ASTROS ENHANCEMENTS

Volume I — ASTROS USER'S MANUAL

D.J. NEILL
D.L. HERENDEEN

Universal Analytics, Inc.
3625 Del Amo Blvd. Suite 370
Torrance, CA 90503

March 1993

INTERIM REPORT FOR THE PERIOD 1/15/87-10/30/92

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED

FLIGHT DYNAMICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-7562
THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.
NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise as in any manner, as licensing the holder or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

STEPHEN M. PITROF, Capt, USAF
Project Engineer
Design & Analysis Methods Section

NELSON D. WOLF, Technical Manager
Design & Analysis Methods Section
Analysis & Optimization Branch

DAVID K. MILLER, Lt Col, USAF
Chief, Analysis & Optimization Branch
Structures Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify WL/FIBRA Bldg 45, 2130 Eighth St Ste 11, Wright-Patterson AFB OH 45433-7552 to help us maintain a current mailing list".

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.
**Title and Subtitle:**

ASTROS ENHANCEMENTS
VOLUME 1 - ASTROS USER'S MANUAL

**Author(s):**

D. J. NEILL
D. L. HERENDEEN

**Performing Organization Name(s) and Address(es):**

UNIVERSAL ANALYTICS INC.
3625 DEL AMO BLVD STE 370
TORRANCE CA 90503

**Sponsoring/Monitoring Agency Name(s) and Address(es):**

FLIGHT DYNAMICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT PATTERSON AFB OH 45433-7562

**Abstract:**

ASTROS (Automated STRuctural Optimization System) is a computer program for the multidisciplinary design and analysis of aerospace structures. ASTROS combines mathematical optimization algorithms with traditional structural analysis disciplines such as static forces, normal modes, static aeroelasticity, and dynamic aeroelasticity (flutter), all in a finite element context, to perform automated preliminary design of an aircraft structure. This report is a complete user's manual that documents the many features of ASTROS through version 10 of the software package. It also provides information on system architecture and other topics of interest. This report is Volume 1 of a set; Volume 2 (WL-TR-93-3038) is the programmer's manual.
FOREWORD

This interim report is submitted in fulfillment of CDRL CLIN 0001, Sequence No. 13. of the ASTROS Enhancements Contract, F33615-87-C-3216, dated 29 January 1987. This volume provides the basic user's documentation for the Automated Structural Optimization System.

This work was performed by Universal Analytics, Inc. and their subcontractor, Northrop Corp. This manual supersedes the original ASTROS User's Manual documented in AFWAL-TR-88-3028, Volume II by D.J. Neill, E.H. Johnson and D.L. Herendeen. The major contributors to this report were D.L. Herendeen, the Program Manager, and D.J. Neill, the Associate Program Manager. E.H. Johnson, previously of Northrop, was a major contributor to the original report.

Capt. S. Pitrof is the Air Force Project Engineer and Dr. V.B. Venkayya the initiator of the effort. This report covers the work performed between 29 January 1987 and 30 October 1992, but also includes updated information valid for Version 10.0 of ASTROS.
# CONTENTS

## 1. THE ASTROS INPUT DATA .............................................. 1

1.1. INTRODUCTION .................................................. 1

1.2. THE INPUT DATA STREAM ......................................... 3

1.3. THE ASSIGN DATABASE ENTRY — GENERAL .......................... 7

1.3.1. ASSIGN DATABASE DESCRIPTIONS FOR HOST COMPUTERS ............ 9

1.3.1.1. VMS IMPLEMENTATION .................................... 9

1.3.1.2. IBM MVS IMPLEMENTATION .................................. 11

1.3.1.3. UNIX SYSTEM IMPLEMENTATION ............................ 12

1.4. THE INCLUDE DIRECTIVE ........................................... 14

1.5. THE DEBUG PACKET ................................................ 15

1.5.1. EXECUTIVE SYSTEM DEBUG COMMANDS .......................... 16

1.5.2. DATABASE AND MEMORY MANAGER DEBUG COMMANDS ............... 17

1.5.3. INTERMEDIATE RESULTS PRINTING COMMANDS ....................... 18

1.5.4. SEQUENCER INTERMEDIATE PRINT COMMANDS ....................... 21

## 2. THE EXECUTIVE SYSTEM AND MAPOL .................................. 23

2.1. THE MAPOL PROGRAM .............................................. 24

2.2. MAPOL EDIT COMMANDS ........................................... 25

2.3. THE STANDARD EXECUTIVE SEQUENCE ................................ 25

2.4. STANDARD EXECUTIVE SEQUENCE STRUCTURE ......................... 26

2.4.1. MAPOL Declarations ........................................... 28

2.4.2. The Solution Algorithm ....................................... 34

2.4.2.1. MAPOL Engineering and Utility Modules ....................... 35

2.4.2.2. The Preface Segment ....................................... 40

2.4.2.3. The Analysis/Optimization Segments ........................ 40

2.4.3. Modifying the Standard MAPOL Sequence ........................ 41

2.4.4. Restart Capability ............................................. 43

2.4.4.1. Introduction ............................................... 43

2.4.4.2. Ensuring proper STATUS of the run-time database ............. 43

2.4.4.3. Suspending/Restarting Execution ................................ 44

2.4.4.4. Resetting MAPOL Parameters ................................ 45

2.5. MAPOL PROGRAM LISTING .......................................... 45
6. MAPOL PROGRAMMING ........................................... 395

6.1. INTRODUCTION AND USER OPTIONS ........................... 395
  6.1.1. USER OPTIONS ........................................... 395
  6.1.2. MAPOL PROGRAM FORM .................................... 396
  6.1.3. THE STANDARD ASTROS SOLUTION .......................... 396
  6.1.4. MODIFYING THE STANDARD SOLUTION ........................ 397
  6.1.5. CREATING MAPOL PROGRAMS ............................... 397
  6.1.6. SUMMARY ................................................. 397

6.2. DATA TYPES AND DECLARATIONS ................................ 398
  6.2.1. DEFINITIONS AND NOTATION ................................ 398
  6.2.2. COMMENTARY .............................................. 399
  6.2.3. SIMPLE DATA TYPES ....................................... 399
    6.2.3.1. Data Type INTEGER .................................. 400
    6.2.3.2. Data Type REAL ...................................... 400
    6.2.3.3. Data Type COMPLEX .................................. 400
    6.2.3.4. Data Type LOGICAL .................................. 400
    6.2.3.5. Data Type LABEL ..................................... 400
  6.2.4. COMPLEX DATA TYPES ...................................... 400
    6.2.4.1. Data Types MATRIX and IMATRIX ......................... 401
    6.2.4.2. Data Type Relation ................................... 401
    6.2.4.3. Data Types UNSTRUCT and IUNSTRUCT ..................... 403
    6.2.4.4. Data Base Entity Declaration Requirements ............... 403

6.3. EXPRESSIONS AND ASSIGNMENTS .................................. 403
  6.3.1. ARITHMETIC EXPRESSIONS .................................. 404
    6.3.1.1. Arithmetic Operators ................................ 404
    6.3.1.2. Arithmetic Operands ................................ 404
    6.3.1.3. Evaluation of Arithmetic Expressions .................... 405
    6.3.1.4. The Uses of Parentheses ................................ 405
    6.3.1.5. Type and Value of Arithmetic Expressions ............... 406
  6.3.2. LOGICAL EXPRESSIONS ..................................... 406
    6.3.2.1. Logical Operators ................................... 406
    6.3.2.2. Logical Operands ..................................... 407
    6.3.2.3. Evaluation of Logical Expressions ........................ 407
  6.3.3. RELATIONAL EXPRESSIONS .................................. 408
    6.3.3.1. Relational Operators ................................ 408
    6.3.3.2. Relational Operands ................................ 409
    6.3.3.3. Evaluation of Relational Expressions .................... 409
  6.3.4. MATRIX EXPRESSIONS ...................................... 409
    6.3.4.1. Matrix Operators .................................... 409
6.3.4.2. Matrix Operands and Expressions .............................................. 410
6.3.5. ASSIGNMENT STATEMENTS .......................................................... 410

6.4. CONTROL STATEMENTS ................................................................. 411
  6.4.1. INTRODUCTION ............................................................................. 411
  6.4.2. THE UNCONDITIONAL GOTO STATEMENT ......................................... 411
  6.4.3. ITERATION ................................................................................... 412
     6.4.3.1. The FOR...DO Loop ............................................................ 412
     6.4.3.2. The WHILE...DO Loop ......................................................... 413
  6.4.4. THE IF STATEMENT ...................................................................... 414
     6.4.4.1. The Logical IF ..................................................................... 414
     6.4.4.2. The Block IF ....................................................................... 414
     6.4.4.3. The IF...THEN...ELSE ......................................................... 415
     6.4.4.4. Nested IF Statements ........................................................... 415
  6.4.5. THE END AND ENDP STATEMENTS ............................................... 415

6.5. INPUT/OUTPUT STATEMENTS .......................................................... 415
  6.5.1. THE PRINT STATEMENT ............................................................... 416

6.6. PROCEDURES AND FUNCTIONS ...................................................... 416
  6.6.1. INTRODUCTION ............................................................................. 416
  6.6.2. PROGRAM UNITS AND SCOPE OF VARIABLES ............................... 416
  6.6.3. DEFINING A PROCEDURE ............................................................ 417
  6.6.4. INVOKING A PROCEDURE ........................................................... 418
  6.6.5. FUNCTION PROCEDURES ............................................................. 419
     6.6.5.1. Examples of Variable Scope .................................................. 419
  6.6.6. INTRINSIC FUNCTION PROCEDURES AND INTRINSIC PROCEDURES ................................................................................. 419
  6.6.7. INTRINSIC MATHEMATICAL FUNCTIONS ...................................... 420
  6.6.8. INTRINSIC RELATIONAL PROCEDURES ....................................... 421
  6.6.9. GENERAL INTRINSIC PROCEDURES .......................................... 422

7. REFERENCES ......................................................................................... 425
LIST OF FIGURES

Figure 1. Structure of the ASTROS Input Data Stream .................................................. 5
Figure 2. Features of a Sample ASTROS Input Stream .................................................. 6
Figure 3. Function of the ASSIGN DATABASE Entry .................................................... 8
Figure 4. Structure of the Standard MAPOL Sequence ................................................ 27
Figure 5. BAR Element Coordinate System .................................................................... 365
Figure 6. BAR Element Forces Sign Conventions .......................................................... 366
Figure 7. IHEX1 Element Geometry ............................................................................... 367
Figure 8. IHEX2 Element Geometry ............................................................................... 369
Figure 9. IHEX3 Element ............................................................................................... 370
Figure 10. ROD Element Coordinate System ............................................................... 371
Figure 11. QDMEM1 Element Coordinate System ......................................................... 372
Figure 12. TRMEM Element Coordinate System .......................................................... 373
Figure 13. QUAD4 Element Coordinate System ........................................................... 374
Figure 14. TRIA3 Element Coordinate System ............................................................. 374
Figure 15. Shear Panel Forces ....................................................................................... 377
Figure 16. Schematic Representation of Relational Data ................................................ 402
Figure 17. MAPOL Program Using Relational Procedures ........................................... 422
LIST OF TABLES

Table 1. Command Syntax Conventions ................................................................. 3
Table 2. MAPOL Debug Commands ........................................................................ 16
Table 3. Database Debug Commands ..................................................................... 17
Table 4. Intermediate Results Debug Commands .................................................. 18
Table 5. Sequencer Debug Commands ................................................................... 21
Table 6. MAPOL Edit Commands ........................................................................... 25
Table 7. Real Parameters in the Standard Sequence ............................................ 29
Table 8. Integer Modelling Parameters .................................................................. 30
Table 9. Integer Design Parameters ...................................................................... 31
Table 10. Integer Aerodynamic Parameters ......................................................... 31
Table 11. Integer Discipline Parameters ............................................................... 32
Table 12. Logical Discipline Parameters ............................................................... 33
Table 13. Summary of ASTROS Modules ............................................................... 35
Table 14. Levels of Solution Control ...................................................................... 100
Table 15. Summary of ASTROS Disciplines ......................................................... 106
Table 16. Summary of Discipline Options ............................................................ 109
Table 17. Response Quantities by Discipline ......................................................... 116
Table 18. DEBUG and ASSIGN DATABASE Output .............................................. 357
Table 19. Boundary Condition Summary ......................................................... 357
Table 20. Active Boundary and Constraint Summary ........................................... 358
Table 21. Resequencing Summary ....................................................................... 358
Table 22. Active Constraint Summary ................................................................. 359
Table 23. Approximate Optimization Summary .................................................... 359
Table 24. Design Iteration History ...................................................................... 360
Table 25. ASTROS Execution Summary ............................................................... 361
Table 26. Aerodynamic and Structural Elements in ASTROS ............................... 364
Table 27. BAR Element Output Quantities ........................................................... 366
Table 28. IHEX1 Element Solution Quantities ...................................................... 368
Table 29. ROD Element Solution Quantities ........................................................ 372
Table 30. QDMEM1 Solution Quantities ............................................................... 373
Table 31. QUAD4 and TRIA3 Solution Quantities ............................................... 375
Table 32. SHEAR Solution Quantities ............................................................... 377
Table 33. Displacement Vector ......................................................................... 378
Table 34. Complex Displacement Vector ........................................................... 378
Table 35. Design Variable Values ...................................................................... 381
Table 36. Flutter Solution Results ...................................................................... 382
Table 38. Real Eigenanalysis Results ................................................................. 383
Table 37. Modal Participation Factors ............................................................... 383
Table 39. Symmetric Trim Results .................................................................... 385
1. THE ASTROS INPUT DATA

1.1. INTRODUCTION

This User's Manual is one of four manuals documenting ASTROS (the Automated Structural Optimization System). The other three are the Application Manual, the Programmer's Manual and the Theoretical Manual. The Application and Theoretical Manuals indicate the range of capabilities of the ASTROS system while the Programmer's Manual is provided to give details of the internal workings of the engineering and programming utility modules. The User's Manual provides a complete description of the user interface to the ASTROS system in order to facilitate the preparation of input data. It introduces the features of the ASTROS system that enable the user to direct this software system and documents the mechanisms by which the user can communicate with the system. It is assumed that the reader is familiar, from a study of the Theoretical Manual, with the engineering capabilities of the ASTROS system and is using this manual to define the form of the particular input that directs the system to perform a desired function. This manual is intended to provide the user with a convenient reference for all forms of input to the system and is therefore organized along the same lines as the input data stream. The discussion of each topic is brief and generic and is followed by detailed documentation of the user inputs. Information on ASTROS output formats is in a separate chapter as is the description of the Matrix Analysis Problem Oriented Language (MAPOL) used for programming ASTROS.

This manual is directed toward the engineer/designer/analyst who is using ASTROS to perform engineering design or analysis. While ASTROS is perfectly capable of performing many tasks not explicitly supported in the standard execution, the user must know the engineering software in considerable detail to direct the procedure to perform these alternative functions. The mechanisms by which these more advanced features are invoked are included in this manual but no attempt is made to provide sufficient information to the user to generate new analysis features or to grossly modify the existing capabilities of the system. These more advanced topics are treated in the Programmer's and Application Manuals which document the individual modules in the system and their interactions. Rudimentary (and typical) modifications to the execution sequence and changes to execution parameters are discussed in detail in this manual.

Machine and installation dependent aspects of ASTROS are also contained in the Application and Programmer's Manuals rather than in the User's Manual. Only those machine dependency issues that are logically related to the preparation of the input are discussed in the User's Manual. Machine dependencies in the input are limited to the naming conventions for the run time database file(s) and the parameters that can be used on the ASSIGN DATABASE entry. Other machine dependencies are handled as part of the installation of the system on each particular host machine. These issues are documented in the Programmer's Manual since they are relevant only to the "system manager," not to the "user."
It will be apparent to many readers that the NASTRAN structural analysis system was used as a guide in the design of the ASTROS procedure. Both NASTRAN and ASTROS comprise large scale, finite element structural analysis in executive driven software systems. Therefore, many of the input and output features are similar. NASTRAN has become an industry standard in finite element structural analysis with many pre- and post-processors developed around NASTRAN data. In order to facilitate acceptance and use of the ASTROS procedure, NASTRAN compatibility was considered desirable in many areas. As a result, many aspects of the ASTROS input are similar in form or purpose to those in NASTRAN and, in many other cases, the same nomenclature has been adopted. In some instances in this document, therefore, ASTROS input will be compared and contrasted to NASTRAN input in order to present a concise picture of the ASTROS input and to assist the reader familiar with NASTRAN in making the connection to the equivalent item in ASTROS. Although familiarity with NASTRAN is not a prerequisite to understanding the ASTROS documentation, sufficient numbers of potential ASTROS users are expected to be familiar with the NASTRAN system to justify the sometimes casual reference to NASTRAN features.

This Chapter contains a description of the ASTROS input file, database assignment and debug control inputs with Chapters 2 through 4 organized to parallel the input file structure. Within each of these chapters, the function of the particular input packet is presented along with illustrations of how the data are prepared. Each packet is described in a generic fashion so as to indicate how the sophisticated user can make use of the more advanced features of the system without cluttering the discussion with details of the input structures. The detailed documentation of the separate input structures of the data packet then follow within each Chapter. This form of user's documentation enables this manual to be useful as a guide to the beginning user as well as a reference for the experienced user. While there are a number of advanced input features, the required input for most jobs is the ASSIGN DATABASE command, described in Section 1.3, and the Solution Control and Bulk Data packets described in Chapters 3 and 4, respectively.

In Chapter 5, following the input stream descriptions, the output features of the ASTROS system are documented. While these features are selected through directives in the input data stream, they are sufficiently numerous and complex to justify a separate chapter devoted solely to output requests. The output capabilities of the system are described in very general terms while the output requests available for each analysis discipline and optimization feature are documented in detail. Most output is selected through Solution Control directives that are documented in Chapter 3, but some are selected through modifications to the executive (MAPOL) sequence. Chapter 2 documents all of the output utilities available to the user through MAPOL directives and gives several examples of modifying the MAPOL sequence to obtain additional output. Other features are described in the MAPOL Programmer's Manual which comprises Chapter 6.
Many examples of user input are used throughout this document. In order to ease the burden of interpretation, the conventions shown in Table 1 will be used in the examples unless otherwise noted. Chapter 6, which describes the MAPOL programming interface, describes additional conventions required for the programming syntax of MAPOL.

<table>
<thead>
<tr>
<th>MAPOL NOGO</th>
<th>Capital letters indicate that the phrase must appear exactly as shown</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAPOL params</td>
<td>Lower case italic symbols act as generic place holders indicating that an option or options can or must be included</td>
</tr>
<tr>
<td>MAPOL [GO NOGO]</td>
<td>Symbol(s) enclosed in brackets [ ] are optional. If more than one symbol is available they will be stacked in vector notation with any defaults denoted by boldface.</td>
</tr>
<tr>
<td>INCLUDE &lt;filename&gt;</td>
<td>A required symbol is enclosed in angle brackets. If the angle brackets surround an option list, at least one of the available options must be selected.</td>
</tr>
<tr>
<td>BEGIN_BULK</td>
<td>The underscore (_) is used to signify a required blank space.</td>
</tr>
</tbody>
</table>

### 1.2. THE INPUT DATA STREAM

The ASTROS user directs the system through an input data stream composed of a command to attach the ASTROS run time database followed by multiple data packets. Each packet contains a set of related data providing the information needed to execute the ASTROS procedure. The packets begin with a keyword indicating the nature of the data within the packet and terminate with an ending keyword or with the start of the next data packet. All the packets in the input data stream are optional, although the order in which they must appear is fixed. The purpose of this section is to document the structure of the input data stream. Detailed documentation of the data within each data packet is then presented in separate chapters.

Figure 1 shows the general form of the input data stream and Figure 2 illustrates the actual input stream features with a sample stream for a ten bar truss model. The first non-blank record of the input file must be the ASSIGN DATABASE entry. This command enables the user to attach the run time database file(s) that are used during the execution of the ASTROS procedure. There are four optional data packets following the ASSIGN DATABASE entry which, if they are present, must appear in the order shown. The first is the DEBUG packet which contains debug commands to control or select specific actions within the executive and database management systems. The second packet is the MAPOL packet containing the executive system control directives consisting of either a standalone MAPOL program or EDIT commands to modify the standard MAPOL program. If the MAPOL packet is absent, the unmodified standard MAPOL sequence directs the execution. The Solution Control commands appear in the third optional packet denoted by the keyword SOLUTION. These commands select the engineering data to be used in each subcase from the set of data provided in the fourth and final data packet: the
BULK DATA packet. The BULK DATA packet contains the engineering data describing the finite element structural model, the aerodynamic model(s), and the design model, as well as all the data needed to perform the specific analysis and/or optimization tasks. The MAPOL, SOLUTION and BULK DATA packets are analogous to the NASTRAN executive control, case control and bulk data decks, respectively.

In interpreting the input data stream, ASTROS recognizes the keywords shown in Figure 1. These keywords must be the first nonblank characters on the line (leading blanks are allowed) and have the structure shown. In some cases the keyword is also a command line that makes up part of the data packet which it initiates. In these cases, the command parameters are documented in the User's Manual chapter discussing the details of the associated data packet. For example, the MAPOL keyword is part of a command that directs the MAPOL compiler to take certain actions. The detailed discussion of the MAPOL command is therefore contained in Chapters 2 and 6 of this manual. ASTROS automatically converts the case of the input data stream when necessary. The only portions of the data which are not converted are file names which are used in the INCLUDE and ASSIGN DATABASE commands, and the Solution Control commands TITLE, SUBTITLE and LABEL. This allows the user to freely enter data in any case. Be aware, however, that file names are never converted and that when using an ASTROS host computer in which case is important, such as Unix, then the correct case must always be used in file names.

