE SCANNING

The **XYPEAK** command allows you scan the data range of the requested solution results and to print a summary of the maximum and minimum values of each curve that you draw. Table 30-5 shows a sample of the output resulting from this command. You obtain a single table that contains all of the data for all of the **XYPEAK** commands that appear in your X-Y plotter command subpacket.

Table 30-5. EXAMPLE OF XYPEAK OUTPUT

X Y - O U T P U T S U M M A R Y

MAXIMUM AND MINIMUM VALUES OF THE DATA

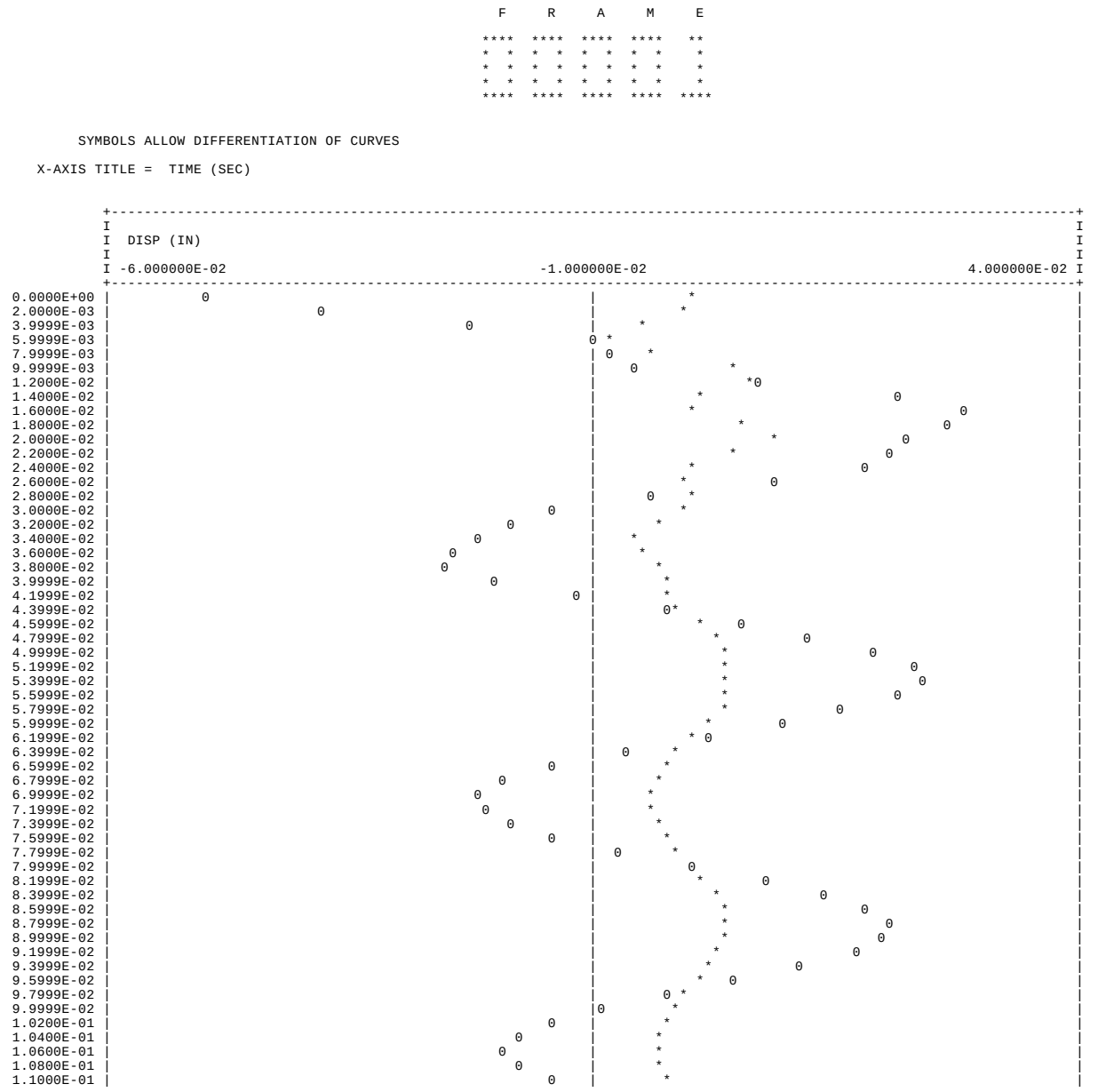
X LIMITS FOR THIS TABLE MAX = 2.000000E-01 MIN = 0.000000E+00

SUBCASE	POINT OR ELEMENT ID	COMPONENT NUMBER	VECTOR TYPE	PLOT TYPE	MAXIMUM Y	CORRESPONDING X	MINIMUM Y	CORRESPONDING X
1	2	T3	DISP	RESP	8.317089E-03	2.000000E-02	-8.386047E-03	6.000000E-03
1	4	T3	DISP	RESP	2.839540E-02	1.600000E-02	-5.000000E-02	0.000000E+00

30.9 THE XYPAPER COMMAND

In the event that your computer site does not have graphics hardware suitable for displaying XY-Plots, you may use the **XYPAPER** command. This command generates the plots on your line printer. The Y-Axis is plotted in 132 columns across the width of the paper and the X-Axis is plotted from page-to-page in an open-ended fashion. Only symbols are placed on these **XYPAPER** plots. The first curve on the frame uses the symbol *. Successive curves are denoted by the symbols **O**, **A**, **B**, **C**, **D**, **E** , **F**, **G**, and **H**. Although the resolution of these plots is low, they still may provide important qualitative information as evidenced by the example shown in Figure 30-4.