Two keyword commands are related only to the input data stream and not to the data within a packet. These are the ASSIGN DATABASE and the INCLUDE commands. Each of these is discussed in detail in the following sections.
Any number of leading blank lines

**ASSIGN DATABASE** <name> <password> <status> [params]

**DEBUG**

DEBUG directives
used for tracing input stream errors

**MAPOL** [option-list] or **EDIT** [option-list]

MAPOL program or EDIT commands
allow user modifications to the standard ASTROS solution sequence

**SOLUTION**

Solution Control Directives
select optimization and analysis disciplines

**BEGIN_BULK**

Bulk Data Entries
defines the structural and aero models, boundary
conditions, loading cases, and other engineering data and the
design model — design variables and constraints — required
when performing design. Similar to NASTRAN

... ...

**ENDDATA**

Bulk Data Terminator

---

_Figure 1. Structure of the ASTROS Input Data Stream_
ASSIGN DATABASE TENBAR SHAZAM NEW DELETE

DEBUG
DESIGN=5

EDIT NL=LIST

INSERT 1463
CALL UMGPF (.,[AMAT]);

SOLUTION

TITLE = TENBAR TRUSS
OPTIMIZE
PRINT DCON=ALL, HIST
BOUNDARY SPC = 1
   LABEL = STATIC ANALYSIS
   STATICS (MECH = 1), CONST (STRESS = 100, GENERAL = 100)
END

ANALYZE

BOUNDARY SPC = 1, METHOD = 2
   STATICS (MECH = 1)
   LABEL = FINAL STATIC ANALYSIS
   PRINT DISP = ALL
   MODES
       LABEL = FINAL MODAL ANALYSIS
       PRINT (MODES=ALL) DISP = ALL, ROOT=ALL
END

BEGIN BULK

GRID, 1, 720.0, 360.0, 0.0
GRID, 2, 720.0, 0.0, 0.0
GRID, 3, 360.0, 360.0, 0.0
GRID, 4, 360.0, 0.0, 0.0
GRID, 5, 0.0, 360.0, 0.0
GRID, 6, 0.0, 0.0, 0.0
CROD, 1, 10, 3, 5
CROD, 2, 10, 1, 3
CROD, 3, 10, 4, 6
   ...
CROD, 9, 10, 2, 3
CROD, 10, 10, 1, 4
PROD, 10, 2, 15.0
MATI, 2, 1.25E7, 0.3, 0.1, ..., 25000.0, -25000.0
SPCI, 1, 123456, 5, 6
SPCI, 1, 3456, 1, THRU, 4
FORCE, 1, 2, -1.0E5, 0.0, 1.0, 0.0
FORCE, 1, 4, -1.0E5, 0.0, 1.0, 0.0
CONVERT, MASS, 2.59E-3
EIGR, 1, GIV, 0.0, 700.0, 2, ..., A BC. +BC. MAX
MPPARM, ISCAL, 1
DESELM, 1, 1, CROD, 6.667E-3, 1000.0, 2.0, ROD1
DESELM, 2, 2, CROD, 6.667E-3, 1000.0, 2.0, ROD2
   ...
DESELM, 9, 9, CROD, 6.667E-3, 1000.0, 2.0, ROD9
DESELM, 10, 10, CROD, 6.667E-3, 1000.0, 2.0, ROD10
DCONSTR, 2, VMISES
DCONVM, 100, 2.5+4, -2.5+4, 2
DCONDS, 100, 1, UPPER, 2.0, POSNOD1, 1, 2, 1.0
DCONDS, 100, 2, UPPER, 2.0, POSNOD2, 2, 2, 1.0
   ...
DCONDS, 100, 8, LOWER, -2.0, NEGNOD1, 4, 2, 1.0
ENDDATA

Figure 2. Features of a Sample ASTROS Input Stream
1.3. THE ASSIGN DATABASE ENTRY — GENERAL

The first line of the ASTROS input data stream must be the **ASSIGN DATABASE** entry. This entry identifies the runtime database files to be used in the current ASTROS execution and specifies certain parameters associated with the files. The format of this entry is:

```
ASSIGN DATABASE <dbname> <password> <status> [params]
```

where,

- **dbname** is a name identifying the runtime database files (maximum of 8 characters or fewer, depending on the local host. Restricted under IBM/MVS DDNAME syntax to a maximum of 5 characters)
- **password** is a user assigned password for the database files (maximum of 8 characters)
- **status** is the status of the database files. Must be either **OLD**, **NEW** or **TEMP**
- **params** are optional (installation dependent) parameters e.g., `DBLKSIZ = n`, `IBLKSIZ = n`, etc.

The entries on the **ASSIGN DATABASE** command are **not** keyword controlled and so must be input in the order shown. They can be separated by any number of blank characters or commas but must reside on no more than 10 physical records of the input data stream. Note that two commas do not imply that a parameter is missing; instead, the second comma will be ignored. Two equivalent examples are shown below:

```
ASSIGN DATABASE, ASTSC, KIMBERLY, OLD
ASSIGN DATABASE ASTSC KIMBERLY OLD
```

Figure 3 is provided to illustrate the function of the **ASSIGN DATABASE** entry. In the case shown, the installation dependent parameter **VOL** has been implemented which allows selection of the physical device on which the requested file(s) DB1 reside.

The optional **params** on the **ASSIGN DATABASE** entry may or may not be keyword controlled and are installation dependent. They provide a mechanism for the user to direct machine or installation dependent file operations to be performed by the ASTROS procedure. At each site, the installation of the code involves a definition of these parameters and the form they must take. The ASTROS system is currently functional on numerous host systems including IBM 370 series, VAX/VMS, VAX/Ultrix, IBM/AIX, SGI 4D series and Crimson/Indigo series, HP/9000 720 and 730 series, CRAY/UNICOS, DEC-Station, Convex, and SunSparcstation. The availability on a specific computer may be obtained on request from the U.S. Air Force.

The next section documents the installation dependent **ASSIGN DATABASE** parameters for some of the more common features/hosts. These features, however, may be customized to a very high degree and may be modified by the local system manager. Further documentation of the **ASSIGN DATABASE** command is left to the local installation or will be included in the delivery material. The Programmer's Manual contains the detailed description of how these and other machine dependent parameters are defined. The following sections are optional. First-time readers may proceed to Chapters 2 or 3.
Figure 3. Function of the ASSIGN DATABASE Entry
1.3.1. ASSIGN DATABASE DESCRIPTIONS FOR HOST COMPUTERS

This section contains the descriptions of the machine and installation dependent parameters on the ASSIGN DATABASE entry for three machines on which ASTROS is currently functional. The parameters that are available at each site are listed and details of their use are presented. The user is cautioned that these are site dependent parameters which may be different for each installation even if the host system is the same. This documentation is provided both as an example to the system programmer and because the features that have been made available on these machines are very likely to exist on most machines that may be used. The user is referred to their ASTROS system manager for the particulars of the interface between the local host system and ASTROS.

The host systems documented in this section are:
(1) DEC VAX computers using VMS
(2) IBM 370/3090 computers using MVS
(3) Computers using Unix-based Operating Systems

1.3.1.1. VMS IMPLEMENTATION

The ASSIGN DATABASE entry supplies the ASTROS system with the root name of the database files, the status of those files, and a set of user parameters. The root name is limited to 8 characters in length, the status is one of NEW, OLD or TEMP and the set of user parameters can be any of the following keyword commands:

- **VOL-name**: Names the device and/or default directory on which the database files reside.
- **DBLKSIZB-n**: CADDB data files block size in words.
- **IBLKSIZZmn**: CADDB index file block size in words.
- **DELETE**: Denoting that the run-time database files are to be deleted after the execution.
- **KEEP**: Denoting that the run-time database files are to be returned after the execution.

**DCL Requirements**:

Under MicroVMS and VMS, the database files can be explicitly named in the FORTRAN open statement or an equivalence between a logical name and the true file name can be established using the DCL ASSIGN command:

```
$ ASSIGN <physical-name> <logical-name>
```

Both options can be used in the VMS version of ASTROS. The logical names `dbrootIDX` and `dbrootD01` are checked to see if they have been ASSIGNED. If so, these names are used in the Fortran OPEN and the ASSIGN provides the hook to the physical files. When no logical names exist, the filenames:

- `dbroot.idx`
- `dbroot.d01`
are used. When a VOL parameter is specified, a file name with the structure of a physical file name is used:

```
voldbroot.IDX
voldbroot.DO1
```

As an example, assuming that the logical names MULTIDX and MULTDO1 do not exist:

```
ASSIGN DATABASE MULT SHAZAM NEW
```

would use MULT.IDX and MULT.DO1 as file names in Fortran OPEN statements whereas:

```
ASSIGN DATABASE MULT SHAZAM NEW VOL = $DISK1:[SCR]
```

would use $DISK1:[SCR]MULT.IDX and $DISK1:[SCR]MULT.DO1 as filenames in the OPEN statements. Note that if the logical name is assigned, VOL will be ignored.

**TEMP Data Base Example:**

When the status is TEMP, a temporary database is created. Internally, the file is not named in the OPEN statement and has a status of SCRATCH. Generally, this means that the files will be created in the user's default directory. The only legal optional parameters are DBLSIZE and ISBLSIZE. In the example:

```
ASSIGN DATABASE TEST PASS TEMP
```

the database files will be created using the default block sizes and deleted upon termination of the execution.

**NEW Data Base Example:**

When the status is NEW, a new database is created. The filename will be formed from the root name and, if present, from the VOL option as discussed. Any of the user parameters may be used. As an example:

```
ASSIGN DATABASE TEST PASS NEW
```

will create either the files TEST.IDX and TEST.DO1 or, if the DCL contains the proper assignments:

```
$ ASSIGN SYSSCRATCH:TEST.IDX TESTIDX
$ ASSIGN SYSSCRATCH:TEST.DO1 TESTDO1
```

the files named in the DCL ASSIGN statements.

The DELETE option will cause the files to be deleted using the DISP='DELETE' option on the OPEN and CLOSE statements. Otherwise the files will be kept.

**OLD Data Base Example:**

When the status is OLD, an old database is used. The physical files that make up the database must exist. Only the VOL parameter is legal and the database files will be kept after the execution has terminated. In the example:
ASSIGN DATABASE TEST PASS OLD VOL = $DISK1:[DB]

the pre-existing database files

$DISK1:[DB]TEST.IDX
$DISK1:[DB]TEST.D01

will be used in the ASTROS execution.

1.3.1.2. IBM MVS IMPLEMENTATION

The ASSIGN DATABASE entry supplies the ASTROS system with the root name of the database files, the status of those files, and a set of user parameters. The root name is limited to 5 characters in length, the status is one of NEW, OLD or TEMP and the set of user parameters can be any of the following keyword commands:

- **DBLKSIZEn**: CADDB data files block size in words.
- **IBLKSIZEn**: CADDB index file block size in words.

**NEW Database Example:**

When the status is NEW, a new database is created. The NEW refers to the fact that the database itself is to be initialized. The physical files that make up the database may exist or be created with the proper DD card. Other legal parameters are DBLKSIZE and IBLKSIZE. Here is an example:

```plaintext
ASSIGN DATABASE RICH RICHPASS NEW
```

JCL should be as follows:

```plaintext
//RICHIDX DD DSN=RICHDB.INDEX,SPACE=(TRK,10),DISP=(,CATLG)
//RICHDOI DD DSN=RICHDB.DATA,SPACE=(TRK,50),DISP=(,CATLG)
```

In this example a database with files **RICHDB.INDW** and **RICHDB.DA** will be created.

**JCL Requirements:**

On the IBM computer the user must supply the appropriate JCL cards for each database file. The DDNAMEs for these cards are built from dbroot root file name as follows:

- **dbrootIDX**: index file
- **dbrootD01**: First data file
- **dbrootD02**: Second data file

The only required DD parameters for new files are SPACE and DISP. The user can control the number of data files by including the desired number of DD statements.
**TEMP Database Example:**

When the status is TEMP, a temporary database is created. Internal to the database, there is no difference between a TEMP and a NEW database on the IBM. The only difference is in the JCL where the user supplies a temporary file instead of a permanent one. The only legal optional parameters are DBLKSZ and IBLKSZ. Here is an example:

```
ASSIGN DATABASE RICH RICHPASS TEMP DBLKSZ=2048 IBLKSZ=256
```

JCL should be as follows:

```
//RICHIDX DD DSN=&INDEX,SPACE=(TRK, 10)
//RICHDO1 DD DSN=&DATA,,SPACE=(TRK, 50)
```

**OLD Database Example:**

When the status is OLD, an old database is used. The physical files that make up the database must exist. No other parameters are legal. Here is an example:

```
ASSIGN DATABASE RICH RICHPASS OLD
```

The JCL should then be as follows:

```
//RICHIDX DD DSN=RICHDB.INDEX,DISP=OLD
//RICHDO1 DD DSN=RICHDB.DATA,DISP=OLD
```

### 1.3.1.3. UNIX System Implementation

The `ASSIGN DATABASE` entry supplies the ASTROS system with the root name of the database files, the status of those files, and a set of user parameters. The root name is limited to 8 characters in length, the status is one of **NEW**, **OLD** or **TEMP** and the set of user parameters can be any of the following keyword commands:

- **VOL=** Name
  - Names the disk volume (directory) on which the database files reside. **VOL** and **DIR** are synonymous and mutually exclusive.

- **DIR=** Name
  - Names the disk volume (directory) on which the database files reside. **VOL** and **DIR** are synonymous and mutually exclusive.

- **DBLKSZ=** N
  - CADDB data files block size in words.

- **BLKSZ=** N
  - CADDB index file block size in words.

- **DELETE**
  - Denoting that the run-time database files are to be deleted after the execution.

- **KEEP**
  - Denoting that the run-time database files are to be returned after the execution.

**TEMP Database Example:**

When the status is TEMP a temporary database is created and no data is kept after the run. The only legal optional parameters are **DBLKSZ** and **IBLKSZ**.
ASSIGN DATABASE RICHDB RICHPW TEMP DBLKSIZEx2048 IBLKSIZEx256

**NEW Data Base Example:**

When the status is NEW, a new database is created. If the files already exist, they will be overwritten.

- voldbroot.IDX - index file
- voldbroot.DO1 - data file 1

The VOL parameter and the root database name are used to form the names. Other legal parameters are DBLKSIZEx and IBLKSIZEx.

ASSIGN DATABASE RICHDB RICHPASS NEW DELETE, VOL=/tmp/

In this example, a database with files /tmp/RICHDB.IDX and /tmp/RICHDB.DO1 will be created. If existing files are found they will be overwritten.

**OLD Data Base Example:**

When the status is OLD, an old database is used. The physical files that make up the database must exist. Two files are used to store the database. The names of these files are as follows:

- voldbroot.IDX - index file
- voldbroot.DO1 - data file 1

The VOL parameters and the root database name are used to form the names. No other parameters are legal. Here is an example:

ASSIGN DATABASE RICHDB RICHPASS OLD VOL=/home/rich/

In this example a database with files /home/rich/RICHDB.IDX and /home/rich/RICHDB.DO1 will be used.
1.4. **THE INCLUDE DIRECTIVE**

The input data stream typically resides in a single file, but the user can direct the input stream interpreter to include other files through the use of the `INCLUDE` directive in the primary input stream. The format of the `INCLUDE` command is:

```
#include <filename>
```

where,

- `filename` is a name identifying the file to be included (maximum of 72 characters).

The filename, which is used in a FORTRAN `OPEN` statement, must satisfy the requirements of the particular host system for file names. Beyond this restriction, the user is free to have any set of contiguous non-blank characters in the filename. In order to avoid the possibility of an infinite recursion, there is a restriction on the include feature that no `INCLUDE` statement can appear in a file that is being included. For example, if the file "TENBAR" is being included, it may not itself contain an `INCLUDE` directive. The input stream interpreter will terminate with an appropriate error message should this occur.

The `INCLUDE` directive can appear anywhere in the input stream after the `ASSIGN DATABASE` entry. The `ASSIGN DATABASE` must always appear as the first non-blank line in order to allow the use of the run time database in the subsequent input stream interpretation. A single data packet can be split among included files or an `INCLUDE` file may contain parts of multiple data packets. The input interpreter merely replaces the `INCLUDE` directive with the data contained in the named file so the only requirement is that the input stream that results from the combination of all `INCLUDEs` have the form of a normal input stream. The `INCLUDE` feature can be very useful in certain circumstances. For example, a special user developed MAPOL sequence can be stored and maintained external to the files containing the engineering data for particular runs; or, conversely, the bulk data representing a large model can be included into the file containing the solution control directives.
1.5. THE DEBUG PACKET

The debug packet represents a development tool and is intended to be used primarily by those responsible for maintaining the software. The debug packet provides the system programmer with the means to invoke or control certain executive and database management system functions that are helpful in tracking the ASTROS execution and/or testing the executive and database management system software. However, because some of the debug options can be useful to the general user, the debug packet is fully documented in the User's Manual rather than in the Programmer's Manual. This section documents each of the debug options and indicates how the option can be useful in debugging the ASTROS procedure. Emphasis is placed on those debug options that are of interest to the general user.

The debug packet is initiated by the keyword DEBUG, which must appear alone on the line of the input stream that follows the ASSIGN DATABASE entry and that precedes any other data packet. Following the initiator, any number of debug lines can be included in the data stream. Each debug command line can be composed of a number of debug commands, appearing in any order, separated by blanks or commas. The DEBUG packet is terminated when a new data packet initiator, or the end of the input stream, is encountered. Most debug commands consist of single keywords which toggle flags activating the debug functions. The appearance of these debug keywords is all that is required to activate the option. Other debug commands select that a flag take on a particular value. These commands have the form:

\[ <\text{command}> = <\text{value}> \]

There can be any number of blanks between the end of the command keyword, the value and the equal sign, but neither the command nor the value can contain imbedded blanks. Any errors in the DEBUG packet input will result in warnings but will not terminate the execution and the erroneous command will be ignored.

The tables shown in this section list the keywords that can be included in the DEBUG packet. The debug commands are grouped into executive system and database management system debugs. Each of these groups is described in greater detail in the following sections.
1.5.1. EXECUTIVE SYSTEM DEBUG COMMANDS

The first four executive system keywords are intended to assist the system programmer in following the actions of the MAPOL compiler and execution monitor. The options are shown in Table 2.

As such, they are of limited value to the general user. The MATRIX option, however, can be useful in tracking the execution of the MAPOL program. It echoes the matrix utility calls for all matrix operations that are in the MAPOL sequence. For example, if the MAPOL program includes the expression:

\[ [A] := \text{TRANS}([B]) \times [C] + [D]; \]

the MATRIX trace echoes the resultant call to the MPYAD large matrix utility with the arguments shown in detail. This trace can be very useful in determining which particular MAPOL instruction is being executed when a problem occurs. Large MAPOL programs with many loops and a large number of matrix expressions can be debugged quite simply using the MATRIX trace. All MAPOL statements that result in calls to any of the large matrix utilities, such as PARTN, MERGE, MPYAD, and MXADD are echoed.

LOGBEGIN and LOGMODULE provide expanded echoes of the module timing summary that is found at the end of each ASTROS output file. When problems cause early termination of the job, these options provide the name of the last module entered prior to the failure. This provides a starting point to diagnose the problem.

---

<table>
<thead>
<tr>
<th>KEYWORD</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSTACK</td>
<td>MAPOL compiler stack output</td>
</tr>
<tr>
<td>MEXEC</td>
<td>MAPOL execution debug flag</td>
</tr>
<tr>
<td>MOBJ</td>
<td>MAPOL object code debug packet</td>
</tr>
<tr>
<td>MTRACE</td>
<td>MAPOL trace debug output</td>
</tr>
<tr>
<td>MATRIX</td>
<td>MAPOL peeper matrix operation trace</td>
</tr>
<tr>
<td>LOGMODULE</td>
<td>Expanded log entries for each module</td>
</tr>
<tr>
<td>LOGBEGIN</td>
<td>Beginning entries for each module in log file</td>
</tr>
</tbody>
</table>
1.5.2. DATABASE AND MEMORY MANAGER DEBUG COMMANDS

The database management system has a number of debug options which can be divided into three categories: trace options, control options and memory manager options. These are shown in Table 3.

The first group of database debug commands contain two tracing options: TRACEx and IOSTAT. The IOSTAT keyword selects either a FULL tracing or a SUMMARY. The first of these options and the IOSTAT=FULL option are further controlled by the ENTITY option which completes the first group of keywords. Note: the tracing keywords generate an overwhelming amount of data which are often of limited use unless the user is familiar with the internal structure of the database files. The ENTITY keyword limits the activation of the tracing options to those times when the named database matrix, relation or unstructured entity is open. If no entity specification is made, the traces are active for all database operations. In addition to their role in debugging the database software, the trace options provide a useful means of debugging the interface between a user written module and the database.

The database control options CALLSTAT and NOCOREDIR provide user control over two internal database functions. The CALLSTAT option compiles a summary of the number of calls made to each database subroutine. This summary, in combination with the IOSTAT option, provides statistics on the number of database operations in the execution. The NOCOREDIR option is made available for machines with limited core memory resources. If NOCOREDIR is selected, the database manager stores the database directories on the database files rather than in core. This can substantially reduce the database memory requirements at the cost of increasing the number of input/output operations.