Figure 30-4. EXAMPLE OF AN XYPAPER PLOT



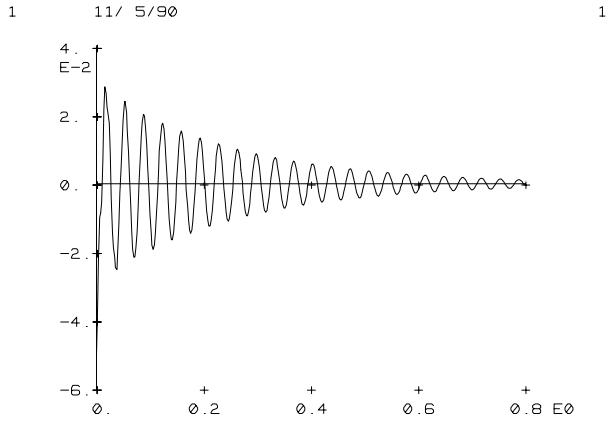
30.10 X-Y PLOTTER EXAMPLES

This section uses tutorial examples to illustrate the capabilities of the X-Y Plotter. The examples are drawn from two models. The first is a transient response analysis of a simple cantilever beam model which is excited by an initial displacement. The second is a material nonlinear analysis used to illustrate the correlation plot.

30

You may create an X-Y Plot very simply if you are select all of the standard defualt values. The plot on the right was created with the commands:

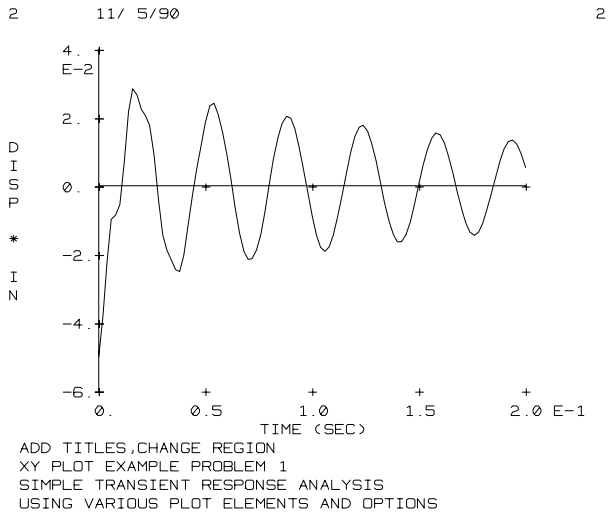
```
OUTPUT(XYPLOT)
FTITLE=WHOLE FRAME ALL DEFAULTS
XYPLOT DISP RESPONSE / 4(T3)
```



Although the displacement results are well represented, usually you will add titling to your plot. Often, you will also want to zoom in on a smaller region of interest. The commands:

```
XTITLE=TIME (SEC)
YTITLE=DISP * IN
FTITLE=ADD TITLES,CHANGE REGION
XMAX = 0.2
XYPLOT DISP RESPONSE / 4(T3)
```

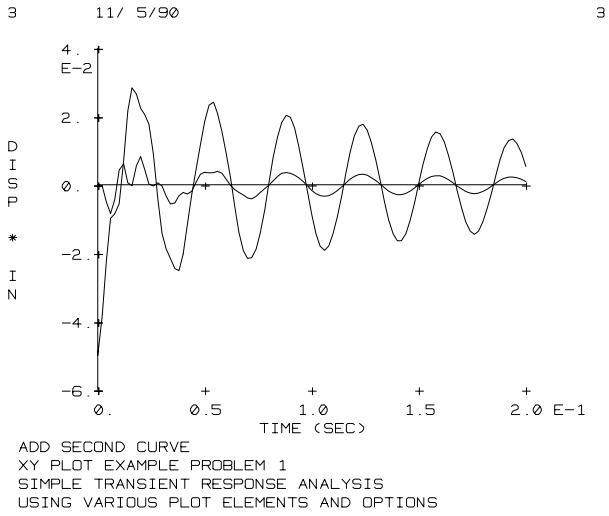
The resulting plot is much more understandable.



You may often wish to compare several responses by plotting them on the same graph. This is done simply by specifying them in a list. For example, to add the **T3** displacement for **GRID 2**, you could use the commands:

```
FTITLE=ADD SECOND CURVE
XYPLOT DISP RESPONSE / 4(T3),2(T3)
```

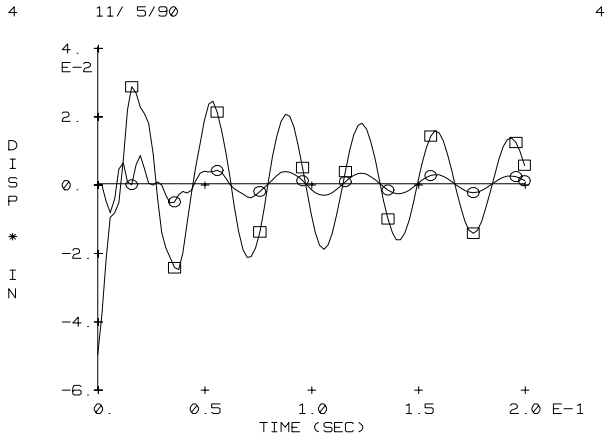
Notice that all of the titling and the plot region are still in effect for this plot.