Table 3. Database Debug Commands

<table>
<thead>
<tr>
<th>KEYWORD</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRACE</td>
<td>Traces all database CALLs</td>
</tr>
<tr>
<td>IOSTAT=pree</td>
<td>Database I/O tracing</td>
</tr>
<tr>
<td>FULL</td>
<td>Full trace of I/O activity</td>
</tr>
<tr>
<td>SUM</td>
<td>Summary of I/O activity</td>
</tr>
<tr>
<td>ENTITY=name</td>
<td>Restricts tracing to entity name</td>
</tr>
<tr>
<td>CALLSTAT</td>
<td>Compiles statistics on the number of calls to each database routine at the end of a job.</td>
</tr>
<tr>
<td>NOCOREDIR</td>
<td>Turns off the option to store directories in core</td>
</tr>
<tr>
<td>NODELAYCRE</td>
<td>Turns off the option that delays entity creation until the entity is opened</td>
</tr>
<tr>
<td>MEMORY</td>
<td>Memory manager debug print</td>
</tr>
</tbody>
</table>
The last group of database debug options consists of the \texttt{MEMORY} command. This option causes an echo of all the memory management calls made in the modules. The user can then track the ASTROS execution into the engineering modules themselves. In addition to the echo, the \texttt{MEMORY} option invokes a checksum operation which checks for the integrity of the memory block headers on every memory manager operation. If the checksum fails, a message is written to the effect that a block header has been overwritten. This option is very effective in uncovering errors in engineering modules that make use of dynamic memory allocation.

1.5.3. \textbf{INTERMEDIATE RESULTS PRINTING COMMANDS}

Many of the ASTROS engineering modules have intermediate output print options that are useful in tracing the details of an analysis or in reviewing the quality of the inputs. These many options are listed in Table 4.

\begin{table}[h]
\centering
\begin{tabular}{|c|p{10cm}|}
\hline
\textbf{KEYWORD} & \textbf{DESCRIPTION} \\
\hline
\texttt{STEADY} & Steady preface \texttt{USSAERO} output \\
\texttt{STEADY} & Prints steady aerodynamic model geometry \\
\texttt{STEADY} & Includes stability coefficient data \\
\texttt{STEADY} & Includes pressure data \\
\texttt{STEADY} & Includes velocity components and matrix output \\
\hline
\texttt{STEADYNP} & Nonplanar preface \texttt{USSAERO} output \\
\texttt{STEADYNP} & Prints steady aerodynamic model geometry \\
\texttt{STEADYNP} & Includes stability coefficient data \\
\texttt{STEADYNP} & Includes pressure data \\
\texttt{STEADYNP} & Includes velocity components and matrix output \\
\hline
\texttt{AMP} & Intermediate unsteady Aero matrices \\
\texttt{AMP} & Prints the \texttt{SKJ} matrix and, if only one group, includes \texttt{AJJ}, \texttt{QKJ} and \texttt{QJJ} if they exist \\
\texttt{AMP} & Includes \texttt{D1JK}, \texttt{D2JK} and \texttt{AJJT} matrices \\
\hline
\texttt{FLUTTRAN} & Additional flutter eigenextraction information \\
\texttt{FLUTTRAN} & Prints the number of iterations required to find each flutter root \\
\texttt{FLUTTRAN} & Includes the estimated roots for each iteration \\
\hline
\end{tabular}
\caption{Intermediate Results Debug Commands}
\end{table}
Table 4. Intermediate Results Debug Commands - Continued

<table>
<thead>
<tr>
<th>KEYWORD</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>SKEVAL=n</td>
<td>Prints additional constraint data</td>
</tr>
<tr>
<td>0</td>
<td>Prints the stress or strain components, the constraint type and the constraint value for each constrained element/layer.</td>
</tr>
<tr>
<td>1</td>
<td>Prints initial design information</td>
</tr>
<tr>
<td>2</td>
<td>Includes function values at each iteration</td>
</tr>
<tr>
<td>3</td>
<td>Includes internal Microdot parameters</td>
</tr>
<tr>
<td>4</td>
<td>Includes search directions</td>
</tr>
<tr>
<td>5</td>
<td>Includes gradient information</td>
</tr>
<tr>
<td>6</td>
<td>Includes scaling information</td>
</tr>
<tr>
<td>7</td>
<td>Includes one-dimensional search information</td>
</tr>
<tr>
<td>MKUSET=n</td>
<td>MKUSET redundant set warnings</td>
</tr>
<tr>
<td>0</td>
<td>Prints warning messages if the same degree of freedom is placed in a set more than once.</td>
</tr>
<tr>
<td>VANGO=n</td>
<td>VANGO algorithm output</td>
</tr>
<tr>
<td>1</td>
<td>Prints all output not controlled by OCPARM bulk data entry.</td>
</tr>
<tr>
<td>OCAPPROX=n</td>
<td>VANGO use of approximate problem</td>
</tr>
<tr>
<td>0</td>
<td>VANGO iterates to convergence on the approximate problem rather than making one design variable change and then requiring a complete reanalysis.</td>
</tr>
<tr>
<td>OCINVERSE=n</td>
<td>VANGO use of inverse design variables</td>
</tr>
<tr>
<td>0</td>
<td>VANGO uses inverse design variables when there are no shape function global variables. Using this with the OCAPPROX debug makes VANGO behave exactly like DESIGN in its treatment of the approximate problem. The only difference is that OC methods are used to update (v).</td>
</tr>
<tr>
<td>SAROGEOM=n</td>
<td>Planar steady aerodynamics geometry option</td>
</tr>
<tr>
<td>0</td>
<td>STEADY stops the execution of ASTROS after the steady aerodynamic geometry has been computed. No printed output is generated unless the STEADY debug is also used.</td>
</tr>
<tr>
<td>KEYWORD</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>NPSGEOM=n</td>
<td>Nonplanar steady aerodynamics geometry option</td>
</tr>
<tr>
<td>&gt;0</td>
<td>STDYNP stops the execution of ASTROS after the nonplanar steady aerodynamic geometry has been computed. No printed output is generated unless the STEADYNP debug is also used.</td>
</tr>
<tr>
<td>MPSCAL=opt</td>
<td>Controls scaling of design variables</td>
</tr>
<tr>
<td>ON</td>
<td>Scales global variables to unity before Microdot is invoked (Default).</td>
</tr>
<tr>
<td>OFF</td>
<td>Does not scale variables.</td>
</tr>
<tr>
<td>PLYDATA=opt</td>
<td>Selects the output of PSHELL/MAT2 Bulk Data entries for elements defined by PCOMP1 data.</td>
</tr>
<tr>
<td>PRINT</td>
<td></td>
</tr>
<tr>
<td>PUNCH</td>
<td></td>
</tr>
<tr>
<td>BOTH</td>
<td></td>
</tr>
</tbody>
</table>
1.5.4. SEQUENCER INTERMEDIATE PRINT COMMANDS

There are a number of print and control options for the grid point sequencer that are shown in Table 5.

<table>
<thead>
<tr>
<th>TABLE 5. SEQUENCER DEBUG COMMANDS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>KEYWORD</strong></td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>SEQPRINT-opt</td>
</tr>
<tr>
<td>DETAIL</td>
</tr>
<tr>
<td>DIAGNOSTIC</td>
</tr>
<tr>
<td>SEQMETH-meth</td>
</tr>
<tr>
<td>CM</td>
</tr>
<tr>
<td>GPS</td>
</tr>
<tr>
<td>ALL</td>
</tr>
<tr>
<td>NOSEQMPC</td>
</tr>
<tr>
<td>SEQCRT=crit</td>
</tr>
<tr>
<td>BAND</td>
</tr>
<tr>
<td>PROF</td>
</tr>
<tr>
<td>RMS</td>
</tr>
<tr>
<td>WAV</td>
</tr>
<tr>
<td>SEQPUNCH</td>
</tr>
<tr>
<td>SEQOFF</td>
</tr>
</tbody>
</table>
2. THE EXECUTIVE SYSTEM AND MAPOL

The ASTROS system is an executive controlled software procedure. One of the functions of the ASTROS executive system, described in detail in Reference 1, is to determine the sequence in which the modules of the procedure are invoked. For ASTROS, the Mat. Analysis Problem Oriented Language (MAPOL) has been developed to perform this executive system task. The MAPOL language has its conceptual roots in the Direct Matrix Abstraction Program (DMAP) capability developed for the NASTRAN structural analysis system (Reference 2). The DMAP "language," used to create NASTRAN's solution algorithms is very crude; however, it has been a major factor in extending the life cycle of the software. It provided a simple method of installing new code and functional capabilities into the system and afforded the user an opportunity to interact with the software. MAPOL provides the same advantages to the ASTROS system and represents a considerable advance over DMAP in that MAPOL is a structured, procedural language that directly supports high order matrix operations, manipulation of data base entities and complex data types. Moreover, the syntax of the language looks much like that of any scientific programming language and so is easily learned by anyone who knows FORTRAN or PASCAL.

From the user's point of view, ASTROS is directed by a sequence of control statements "coded" in the MAPOL language just as a NASTRAN rigid format is coded in the DMAP "language." As with the NASTRAN Rigid Formats, the majority of users will use the standard MAPOL sequence. This is the default, and as such it requires no special action. Advanced users may optionally edit the standard sequence or write their own "program". The methods used to do this are described in this Chapter. Because changes to the executive system are an advanced topic, first-time users may proceed directly to Chapter 3.

The executive system within ASTROS compiles the MAPOL program and executes the resultant "ASTROS machine code" which directs the execution of the ASTROS procedure. (Note that ASTROS is NOT written in MAPOL, only the executive control algorithm is written in the MAPOL language. In fact, ANSI standard FORTRAN was used to write the compiler for MAPOL and for all the engineering software of the ASTROS system.) MAPOL allows the user to manipulate the software system in many ways to tailor the available capabilities to perform particular tasks. At a higher level of sophistication, the user may add modules to the system or replace modules that already exist. Obviously, some of these features require a knowledge of the ASTROS system that is beyond the scope of the User's Manual. Those features that require detailed information are more fully discussed in the Programmer's Manual but their existence is emphasized here in order to introduce the user to the flexibility that the executive system provides.

This section serves two functions. First, it presents the mechanics of the MAPOL packet and, second, it presents the standard MAPOL sequence that has been developed to direct the optimization and analysis tasks for which ASTROS has been designed. The potential of the executive system to tailor the
ASTROS procedure will be explored in this discussion of the standard sequence. In addition to this Chapter, Chapter 6 presents a detailed description of the MAPOL language, its syntax and features. Included in this chapter is a listing of the most recent ASTROS standard executive sequence. It cannot be overemphasized that, while the capabilities implemented in the ASTROS software are significant, the true power embodied in the ASTROS system is its immense flexibility, largely provided by the executive system and its MAPOL language.

The MAPOL packet is initiated either by the keyword MAPOL or by the keyword EDIT and is terminated upon encountering the SOLUTION CONTROL packet, the BULK DATA packet or the end of the input stream. In addition, each of the initiator keyword commands act as directives to the MAPOL compiler to take specific actions. The exact form of the MAPOL and EDIT commands is:

\[
\text{MAPOL} \begin{bmatrix} \text{GO} \end{bmatrix} \begin{bmatrix} \text{LIST} \end{bmatrix} \begin{bmatrix} \text{NOGO} \end{bmatrix} \begin{bmatrix} \text{NOLIST} \end{bmatrix}
\]

\[
\text{EDIT} \begin{bmatrix} \text{GO} \end{bmatrix} \begin{bmatrix} \text{LIST} \end{bmatrix} \begin{bmatrix} \text{NOGO} \end{bmatrix} \begin{bmatrix} \text{NOLIST} \end{bmatrix}
\]

where:

\[
\begin{bmatrix} \text{GO} \end{bmatrix} \begin{bmatrix} \text{NOGO} \end{bmatrix} \quad \text{selects whether the MAPOL program is to be executed after compilation.}
\]

\[
\begin{bmatrix} \text{LIST} \end{bmatrix} \begin{bmatrix} \text{NOLIST} \end{bmatrix} \quad \text{selects whether the MAPOL source code is to be written to the output file.}
\]

The MAPOL command is followed by a MAPOL program which can be any syntactically complete set of MAPOL statements as described in the chapter on MAPOL Programming (Chapter 6). The EDIT command indicates that the MAPOL packet will consist of edit commands that INSERT, DELETE or REPLACE lines of the standard executive sequence.

2.1. THE MAPOL PROGRAM

If the MAPOL packet begins with the MAPOL command line, the compiler assumes that the remaining statements in the packet constitute a complete MAPOL program. That program can be any set of MAPOL statements that satisfy the rules of the language as presented in Chapter 6. The program can call any of a number of intrinsic functions (including most of the common FORTRAN intrinsic functions) and any of the "engineering" utilities and modules that have been defined to the compiler. The user can access these modules in any desired order, subject only to limits imposed by the engineering modules themselves. In addition, the user can write special purpose modules and define them to the compiler through the SYSTEM GENERATION (SYSGEN) program discussed in the Programmer’s Manual. Thus, a wide range of tasks can be performed using the ASTROS system in combination with a user’s MAPOL program.

The MAPOL language can be read and written easily by anyone familiar with a scientific programming language. This feature opens the advantages of the executive system to the average user without requiring specialized knowledge in computer science or requiring effort to learn a radically
different programming language. The user will often find the simplicity and power of the MAPOL language enables many tasks to be performed using the ASTROS system that are not explicitly supported in the standard executive sequence.

2.2. MAPOL EDIT COMMANDS

If the MAPOL packet begins with the EDIT command line, the compiler assumes that the remainder of the packet (if any) is composed of MAPOL edit commands and new MAPOL statements that modify the standard executive sequence. The set of edit commands is given in Table 6 (and in Chapter 6). They allow the user to insert, delete and replace lines of the standard MAPOL sequence. All the edit commands reference a line number or range of line numbers. The line numbers are those in a compiled listing of the standard MAPOL sequence which is written as part of the system generation task. When editing the standard sequence, the user is cautioned to obtain the most recent listing either from the SYSGEN output or by executing ASTROS with an input stream containing only an ASSIGN DATABASE entry and the one line MAPOL packet:

```
EDIT LIST NOGO
```

This input stream will result in an output file containing the current listing of the standard executive sequence.

Table 6. MAPOL Edit Commands

<table>
<thead>
<tr>
<th>STATEMENT</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDIT</td>
<td>Modify the standard solution</td>
</tr>
<tr>
<td>DELETE a[,b]</td>
<td>Remove lines a through b inclusive</td>
</tr>
<tr>
<td>REPLACE a[,b]</td>
<td>Removes lines a through b inclusive and replaces them with the following lines</td>
</tr>
<tr>
<td>INSERT a</td>
<td>Insert the lines following the command after line a</td>
</tr>
</tbody>
</table>

2.3. THE STANDARD EXECUTIVE SEQUENCE

As previously mentioned, the MAPOL language has its conceptual roots in the DMAP "language". In order to allow the user of NASTRAN to perform certain predefined analyses, a set of "rigid formats" or DMAP algorithms were written, alleviating the user of the need to learn the details of the control language. Each rigid format allowed the user to perform analyses in a different engineering discipline; for example, static structural analyses, normal modes analyses, or transient analyses. In a similar manner, a standard executive sequence or MAPOL algorithm has been developed for the ASTROS system which supports all the engineering disciplines and optimization features of the procedure. Unlike the multiple DMAP rigid formats, however, there is a single MAPOL sequence that supports all the available engineering disciplines as well as optimization. This fundamental difference is necessary to permit multidisciplinary optimization.
One consequence of having a single multidisciplinary algorithm is that the standard sequence appears to be very complicated. The purpose of this subsection is to present the internal structure and flow of the standard MAPOL sequence thereby providing the user with sufficient information to tailor the standard sequence to suit individual needs. The discussion in this section will be general in order to provide the necessary overview and to introduce the concepts embodied in the standard sequence. Modifications to the standard sequence will be presented primarily in terms of capabilities but the presentation will be supported by examples that represent both simple and more complex modifications. Finally, the Chapter closes with a detailed line-by-line presentation of the standard executive sequence. The reader is also referred to the Programmer's Manual for information on the addition of modules to the ASTROS engineering library.

2.4. STANDARD EXECUTIVE SEQUENCE STRUCTURE

The standard MAPOL sequence consists of two major components: the variable declarations and the solution algorithm. The solution algorithm can be further divided into preface modules, the optimization segment and the final analysis segment. The declaration segment declares all variables used in the MAPOL sequence. This includes all integer and real scalar variables as well as high order variables: relations, matrices and unstructured data base entities. Within the solution algorithm, the preface modules comprise a group of engineering modules exercised prior to the boundary condition loops to perform a number of system initialization tasks; e.g. loads generation and the computation of invariant aerodynamic matrices. The separate optimization and analysis segments consist of a loop on the number of (optimization or analysis) boundary conditions in the current execution. In the optimization segment, a second boundary condition loop is performed to obtain the sensitivities of active boundary condition dependent constraints in preparation for the optimization task.

Figure 4 provides the standard algorithm structure showing how multidisciplinary optimization is performed in ASTROS. It is readily apparent that the structure of the standard MAPOL sequence has been determined by the requirement to perform multidisciplinary optimization. Each of the segments of the standard sequence are discussed in greater detail in the following subsections.
<table>
<thead>
<tr>
<th>PREFACE SEGMENT</th>
<th>Initialization (PREFACE) Segment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>WHILE NOT CONVERGED DO</strong></td>
</tr>
<tr>
<td><strong>ANALYSIS PHASE</strong></td>
<td><strong>For Each Boundary Condition Do</strong></td>
</tr>
<tr>
<td></td>
<td>Discipline 1 Constraints</td>
</tr>
<tr>
<td></td>
<td>Subcase 1 Constraints</td>
</tr>
<tr>
<td></td>
<td>Subcase 2 Constraints</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>Discipline 2 Constraints</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td><strong>OPTIMIZATION SEGMENT</strong></td>
<td><strong>End Do</strong></td>
</tr>
<tr>
<td><strong>SENSITIVITY PHASE</strong></td>
<td><strong>Select Active Constraints</strong></td>
</tr>
<tr>
<td></td>
<td><strong>For Each Active Boundary Condition Do</strong></td>
</tr>
<tr>
<td></td>
<td>Active Discipline 1</td>
</tr>
<tr>
<td></td>
<td>Active Subcase 1 Constraint Sensitivities</td>
</tr>
<tr>
<td></td>
<td>Active Subcase 1 Constraint Sensitivities</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>Active Discipline 2</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td><strong>OPTIMIZATION PHASE</strong></td>
<td><strong>End Do</strong></td>
</tr>
<tr>
<td><strong>FINAL ANALYSIS SEGMENT</strong></td>
<td><strong>Redesign Based on Current Active Constraints and Constraint Sensitivities</strong></td>
</tr>
<tr>
<td><strong>END DO</strong></td>
<td></td>
</tr>
<tr>
<td><strong>FOR EACH Boundary Condition Do</strong></td>
<td></td>
</tr>
<tr>
<td>Discipline 1</td>
<td></td>
</tr>
<tr>
<td>Subcase 1</td>
<td></td>
</tr>
<tr>
<td>Subcase 2</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>Discipline 2</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 4. Structure of the Standard MAPOL Sequence**
2.4.1. MAPOL Declarations

MAPOL is a strongly typed language that requires all variables used in a program unit (either the main program or a procedure) to be declared. This applies to both simple variables like real and integer scalar or array variables and to high order variables (like MATRIX) that refer to data base entities. The first several hundred lines of the standard sequence consist solely of these variable declarations. Tables 7 through 12 give a summary of the scalar parameters used in the standard MAPOL sequence. These parameters, set in engineering modules or in the MAPOL sequence, are used as logic control flags and/or arguments to the engineering modules. The tables, which are separated by function, provide a brief description of each variable and a list of modules (where applicable) that use the parameter. For a description of all the variables used as arguments of the engineering modules, the reader is referred to the ASTROS Programmer's Manual. It should be noted that all of these variables can be directly modified within the MAPOL algorithm at the user's discretion. A discussion of those parameters that the user is most likely to want to modify is given in Section 2.4.3, but the experienced user is free to change any variable in the MAPOL sequence.