More than one plot is difficult to interpret. So, you may place different symbols on each curve:

```
FTITLE=ADD SYMBOLS
SYMBOL 6,10
XYPLOT DISP RESPONSE / 4(T3),2(T3)
```

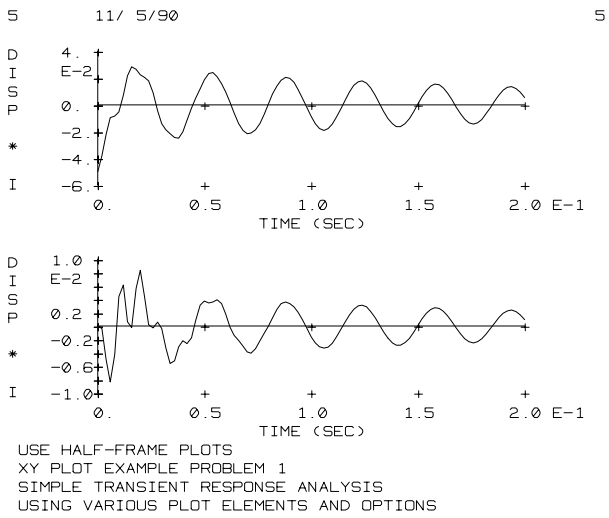
The **SYMBOL** command selects the type of the first symbol. The others are then selected using the rules described in the ***User's Reference Manual***. The value **10** indicates that a symbol will be placed every ten data points.



Your plots may be more readable if you plot responses as separate upper and lower half frames. The example plot was created with the commands:

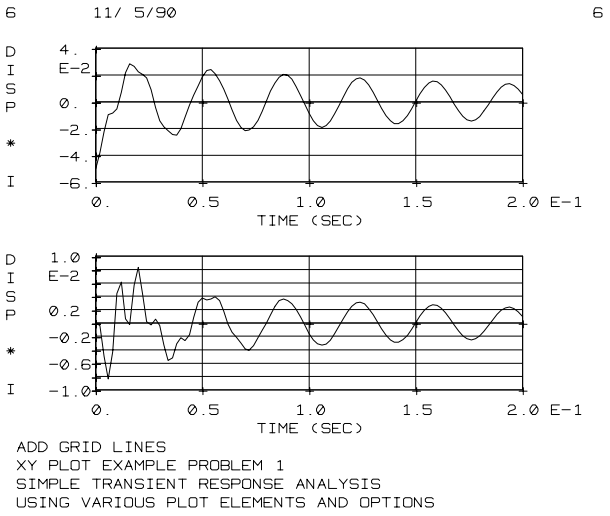
```
FTITLE=USE HALF-FRAME PLOTS
XYPLOT DISP RESPONSE
       / 4(T3,),2(,T3)
```

Now, each response is plotted individually. The smaller plots have less accuracy that the whole frame plots.



You may also add grid lines to your plots to simulate graph paper. The example was created with the commands:

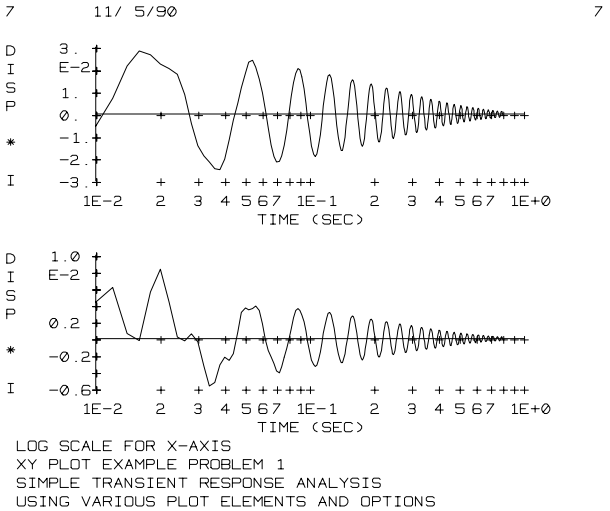
```
FTITLE=ADD GRID LINES
XGRID ON
YGRID ON
XYPLOT DISP RESPONSE
       / 4(T3,),2(,T3)
```



You may also select logarithmic scales in one or both coordinate directions. This is often used in transient response analyses. The example was created with:

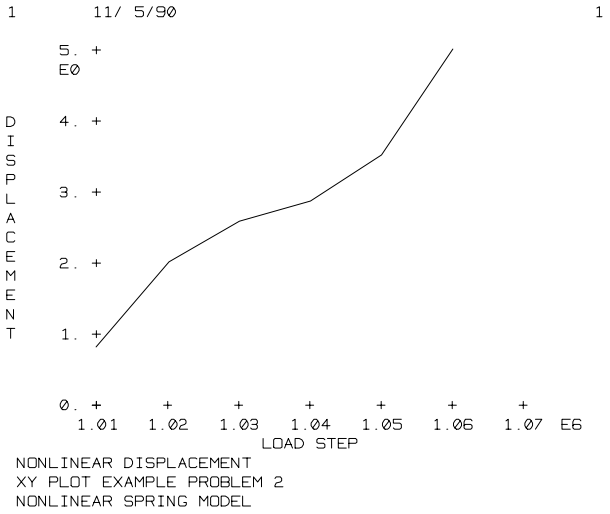
```
FTITLE=LOG SCALE FOR X-AXIS
XLOG ON
XYPLOT DISP RESPONSE
       / 4(T3,),2(,T3)
```

Note that when you use log scales, **UAI/*NASTRAN*** automatically selects the number of cycles. You may not control this value unless you reduce the data range requested.



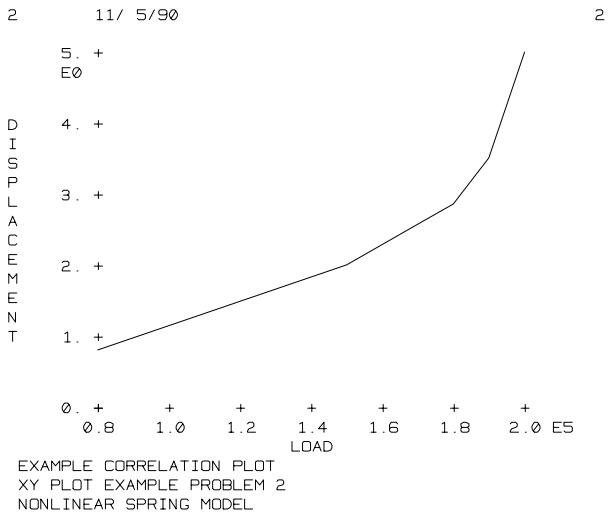
The second example problem is the nonlinear analysis of a simple spring system. The first plot shows the displacement as a function of the nonlinear load step. It was created with the commands:

```
FTITLE=NONLINEAR DISPLACEMENT
XTITLE=LOAD STEP
YTITLE=DISPLACEMENT
XYPLOT DISP RESP / 1(T1)
```



Since one is often interested in the force-deflection results for nonlinear analyses, a correlation plot can be created with the command:

```
FTITLE=EXAMPLE CORRELATION PLOT
XTITLE=LOAD
XYPLOT CORRELATION
       DISP VS OLOAD /
       1(T1) VS 1(T1)
```



30

30

This page is intentionally blank.

Chapter 31

INTERFACING WITH UAI/*NASTRAN*

UAI/*NASTRAN* has a number of interfaces with other third-party commercial programs and with other UAI products. This Chapter provides you with the basic data needed to interact with these other software systems.

31.1 THIRD-PARTY PRODUCTS

UAI/*NASTRAN* has interfaces to a variety of commercial software products including pre- and post-processors and tools for other types of analyses. Table 31-1 presents a summary of these. The following sections describe the manner in which you interface with each of them.

Table 31-1. INTERFACES TO THIRD-PARTY SOFTWARE

VENDOR	PRODUCT	DESCRIPTION
ESP	FEMAP®	ESP and UAI have developed an ASCII file interface between ESP's FEMAP pre- and post-processing system and **UAI/*NASTRAN***. FEMAP can create complete **UAI/*NASTRAN*** input files and performs comprehensive post-processing of solution results.
MSC	PATRAN® Version 2.5 and higher	The **UAI/*NASTRAN*** Rigid Formats have integrated support for PATRAN. The **POST** Case Control command is used to control the generation of the binary files which are required by the PATRAN program. The **POST** command allows you to specify the PATRAN version. In Multidisciplinary Rigid Formats such as Buckling, the **POST** command may be used in each discipline to select different output.
	PATRAN® Versions 5,6 and 7	UAI also provides an optional PATRAN Preference. Contact UAI directly for up-to-date information about this product.
SDRC	I-DEAS Master Series™ Simulation	The **UAI/*NASTRAN*** Rigid Formats have integrated support for I-DEAS Master Series Simulation. The **POST** Case Control command is used to control the generation of the binary files which are required by the UAI version of the I-DEAS™ Data Loader program. This program then creates a universal file required to as input to I-DEAS Simulation.
	I-DEAS Master Series™ Simulation Dataloader	UAI has developed, and delivers, a customized version of the Dataloader which is tuned for **UAI/*NASTRAN***.
	I-DEAS Master Series™ System Dynamics Analysis	There is an ALTERLIB member, **SDRCSDA**, which allows you to automatically create the **OUTPUT2** files needed by this program. These files are then input to the System Dynamics Analysis program.

31.1.1 ESP

PRE-PROCESSING. You may use any of the FEMAP model development tools to create your finite element models. These models are then communicated to **UAI/*NASTRAN*** through a Bulk Data packet which is created directly from FEMAP. You must consult the appropriate ESP documentation for information describing this capability.

POST-PROCESSING. FEMAP uses an ASCII file interface developed by ESP and UAI, to capture both geometry and solution results for post-processing. No special intervention is required when performing your **UAI/*NASTRAN*** execution.