Higher order variables fall into two categories: MAPOL entities and hidden entities. MAPOL entities are those that actually appear in the MAPOL sequence while hidden entities are those that are declared but do not subsequently appear in the sequence. Their declaration ensures that the corresponding data base entity is created and can be used by a number of engineering modules without requiring the entity name to appear in the argument list. Hidden entities are typically those that contain the raw data needed by many modules; e.g. bulk data, geometry data and connectivity data. The declarations of the higher order variables are arranged to place logically related entities together. Several of the matrix entities, it should be noted, are subscripted, for example [KLLINV(1000)]. The subscripted matrix entity allows the ASTROS procedure to perform multiple analyses in several boundary conditions and retain the information needed to compute the sensitivities of the active constraints retained from each of these boundary conditions. The ASTROS executive system generates a name for each subscripted variable and that name is used by all the engineering modules receiving the subscripted entity name as an argument. The actual data base entity name need not be known. This does, however, impose the following restriction: a subscripted entity may not be used as a hidden entity in any engineering module; it must appear in the calling list for the module because only the executive system knows the actual name of the data base entity corresponding to the current subscript value. In the standard sequence, provision has been made for up to 1000 entities (doubly subscripted arrays of entities are set up for 30 boundary conditions and 33 secondary subscript values), but the user can change the declared number of subscripts to match the required range of indices.
<table>
<thead>
<tr>
<th>PARAMETER NAME</th>
<th>USED IN MODULES</th>
<th>DESCRIPTION</th>
<th>DEFAULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPHA</td>
<td>SOLUTION, FSD</td>
<td>Exponential move limit for fully stressed design. Set through the Solution Control OPTIMIZE command.</td>
<td>0.90</td>
</tr>
<tr>
<td>BQDP</td>
<td>BLASTFIT</td>
<td>Dynamic pressure value used in the current nuclear blast subcase. Output from BLASTFIT and used subsequently in MAPOL expressions.</td>
<td></td>
</tr>
<tr>
<td>CNVRGLIM</td>
<td>DESIGN, FSD, VANGO</td>
<td>Convergence test limit specifying the maximum percent objective change for the appropriate problem to be considered converged. Output from SOLUTION.</td>
<td>0.50</td>
</tr>
<tr>
<td>CTL</td>
<td>ACTCON, DESIGN, FSD, VANGO</td>
<td>Criteria for defining a constraint to be active in determining convergence in ACTCON. If value &gt; CTL, the constraint is active. Set in DESIGN, VANGO or FSD.</td>
<td></td>
</tr>
<tr>
<td>CTLMIN</td>
<td>ACTCON, DESIGN, FSD, VANGO</td>
<td>Criteria for denoting a constraint to be feasible in determining convergence in ACTCON. If maximum constraint value &lt; CTLMIN, the design is feasible. Set in DESIGN, VANGO, or FSD.</td>
<td></td>
</tr>
<tr>
<td>EPS</td>
<td>SOLUTION, ACTCON</td>
<td>Criteria used in ACTCON for selecting active constraints. All constraints with values greater than EPS will be retained. Set through the Solution Control OPTIMIZE command. (See also NRFAC)</td>
<td>-0.10</td>
</tr>
<tr>
<td>FMAX</td>
<td>GRD1, GRD2</td>
<td>The maximum frequency value associated with the HEIV eigenvalues computed for dynamic reduction in the current boundary condition. Output from GRD1.</td>
<td></td>
</tr>
<tr>
<td>MACH</td>
<td>SAERODRV</td>
<td>Mach number for the current case. Set in SAERODRV.</td>
<td></td>
</tr>
<tr>
<td>MOVLM</td>
<td>SOLUTION, DESIGN, FSD, TCEVAL</td>
<td>A move limit applied to the physical design variable (v) for mathematical programming methods. The move is: V/MOVLM &lt; V &lt; V * MOVLM. Set through the Solution Control OPTIMIZE command.</td>
<td>2.0</td>
</tr>
<tr>
<td>NRFAC</td>
<td>SOLUTION, ACTCON</td>
<td>Criteria used in ACTCON for selecting active constraints. At least NRFAC times MDV (see Table 5) constraints will be retained. Set through the Solution Control OPTIMIZE command. (See also EPS)</td>
<td>3.0</td>
</tr>
<tr>
<td>OCMOVLM</td>
<td>SOLUTION, VANGO</td>
<td>A move limit applied to the physical design variable (v) for the optimality criteria method. See MOVLM.</td>
<td>10⁻⁵</td>
</tr>
<tr>
<td>QDP</td>
<td>SAERODRV, SAERO, others</td>
<td>Dynamic pressure value used in the current steady aeroelastic subcase. Output from SAERODRV used subsequently in MAPOL expressions and modules.</td>
<td></td>
</tr>
<tr>
<td>WINDOW</td>
<td>TCEVAL</td>
<td>The window in which the MOVLM bound is overridden to allow local variables to change sign. If WINDOW is 0.0, then the local variable may not change sign. If it is nonzero, the half-width of a band around zero, called EPS, is computed by:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>EPS = WINDOW/100 * MAX(ABS(TMIN), ABS(TMAX))</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the local variable falls within the band, then the new minimum or maximum for the current iteration is changed to lie on the other side of zero from the local variable. Output from SOLUTION.</td>
<td></td>
</tr>
</tbody>
</table>

29
## Table 8. Integer Modelling Parameters

<table>
<thead>
<tr>
<th>NAME</th>
<th>MODULES</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASIZE</td>
<td>GDR3</td>
<td>The number of a-set degrees of freedom.</td>
</tr>
<tr>
<td>BC</td>
<td>N/A</td>
<td>Boundary condition loop counter.</td>
</tr>
<tr>
<td>DDFLG</td>
<td>DDLOAD</td>
<td>Indicates if the current statics subcases contain design dependent (gravity or thermal) loads. Output by DDLOAD.</td>
</tr>
<tr>
<td>HSIZE</td>
<td>BCBGPDT others</td>
<td>The number of extra points in the boundary condition.</td>
</tr>
<tr>
<td>GNORM</td>
<td>GDR3</td>
<td>The sum of LJSAT and LKSET.</td>
</tr>
<tr>
<td>GSIZEB</td>
<td>IFF others</td>
<td>The number of structural degrees of freedom in the model. Output from IFF and subsequently used in many modules.</td>
</tr>
<tr>
<td>GSIZE</td>
<td>GDR4 others</td>
<td>GSIZEB modified subject to dynamic reduction.</td>
</tr>
<tr>
<td>HSIZE</td>
<td>FLUTTRAN others</td>
<td>Number of eigenvectors extracted by the NEIG module. Set in NEIG.</td>
</tr>
<tr>
<td>LJSAT</td>
<td>GDR1</td>
<td>Number of degrees of freedom in the j-set in dynamic reduction. Set in GDR1.</td>
</tr>
<tr>
<td>LKSET</td>
<td>GDR1</td>
<td>Number of degrees of freedom in the j-set in dynamic reduction. Set in GDR1.</td>
</tr>
<tr>
<td>LSIZE</td>
<td>GDR1</td>
<td>The number of l-set degrees of freedom.</td>
</tr>
<tr>
<td>REIV</td>
<td>GDR1</td>
<td>An output from GDR1 indicating the number of eigenvalues below the maximum frequency specified for dynamic reduction.</td>
</tr>
<tr>
<td>NDR     BOUND</td>
<td></td>
<td>Logical flag equal to negative one if dynamic reduction is selected for the current boundary condition.</td>
</tr>
<tr>
<td>NMPC</td>
<td>BOUND ABOUND</td>
<td>Logical flag equal to the number of degrees of freedom in the multipoint constraint set for the current boundary condition.</td>
</tr>
<tr>
<td>NOMIT</td>
<td>BOUND ABOUND</td>
<td>Logical flag equal to the number of omitted degrees of freedom in the current boundary condition.</td>
</tr>
<tr>
<td>NRSET</td>
<td>BOUND ABOUND</td>
<td>Logical flag equal to the number of support degrees of freedom in the current boundary condition.</td>
</tr>
<tr>
<td>NSPC</td>
<td>BOUND ABOUND</td>
<td>Logical flag equal to the number of single point constraint degrees of freedom in the current boundary condition.</td>
</tr>
<tr>
<td>PSIZE</td>
<td>BCBULK others</td>
<td>The number of grid and extra point degrees of freedom.</td>
</tr>
<tr>
<td>SINGASET</td>
<td>SDCOMP</td>
<td>Used to process singular matrix columns.</td>
</tr>
<tr>
<td>SINGLET</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SINGOSAT</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 9. Integer Design Parameters

<table>
<thead>
<tr>
<th>NAME</th>
<th>MODULES</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSDE</td>
<td>SOLUTION</td>
<td>The last iteration to use fully stressed design. Output from SOLUTION.</td>
</tr>
<tr>
<td>FSDS</td>
<td>SOLUTION FSD</td>
<td>The first iteration to use fully stressed design. Output from SOLUTION.</td>
</tr>
<tr>
<td>MAXITER</td>
<td>SOLUTION ACTCON</td>
<td>Parameter set in the MAPOL sequence indicating the maximum number of resizing cycles that are to be performed. Set through the Solution Control OPTIMIZE command. (Def = 15)</td>
</tr>
<tr>
<td>MPE</td>
<td>SOLUTION</td>
<td>The last iteration to use mathematical programming. Output from SOLUTION.</td>
</tr>
<tr>
<td>MPS</td>
<td>SOLUTION FSD</td>
<td>The first iteration to use mathematical programming. Output from SOLUTION.</td>
</tr>
<tr>
<td>NACSD</td>
<td>ABOUND</td>
<td>The number of active stress and displacement constraints in the current active boundary condition. Used to select either the virtual load or gradient method in sensitivity analysis. Set in ABOUND.</td>
</tr>
<tr>
<td>NAUS</td>
<td>ABOUND</td>
<td>The number of active displacement vectors for statics. Set in ABOUND.</td>
</tr>
<tr>
<td>NBNDCOND</td>
<td>SOLUTION</td>
<td>The total number of boundary conditions in the solution control packet. Equal to the number of optimization boundary conditions plus the number of analysis boundary conditions. Output from SOLUTION.</td>
</tr>
<tr>
<td>NDV</td>
<td>MAKEST, others</td>
<td>The number of global design variables in the design model. Set by MAKEST and used in many subsequent modules.</td>
</tr>
<tr>
<td>NITER</td>
<td>N/A</td>
<td>The current optimization iteration number.</td>
</tr>
<tr>
<td>NUMOPTBC</td>
<td>SOLUTION</td>
<td>The number of optimization boundary conditions in the solution control packet. Set in SOLUTION.</td>
</tr>
<tr>
<td>OCE</td>
<td>SOLUTION</td>
<td>The last iteration to use optimality criteria. Set in SOLUTION.</td>
</tr>
<tr>
<td>OCS</td>
<td>SOLUTION FSD</td>
<td>The first iteration to use optimality criteria. Set in SOLUTION.</td>
</tr>
</tbody>
</table>

### Table 10. Integer Aerodynamic Parameters

<table>
<thead>
<tr>
<th>NAME</th>
<th>MODULES</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINDEX</td>
<td>ABOUND, AEROSENS, BOUND, PFAERO</td>
<td>The index value for the Mach number dependent subscripted steady aerodynamic matrices. Typically has a value used to select the proper matrices for the current boundary condition.</td>
</tr>
<tr>
<td>SUBS</td>
<td>SAERODRV SAERO others</td>
<td>Identifies the subcase subscript. SAERO subcases with the same symmetry Mach number, MINDEX, trim type, and dynamic pressure are processed using the same subscript. This occurs with multiple load conditions with the same aero correction.</td>
</tr>
<tr>
<td>SYM</td>
<td>BOUND</td>
<td>A control flag denoting whether the symmetric (SYM=1) or antisymmetric (SYM=-1) steady aeroelastic matrices are to be used are to be used in the current boundary condition.</td>
</tr>
<tr>
<td>NAME</td>
<td>MODULES</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BBLAST</td>
<td>BOUND</td>
<td>Indicates if there are any nuclear blast subcases in the current boundary condition.</td>
</tr>
<tr>
<td>BDFR</td>
<td>BOUND</td>
<td>Indicates if there are c:y direct frequency response subcases in the current boundary condition.</td>
</tr>
<tr>
<td>BDRSF</td>
<td>BOUND</td>
<td>Indicates if there are either transient or frequency response disciplines in the current boundary condition.</td>
</tr>
<tr>
<td>BDTR</td>
<td>BOUND</td>
<td>Indicates if there are any direct transient response subcases in the current boundary condition.</td>
</tr>
<tr>
<td>BDYN</td>
<td>BOUND</td>
<td>Indicates if there are any dynamic analyses (flutter, transient or frequency) in the current boundary condition.</td>
</tr>
<tr>
<td>BFLUTR</td>
<td>BOUND</td>
<td>Indicates if there are any flutter analyses in the current boundary condition.</td>
</tr>
<tr>
<td>BGUST</td>
<td>BOUND</td>
<td>Indicates if there are any gust loads for either transient or frequency disciplines in the current boundary condition.</td>
</tr>
<tr>
<td>BLOAD</td>
<td>BOUND</td>
<td>Indicates if there are any mechanical, thermal or gravity static applied loads in the current boundary condition.</td>
</tr>
<tr>
<td>BMASS</td>
<td>BOUND</td>
<td>Indicates if a mass matrix exists in the current boundary condition.</td>
</tr>
<tr>
<td>BMFR</td>
<td>BOUND</td>
<td>Indicates if there are any modal frequency response subcases in the current boundary condition.</td>
</tr>
<tr>
<td>BMODES</td>
<td>BOUND</td>
<td>Indicates if there are any disciplines that require that a normal modes analysis be performed.</td>
</tr>
<tr>
<td>BMTR</td>
<td>BOUND</td>
<td>Indicates if there are any modal transient response subcases in the current boundary condition.</td>
</tr>
<tr>
<td>BSARRO</td>
<td>BOUND</td>
<td>Indicates if there are any steady aeroelastic subcases in the current boundary condition.</td>
</tr>
<tr>
<td>DMODES</td>
<td>BOUND</td>
<td>Indicates if there are any modal disciplines in the current boundary condition.</td>
</tr>
<tr>
<td>NQDR</td>
<td>BOUND</td>
<td>Indicates if dynamic reduction is selected for the current boundary condition.</td>
</tr>
</tbody>
</table>
Table 12. Logical Discipline Parameters

<table>
<thead>
<tr>
<th>NAME</th>
<th>MODULES</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTAEFF</td>
<td>ABOUND</td>
<td>TRUE if the current boundary condition has any active aeroelastic effectiveness constraints.</td>
</tr>
<tr>
<td>ACTAERO</td>
<td>ABOUND</td>
<td>TRUE if the current boundary condition has any active constraints associated with SAERO analyses.</td>
</tr>
<tr>
<td>ACTBOUND</td>
<td>ABOUND</td>
<td>TRUE if the current boundary condition has any active constraints.</td>
</tr>
<tr>
<td>ACTDYN</td>
<td>ABOUND</td>
<td>TRUE if the current boundary condition has any active frequency constraints.</td>
</tr>
<tr>
<td>ACTFLUT</td>
<td>ABOUND</td>
<td>TRUE if the current boundary condition has any active flutter constraints.</td>
</tr>
<tr>
<td>AEFLG</td>
<td>SAERO</td>
<td>Logical array which indicates whether the current SAERO subscript value has aeroelastic effectiveness constraints applied to it.</td>
</tr>
<tr>
<td>APPCNVRG</td>
<td>DESIGN</td>
<td>TRUE when the approximate problem was converged in a previous iteration.</td>
</tr>
<tr>
<td></td>
<td>ACTCON</td>
<td>TRUE when global convergence has been reached.</td>
</tr>
<tr>
<td>GLECNVRG</td>
<td>ACTCON</td>
<td>Set TRUE in MK2GG if a MK2GG matrix is input for the current boundary condition.</td>
</tr>
<tr>
<td>LOOP</td>
<td>—</td>
<td>General logical used to control DO-WHILE loops.</td>
</tr>
<tr>
<td>M2GGFLG</td>
<td>MK2GG</td>
<td>Set TRUE in MK2GG if an MK2GG matrix is input for the current boundary condition.</td>
</tr>
<tr>
<td>NONPONLY</td>
<td>NP4AERO</td>
<td>TRUE if the run contains any NP4AERO analyses.</td>
</tr>
<tr>
<td>PFLAG</td>
<td>ACTCON</td>
<td>Set TRUE in ACTCON if DESPUNCH needs to punch a new model.</td>
</tr>
</tbody>
</table>
2.4.2. The Solution Algorithm

Finite element structural analysis, which forms the core of the ASTROS system, requires the manipulation of large matrices. The MAPOL control language was designed with this requirement in mind and, therefore, is able to directly support the manipulation of matrices. Consequently, the majority of the MAPOL sequence consists of matrix equations. The algorithmic nature of the MAPOL syntax allows the reader to follow these matrix operations fairly easily, and the notation roughly follows that used in the Theoretical Manual. Therefore, the focus of this section will be the description of modules called by the MAPOL sequence.

There are a number of engineering and utility modules called to perform tasks associated with the several analysis disciplines supported by the ASTROS system. Table 13 of Section 2.4.2.1 lists the modules defined to the ASTROS executive system, and provides a brief description of each. Not all of these modules appear in the standard solution sequence, but are included in the table to ensure its completeness and usefulness in modifying the standard sequence. The use of these modules is discussed in more detail in the section on modifying the standard MAPOL sequence and are more fully documented in the Programmer's Manual. The brief descriptions of the remaining segments of the standard algorithm that follow, coupled with the inherent readability of MAPOL syntax, are felt to provide a reasonably complete picture of the flow through the standard sequence.
2.4.2.1. MAPOL Engineering and Utility Modules

This section contains a brief description, shown in Table 13, of each of the MAPOL addressable modules defined to the ASTROS executive system. The intrinsic mathematical functions of the MAPOL language are not included.

<table>
<thead>
<tr>
<th>MODULE</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABOUND</td>
<td>ENGINEERING</td>
<td>Generates flags for the current boundary condition during the sensitivity calculation. These are then returned to the executive sequence to direct the execution of the required sensitivity analyses.</td>
</tr>
<tr>
<td>ACTCON</td>
<td>ENGINEERING</td>
<td>Determines whether the design task has converged. If the optimization has not converged, this module selects which constraints are to be included in the current redesign. On termination or print request, this routine computes the values of the local design variables.</td>
</tr>
<tr>
<td>AEROEFPSS</td>
<td>ENGINEERING</td>
<td>Evaluates aerelastic effectiveness sensitivities.</td>
</tr>
<tr>
<td>AEROSENS</td>
<td>ENGINEERING</td>
<td>Computes the sensitivities to active strength constraints and/or aerelastic effectiveness constraints for active steady aerelastic optimization boundary conditions.</td>
</tr>
<tr>
<td>AMP</td>
<td>ENGINEERING</td>
<td>Computes the discipline dependent unsteady aerodynamic matrices for flutter, gust and blast analyses.</td>
</tr>
<tr>
<td>ANALINIT</td>
<td>ENGINEERING</td>
<td>Initializes the final analysis pass.</td>
</tr>
<tr>
<td>APPEND</td>
<td>MATRIX</td>
<td>Appends one matrix to another.</td>
</tr>
<tr>
<td>AROSNSDR</td>
<td>ENGINEERING</td>
<td>Driver for SAERO sensitivity analysis.</td>
</tr>
<tr>
<td>AROSNMR</td>
<td>ENGINEERING</td>
<td>Merges SAERO sensitivities for each subscript into (MATOUT) in case order for active subcases.</td>
</tr>
<tr>
<td>BCBGPDT</td>
<td>ENGINEERING</td>
<td>Builds the boundary condition dependent grid point coordinate relation, BCBGPDT, for the specified boundary condition.</td>
</tr>
<tr>
<td>BCBULK</td>
<td>ENGINEERING</td>
<td>Builds boundary condition dependent matrices, transfer functions and initial conditions.</td>
</tr>
<tr>
<td>BLSTDRV</td>
<td>ENGINEERING</td>
<td>Performs the response of an aircraft to a nuclear blast. Computes modal displacements, velocities, and accelerations.</td>
</tr>
<tr>
<td>BLASTFIT</td>
<td>ENGINEERING</td>
<td>Computes the interpolated time domain steady state and time dependent unsteady aerodynamic influence coefficients for blast analyses.</td>
</tr>
<tr>
<td>BLASTRIM</td>
<td>ENGINEERING</td>
<td>Performs a trim analysis of an aircraft in order to establish initial conditions for a nuclear blast response calculation.</td>
</tr>
<tr>
<td>BOUND</td>
<td>ENGINEERING</td>
<td>Returns flags to the MAPOL sequence that define the matrix reduction path for the current boundary condition.</td>
</tr>
<tr>
<td>MODULE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>------------</td>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CUIb</td>
<td>ENGINEERING</td>
<td>Computes the complex eigenvalues and eigenvectors of a matrix.</td>
</tr>
<tr>
<td>COLMERGE</td>
<td>MATRIX</td>
<td>Merges two or more submatrices into a single matrix based on column partitioning vectors.</td>
</tr>
<tr>
<td>COLPART</td>
<td>MATRIX</td>
<td>Partitions a matrix into two or more submatrices based on column partitioning vectors.</td>
</tr>
<tr>
<td>DCEVAL</td>
<td>ENGINEERING</td>
<td>Evaluates displacement constraints in the current boundary condition.</td>
</tr>
<tr>
<td>DDLOAD</td>
<td>ENGINEERING</td>
<td>Computes the sensitivities of design dependent loads for active boundary conditions.</td>
</tr>
<tr>
<td>DECOMP</td>
<td>MATRIX</td>
<td>Decomposes a matrix into its triangular factors.</td>
</tr>
<tr>
<td>DESIGN</td>
<td>ENGINEERING</td>
<td>Performs redesign by math programming methods based on the current set of active constraints and constraint sensitivities.</td>
</tr>
<tr>
<td>DESPUNCH</td>
<td>UTILITY</td>
<td>Writes new modified Bulk Data entries for the current design iteration to the PUNCH file.</td>
</tr>
<tr>
<td>DMA</td>
<td>ENGINEERING</td>
<td>Assembles the direct and/or modal stiffness, mass and/or damping matrices including extra point degrees of freedom for dynamic analysis disciplines.</td>
</tr>
<tr>
<td>DYNLOAD</td>
<td>ENGINEERING</td>
<td>Assembles the direct and/or modal time and/or frequency dependent loads including extra point degrees of freedom for dynamic response disciplines.</td>
</tr>
<tr>
<td>DYNRESP</td>
<td>ENGINEERING</td>
<td>Computes the direct or modal displacements, velocities and accelerations for transient, frequency and blast analyses.</td>
</tr>
<tr>
<td>EDR</td>
<td>ENGINEERING</td>
<td>Computes the stresses, strains, grid point forces and strain energies for elements selected for output for the particular boundary condition.</td>
</tr>
<tr>
<td>ENA1</td>
<td>ENGINEERING</td>
<td>Assembles the element stiffness and mass matrices (stored in the KELM and MELM entities) into the design sensitivity matrices DKVI, DMVI.</td>
</tr>
<tr>
<td>ENA2</td>
<td>ENGINEERING</td>
<td>Assembles the element stiffness and mass matrix sensitivities (stored in the DKVI and DMVI entities) into the global stiffness and mass matrices for the current design iteration.</td>
</tr>
<tr>
<td>EMG</td>
<td>ENGINEERING</td>
<td>Computes the element stiffness, mass, thermal load and stress component sensitivities for all structural elements.</td>
</tr>
<tr>
<td>EXIT</td>
<td>UTILITY</td>
<td>Terminates the execution of the MAPOL sequence. Useful to terminate modified MAPOL sequences.</td>
</tr>
<tr>
<td>PBS</td>
<td>MATRIX</td>
<td>Performs the forward-backward substitution to solve systems of linear equations.</td>
</tr>
<tr>
<td>PCEVAL</td>
<td>ENGINEERING</td>
<td>Evaluates the current value of all frequency constraints.</td>
</tr>
<tr>
<td>FLUTDMA</td>
<td>ENGINEERING</td>
<td>Assembles the dynamic matrices for the FLUTTER disciplines.</td>
</tr>
<tr>
<td>FLUTDRV</td>
<td>ENGINEERING</td>
<td>Driver for FLUTTER analyses.</td>
</tr>
<tr>
<td>FLUTQHKL</td>
<td>ENGINEERING</td>
<td>Processes the [QKL] matrix with normal modes for FLUTTER.</td>
</tr>
<tr>
<td>FLUTSENS</td>
<td>ENGINEERING</td>
<td>Computes the sensitivities of active flutter constraints in the current active boundary condition.</td>
</tr>
<tr>
<td>FLUTTRAN</td>
<td>ENGINEERING</td>
<td>Performs flutter analyses in the current boundary condition and evaluates any flutter constraints if it is an optimization boundary condition with applied flutter constraints.</td>
</tr>
<tr>
<td>MODULE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>-------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>FREDUCE</td>
<td>ENGINEERING</td>
<td>Reduces the symmetric or asymmetric f-set stiffness, mass and/or loads matrix to the a-set if there are omitted degrees of freedom.</td>
</tr>
<tr>
<td>FREQSENS</td>
<td>ENGINEERING</td>
<td>Computes the sensitivities of active frequency constraints in the current active boundary condition.</td>
</tr>
<tr>
<td>PSD</td>
<td>ENGINEERING</td>
<td>Performs redesign by fully stressed design methods based on the set of applied stress constraints. All other applied constraints are ignored.</td>
</tr>
<tr>
<td>GDR1</td>
<td>ENGINEERING</td>
<td>Computes the shifted stiffness matrix and the rigid body transformation matrix ([OEM]) to be used in Phase 2 of Generalized Dynamic Reduction.</td>
</tr>
<tr>
<td>GDR2</td>
<td>ENGINEERING</td>
<td>Computes the orthogonal basis ([KPRZOX]) for the general Krylov subspace to be used in Phase 3 of Generalized Dynamic Reduction.</td>
</tr>
<tr>
<td>GDR3</td>
<td>ENGINEERING</td>
<td>Computes the transformation matrix ([GDR3O]) for Generalized Dynamic Reduction.</td>
</tr>
<tr>
<td>GDR4</td>
<td>ENGINEERING</td>
<td>Computes transformations between displacement sets useful for data recovery from Generalized Dynamic Reduction.</td>
</tr>
<tr>
<td>GPWG</td>
<td>ENGINEERING</td>
<td>Grid point weight generator module.</td>
</tr>
<tr>
<td>GREDUCE</td>
<td>ENGINEERING</td>
<td>Reduces the symmetric g-set stiffness, mass or loads matrix to the n-set if there are multipoint constraints in the boundary condition.</td>
</tr>
<tr>
<td>GTLOAD</td>
<td>ENGINEERING</td>
<td>Assembles the current static applied loads matrix for any statics subcases in the current boundary condition from the constant simple load vectors and the design dependent load sensitivities.</td>
</tr>
<tr>
<td>IFP</td>
<td>ENGINEERING</td>
<td>Reads the Bulk Data File and loads the input data to relations. Computes the external coordinate system transformation matrices and creates the basic grid point data. Also performs bandwidth minimization.</td>
</tr>
<tr>
<td>INERTIA</td>
<td>ENGINEERING</td>
<td>Computes the rigid body accelerations for statics analyses with inertia relief.</td>
</tr>
<tr>
<td>ITERINIT</td>
<td>ENGINEERING</td>
<td>Initializes the CONUT relation for the current iteration.</td>
</tr>
<tr>
<td>LODGEN</td>
<td>ENGINEERING</td>
<td>Assembles the simple load vectors and simple load sensitivities for all applied loads in the Bulk Data File.</td>
</tr>
<tr>
<td>MAKDFU</td>
<td>ENGINEERING</td>
<td>Assembles the sensitivities to the displacements of active stress and displacement constraints in the current active boundary condition.</td>
</tr>
<tr>
<td>MAKDFV</td>
<td>ENGINEERING</td>
<td>Assembles the sensitivities of active thickness constraints.</td>
</tr>
<tr>
<td>MAKDVU</td>
<td>ENGINEERING</td>
<td>Multiplies the stiffness or mass design sensitivities by the active displacements or accelerations.</td>
</tr>
<tr>
<td>MAKEST</td>
<td>ENGINEERING</td>
<td>Generates the element summary relational entities for all structural elements. Determines the design variable linking and generates sensitivities for any thickness constraints.</td>
</tr>
<tr>
<td>MERGE</td>
<td>MATRIX</td>
<td>Merges two or more submatrices into a single matrix based on row and column partitioning vectors.</td>
</tr>
<tr>
<td>MKAMAT</td>
<td>ENGINEERING</td>
<td>Assembles the constraint sensitivity matrix from the sensitivity matrices formed by MAKDFU and the sensitivities of the displacements for active static load conditions in the current active boundary condition.</td>
</tr>
</tbody>
</table>
Table 13. Summary of ASTROS Modules — Continued

<table>
<thead>
<tr>
<th>MODULE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>MKUglIT</td>
<td>ENGINEERING</td>
<td>Generates the structural set definition entity USET for each boundary condition and forms the partitioning vectors and transformation matrices used in matrix reduction.</td>
</tr>
<tr>
<td>MK20G</td>
<td>ENGINEERING</td>
<td>Interprets solution control and generates the [K20G] and [K2001] matrices if necessary.</td>
</tr>
<tr>
<td>NREDUCE</td>
<td>ENGINEERING</td>
<td>Reduces the symmetric n-set stiffness, mass or loads matrix to the f-set if there are single point constraints in the boundary condition.</td>
</tr>
<tr>
<td>NULLMAT</td>
<td>ENGINEERING</td>
<td>Breaks data base equivalences from previous boundary conditions.</td>
</tr>
<tr>
<td>OPPAEROM</td>
<td>ENGINEERING</td>
<td>Solves for the SAERO applied loads and displacements on aero boxes for output requests.</td>
</tr>
<tr>
<td>OPPDISP</td>
<td>ENGINEERING</td>
<td>Prints selected displacements, velocities and/or accelerations from any analyses in the current boundary condition.</td>
</tr>
<tr>
<td>OPPALOAD</td>
<td>ENGINEERING</td>
<td>Solves for the SAERO applied loads and constraint forces for output processing.</td>
</tr>
<tr>
<td>OPFFEDR</td>
<td>ENGINEERING</td>
<td>Prints selected element stress, strain, force and/or strain energies from any analyses in the current boundary condition.</td>
</tr>
<tr>
<td>OPFGRAD</td>
<td>ENGINEERING</td>
<td>Processes output requests for objective and constraint gradients.</td>
</tr>
<tr>
<td>OPFLOAD</td>
<td>ENGINEERING</td>
<td>Prints selected applied external loads from any analyses in the current boundary condition.</td>
</tr>
<tr>
<td>OPPMROOT</td>
<td>ENGINEERING</td>
<td>Processes output requests for normal modes.</td>
</tr>
<tr>
<td>OPPSPCF</td>
<td>ENGINEERING</td>
<td>Processes output requests for single-point constraint forces.</td>
</tr>
<tr>
<td>PARTN</td>
<td>MATRIX</td>
<td>Partitions a matrix into two or more submatrices based on row and column partitioning vectors.</td>
</tr>
<tr>
<td>PFBULK</td>
<td>ENGINEERING</td>
<td>Performs a number of preface operations to form additional collections of data.</td>
</tr>
<tr>
<td>QHHLGEN</td>
<td>ENGINEERING</td>
<td>Computes the discipline dependent unsteady aerodynamic matrices for GUST and BLAST analyses in the modal structural system.</td>
</tr>
<tr>
<td>RECEND</td>
<td>CADDB</td>
<td>Terminates setting conditions on a MAPOL relational access.</td>
</tr>
<tr>
<td>RECOVA</td>
<td>ENGINEERING</td>
<td>Recovers the symmetric or asymmetric f-set displacements or accelerations if there are omitted degrees of freedom.</td>
</tr>
<tr>
<td>REIG</td>
<td>ENGINEERING</td>
<td>Computes the eigenvalues and eigenvectors of the system as directed by the boundary METHOD selection.</td>
</tr>
<tr>
<td>RELCND</td>
<td>CADDB</td>
<td>Sets conditions on attribute values for MAPOL retrieval of relational entities.</td>
</tr>
<tr>
<td>RELADD</td>
<td>CADDB</td>
<td>Adds a tuple to an entity opened with RELUSE.</td>
</tr>
<tr>
<td>RELEND</td>
<td>CADDB</td>
<td>Closes an entity opened from the MAPOL sequence using RELUSE.</td>
</tr>
<tr>
<td>RELGET</td>
<td>CADDB</td>
<td>Retrieves a relational tuple into execution memory for a relation opened for use in the MAPOL sequence.</td>
</tr>
<tr>
<td>RELUPD</td>
<td>CADDB</td>
<td>Performs a relational update from execution memory of a tuple retrieved using RELGET.</td>
</tr>
<tr>
<td>RELUSE</td>
<td>CADDB</td>
<td>Opens a relational entity for access from the executive sequence.</td>
</tr>
<tr>
<td>ROWMERGE</td>
<td>MATRIX</td>
<td>Merges two or more submatrices into a single matrix based on row partitioning vectors.</td>
</tr>
<tr>
<td>MODULENAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>------------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>ROWPART</td>
<td>MATRIX</td>
<td>Partitions a matrix into two or more submatrices based on row partitioning vectors.</td>
</tr>
<tr>
<td>SAERO</td>
<td>ENGINEERING</td>
<td>Solves the trim equation for steady aeroelastic trim analyses. Computes the rigid and flexible stability coefficients for steady aeroelastic analyses and the aerodynamic effectiveness constraints for constrained optimization steady aerodynamic analyses.</td>
</tr>
<tr>
<td>SAERODRV</td>
<td>ENGINEERING</td>
<td>Driver for SAERO disciplines.</td>
</tr>
<tr>
<td>SAEROMRG</td>
<td>ENGINEERING</td>
<td>Merges the SAERO results into (MATOUT) in case order.</td>
</tr>
<tr>
<td>SCEVAL</td>
<td>ENGINEERING</td>
<td>Computes the stress and/or strain constraint values for the statics or steady aeroelastic trim analyses in the current boundary condition.</td>
</tr>
<tr>
<td>SOLUTION</td>
<td>ENGINEERING</td>
<td>Interprets the solution control packet, forms the CASE entity and outputs certain key parameters to the executive sequence.</td>
</tr>
<tr>
<td>SPLINES</td>
<td>ENGINEERING</td>
<td>Generates interpolation matrix relating displacements and forces between the steady aero and structural models.</td>
</tr>
<tr>
<td>SPLINEU</td>
<td>ENGINEERING</td>
<td>Generates interpolation matrix relating displacements and forces between the unsteady aero and structural models.</td>
</tr>
<tr>
<td>STEADY</td>
<td>ENGINEERING</td>
<td>Computes rigid unit forces and aeroelastic corrections for steady aero.</td>
</tr>
<tr>
<td>STEADYNP</td>
<td>ENGINEERING</td>
<td>Computes rigid trimmed forces for non-planar models.</td>
</tr>
<tr>
<td>TCEVAL</td>
<td>ENGINEERING</td>
<td>Computes the current values of thickness constraints for this optimization iteration.</td>
</tr>
<tr>
<td>TRANSPOSE</td>
<td>MATRIX</td>
<td>Transposes a matrix.</td>
</tr>
<tr>
<td>UNSTEADY</td>
<td>ENGINEERING</td>
<td>Computes unsteady generalised forces.</td>
</tr>
<tr>
<td>USETPRT</td>
<td>UTILITY</td>
<td>Prints the structural set definition table from the USBT entity for the specified boundary condition.</td>
</tr>
<tr>
<td>UTOPRT</td>
<td>UTILITY</td>
<td>Prints several specific matrix entities in an interpretable form.</td>
</tr>
<tr>
<td>UTMPRNG</td>
<td>UTILITY</td>
<td>Purges matrix entities.</td>
</tr>
<tr>
<td>UTMPRT</td>
<td>UTILITY</td>
<td>To print any matrix entity.</td>
</tr>
<tr>
<td>UTNRPT</td>
<td>UTILITY</td>
<td>To print any relational entity. Only the first twelve attributes are printed and character attributes must be eight characters in length or they will be ignored.</td>
</tr>
<tr>
<td>UTNRPRG</td>
<td>UTILITY</td>
<td>Purges relational entities.</td>
</tr>
<tr>
<td>USEUPRT</td>
<td>UTILITY</td>
<td>To print any unstructured entity.</td>
</tr>
<tr>
<td>VANGO</td>
<td>ENGINEERING</td>
<td>Performs redesign using optimality criteria methods.</td>
</tr>
<tr>
<td>YSMERGE</td>
<td>ENGINEERING</td>
<td>A special purpose merge utility for merging YS-like vectors (vectors of enforced displacements) into matrices for data recovery.</td>
</tr>
</tbody>
</table>
2.4.2.2. The Preface Segment

In the context of optimization, it is obvious that invariant data should be computed only once and re-used subsequently for each iteration. This is the underlying principle involved in the determination of which modules constitute preface modules. In each instance, the data generated are invariant with respect to the design variables.

The preface segment begins with a call to the solution control interpreter to determine the number and types of analyses to be performed. The input file processor (IFP) is then called. The element connection data and element matrices are then formed. PFBULK is then called to perform error checking operations on a variety of user input data. If any nonplanar steady aerodynamic analyses are requested in the solution control, they are analyzed next. Then the DMD1 is called to compute the design invariant stiffness and mass sensitivities to the global design variables. Then the simple loads and load sensitivities are computed in LODGEN. If any planar static aerodynamic analyses are requested in the solution control, the STEADY and SPLINES modules are called to create the aerodynamic matrices required for the aeroelastic analysis. Finally, unsteady aerodynamics matrices are computed for BLAST, GUST and FLUTTER analyses in UNSTEADY, AMP and SPLINEU.

2.4.2.3. The Analysis/Optimization Segments

The remainder of the MAPOL algorithm consists of the optimization and analysis segments. Any particular boundary condition is either an optimization boundary condition (implying that the quantities computed in the disciplines selected in the solution control are constrained and that the structure is to be optimized subject to those constraints) or an analysis boundary condition. The design of the ASTROS system requires that all optimization boundary conditions precede any analysis boundary conditions. The analysis segment (labeled the "final analysis") is intended to follow an optimization with analyses in disciplines whose output values are not constrained but are of interest to the designer or to provide the user with an opportunity to view additional output not desired within the optimization loop. Also the analysis segment can be used on a stand alone basis to perform any desired analyses.

Both the optimization and analysis segments consist of an initial loop on the number of boundary conditions. The analyses in these loops support all the disciplines currently available in the ASTROS system and differ only in the respect that the analysis segment does not have calls to constraint evaluation modules and the optimization segment has convergence tests and design iteration initialization outside the analysis boundary condition loop. The first step in these loops is to assemble the boundary condition dependent number of degrees of freedom (extra points are BC selectable in ASTROS). Then additional PFBULK-like operations are performed in SCBULK to ensure that BC-dependent user input is correct. Then the global stiffness and/or mass matrices are assembled and, if needed, the global loads matrix. Following these tasks, there are several BLOCK IF statements on the various dependent structural sets. In executing each block, the required matrix partitions and reductions are performed. Once the reduced matrices have been obtained for the analyses being performed within the loop, the lowest level response quantities (e.g. displacements, eigenvalues, etc.) are computed. Following the solution, the execution proceeds through another group of dependent set BLOCK IF's to recover the solution vectors to the global set. At this point, the analysis segment is completed with calls to the output file processor modules to compute and output high level response quantities (e.g. stresses).
In the optimization phase of the optimization segment, the ACTCON module determines the status of the global convergence flag CONVERGE and, if the optimization is not complete, the redesign task is performed. Three redesign methods are supported by the standard sequence and selected through the Solution Control. If the option for Fully Stressed Design (FSD) is selected, the redesign is performed in the FSD modules. The two alternative methods (mathematical programming and generalized optimality criteria) employ methods that require sensitivity information. In this case, the sensitivities of the active constraints (chosen by ACTCON based on the NRPAC and EPS parameters) are computed.

The sensitivities of the active constraints which are explicit functions of the design variables are computed first in the MAKDFV module. Then the second boundary condition loop within the optimization segment begins. The ABOUND routine determines the types of active constraints in each boundary condition and outputs logic flags to control the subsequent sensitivity computations. Then boundary condition dependent constraints which are explicit functions of the design variables (frequency and flutter) are computed. Next, the sensitivities of the constraints to the displacements for those STATICS constraints which are explicit functions of the displacements (e.g., stress and displacement constraints) are computed using the MAKDFU module. For these types of constraints, the product of the stiffness sensitivities and the displacements and the mass sensitivities and the accelerations are also computed and modified appropriately to account for design dependent loads and inertia relief. The resulting matrix is then reduced and used to solve for the sensitivities of the displacements to the design variables. This matrix is recovered to the free displacement set in a manner similar to the recovery of the outputs in the analysis phase of the optimization segment. The final module within the boundary condition loop for sensitivity evaluation is MXAMAT. Within this module the constraint sensitivities to the design variables are formed from the product of the two sensitivity matrices previously obtained.

For static aeroelastic analyses, a procedure similar to that for STATICS is used twice: once for "pseudo-displacements" that allow computation of aeroelastic effectiveness derivatives and once for real displacements that support the static strength constraints. The static aeroelastic sensitivity code is further complicated by the generality of the aeroelastic correction matrix selections, which are subcase dependent.

After all the active optimization boundary conditions have been processed, the DESIGN or the VANGO module is called. Within these modules, the approximate design problem is arranged for use by the optimizer and is solved. Following convergence of the approximate problem, execution returns to the top of the optimization loop and a complete reanalysis of all the boundary conditions is performed. Once completed, the ACTCON module determines if the global problem is converged and, if so, sets the global convergence flag to TRUE causing the execution to pass to the top of the analysis segment. If any analysis boundary conditions exist, they will be processed in a manner similar to the analysis phase of the optimization segment. After performing the requested final analyses (if any) the executive system terminates the ASTROS execution.

2.4.3. Modifying the Standard MAPOL Sequence

The standard MAPOL sequence is provided to allow a user to run the ASTROS system without detailed knowledge of the MAPOL language or the standard sequence. There is no* however, any requirement that the standard sequence be used. Chapter 6 outlines the procedure for writing a valid MAPOL sequence, and any series of syntactically correct MAPOL statements may be used to direct the ASTROS procedure. All the engineering, utility and matrix manipulation modules shown in Subsection
2.4.2.1 are available to any MAPOL sequence used to direct the system. In addition, there are a number of intrinsic functions such as \texttt{SIN} and \texttt{ABS} that are also available. Their use is detailed in the MAPOL Programming chapter. The sophisticated MAPOL user is thus provided with a very flexible control language to manipulate the ASTROS system. This subsection describes simple modifications to the standard algorithm to print out additional data items, to fine tune the optimization algorithm and to restore an ASTROS analysis that was partially executed on a previous run. No set of examples, however, can possibly indicate the full range of available capabilities; the user is therefore cautioned not to be overly constrained by this discussion.

In order to avoid vast quantities of output and to limit the execution time, the standard output is kept to a minimum. Several utilities, listed in Section 2.4.2.1, can, however, be inserted in the standard sequence to output data stored on the data base. In addition, a utility has been written to print out the structural set definition table to aid in the debugging of the structural model. The \texttt{UTRPRT}, \texttt{UTGPRT}, \texttt{UTRT2RT} and \texttt{UTUPRT} print utilities dump the contents of specified data base entities to the user's output file. These can be used anywhere in the MAPOL sequence after the specified entity has been filled with data. The \texttt{UDETPRT} utility provides the user with the ability to print the structural set definition table (\texttt{USET}) in a format which aids in debugging the structural model. These utilities provide the user with some simple tools to allow closer interaction with the data stored on the data base and to provide capability to more closely track the execution.

The print utilities provide data visibility without modifying the basic execution of the standard sequence. At a slightly more complex level, the user might desire to fine tune the optimization procedure or to track the iterations of the optimizer more closely. Table 4 includes a number of parameters which are used by ASTROS to direct the optimization. All of these parameters can be modified through the \texttt{OPTIMIZR} command in solution control. That modification, however, only occurs once. Any of these parameters can be changed by the user at any point in the MAPOL sequence. For example, the \texttt{MOVLIM} parameter could be changed to a different value after the fifth iteration by placing the following statement immediately after the \texttt{WHILE} test on \texttt{QLBCNVRG}:

\begin{verbatim}
IF NITER > 5 MOVLIM = 1.5:
\end{verbatim}

Obviously, the conditional testing could become as complex as the MAPOL programmer desires.

The brief discussion above does not begin to describe all the options open to the sophisticated ASTROS user. It does, however, outline some of the most commonly performed modifications to the standard MAPOL algorithm. The concepts described can be extended to a large number of similar changes; e.g., modifying the input dynamic pressure value within the MAPOL sequence could be done to avoid re-running the base run of an ASTROS execution. At a more advanced level, the MAPOL relational data base entity utilities can be used to directly modify the design variable values or objective sensitivities.
2.4.4. Restart Capability

The ASTROS system has an informal restart capability which is described in the following sections.