31.1.2 MSC/PATRAN

For users having the latest versions of MSC/PATRAN®, UAI provides an optional Preference. This Preference allows you to submit **UAI/*NASTRAN*** jobs directly from within PATRAN and to post-process the analysis results. Versions 5, 6 and 7 of PATRAN are supported.

UAI/*NASTRAN* also supports both PATRAN 2.5 and PATRAN 3 for pre- and post-processing as described below.

PRE-PROCESSING. You may use any of the PATRAN model development tools to create your finite element models. These models are then communicated to **UAI/*NASTRAN*** through a Bulk Data packet which is created directly from PATRAN. You must consult the appropriate PDA documentation for information describing this capability.

POST-PROCESSING. PATRAN uses a **UAI/*NASTRAN*** OUTPUT2 file to capture solution results for post-processing. When you execute your **UAI/*NASTRAN*** job, you include the **POST** command with the **PDA** or **PDA3** option. See the **UAI/*NASTRAN User's Reference Manual*** for a complete description of this command. This command automatically creates an OUTPUT2 file which contains the data needed by PATRAN. Note that your Bulk Data packet is also required as input to versions of PATRAN prior to P3.

31.1.3 SDRC

There are **UAI/*NASTRAN*** interfaces to the SDRC products I-DEAS Simulation and I-DEAS System Dynamics Analysis. The following sections describe how you use these programs.

31.1.3.1 I-DEAS Simulation (Master Series) and I-DEAS Version VI

PRE-PROCESSING. You may use any of the SDRC I-DEAS model development tools to create your finite element models. These models are then communicated to **UAI/*NASTRAN*** through a Bulk Data packet which is created directly from I-DEAS. You must consult SDRC documentation for information describing this capability.

POST-PROCESSING. The I-DEAS post-processing programs use an SDRC formatted file called the *universal file*. There are two steps required to create this file. When you execute your **UAI/*NASTRAN*** job, you include the **POST** command with the **SDRC** option. See the **UAI/*NASTRAN User's Reference Manual*** for a complete description of this command. This command automatically creates an **OUTPUT2** file which contains the data needed by I-DEAS Simulation. This file must then be used as input to a program called the Dataloader which transforms it into a universal file.

UAI provides you with a customized version of this Dataloader program which is especially tailored to **UAI/*NASTRAN***. The manner in which you use this program is described below.

POST-PROCESSING: UNIX-BASED COMPUTERS. The Dataloader program **dlnast** reads an OUTPUT2 file generated from **UAI/*NASTRAN*** and converts it to the SDRC universal file format. This universal file may then be imported into I-DEAS for post-processing. To execute the dlnast program, you enter:

```
dlnast { -v4 } output2_file universal_file [option_codes]
       { -v6 }                                [first_case] [last_case]
```

where:

v4	Specifies conversion to version 4 of the universal file format		
v6	Specifies conversion to version 6 of the universal file format. You also use this option for the Master Series.		
output2_file	Name of **UAI/*NASTRAN*** OUTPUT2 file name		
universal_file	Name of resulting universal file		
option_codes	option list, separated by commas, selected from:		
	AL Select all data from the OUTPUT2 file (Default).	**GD**	Select geometry, grid point definitions and element connectivity, from the OUTPUT2 file.
	AD Select all the analysis vectors from the OUTPUT2 file.	**ND**	Select the displacement results from the OUTPUT2 file.
	ST Select the stress components from the OUTPUT2 file.	**EN**	Select the elemental energy data from the OUTPUT2 file.
	NG Select grid point definitions from the OUTPUT2 file.	**EC**	Select element connectivity from the OUTPUT2 file.
first_case	Start of the range for selective processing of cases for subcases or mode numbers (Default=1)		
last_case	End of the range for selective processing of cases for subcases or mode numbers (Default=999999999)		

POST-PROCESSING: DEC VAX VMS COMPUTERS. The Dataloader program for DEC VAX VMS is similar. To execute the **dlnast** program, you enter:

```
dlnast { /v4 } output2_file universal_file [option_codes]
       { /v6 }                              [first_case] [last_case]
```

where:

v4	Specifies conversion to version 4 of the universal file format		
v6	Specifies conversion to version 6 of the universal file format. You also use this option for the Master Series.		
output2_file	Name of **UAI/*NASTRAN*** OUTPUT2 file name		
universal_file	Name of resulting universal file		
option_codes	option list, separated by commas, selected from:		
	AL Select all data from the OUTPUT2 file (Default).	**GD**	Select geometry, grid point definitions and element connectivity, from the OUTPUT2 file.
	AD Select all the analysis vectors from the OUTPUT2 file.	**ND**	Select the displacement results from the OUTPUT2 file.

	ST Select the stress components from the OUTPUT2 file.	**EN** Select the elemental energy data from the OUTPUT2 file.
	NG Select grid point definitions from the OUTPUT2 file.	**EC** Select element connectivity from the OUTPUT2 file.
first_case	Start of the range for selective processing of cases for subcases or mode numbers (Default=1)	
last_case	End of the range for selective processing of cases for subcases or mode numbers (Default=999999999)	

31.1.3.2 System Dynamics Analysis.

PRE-PROCESSING. You may use any of the SDRC I-DEAS model development tools to create your finite element model. In generating the input file for **UAI/*NASTRAN***, you must remember that the the Craig-Bampton Exact Representation of a Constraint Mode problem imposes strict requirements on the finite element model. The requirements are:

- All boundary degrees of freedom must be placed on **SUPORT** Bulk Data entries.
- The model must contain only elements and grids within the substructure of interest.
- Craig-Bampton modes and matrices will be obtained relative to the fixed boundary grid points.
- The **LANCZOS** eigensolution with the Mass normalization option must be used. Do ***not*** use generalized dynamic reduction.