2.4.4.1. Introduction

Currently, ASTROS does not support a formal restart capability but this does not imply that restarts cannot be performed in ASTROS. The restart capability in ASTROS is limited in that the user must use a modified MAPOL sequence in order to terminate the system early and the restarted job MUST use a tailored MAPOL sequence to restart the job at the desired point. Otherwise, there are no limits to what can be done by the experienced user. The ASTROS restart capability is best described as a full featured Manual Restart — ASTROS does not have an Automated Restart.

There are several reasons why users may wish to suspend an ASTROS execution and then perform a restart and the code supports this basic capability. For example, a user may wish to examine the progress of a design after each optimization iteration. With the run stopped, the user would then have the freedom to use ICE (Reference 7) and alter ASTROS data to redirect the optimization path if desired. As another example, the user may suspend execution and, again using ICE, replace ASTROS-computed data with their external equivalent (such as the QHHL or QKKL matrices of unsteady aerodynamic influence coefficients). Clearly, the ability to suspend/restart executions in combination with the ICE environment opens limitless possibilities.

Without an automated restart, however, the user is responsible for ensuring that several requisite tasks are completed. These are

- Ensuring that the run-time database has the proper STATUS on suspension and on restart.
- Selecting where in the MAPOL sequence to suspend execution.
- Writing a MAPOL sequence to restart execution. This may or may not be a modification to the standard sequence.
- Ensuring that those scalar variable(s) that are common to both the original and the restart MAPOL sequences are initialized to the correct value(s)

Each of these tasks are discussed in the following sections.

2.4.4.2. Ensuring proper STATUS of the run-time database

When suspending execution, the run-time database must be saved. ASTROS stores all the information that it has generated during the execution on the run-time database and, on any restart, the downstream modules will expect that those data will exist when they are executed in the restart environment. The run-time database is also the location of the data that the user may wish to modify or add to using ICE prior to initiating the restart. Saving the run-time database is done by selecting a STATUS of NEW (with the optional user parameter, KEEP, if required on the local host) on the ASSIGN DATABASE entry. For example,

```
ASSIGN DATABASE CALVIN HOBSES NEW
```
ASSIGN DATABASE CALVIN HOBBES NEW KEEP

When restarting ASTROS using an existing database whose contents are to be preserved, ASTROS must be notified to attach the existing run-time database files without re-initializing them. This is done by selecting a STATUS of OLD on the ASSIGN DATABASE entry. For example

ASSIGN DATABASE CALVIN HOBBES OLD

If the STATUS of OLD is not given, existing database files are typically overwritten by the system. The STATUS flag indicates the status of the data not of the files so the files may exist with a STATUS of NEW and will result in the database contents being replaced by the new execution.

2.4.4.3. Suspending/Restarting Execution

ASTROS execution is controlled by the MAPOL sequence that is supplied in the MAPOL packet. This may be the standard sequence (if the packet is omitted), an edited version of the standard sequence, or a user supplied sequence. To suspend execution, a MAPOL sequence must be defined which results in clean termination (one without fatal errors) of the ASTROS execution. This may be the standard execution or, more typically, an edited standard sequence or even a standalone MAPOL program.

Most commonly, the suspension is performed by editing the standard MAPOL sequence and inserting an EXIT call after the last line that is to be executed in the current execution. Alternatively, if no missing IF-THEN-ENDIFs or ENDDO's result, portions of the sequence can simply be deleted. Some care should be taken in suspending execution in the middle of DO and DO WHILE loops or block IFs. Although possible to do, suspensions during execution of these repetitive segments can leave the system in a state that is more difficult to reinitialize on the restart execution. Some experience with MAPOL and with ASTROS is needed before attempting these more complex suspensions. Suspending execution at the beginning or end of the preface, analysis phase (of either the optimization or final analysis segments) or the sensitivity phase is most likely to yield success.

To restart the execution, the user must generate a special MAPOL program either by editing the standard sequence or by writing a new sequence. The restart execution of ASTROS does not have any information on where the initial execution terminated. Only the data on the database is saved (i.e., available for the current execution). Obviously, the new execution may start up at any point the user wishes and need not be associated with the area where the initial run terminated (although only experienced ASTROS users should attempt to drastically alter the flow of the MAPOL sequence).

To generate the special MAPOL sequence, the user may use a GOTO statement to jump ahead in the standard sequence to the restart point or the user may use the EDIT commands to delete those initial sections that no longer need execution. The latter is typically the case when the preface segment is "saved" for restart. If the user deletes lines, care should be taken not to delete half of a looping construct or block IF since that will result in a MAPOL compilation error. The restart MAPOL sequence must also contain any new statements that are required to reset values of MAPOL scalar parameters.
2.4.4.4. Resetting MAPOL Parameters

In the ASTROS system, all the values of MAPOL variables are stored on the database. For the complex data types like RELATIONs and MATRICEs, this is obvious, since their data resides on the database for ICE execution or other processing. Less obviously, the simple data types like REAL and INTEGER (including arrays) are also saved on the database. These data are not easily viewed in the ICE context, but are saved in a way that the MAPOL compiler can recognize. When a restart job is performed, the existence of these old data causes the MAPOL compiler to determine the correspondence between the original data and the new MAPOL sequence. Whenever a variable of the same name and type is found, its initial value is recovered from the old data thus "restoring" the value of the original variable to the last value it contained.

In the restart execution, however, the user must make sure that the last value of the variable is the desired initial value for restart. In some cases, the variables contain "invariants" like the variable NDV which contains the number of global design variables. In other cases, like BC, the variable is a loop counter that should be reset. The MAPOL sequence may perform the reinitialization automatically (for example if a DO loop is re-executed for values 1 through 10, the do loop counter will be reset to 1 no matter what value it contains). If, however, the restart MAPOL omits the loop, the last loop counter that was achieved will be stored in the loop counter on restart.

Determining which MAPOL parameters should be left alone and which should be reset (rather than default to their last value) is the challenge of the manual restart. Tables 7 through 12 (section 2.4.1) of this Manual have a list of all the MAPOL parameters. These tables list each parameter and give a description of how and where the parameter is used. Together with the ASTROS Programmer's Manual, which document the actions that occur in each ASTROS module, the user can decide which parameters should be reset and which should be allowed to default to the value set in the initial execution.

2.5. MAPOL PROGRAM LISTING

The Version 10 MAPOL listing of the standard solution sequence is given on the following pages.
Standard MAPOL Sequence Listing

STAT LEVEL
1:*** $S
2: CSCID <$#> MC0083-MAPOLSEQ 10.1 7/1/93 11:27:52 $S
3:*** $S
4:**********************************************************$S
5: EXECUTIVE SEQUENCE FOR ASTROS $S
6:**********************************************************$S
7:**********************************************************$S
8: CONSTANTS FOR SDCOMP SET SINGULARITY MESSAGES $S
9:**********************************************************$S
10:INTEGER SINGOSET, SINGASET, SINGLSET; $S
11:**********************************************************$S
12:VARIABLE DECLARATION SEGMENT $S
13:**********************************************************$S
14:**********************************************************$S
15:INTEGER GSIZE, NDV, NITER, BC, $S
16:REAL CTL, CTLMIN; $S
17:LOGICAL GLCNVRC, APPCNVRG, PFLAG; $S
18:UNSTRUCT DCENT, GRIDTEMP, SMPLOD; $S
19:RELATION DESHIST, CONI, NPPAR, CONVE, OCPAR, $S
20:REAL MFORM, GRID, SPOINT, EPOINT, SROOP, $S
21:REAL MOMENT1, PLOAD, GRAV, LOAD, EIGR, $S
22:REAL TEMP, TEMPD, $S
23:REAL CORDIC, CORDIR, CORDIS, CORD2C, CORD2R, $S
24:REAL CORD2S, GWPGRID, OGPWG, GRADIENT, $S
25:REAL $S
26:DECLARATIONS FOR MODULE MKUSET $S
27:**********************************************************$S
28:**********************************************************$S
29:DECLARATIONS FOR MODULES MAKEST AND EMG $S
30:**********************************************************$S
31:**********************************************************$S
32:RETURN $S
33:**********************************************************$S
34:**********************************************************$S
35:**********************************************************$S
36:**********************************************************$S
37:**********************************************************$S
38:**********************************************************$S
39:**********************************************************$S
40:**********************************************************$S
41:**********************************************************$S
42:**********************************************************$S
43:**********************************************************$S
44:**********************************************************$S
45:**********************************************************$S
46:**********************************************************$S
47:**********************************************************$S
48:**********************************************************$S
49:**********************************************************$S
50:**********************************************************$S
51:**********************************************************$S
52:**********************************************************$S
Standard MAPOL Sequence Listing — Continued

STAT LEVEL

53  $POMP1,  POMP2,  PIHEX,  MAT1,  MAT2.
54  $MAT8,  MAT9,  CTRIA3,  TRIAL3;
55  
56  $DECLARATIONS FOR DESIGN VARIABLES/CONSTRAINTS AND LINKING
57  $;
58  $;
59  $;
60  $RELATION DESLAM,  DESVAP,  DESVAP,  PLIST,  ELIST;
61  $SHAPE,  SHIPDEN;
62  $RELATION DCONVM,  DCONTM,  DCONEM,  DCONFT,  DCONVM;
63  $DCONVM,  DCONEM,  DCONFT,  DCONVM,  DCONTP;
64  $DCONEM,  DCONTP;
65  $RELATION DCONSP,  DCONFM,  DCONLM,  DCONGP;
66  $RELATION DCWMB,  DCTIAM,  TPWED,  LOCLVAR,  DVT;
67  $MATRIX [PTRANS];
68  $MATRIX [PMINT],  [PHAXT],  [PSMAT];
69  $;
70  $;
71  $DECLARATIONS FOR OUTPUT FILE PROCESSING (EDR/OFP)
72  $;
73  $;
74  $;
75  $RELATION GRIDLIST,  MODELIST,  ELEMIST,  FREQLIST,  TIMELIST;
76  $ITERLIST,  GOVLIST,  LDVLIST,  DCONLIST,  PLYLIST;
77  $;
78  $RELATION OGRIDLO,  OGRIDSP,  OLOCALDV,  OAGRPD,  OAGRDLO;
79  $MATRIX [FLMDM],  [PLGLOAD],  [PFGLOAD],  [PHLOAD];
80  $;
81  $DECLARATIONS FOR MODULES EMA1,  EMA2 AND GLOBAL
82  $;
83  $MATRIX PARTITION/REDUCTION
84  $;
85  $;
86  $;
87  $;
88  $;
89  $;
90  $;
91  $INSTRUCT DRLV,  DRLV;
92  $RELATION GNCRT,  GNNCT;
93  $MATRIX [KGG],  [KNN],  [KFF],  [KAA],  [KLL],
94  $[MGG],  [MNN],  [MFF],  [MAA],  [MLL],
95  $[MRRBAR],  [MLR],  [KFF],  [KSS],  [XKOINV(1000)],
96  $[GSUBO(1000)],  [KLINV(1000)],  [MRINV(1000)],
97  $[IPM(1000)],  [MGG],  [IFR(1000)],  [KRR],  [D(1000)],
98  $[KLR],  [KGG],  [LHS(1000)],  [W2(1000)],  [WOO],
99  $[MOA],  [K2(1000)],  [MAABAR];
100  $INSTRUCT TMIP,  [TMP2];
101  $MATRIX [PO],  [PN],  [PF],  [PA],
102  $[PO],  [PLBAR],  [PR],  [RHS(1000)],  [UG(1000)].
Standard MAPOL Sequence Listing — Continued

### STAT LEVEL

103 | INTEGER | [UN], [UP], [UA], [UL], [UN], !
104 | [AN], [AF], [AA], [AR], !
105 | [AI], [AD], [AUO], [PS]; !
106 | LOGICAL M2GFLAG, K2GFLAG; !
107 | $!
108 | $!
109 | DECLARATIONS FOR SOLUTION CONTROL $!
110 | $!
111 | $!
112 | INTEGER NMOPTEC, NNBCOND, MAXITER, !
113 | MPS, MPE, !
114 | OCS, OCE, !
115 | FSDD, FSDE; !
116 | INTEGER BLOAD, BMASS, BMODES, BSAERO, BFLUTR, !
117 | BDN, BDRSP, BBN, BNSP, !
118 | BMFR, BBLAST, BNSP, BNSP, !
119 | NONIT, NRSET, DMODES; !
120 | REAL MOVLIN, WINDOW, OCMOVLIN, ALPHA, CHWGLIM, !
121 | NRIF, NDF, EPS; !
122 | RELATION JOB, OPTIMIZE, CASE; !
123 | $!
124 | $!
125 | DECLARATIONS FOR SENSITIVITY EVALUATION $!
126 | $!
127 | $!
128 | INTEGER DDFLG, NACSD, NAUS, NAU; !
129 | ACTBOUND, ACTFLUT, ACTDYN, ACTAERO, ACTAEEF, !
130 | $!
131 | UNSTRUCT PCAS, PCA, PCAE; !
132 | MATRIX [DFDU], [PDQ], [UGA], [DUG], [IMUG], !
133 | ![DUG], [DUG], [DUG], !
134 | ![DUG], [DUG], [DUG], [DUG], !
135 | ![DUG], [DUG], [DUG], [DUG], !
136 | ![DUG], [DUG], [DUG], [DUG], !
137 | ![DUG], [DUG], [DUG], [DUG], !
138 | ![DUG], [DUG], [DUG], [DUG], !
139 | ![DUG], [DUG], [DUG], [DUG], !
140 | ![DUG], [DUG], [DUG], [DUG], !
141 | ![DUG], [DUG], [DUG], [DUG], !
142 | ![DUG], [DUG], [DUG], [DUG], !
143 | ![DUG], [DUG], [DUG], [DUG], !
144 | AERODYNAMIC ENTITIES $!
145 | $!
146 | $!
147 | INTEGER SYM, MINDEX, SUB, $!
148 | REAL QDP, MACH, !
149 | LOGICAL LOOP, AEFLG(100), NONONLY, !
150 | UNSTRUCT ACPT, UNMK, !
151 | RELATION AEROSURF, AEROFIOL, AEROS, AERFACT, AXSTA, !
152 | BODY, SPLINE1, SET1, SET2, ATTACH, !
153 | TRIM, AERO, BLAST, CAERO6, PAERO6, !
154 | GEOMSA, AECOMPS, STABCF, CAERO1, PAERO1, !
Standard MAPOL Sequence Listing — Continued

STAT LEVL

155 1: CAERO2, PAERO2, MAKERO1, MAER02, FLUTTER, !
156 1: FLFACT, CLAMDA, DCONATE, DCONCLA, DCONFLT, !
157 1: CONEPPS, CONIINK, GROMUA, AECOMPU, SPLINE2, !
158 1: CONEPPF, DCONTRN, DCONSCF, AEROGEOM, CAROGEOM; !
159 1: MATRIX [AIRFRC(1000)], [AICMAT(1000)], [AICMAT(1000)]; !
160 1: [AICS], [KAFF], [PAP], [KAAA], [PA]; !
161 1: [GASUB2(30,33)], [SKJ], [DJLJK], [DJUK]; !
162 1: [KJL], [R1], [K21(30,33)], [PABAR], [PAL]; !
163 1: [PAR(30,33)], [K112(30,33)], [AIRFORCE], [K22]; !
164 1: [GTRK], [GTRK], [GTRK], [GTRK], [GFRTK]; !
165 1: [GFRTK], [GTRK], [UQRTK], [UQRTK], [UQRTK]; !
166 1: [UQRTK], [UQRTK], [UQRTK], [UQRTK], [UQRTK]; !
167 1: [R12(30,33)], [R22], [R21(30,33)], [R11], [K12(30,33)]; !
168 1: [P1], [R21(30,33)], [R31(30,33)], [P111(30,33)]; !
169 1: [R11(30,33)], [R21], [NAAA], [IPMA(30,33)]; !
170 1: [R13(30,33)], [R3], [DEL], [PRIDI]; !
171 1: [AAC], [AAP], [AAB], [AAB], [AAB]; !
172 1: [PAO(1000)], [AAPT], [UAPT], [UAPT], [UAPT]; !
173 1: [UAG(1000)], [AAG(1000)], [AAL], [AAP], [UAP]; !
174 1: [KGG(30,33)], [KGG(30,33)], [HHA(30,33)]; !
175 1: [POAO(30,33)], [PAR(30,33)], [PAR(30,33)]; !
176 1: [DELTA(1000)], [PAOC(1000)], [UAPC(1000)], [AAPC(1000)]; !
177 1: [UAPC(1000)], [UAPC(1000)], [UAPC(1000)], [AAPC(1000)]; !
178 1: [AAPC(1000)], [AAPC(1000)], [K112(30,33)], [K111(30,33)]; !
179 1: [R112(30,33)], [R111(30,33)], [R11(30,33)]; !
180 1: [R111(30,33)], [R111(30,33)], [R11(30,33)]; !
181 1: MATRIX [AATT], [QJL], [QKLL], [QHLL]; !
182 1: INTEGER HSIZE(1000); !
183 1: UNSTRUCT TDATA, ICMDATA, UDLOLY; !
184 1: RELATION LAMBDA, KEIGS, DCONLY, DLOAD, TABLED1; !
185 1: IC, TLGD1, TLGD2, RLDGI, RLOAD2; !
186 1: TSTEP, VSVDMP, TARDMP, DLAGS, TF; !
187 1: DMIG, GUST, FREQ, FREQ1, FREQ2; !
188 1: FFT, FLUTR; !
189 1: MATRIX [PMIK], [QJQL], [QKLL], [PHIA], [M11]; !
190 1: [PH1], [PH1], [PH1], [PH1]; !
191 1: [PH1], [PH1], [PH1], [PH1], [PH1]; !
192 1: [PH1], [PH1], [PH1], [PH1], [PH1]; !
193 1: [PH1], [PH1], [PH1], [PH1], [PH1]; !
194 1: [PH1], [PH1], [PH1], [PH1], [PH1]; !
195 1: [PH1], [PH1], [PH1], [PH1], [PH1]; !
196 1: [PH1], [PH1], [PH1], [PH1], [PH1]; !
197 1: [PH1], [PH1], [PH1], [PH1], [PH1]; !
198 1: [PH1], [PH1], [PH1], [PH1], [PH1]; !
199 1: [PH1], [PH1], [PH1], [PH1], [PH1]; !
200 1: [PH1], [PH1], [PH1], [PH1], [PH1]; !
201 1: [PH1], [PH1], [PH1], [PH1], [PH1]; !

50
Standard MAPOL Sequence Listing — Continued

STAT LEVL
202 1:S
203 1:S******************************************************************************$!
204 1:S DECLARATIONS FOR GENERALIZED DYNAMIC REDUCTION (GDR) $!
205 1:S******************************************************************************$!
206 1:S $!
207 1:INTEGER LKSET, LJSET, NEIV, GNORM, NGDR, $!
208 1! ASIZE, LSIZE; $!
209 1:REAL FNAX; $!
210 1:RELATION DYNRED; $!
211 1:MATRIX [PGDRG(1000)], [PHI0X], [X00], [GG0], [XSOO], $!
212 11 [KO], [LS00], [PAJK], [PFJK], [UPGDR], $!
213 11 [AFGDR], [UJK], [GTMF], $!
214 11 $!
215 11******************************************************************************$!
216 11 $!
217 11******************************************************************************$!
218 11 $!
219 11 REAL BQDP; $!
220 11 MATRIX [MPART], [ID2], [PHIE], [PHIE], [PHIB], $!
221 11 [GENM], [GENK], [GENF], [GENQ], [GENQL], $!
222 11 [DSLP], [PTF], [QRE], [QEE], [REQ], $!
223 11 [JLQ], [UQK], [GPR], [GPR], [GTPM], $!
224 11 [BLSTJAI], [BLSTJAI], [BPFC], [MATTR], [MATSS], $!
225 11 [KEE], [DELB], [DEL], [URDB], [GENFA], $!
226 11 [DINN], [ELAS], [SPECT], [QRR], [UBLAST]; $!
227 11 [UBLASTG], [UBLASTF]; $!
228 11 $!
229 11******************************************************************************$!
230 11 $!
231 11******************************************************************************$!
232 11 $!
233 11******************************************************************************$!
234 11 $!
235 11******************************************************************************$!
236 11 SINGOSET := 1; $!
237 11 SINGASET := 2; $!
238 11 SINGLET := 3; $!
239 11 $!
240 11 INITIALIZE SUBSCRIPT VALUES TO "1" TO AVOID RUN TIME PROBLEMS $!
241 11 $!
242 11 SUB := 1; $!
243 11 PRINT(’LOG’(’BEGIN PREFACE MODULES’)); $!
244 11 CALL SOLUTION (NUMOPTBC, NBNDCOND, MPS, MPE, OCS, OCE, FSDS, FSDE, $!
245 11 MAXITER, NOVL1, M:IPW0X, OCHCR:IM, ALPHA, CNVRGLIM, $!
246 11 NRFAC, EPS ); $!
247 11 CALL IFP (GSIZE); $!
248 11 $!
249 11 GENERATE THE ELEMENT MATRICES $!
250 11 $!
251 11 PRINT(’LOG’(’ELEMENT MATRIX GENERATION’)); $!
252 11 $!
Standard MAPOL Sequence Listing — Continued

```
STAT LEVEL
253 1!CALL NAKIST (NDV, GLBDES, [PTANS], [PRINT]. [PMAXT], LOCLVAR.
254 1! TFIXED, DESLINK );
255 1!$ $!
256 1!CALL EMG ( NDV, GSIZER, GLBDES, DESLINK, [SMAT], DVCT, DVSIZE.
257 1! KELM, HELM, TELM, TREF );
258 1!$ $!
259 1!CALL PFBULK ( GSIZER, EOSUMMY, EODISC, GPFELEM );
260 1!$ $!
261 1!$ $!
262 1!$ $!
263 1!$ $!
264 1!PRINT('LOG=('NON-PLANAR STEADY AERODYNAMICS')*);
265 1!CALL STEADYN ( NONONLY, AECOMPS, GEOMSA, STABCF, [AIRFORCE], AEROGEOM.
266 1! CAROGEOM, OAGRDLOD );
267 1!IF NONONLY CALL EXIT;
268 1!$ $!
269 1!$ $!
270 1!$ $!
271 1!$ $!
272 1!$ $!
273 1!$ $!
274 1!$ $!
275 1!$ $!
276 1!$ $!
277 1!$ $!
278 1!$ $!
279 1!$ $!
280 1!$ $!
281 1!$ $!
282 1!$ $!
283 1!$ $!
284 1!$ $!
285 1!$ $!
286 1!$ $!
287 1!$ $!
288 1!$ $!
289 1!$ $!
290 1!$ $!
291 1!$ $!
292 1!$ $!
293 1!$ $!
294 1!$ $!
295 1!$ $!
296 1!$ $!
297 1!$ $!
298 1!$ $!
299 1!$ $!
300 1!$ $!
301 1!$ $!
302 1!$ $!
303 1!$ $!
304 1!$ $!
```
1: IF NUMOPTBC > 0 THEN
2: PRINT('LOG=('BEGIN OPTIMIZATION')');
3: INITIALIZE MAPOL PARAMETERS
4: GLBCNVRG := FALSE;
5: APPCNVRG := FALSE;
6: BEGIN CONVERGENCE LOOP
7: WHILE NOT GLBCNVRG AND NITER <= MAXITER DO
8: ASSEMBLE THE GLOBAL MATRICES
9: BEGIN BOUNDARY CONDITION LOOP FOR OPTIMIZATION
10: FOR BC = 1 TO NUMOPTBC DO
11: PRINT('LOG=('BOUNDARY CONDITION',131',BC);
CALL BOUND (BC, GSIZB, ESIZE(BC), BLOAD, BMASS, DMODES, 
BMODES, BSAERO, BFLUTR, BGYN, BDRSF, BDTR, BNTR, BDFR, 
BNFR, BUST, BBLAST, NMPC, NMPIT, NMRIT, NRSET, NGRK); 

Determine if any M2GG/K2GG input data are to be added 

CALL NULLMAT ([KGG], [MGG]); 

CALL M2GG (BC, GSIZB, [M2GG], M2GGFLAG, [K2GG], K2GGFLAG ); 

IF M2GGFLAG THEN 
(MGG) := [M2GG] + [MGG]; 
ELSE 
(MGG) := [MGG]; 
ENDIF; 

IF K2GGFLAG THEN 
(MGG) := [M2GG] + [K2GG]; 
ELSE 
(MGG) := [K2GG]; 
ENDIF; 

CALL the grid point weight generator for this boundary condition 

CALL GPGW (NITER, BC, GPGWGGRID, [MGG], GCPWG ); 

IF BLOAD <> 0 CALL GTLOAD (NITER, BC, GSIZB, BGDPT(BC), GLEBES, 
SMPLD, [DPTHVI], [DPGRVI], [PG], OGRIDLOD); 

PARTITION-REDUCTION OF GLOBAL MATRICES 

IF NMPC <> 0 THEN 

[GTKN], [GSTKN], [UTKNI ]; 

IF NMPC <> 0 THEN 

PERFORM MPC REDUCTION 

PRINT("LOG("MPREDUCION");); 

CALL GREDUCE ( [KGG], [PG], [PONN(BC)], [TMN(BC)], [KN], [PN]); 

IF BMASS <> 0 CALL GREDUCE ( [MGG], [PONN(BC)], [TMN(BC)], [MN]); 

IF BSAERO <> 0 THEN 

CALL GREDUCE ( [GTKG], [PONN(BC)], [TMN(BC)], [GTKN]); 

CALL GREDUCE ( [GSTKG], [PONN(BC)], [TMN(BC)], [GSTKN]); 

ENDIF; 

IF BFLUTR <> 0 OR BGYST <> 0 OR BBLAST <> 0 

CALL GREDUCE ( [GTKN], [PONN(BC)], [TMN(BC)], [GTKK] ); 

CALL GREDUCE ( [GTKN], [PONN(BC)], [TMN(BC)], [GTKK] ); 

ELSE 

NO MPC REDUCTION 

[KNN] := [KGG]; 

IF BLOAD <> 0 [PN] := [PG]; 

IF BMASS <> 0 [MN] := [MGG]; 

IF BSAERO <> 0 THEN 

[GTKK] := [GTKG];
STAT LEVEL
409 6:
    [GSTKN] := [GSTKG];
ENDIF;
410 6:
END;
411 5:
    IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 [UGTKN] := [UGTKG];
ENDIF;
412 5:
413 4:
    IF NUMOPTBC > 1 CALL NULLMAT ([KFF]. [PF]. [MFF]. [GTKF]. [GSTKF],
        [UGTKF]);
414 5:
    IF NSPC <> 0 THEN
        PERFORM SPC REDUCTION
    ENDIF;
415 4:
416 5:
    IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 [UGTKN] := [UGTKG];
ENDIF;
417 5:
    IF NSPC <> 0 THEN
        PRINT('LOG- SPC REDUCTION');
        CALL NREDUCE ([KNN], [PFN], [PNF(BC)], [YS(BC)], [KFF], [KFS],
            [KSS], [PF], [PS]);
        IF BMASS <> 0 CALL NREDUCE ([MNN], [MNF(BC)], [MFP]);
        IF BSAERO <> 0 THEN
            CALL NREDUCE ([GKTN], [PNF(BC)], [GTKF], [GSTKF]);
        ENDIF;
    ELSE
        NO SPC REDUCTION
    ENDIF;
418 5:
420 5:
    PRINT('LOG- SPC REDUCTION');
    CALL NREDUCE ([KNN], [PFN], [PNF(BC)], [YS(BC)], [KFF], [KFS],
        [KSS], [PF], [PS]);
    IF BMASS <> 0 CALL NREDUCE ([MNN], [MNF(BC)], [MFP]);
    IF BSAERO <> 0 THEN
        CALL NREDUCE ([GKTN], [PNF(BC)], [GTKF], [GSTKF]);
    ENDIF;
421 5:
422 5:
    IF BLOAD <> 0 [PF] := [PN];
423 5:
    IF BSAERO <> 0 THEN
        CALL NREDUCE ([KAA], [PA], [MAA],
            [KAAA], [PAA], [UGTKA]);
    ENDIF;
424 5:
425 5:
    IF NGDR <> 0 THEN
        PRINT('LOG- DYNAMIC REDUCTION');
        CALL PARTN ([KFF], [KKO], [KOA], [PFOA(BC)]);
        CALL PARTN ([MFF], [MMO], [PFOA(BC)]);
        ASIZE := GSIZE - NMPC - NSPC - NOMIT;
        LSIZE := ASIZE - NRSET;
426 5:
427 5:
428 5:
429 5:
430 5:
431 5:
432 5:
433 5:
434 5:
435 5:
436 5:
437 6:
438 6:
439 6:
440 5:
441 5:
442 4:
443 4:
444 5:
445 4:
446 4:
447 4:
448 5:
449 5:
450 5:
451 5:
452 5:
453 5:
454 5:
455 5:
456 5:
457 5:
458 5:
459 5:
Standard MAPOL Sequence Listing — Continued

STAT LEVL

FMX, BC, BGPDT(BC), USET(BC), NOMIT, LSIZE );

LKSET MEANING

< 0 APPROX. MODE SHAPES SELECTED

= 0 NO APPROX. MODE SHAPES IN GDR

IF LKSET <> 0 THEN

CALL SDCOMP ( [KSOO], [LSOO], USET(BC), SINGOSET );

CALL GDR2 ( [LSOO], [SOO], [PHIOK], LKSET, LJSET,

NEIV, FMX, BC );

ENDIF;

CALL GDR3 ( [KOO], [KO], [M1], [PHIO], [TMN(BC)], [GGO],

[PGMN(RC)], [PNSF(BC)], [PFOA(BC)], BGPDT(BC), USET(BC),

LKSET, LJSET, ASIZE, GNORM, BC );

CALL GDR4 ( BC, GSIZE, PSIZE(BC), LKSET, LJSET, NUMOPTBC, NBNDCOND,

[PGMN(BC)], [TMN(BC)], [PNSF(BC)], [PFOA(BC)], [PARL(BC)], [PDRG(BC)], [PAJK], [PFJK], BGPDT(BC),

USET(BC) );

ENDIF;

IF BLOAD <> 0 OR BMODES <> 0 OR BFLUTR <> 0 OR BDYN <> 0 THEN

REDUCE THE MATRICES WITHOUT AEROELASTIC CORRECTIONS

IF NGDR <> 0 THEN

PERFORM THE GENERAL DYNAMIC REDUCTION

PRINT("LOG=(' SYMMETRIC DYNAMIC REDUCTION ')");

[KAA] := TRANS ( [GSUBO(BC) ] ) * [ MFF ] * [GSUBO(BC) ];

[KAA] := TRANS ( [GSUBO(BC) ] ) * [ KFF ] * [GSUBO(BC) ];

IF BLOAD <> 0 OR [PA] :: TRANS ( [GSUBO(BC) ] ) * [PF ];

IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 THEN

[TMP1I := TRANS ( [UGTKF ] ) * [GSUBO(BC) ];

CALL TRANSPOSE ( [TMP1I, [UGTKA] );

ENDIF;

ELSE

IF NOMIT <> 0 THEN

PERFORM THE STATIC REDUCTION

PRINT("LOG=(' STATIC CONDENSATION ')");

CALL REDUCE ( [KFF], [PF], [PFOA(BC)], [KOOINV(BC)],

[GSUBO(BC)], [KAA], [PA], [PO], USET(BC) );

IF BMSS <> 0 THEN

PERFORM GUYAN REDUCTION OF THE MASS MATRIX

56
STAT LEVEL

CALL PARTN ([KFF], [MOO], [MOA], [MAABAR], [PFOA(BC)]);

[MAA] := [MAABAR] * TRANS([MOA]) * [GSUBO(BC)] * 
TRANS([GSUBO(BC)]) * [MOA] * 
TRANS([GSUBO(BC)]) * [MOO] * [GSUBO(BC)];

IF NRSET. <> 0 [IFM(BC)] := [MOO] * [GSUBO(BC)] * [MOA];

ENDIF;

IF BFOUT.< 0 OR BGUST <> 0 OR BBLAST <> 0 THEN
CALL ROWPAST ([UGTKP], [UGTKO], [UGTKAB], [PFOA(BC)]);

[TRANS] := TRANS([UGTKO]) * [GSUBO(BC)];

CALL TRANSPOSE ([TRANS], [TRANS]);

[UGTKA] := [UGTKAB] + [TRANS];

ENDIF;

ELSE

NO F-SET REDUCTION

[KAA] := [KFF];

IF BLOAD <> 0 [PA] := [PF];

IF BFOUT. <> 0 OR BGUST <> 0 OR BBLAST <> 0 [UGTKA] := [UGTKF];

IF BMASS <> 0 IMMAI;

ENDIF;

ENDIF;

ENDIF;

IF NRSET <> 0 THEN

PERFORM THE SUPPORT SET REDUCTION

PRINT('LOG=('SUPPORT REDUCTION');

IF NITER = 1 THEN

CALL PARTN ([KAA], [KRR], [KLR], [KLL], [PARL(BC)]);

CALL SDCOMP ([KLL], [KLLINV(BC)], USET(BC), SINGLSET);

CALL PBS ([KLLINV(BC)], [KLR], [D(BC)], -1);

CALL RBCHECK (BC, USET(BC), USET(BC), D(BC), [KLL],
[KRR], [KLR]);

ELSE

IF BLOAD <> 0 THEN

CALL PARTN ([KAA], [KLR], [KLL], [PARL(BC)]);

CALL SDCOMP ([KLL], [KLLINV(BC)], USET(BC), SINGLSET);

ENDIF;

ENDIF;

ENDIF;

CALL PARTN ([MBA], [MRRBAR], [MLR], [MRL], [PARL(BC)]);

[IFR(BC)] := [MRL] * [D(BC)] * [MLR];

[MRR(BC)] := [MRRBAR] * TRANS ([MLR]) * [D(BC)] * 
TRANS ([D(BC)]) * [IFR(BC)];

[R32] := TRANS ([D(BC)]) * [MLR] * [MRRBAR];

ENDIF;

IF BLOAD <> 0 THEN

PROCESS STATICS WITH INERTIA RELIEF

57
STAT LEVL
565 7:
566 7:
567 7:
568 7:
569 7:
570 7:
571 7:
572 7:
573 7:
574 7:
575 7:
576 7:
577 7:
578 7:
579 7:
580 7:
581 7:
582 7:
583 7:
584 7:
585 7:
586 7:
587 7:
588 7:
589 7:
590 7:
591 7:
592 7:
593 7:
594 7:
595 7:
596 7:
597 7:
598 7:
599 7:
600 7:
601 7:
602 5:
603 4:
604 5:
605 5:
606 5:
607 5:
608 5:
609 5:
610 5:
611 5:
612 6:
613 6:
614 6:
615 6:
616 6:
58

PRINT(***DISCIPLINE: STATICS(INERTIA RELIEF)***):
CALL ROWPART ([PA], [PR], [PLBAR], [PARL(BC)]);
[LHS(BC)] := [MRP(BC)];
[RHS(BC)] := TRANS([D(BC)]) * [PLBAR] + [PR];
CALL INERTIA ([LHS(BC)], [RHS(BC)], [AR]);
[AL] := [D(BC)] * [AR];
CALL ROWMERGE ([AA], [AR], [AL], [PARL(BC)]);
[RHS(BC)] := [PLBAR] - [IFR(BC)] * [AR];
CALL FBS ([KLLINV(BC)], [RHS(BC)], [UL]);
CALL YSMERGE ([UA], [UL], [PARL(BC)]);
ENDIF;
IF BMODES <> 0 THEN
PRINT('LOG(***DISCIPLINE: NORMAL MODES***));
CALL REIG (NITER, BC, USET(BC), [KAA], [MAA], [MRR(BC)]);
[D(BC)] , LAMBDA, [PHIA], [MII], HSIZEB(BC));
CALL OPFMRGE (NITER, BC, NUMOPTBC, LAMBDA);
CALL FCEVAL (NITER, BC, LAMBDA, CONST);
ENDIF;
ELSE
ENDIF;
ENDIF;
IF BLOAD <> 0 THEN
PRINT('LOG(***DISCIPLINE: STATICS***));
CALL SDCOMP ([KAA], [KLLINV(BC)], USET(BC), SINGASET);
CALL FBS ([KLLINV(BC)], [PA], [UA]);
ENDIF;
IF BMODES <> 0 THEN
PRINT('LOG(***DISCIPLINE: NORMAL MODES***));
CALL REIG (NITER, BC, USETBC, [KAA], [MII], LAMBDA, [PHIA], [MII], HSIZEB(BC));
CALL OPFMRGE (NITER, BC, NUMOPTBC, LAMBDA);
CALL FCEVAL (NITER, BC, LAMBDA, CONST);
ENDIF;
ENDIF;
ENDIF;
IF BSAERO <> 0 THEN
PERFORM STATIC AEROELASTIC ANALYSES
PRINT('LOG(***SAERO INITIALIZATION***)');
CALL TRNSPOSE ([GSTKF], [GSKF]);
LOOP := TRUE;
SUB := 0;
WHILE LOOP DO
SUB := SUB + 1;
CALL SAERODRV (BC, SUB, LOOP, MINDEX, SYM, MACH, QDP, 1); ADJUST THE KFF MATRIX AND DETERMINE THE RIGID AIR LOADS
IF SYM = 1 [AICS] := [GTKF]*[TRANS([AICMAT(MINDEX)])]*[GSKF]]; $1$

IF SYM = -1 [AICS] := [GTKF]*[TRANS([AICMAT(MINDEX)])]*[GSKF]]; $1$

[PAF] := (QDP) [GTKF] * [AIRFRC(MINDEX)]]; $1$

[KAFF] := [KFF] - (QDP) [AICS]; $1$

REDUCE THE MATRICES WITH AEROELASTIC CORRECTIONS $1$

SAVE THE SUBCASE/BC DEPENDENT DATA FOR SENSITIVITY ANALYSIS $1$

IF NGDR <> 0 THEN $1$

PERFORM THE GENERAL DYNAMIC REDUCTION $1$

PRINT("LOG=(' SAERO DYNAMIC REDUCTION')"); $1$

PRINT("LOG=(' SAERO STATIC CONDENSATION')"); $1$

IF NITER = 1 AND SUB = 1 AND NRSET <> 0 AND BLOAD = 0 AND BMODES = 0 AND BFUTR = 0 AND BDYN = 0 THEN $1$

FORM [KAA] ON FIRST PASS SO [E] CAN BE FORMED $1$


ENDIF; $1$

IF BMSS <> 0 THEN $1$

PERFORM GUYAN REDUCTION OF THE MASS MATRIX $1$

CALL PARTN ([KFF]. [MOO]. [MOA]. [MAABAR]. [MOO]). $1$

[MAAA] := [MAABAR] * TRANS([MOA]) * [GASUBO(BC,SUB)] + $1$

[MOA] + $1$

TRANS([GASUBO(BC,SUB)]) * [GASUBO(BC,SUB)]; $1$

IF NRSET <> 0 $1$

[IFMA(BC,SUB)] := [MOO]*[GASUBO(BC,SUB)] + [MOA]; $1$

ENDIF; $1$

ELSE $1$

ENDIF; $1$

ENDIF; $1$
STAT LEVL

NO F-SET REDUCTION

IF NITER = 1 AND SUB = 1 AND NRSET <> 0 AND BLOAD = 0 AND BMODES = 0 AND BFLTR = 0 AND BOND = 0 THEN

FORM [KAA] ON FIRST PASS SO [D] CAN BE FORMED

[KAA] := [KFF];

ENDIF;

IF NRSET <> 0 THEN

PERFORM THE SUPPORT SET REDUCTION

PRINT("LOG=(' SAERO SUPPORT REDUCTION')*");

IF NITER = 1 AND SUB = 1 AND BLOAD = 0 AND BMODES = 0 AND BFLTR = 0 AND BOND = 0 THEN

[D] WAS NOT COMPUTED FOR NON-SAERO DISCIPLINES SO NEED TO COMPUTE IT NOW

CALL PARTN([KAA],[KRR],[KLR],[KLL],[PARL(BC)]);

CALL SDCOMP([KLL],[KLLINV(BC)],USET(BC),SINGLSET);

CALL FBS([KLLINV(BC)],[KLR],[DIBC],-1);

CALL RBCHECK(BC,USET(SC),BGPDT(SC),[D(SC)],[KLL],[KRR],[KLR]);

ENDIF;

CALL PARTN([KAAA],[MRBAR],[MLR],[MLL],[PARL(BC)]);

[R13(BC,SUB)] := [MLL] * [D(BC)] + [MLR];

[R3] := [MRBAR] + TRANS([MLR]) * [D(BC)] +

TRANS([D(BC)] *[R13(BC,SUB)]);

[R22] := TRANS([D(BC)] *[MLR] + [MRBAR]);

CALL TRNSPOSE([R13(BC,SUB)],[R21(BC,SUB)]);

PROCESS STEADY AEROELASTIC DISCIPLINE

PRINT("LOG=(' >>>DISCIPLINE: STEADY AERO')*");

CALL PARTN([KAAA],[KRR],[R12(BC,SUB)],[KARL],[R11],[PARL(BC)]);

[R32(BC,SUB)] := TRANS([D(BC)]) *[R12(BC,SUB)] + [KARR];

[R31(BC,SUB)] := TRANS([D(BC)]) *[R11] + [KARL];

CALL DECOMP([R11],[R111(BC,SUB)],[R111(BC,SUB)]);
CALL ROWPART ([PAA], [PARBAR], [PAL], [PARL(BC)])

CALL GFBS ([RL11(BC,SUB)], [RU11(BC,SUB)], [PAL],
          [R11PAL(BC,SUB)], -1);

(PRIGID) := [PARBAR] + TRANS(D(BC)) * [PAL];
(P1) := [R21(BC,SUB)] * [R11PAL(BC,SUB)];
(P2) := [PRIGID] * [R31(BC,SUB)] * [R11PAL(BC,SUB)];

CALL GFBS ([RL11(BC,SUB)], [RU11(BC,SUB)], [R112(BC,SUB)],
          [K11], [K112(BC,SUB)], [K112(BC,SUB)], -1);

[LHSA(BC,SUB)] := [K22] * [K21(BC,SUB)] * [K112(BC,SUB)];
[RHSA(BC,SUB)] := [P2] - [K21(BC,SUB)] * [PAR(BC,SUB)];

CALL SAERO (NITER, BC, MINDEX, SUB, SYM, QDP, STABCF,
            [AAR], [DELTA(SUB)], [PRIGID], [R33],
            [AAL], [DELTA(SUB)], [PRIGID], [R33],
            [AAL], [DELTA(SUB)], [PRIGID], [R33],
            [AAL], [DELTA(SUB)], [PRIGID], [R33],
            [AAL], [DELTA(SUB)], [PRIGID], [R33],
            [AAL], [DELTA(SUB)], [PRIGID], [R33],
            [AAL], [DELTA(SUB)], [PRIGID], [R33],
            [AAL], [DELTA(SUB)], [PRIGID], [R33],
            [AAL], [DELTA(SUB)], [PRIGID], [R33],
            [AAL], [DELTA(SUB)], [PRIGID], [R33],
            [AAL], [DELTA(SUB)], [PRIGID], [R33],
            IF NOMIT <> 0 [PAO(SUB)] := [POARO(BC,SUB)] * [DELTA(SUB)];
            IF AEFLG(SUB) THEN
                [AAL] := [D(BC)] * [AAR];
                CALL ROWMERGE ([AAASUB], [AAR], [AAL], [PARL(BC)]);
                [UAR] := [K112(BC,SUB)] * [AAR] + [PAR(BC,SUB)] *
                        [DELTA(SUB)];
                [UAL] := [R112(BC,SUB)] * [UAR] + [R1113(BC,SUB)] * [AAR] -
                        [R11PAL(BC,SUB)] * [DELTA(SUB)];
                CALL ROWMERGE ([UAA(SUB)], [UAR], [UAL], [PARL(BC)]);
                IF NOMIT <> 0 [PAO(SUB)] := [POARO(BC,SUB)] * [DELTA(SUB)];
            ENDIF;
            ELSE
                NO SUPPORT SET REDUCTION
            ENDIF;

PROCESS STEADY AEROELASTIC DISCIPLINE
Standard MAPOL Sequence Usting - Continued

STAT LEVL
773
7!$
774
7!
775
7!$$
776
7!
777
6!
778
5!
779
4!$
780
4!$
781
4!$
782
4!$
783
4!
784
5!
785
6!