Once the model generation is complete, the **UAI/*NASTRAN*** Bulk Data packet can be generated directly from SDRC I-DEAS. You must consult the appropriate SDRC documentation for the information describing this capability.

POST-PROCESSING. The I-DEAS System Dynamics Analysis program uses OUTPUT2 files generated from **UAI/*NASTRAN*.** To recover data for the Craig-Bampton representation you **INCLUDE** the alter **SDRCSDA** from the ALTERLIB in your **UAI/*NASTRAN*** data stream. This alter will generate four **OUTPUT2** files which contain the geometry and modal properties data.

The appropriate Read menus in the Component Definition task of the I-DEAS System Dynamics Analysis program must be used to recover the geometry and matrices for the component. These Read menus allow you to open the **UAI/*NASTRAN*** OUTPUT2 files and copy the geometry and modal properties into the active component. See the I-DEAS System Dynamics Analysis User's Guide for information describing this capability.

31.2 UAI PRODUCTS

UAI provides a number of additional products that assist you in performing more efficient analysis. These include programs for processing **UAI/*NASTRAN*** graphics output and those which can create and access ***eBase*** ARCHIVE database. These products are summarized in Table 31-2.

Table 31-2. INTERFACES TO OTHER UAI SOFTWARE

PRODUCT	DESCRIPTION
nastplot **nastplotps** **nastplotgl** **tekplot**	UAI has developed four post-processing programs which are used to process plot files created by **UAI/*NASTRAN***. These programs display images on graphics devices or convert the plot file to a form suitable for printing on hard-copy devices.
eShell	The eBase Interactive Shell program allows you to access, modify and manage information in an eBase database. The program includes a powerful query language based on the SQL database standard.
eBase:applib **eBase:matlib**	The **eBase** Application Programming Interfaces provide a powerful suite of development tools. **applib** includes routines for manipulating database entities, and **matlib** includes routine for performing high-performance matrix operations. These tools may be used to design and implement complex scientific software systems which may or may not use **UAI/*NASTRAN*** data.

31.2.1 The Plotting Programs

Four post-processing plotting programs, **nastplot**, **nastplotps**, **nastplotgl** and **tekplot** are provided for use with **UAI/*NASTRAN***. Each of these programs may: create plot displays on graphics terminals or popular hardcopy devices; or create new files using either the PostScript or HP-GL languages. Additionally, source code is provided for the program **tekplot** which provides your facility with a starting point for creating your own customized plotting program.

31.2.1.1 Creating Plots and Plot Files

You create plots of your finite element model and solution results from within **UAI/*NASTRAN*** by using either or both of the plot packets indicated by the Case Control commands:

```
OUTPUT(PLOT)
OUTPUT(XYPLOT)
```

Details of defining both structural plots and X-Y plots are found in Chapters 29 and 30 of this manual. If plots have been selected, then your **UAI/*NASTRAN*** execution creates a plot file. This plot file may be used with a number of utility programs which are described in the following sections.

31.2.1.2 The X-Window, Motif Interface Plot Program

For computer systems which support the X-Window system, the plotting program **nastplot** is provided. This program operates in the X-Window environment, and it uses a Motif interactive interface. **nastplot** provides the following functional capability for viewing and processing **UAI/*NASTRAN*** plot files:

- ❒ Automatic recognition and processing of binary or formatted plot files.
- ❒ Full support of the **UAI/*NASTRAN*** **LINESTYLE** command using user selectable display colors.
- ❒ Direct selection of display for any plot in the plot file.
- ❒ Zooming of the plot display.
- ❒ Export of plots to either a printer or a file, using either PostScript or HP PCL display languages.

nastplot is executed with the command:

```
nastplot [ file_name ]
```

where *file_name* is the name of the plot file created in your **UAI/*NASTRAN*** job. A menu of options then appears. These options allow you to perform the selected functions.

Special Versions of the NASTPLOT Program. On HP/Apollo and Sun workstations special versions of **nastplot** are delivered which operate under the normal window system found on those computers, Display Manager and SunView, respectively.

31.2.1.3 The Plotter Conversion Programs

There are two special plotter conversion programs which allow you to convert the plot files created by **UAI/*NASTRAN*** into files which can be printed directly on hardcopy devices. There are two available conversions: to PostScript and to HP GL. These programs are described below.

POSTSCRIPT CONVERSION. The program **nastplotps** reads both binary and formatted plot files generated by **UAI/*NASTRAN*** and generates an Encapsulated PostScript file. This PostScript output can then be either sent to a printer or imported into a text formatting program which accepts Encapsulated PostScript input. Importing the plot only makes sense when the plot file contains a single frame or if you use the -pn option to explicitly create a single plot. The program allows you to select fonts, control paper size and to determine output orientation. The different versions of this program are described below.

nastplotps: UNIX SYSTEMS. Detailed documentation on the plotter options is available by executing the following command with no arguments:

```
nastplotps
```


The on-line help is:

```
Usage: nastplotps options file_name_1 file_name_2 ...
          -b  = plot files are binary (default)
          -f  = plot files are formatted
          -nf = suppress frame around plot
         -pn# = only plot number # is processed
         -pw# = paper width (default -pw8.5)
         -mw# = unplottable margin width (default -mw0.25)
         -ph# = paper height (default -ph11.0)
         -mh# = unplottable margin height (default -mh0.25)
         -por = portrait orientation (default)
         -lan = landscape orientation
          -tx = typeface (default -tHelvetica)
```

The output of **nastplotps** is to Unix standard output. Normally, you should redirect standard output to a file or pipe it to a print spooling program as desired. The following illustrates a typical use of **nastplotps**:

```
nastplotps -b -por userplot.plt | lpr -Pps
```

This indicates that the input plot file, `userplot.plt`, is binary, that it is to be printed in **PORTRAIT** orientation, and the output of `nastplotps` is piped directly to a PostScript printer.

nastplotps: VAX VMS. A command procedure called **NASTPLOTPS** is provided to allow you to create these PostScript files. To execute, you enter:

```
NASTPLOTPS[/TYPE = { BINARY } ][/ORIENT = { PORTRAIT  } ][ /NOFRAME ]
                   { FORMAT }            { LANDSCAPE }

        [/PN = #][/PW = #][/MW = #][/PH = #][/MH = #]

        [TYPEFACE = tf][/OUTPUT = filename]  plotfile_1  plotfile_2 ...
```

where:

{ BINARY } { FORMAT }	Specifies the plot file format. Default is **BINARY**.
{ PORTRAIT } { LANDSCAPE }	Specifies the paper orientation. Default is **PORTRAIT**.
/NOFRAME	Suppresses the frame around the plot.
/PN=#	Processes single plot with sequence number #.
/PW=#	Sets paper width. Default is 8.5 in.
/MW=#	Specifies unplottable margin width. Default 0.25 in.
/PH=#	Specifies paper height. Default 11.0 in.
/MH=#	Specifies unplottable margin height. Default 0.25 in.
[/TYPEFACE=*tf*]	Selects a PostScript typeface. Default is Helvetica.
/OUTPUT=*filename*	Specifies the file name which will contain the resulting Encapsulated PostScript file. If omitted, output is routed to **SYS$OUTPUT**.
plotfile_i	Specifies the file names which contains your plot files created by a **UAI/*NASTRAN*** job.

Normally you should redirect **SYS$OUTPUT** to a file. This file may then be routed to the PostScript printer, if available, at your facility. For example:

```
NASTPLOTPS /OUTPUT=plot.pst -lan mydata.plt
```

This indicates that the input plot file, `mydata.plt`, is to be printed in **LANDSCAPE** orientation, and the output of **NASTPLOTPS** is redirected to a file `plot.pst` which could then be routed to a PostScript printer.

HP GL CONVERSION. The program **nastplotgl** also reads both binary and formatted plot files generated by **UAI/*NASTRAN*** and generates an HP GL file. This file can then be either sent to a printer or plotter which is compatible with the GL language, or it can be imported into a text formatting program which accepts this format. Importing the plot only makes sense when the plot file contains a single frame or if you use the **-pn** option to explicitly create a single plot. The different versions of this program are described below.

nastplotgl: UNIX SYSTEMS. Detailed documentation on the plotter options is available by executing the following command with no arguments:

```
nastplotgl
```

The on-line help is:

```
Usage: nastplotgl options file_name_1 file_name_2 ...
          -b  = plot files are binary (default)
          -f  = plot files are formatted
          -nf = suppress frame around plot
         -pn# = only plot number # is processed
         -pw# = paper width (default -pw8.5)
         -mw# = unplottable margin width (default -mw0.25)
         -ph# = paper height (default -ph11.0)
         -mh# = unplottable margin height (default -mh0.25)
```

The output of **nastplotgl** is to Unix standard output. Normally, you should redirect standard output to a file or pipe it to a print spooling program as desired.

nastplotgl: VAX VMS. A command procedure called **UAIPLOTGL** is provided to allow you to create these PostScript files. To execute, you enter:

```
NASTPLOTGL [/TYPE = { BINARY } ]  [ /NOFRAME ] [/PN = #][/PW = #]
           [ FORMAT ]

           [/MW = #][/PH = #][/MH = #] [/OUTPUT=filename ]  plotfile_1  plotfile_2  ...
```

where:

{ BINARY } { FORMAT }	Specifies the plot file format. Default is **BINARY**.
/NOFRAME	Suppresses the frame around the plot.
/PN=#	Processes single plot with sequence number #.
/PW=#	Sets paper width. Default is 8.5 in.
/MW=#	Specifies unplottable margin width. Default 0.25 in.
/PH=#	Specifies paper height. Default 11.0 in.
/MH=#	Specifies unplottable margin height. Default 0.25 in.
/OUTPUT=*filename*	Specifies the file name which will contain the resulting Encapsulated PostScript file. If omitted, output is routed to **SYS$OUTPUT**.
plotfile_i	Specifies the file names which contains your plot files created by a **UAI/*NASTRAN*** job.

Normally you should redirect **SYS$OUTPUT** to a file. This file may then be routed to the PostScript printer, if available, at your facility.

31.2.1.4 The Tektronix PLOT10 Plot Program

The Fortran program, **tekplot**, is provided as source code. You may modify and use this program to process **UAI/*NASTRAN*** plot files and create displays on graphics terminals connected to your host computer which support the Tektronix PLOT10 graphics instructions. Contact your **UAI/*NASTRAN*** System Support Specialist for additional information.

31.2.2 The eBase Interfaces

UAI provides several products which allow you to query and manipulate an **eBase** database. These products, which include the **eSHELL** Interactive Interface program and the **eBase:applib** and **matlib** Application Programming Libraries, operate using an ARCHIVE database that is created during your **UAI/*NASTRAN*** execution. The following sections describe the methodology required to do this.

31.2.2.1 Creating an ARCHIVE database

In order to use the eBase software tools, you must capture your **UAI/*NASTRAN*** model geometry and solution results on an ARCHIVE database. This is accomplished by including one or more **ARCHIVE** Case Control commands in your input data stream. Several commands may be used to ARCHIVE different sets of data on different databases. Once the ARCHIVE database exists, you may use the eShell Interactive Interface program to access it, or you may use the **eBase:applib** and **matlib** Application Programming Libraries to write Fortran programs which may access it.

*31.2.2.2 The **eShell** Program*

If your site has the ***eShell*** interactive ***eBase*** interface program, then to execute this program on all Unix-based computers and VAX VMS computers, you enter:

```
eshell [ pref_file ]
```

where:

pref_file Specifies the substitution string used to generate the Preference File names.

This command executes ***eShell*** in the interactive mode. Unless directed otherwise by ***eShell*** commands, all subsequent output will be sent to the terminal device. The eShell Tutorial Problem library is available. Contact your Systems Support Specialist to obtain the name of the directory where these problems may be found. A description of how you may use them is given in the ***eShell User's Manual***.

31.2.2.3 Building Applications with eBASE

If your site has the ***eBase:applib*** and ***eBase:matlib*** Applications Programming Interfaces, you may create your own Fortran programs and use a **UAI/*NASTRAN*** database directly. Complete details are found in the ***eBase:applib Programmer's Manual*** and the ***eBase:matlib Programmer's Manual***. Additional information relative to linking program with these libraries is found in the System Support Manual.

This page is intentionally blank.

REFERENCES

[Bjilaard54] Bjilaard, P.P., "Stresses from Radial Loads in Cylindrical Pressure Vessels," *Weld J.*, Vol. 33, December 1954.

[Francis61] Francis, J.G.F., "The QR Transformation - a Unitary Analogue to the LR Transformation," *Computer Journal*, Vol. 4, No. 3 et. seq., 1961.

[Grimes86] Grimes, R.G., Lewis, J.G., and Simon, H., "The Implementation of a Block Shifted and Inverted Lanczos Algorithm for Eigenvalue Problems in Structural Engineering," Applied Matematics Technical Report, Boeing Computer Services, ETA-TR-39, August, 1986.

[Grimes91] Grimes, R.G., Lewis, J.G., and Simon, H., "A Shifted Block Lanczos Algorithm for solving Sparse Symmetric Generalized Eigenproblems," Boeing Computer Services, AMS-TR-166, July, 1991.

[Harris61] *Shock and Vibration Handbook*, Harris and Crede, Eds, McGraw-Hill, 1961

[Hughes87] Hughes, T.J.R., *The Finite Element Method*, Prentice-Hall, 1987.

[Kreith73] Kreith, F., *Principles of Heat Transfer*, Intext Educational Publishers, 1973.

[Naeim89] Naeim, F. (ed.), *The Seismic Design Handbook*, Van Nostrand Reinhold, 1989.

[NASTRAN79] *The NASTRAN Theoretical Manual*, Level 17.0, NASA SP-221(04), 1979

[Parlett80] Parlett, B.N., *The Symmetric Eigenvalue Problem*, Prentice-Hall, 1980.

[Popov68] Popov, E.P., *Introduction to Mechanics of Solids*, Prentice-Hall, 1968.

[Spiegel58] Spiegel, M.R., *Applied Differential Equations*, Prentice-Hall, 1958.

[Timoshenko36] Timoshenko, S.P., *Theory of Elastic Stability*, McGraw-Hill, 1936.

[Timoshenko61] Timoshenko, S.P. and Gere, J.M., *Theory of Elastic Stability*, McGraw-Hill, 1961.

[Wilkinson65] Wilkinson, J.H., *The Algebraic Eigenvalue Proble*m, Oxford: Clarendon Press, 1965.

[Young89] Young, W.C., *Roark's Formulas for Stress and Strain*, McGraw-Hill, 1989.

Note that Chapter 17, Geometric Nonlinear Analysis, Chapter 23, Three-Dimensional Fluid-Structure Interaction, Chapter 25, Multidisciplinary Design Optimization, and Chapter 28, Aeroelastic Flutter, have their own sets of references because of their highly specialized nature.

INDEX

A

B

C

D

E

F

G

H

I

K

L

M

Q

R

S

T

U

V

W

X

Y

Z