736

6!

787
783
719
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824

6!
6!
7!
71
7!
7!
7!
71
7!
7!
7!
7!
7!
7!
7!
7!
7!
6!
5!$
5!
6!
71
7!
6!
7!
7!
6!
6!
6!
6!
6!
6!
6!
6!

6!
6!
6!
6!

PRINT('LOG=(

>>>DISCIPLINE: STEADY AERO')');

ENDIF;
ENDDO3
ENDIP;
PERFORM ANY DYNAMIC ANALYSES
OF THE SUPPORT SET
IF NDYN <>0 THEN
IF BFLUTR <> 0 THEN
PRINT('LOG=('

SUN

--

NOTE THAT THESE ARE INDEPENDENT

>>>DISCIPLINE:

S!
S
5

FLUTTER')');

0;

LOOP
TRUE;
WHIL.E LOOP DO
SUN :=53
1UN
CALL FLUTDRV
SC, SUN3,LOOP C
CALL FLUTOHHL CNITER. BC. SUB8,ESIZE(BC), PSIZE(BC). rOKKL].
IUOTKA). (PHIA]. USET(BC),
[TMN(BC)I. fGSU3O(BCCJ. NGDR, AECOMPU, GEOMUA.
[PHIKH], [QHHLFL(BC,SUDC I. OAGRDDSP );
CALL FLUTDMA CNITER, 3C, SUB, ESIZEC3C), PSIZECBC),
3GPDrumcDCC USETIUC). [NAAC. (KAAI. (TMNNICDCC,
(GSCJSO(3CC]. NGDR, LAMBDA, [PHIA).
IMHHFL(3C,SUBC], (3HHFL(&CSUBCI. [KHHFLCDC,SUBCC
;
CALL FLUTTRAN ( NITER, 5C, SUB3,(QHHLFL(BC.SUBCI, LAMBDA.
HSIZE(C)C, ESIZECIC), IMHHFLCBC,SUSC J,
(BHHFL(DC,SC.CDI,
(J(HHFL(MCSUB)].
CLAMBDA, CONST C
ENDDO;
ENDIF;
S
IF XDRSP <> 0 THIEN
IF BMTR
<>0 OR DTR <> 0THEN
PRINTCILOG=C'
>>>DISCIPLINE: TRANSIENT RESPONSE')*);
ENDIF;
IF BMFR ý>0 OR SDFR <>0 THEN
PRINT('LOG=('
>>>DISCIPLINE: FREQUENCY RESPONSE*)';
ENDIF;
CALL QHHLGEN (BC, ESIZE(BC), [OKKL). (QKJLI, (UGTKA). (PHIA).
IPHII(H), (OHHLL, IOHJLJC;
(MAA(, ýKAAI, (TMNCDCC)., [GSUBO(CDCJ, NGDR.
LAMBDA, [PHIA), 1MODDC.(BDDI. (KDDTJ. (KDDF1.
[MHH), [BHH].
[KHHT]. lKHHFJ );
CALL DYNLOAD ( NITER, BC, GSIZE. ESIZE(BC), PSIZE(BC). SMPLOD,
BGPD'rNCUC,
USETBC3CC
lTMN(BCCI. (GSUDO(CDCCI
NGDP. [PHIAl. (QHJL]. (PDTJ. IPDF),
JPTGLOADI, [PTHLOADI. IPFGLOAD). (PFHLOADJ C
CALL DYNRSP (8C, ESIZECUC), [MDDI, (UDDI. (KDDTI. (KDDF).
(MHH), (8HHJ, (KHHT), [KHHF], (PDTI. LPDF],

62


Standard MAPOL Sequence Listing — Continued

STAT LEVL, [QHHL], [UTRANA], [UFREQA], [UTRANI], [UFREQI],

IF BMTF <> 0 [UTRANA] := [PHIA] * [UTRANI];

ENDIF;

IF BMFR <> 0 [UFREQA] := [PHIA] * [UFREQI];

ENDIF;

IF BMTR <> 0 [UTRAGA], [UFREAG];

ENDIF;

IF OBLAST <> 0 THEN

PRINT("LOG=(' DISCIPLINE: BLAST')");

CALL BLASTFIT ( BC, [QJL], [MATTR], [MATSS], [BQDP], [BFRC], [DWMOP], [MIZEBC], [ID2], [MPART], [UCTKA], [BLGTJAJ], [BLSTJAJ]);

CALL COLPART ( [PHIA], [PHIE], [MPART]);

CALL ROWERGE ( [PHIR], [ID2], [DIBC], [PARLBC]);

CALL COLMERGE ( [PHIB], [PHIR], [PHIE], [MPART]);

[GENM] := TRANS ( [PHIB] ) * [KAA] * [PHIB];

[GENK] := TRANS ( [PHIB] ) * [KAA] * [PHIB];

[TSPL] := TRANS ( [JLSLTA] ) * [PHIB];

[ETF] := TRANS ( [PHIB] ) * [BLGTJAJ];

[GENF] := [BQDP] * [ETF] * [BFRC];

[GENFA] := [BQDP] * [ETF] * [MATSS];

[GENQ] := [GENFA] * [TSPL];

[GENQL] := [BQDP] * [ETF] * [MATSS];

CALL PARTN ( [GEN], [QRR], [QRE], [QEE], [MPART]);

[KEQE] := [QEE] * [KEE];

CALL DECOMP ( [KEQE], [LQK], [UQK]);

CALL ROWPART ( [GENQ], [GFR], [GFE], [MPART]);

CALL GFBS ( [LQK], [UQK], [GFE], [BTEM]);

[DEL] := [QRE] * [BTEM] * [GFR];

CALL BLASTRIM ( BC, [DELM], [XRRBC], [URDB], [DELB]);

[ELAS] := [BTEM] * [DELB];

[SLPMOD] := TRANS ( [BLSTJAJ] ) * [PHIE];

CALL BLASTDRV ( BC, [GENM], [GENK], [GENFA], [GENQL], [DELB]);

[UKDB], [OWNMSH], [SLPMOD], [ELAS], [UBLASTI]);

ENDIF;

BEGIN THE DATA RECOVERY OPERATIONS;

PRINT("LOG=(' DATA RECOVERY')");

IF NUMOPTBC > 1 CALL NULLMAT ([UF], [AP], [PHIF], [UTRANF], [UFREQF]);

IF NGDR <> 0 THEN

DATA RECOVERY WITH GDR

APPEND THE GDR-GENERATED DOFS TO THE F-SET;

PRINT("LOG=(' DYNAMIC REDUCTION RECOVERY')");

IF BLOAD <> 0 THEN

[UPGDR] := [GSUBOBC] * [UA];

CALL ROWPART ( [UA], [UJX], [PAJX]);

CALL ROWERGE ( [UF], [UJX], [UPGDR], [PFJK]);

IF NRSET <> 0 THEN

[AFGDR] := [GSUBOBC] * [AA];

63
CALL ROWPART ([AA], [UJK], [PAJK]);
CALL ROWMERGE ([AF], [UJK], [APGD], [PFJK]);
ENDIF;
IF BSAERO <> 0 THEN
FOR S = 1 TO SUB DO
  [UFGR] := [GSUBO[BC]] * [UAA(S)];
  CALL ROWPART ([UAA(S)], [UJK], [PAJK]);
  CALL ROWMERGE ([UAFMP], [UJK], [APGD], [PFJK]);
ENDIF;
MERGE THE CURRENT SUBCASE DEPENDENT RESULTS INTO A SINGLE $;
MATRIX OF RESPONSE QUANTITIES FOR FURTHER RECOVERY $;
CALL SAEROMRG ( BC, S, [UAF], [UAFMP] );
IF NRSET <> 0 THEN
  [APGD] := [GSUBO[BC]] * [AAA(S)];
  CALL ROWPART ([AAA(S)], [UJK], [PAJK]);
  CALL ROWMERGE ([UAFMP], [UJK], [APGD], [PFJK]);
  CALL SAEROMRG ( BC, S, [AAF], [UAFMP] );
ENDIF;
IF AEFLG(S) THEN
  [UFGDR] := [GSUBO(BC)] * [UAAC(S)];
  CALL ROWPART ([UAAC(S)], [UJK], [PAJK]);
  CALL ROWMERGE ([UAFMP], [UJK], [APGD], [PFJK]);
ENDIF;
ENDIF;
IF BMODES <> 0 THEN
  [UFGR] := [GSUBO[BC]] * [PHIA];
  CALL ROWPART ([PHIA], [UJK], [PAJK]);
  CALL ROWMERGE ([PHIF], [UJK], [APGD], [PFJK]);
ENDIF;
IF BDFR <> 0 OR BMFR <> 0 THEN
  [UFGR] := [GSUBO[BC]] * [UTRANA];
  CALL ROWPART ([UTRANA], [UJK], [PAJK]);
  CALL ROWMERGE ([UTRANF], [UJK], [APGD], [PFJK]);
ENDIF;
IF BDFR <> 0 OR BMFR <> 0 THEN
  [UFGR] := [GSUBO[BC]] * [UFREQA];
  CALL ROWPART ([UFREQA], [UJK], [PAJK]);
  CALL ROWMERGE ([UFREQF], [UJK], [APGD], [PFJK]);
ENDIF;
ELSE
  IF NOMIT <> 0 THEN
    DATA RECOVERY WITH STATIC CONDENSATION $;
    PRINT('LOG=i' STATIC CONDENSATION RECOVERY');
    IF BLOAD <> 3 THEN
      $;
    $;
  $;
  $;
$;
STATE LEVEL

CALL RECOVA ( [UA], [PO], [GSUBO(BC)], NRSET, [AA],
([IPM(B)],[KOOINV(B)],[PFOA(B)]), [UP]

IF NRSET <> 0 CALL RECOVA ( [AA], [GSUBO(BC)],
[PFOA(B)], [AF] );

ENDIF;

IF BSAERO <> 0 THEN
FOR S = 1 TO SUB DO
CALL RECOVA ( [UAA(S)], [PAO(S)], [GSUBO(BC,S)],
[NRSET, [AA(S)], [IPMA(B,S)], BSAERO,
[KOOL(B,S)], [KOOU(B,S)],
[PFOA(B)], [UAPTMP] );
MERGE THE CURRENT SUBCASE DEPENDENT RESULTS INTO A SINGLE
MATRIX OF RESPONSE QUANTITIES FOR FURTHER RECOVERY
CALL SAEROMRG ( BC, S, [UAP], [UAPTMP] );
ENDIF;

IF AEFLG(S) THEN
CALL RECOVA ( [UAAC(S)], [PAAC(S)], [GSUBO(BC,S)],
[NRSET, [AAAC(S)], [IPMA(B,S)], BSAERO,
[KOOL(B,S)], [KOOU(B,S)],
[PFOA(B)], [UAPC(S)] );
CALL RECOVA ( [AAAC(S)], [GSUBO(BC,S)],
[PFOA(BC)], [AAFC(S)] );
ENDIF;

ENDIF;

ENDIF;

IF BMODES <> 0 THEN
PHI0 := [GSUBO(BC)] * [PHIA];
CALL ROWMERGE ( [PHIF], [PHI0], [PHIA], [PFOA(B)]

ENDIF;

IF BDRY <> 0 OR BMTR <> 0 THEN
CALL RECOVA ( [UTRANA], [GSUBO(BC)],
[PFOA(B)], [UTRANF] );
ENDIF;

IF BDFR <> 0 OR BMFR <> 0 THEN
CALL RECOVA ( [UFREQA], [GSUBO(BC)],
[PFOA(B)], [UFREQP] );
ENDIF;

ELSE
REFERENCE DATA RECOVERY WITHOUT P-SET REDUCTION
ENDIF;

IF BLOAD <> 0 THEN
[UP] := [UA];
IF NRSET <> 0 [AF] := [AA];
ENDIF;

IF BSAERO <> 0 THEN
STAT LEVL

FOR S = 1 TO SUB DO

MERGE THE CURRENT SUBCASE DEPENDENT RESULTS INTO A SINGLE
MATRIX OF RESPONSE QUANTITIES FOR FURTHER RECOVERY

CALL SAEROMRG ( BC, S, [UAAC(S)];
IF NRSET <> 0 CALL SAEROMRG ( BC, S, [AAAS(S)];
IF AEXLG(S) THEN
[AAFC(S)] := [AAAC(S)];
ENDIF;
ENDIF;
ENDIF;
IF BMODES <> 0
IF BDTR <> 0 OR BMTR <> 0 [UTRANA];
IF BDFR <> 0 OR BMFR <> 0 [UFREQ];
ENDIF;
ENDIF;
IF BMODES <> 0
IF BDTR <> 0 OR BMTR <> 0 [UTRANA];
IF BDFR <> 0 OR BMFR <> 0 [UFREQ];
ENDIF;
ENDIF;
ENDIF;
IF BMODES <> 0
IF BDTR <> 0 OR BMTR <> 0 [UTRANA];
IF BDFR <> 0 OR BMFR <> 0 [UFREQ];
ENDIF;
ENDIF;
IF BMODES <> 0
IF BDTR <> 0 OR BMTR <> 0 [UTRANA];
IF BDFR <> 0 OR BMFR <> 0 [UFREQ];
ENDIF;
ENDIF;
De...
STAT LEVL
1033 5!
1034 6!
1035 6!
1036 5!
1037 6!
1038 6!
1039 5!
1040 6!
1041 6!
1042 6!
1043 6!
1044 6!
1045 5!
1046 5:5!
1047 5:5!
1048 5:5!
1049 5!
1050 6!
1051 6!
1052 6!
1053 5!
1054 6!
1055 6!
1056 6!
1057 7!
1058 8!
1059 8!
1060 8!
1061 7!
1062 6!
1063 5!
1064 5!
1065 5!
1066 5!
1067 4:5!
1068 4:4!
1069 5!
1070 4:4!
1071 4:
1072 5:5!
1073 5:5!
1074 5:5!
1075 5:
1076 5:
1077 6:
1078 6:
1079 6:
1080 7:
1081 7:
1082 7:
1083 6:
1084 5!

IF BDTR <> 0 OR BMTR <> 0
CALL YSMERGE ([UTRANN], [YS(BC)], [UTRANF],
[PNFSF(BC)], BDTR);

IF BDTR <> 0 OR BMTR <> 0
CALL YSMERGE ([UTRANN], [YS(BC)], [UTRANF],
[PNFSF(BC)], BDTR);

IF BDFR <> 0 OR BMFR <> 0
CALL YSMERGE ([UFREQN], [YS(BC)], [UFREQF],
[PNFSF(BC)], BDFR);

IF BBLAST > 0 THEN
JUBLAST := [PHIF]*[UBLASTI];
CALL OPPSPCF ( NITER, BC, 0, GSIZE, ESIZE(BC), NGDR,
KFS), [UBLASTF], [PNFSF(BC)], [PGMN(BC)],
[PFJK], 0, [GGRIDLOD]);

ENDIF;
ELSE

DATA RECOVERY WITHOUT SPC-REDUCTION

IF BLOAD <> 0 THEN
[UN] := [UF];
IF NRSET <> 0 [AN] := [AF];
ENDIF;

IF BSAERO <> 0 THEN
[UN] := [UF];
IF NRSET <> 0 [AN] := [AF];
FOR S = 1 TO SUB DO
IF AEFLG(S) THEN
[UANC(S)] = [UAFC(S)];
[AAFC(S)] = [AAFC(S)];
ENDIF;
ENDIF;

IF BMODES <> 0 [PHIN] := [PHIF];
IF BDTR <> 0 OR BMTR <> 0 [UTRANF] := [UTRANA];
ENDIF;

IF NUMOPTBC > 1 CALL NULLMAT ([UG(BC)], [AG(BC)], [UAG(BC)],
[GAAG(BC)], [PHIG(BC)]);

IF NMPC <> 0 THEN

PRINT("LOG=(' MPC RECOVERY')");

IF BLOAD <> 0 THEN
[UN] := [TMN(BC)] *[UN];
CALL ROWMERGE ([UG(BC)], [UN], [UN], [PGMN(BC)]);
ENDIF;

ENDDO;
ENDIF;

DATA RECOVERY WITH MPC-REDUCTION

IF BSAERO <> 0 THEN
Standard MAPOL Sequence Listing — Continued

STAT LEVL
1085 61 [UM] := [TMN(BC)] * [UAN];
1086 61 CALL ROWMERGE ([UAG(BC)], [UM], [UAN], [PGNN(BC)]);
1087 61 IF NRSET <> 0 THEN
1088 71 [UM] := [TMN(BC)] * [AAN];
1089 71 CALL ROWMERGE ([AAG(BC)], [UM], [AAN], [PGNN(BC)]);
1090 71 ENDIF;
1091 71 FOR S = 1 TO SUB DO
1092 71 IF AEFLO(S) THEN
1093 81 [UM] := [TMN(BC)] * [UANC(S)];
1094 81 CALL ROWMERGE ([UAGC(BC,S)], [UM], [UANC(S)], [PGNN(BC)]);
1095 81 [UM] := [TMN(BC)] * [AANC(S)];
1096 81 CALL ROWMERGE ([AAGC(BC,S)], [UM], [AANC(S)], [PGNN(BC)]);
1097 81 ENDIF;
1098 71 ENDOD;
1099 61 ENDIF;
1100 51 IF BNODES <> 0 THEN
1101 61 [UM] := [TMN(BC)] * [PHIN];
1102 61 CALL ROWMERGE ([PHIG(BC)], [UM], [PHIN], [PGNN(BC)]);
1103 61 ENDIF;
1104 51 IF BTR <> 0 OR BMTR <> 0 THEN
1105 61 [UM] := [TMN(BC)] * [UTRANN];
1106 61 CALL ROWMERGE ([UTRANG], [UM], [UTRANN], [PGNN(BC)]);
1107 61 ENDIF;
1108 51 IF BDTR <> 0 OR BMTR <> 0 THEN
1109 61 [UM] := [TMN(BC)] * [UFREQN];
1110 61 CALL ROWMERGE ([UFREOG], [UM], [UFREQN], [PGNN(BC)]);
1111 61 ENDIF;
1112 51 ELSE
1113 51 DATA RECOVERY WITHOUT NPC-REDUCTION
1114 51 DATA RECOVERY WITHOUT NPC-REDUCTION
1115 51 ELSE
1116 51 IF BLOAD <> 0 THEN
1117 61 [UG(BC)] := [UM];
1118 51 IF NRSET <> 0 [AG(BC)] := [AN];
1119 61 ENDIF;
1120 51 IF BSAERO <> 0 THEN
1121 61 [UAG(BC)] := [UAN];
1122 61 IF NRSET <> 0 [AAG(BC)] := [AAN];
1123 61 FOR S = 1 TO SUB DO
1124 71 IF AEFLO(S) THEN
1125 81 [UAGC(BC,S)] := [UANC(S)];
1126 81 [AAGC(BC,S)] := [AANC(S)];
1127 81 ENDIF;
1128 71 ENDOD;
1129 61 ENDIF;
1130 51 IF BMODES <> 0 [PHIG(BC)] := [PHIN];
1131 51 IF BTR <> 0 OR BMTR <> 0 [UTRANG] := [UTRANN];
1132 51 IF BDTR <> 0 OR BMTR <> 0 [UFREQN] := [UFREOG];
1133 51 ENDIF;
1134 41 RECOVER PHYSICAL BLAST DISCIPLINE DISPLACEMENTS
1135 41 RECOVER PHYSICAL BLAST DISCIPLINE DISPLACEMENTS
1136 41 RECOVER PHYSICAL BLAST DISCIPLINE DISPLACEMENTS

68
STAT LEVEL
1137 4: IF BBLAST <> 0 [UBLASTG] := [FHIG(BC)] * [UBLASTI];
1138 4: $!
1139 4: $!
1140 4: $!
1141 4: $!
1142 4: $!
1143 5: CALL DCEVAL ( NITER, BC, [UG(BC)], CONST );
1144 5: $!
1145 5: ENDIF;
1146 4: $!
1147 5: CALL DCEVAL ( NITER, BC, [UG(BC)], CONST );
1148 5: $!
1149 5: $!
1150 5: $!
1151 4: $!
1152 4: $!
1153 4: $!
1154 4: $!
1155 4: $!
1156 5: $!
1157 5: $!
1158 5: $!
1159 5: $!
1160 5: $!
1161 5: $!
1162 6: $!
1163 6: $!
1164 6: $!
1165 6: $!
1166 6: $!
1167 6: $!
1168 7: IF SYM = 1 THEN
1169 7: CALL OFFLOAD ( NITER, BC, MINDEX, SUB, GSIZE, BGPDT(BC),
1170 7: [GTKG], [GSTKG], QDP, [AIRFRC(MINDEX)],
1171 7: [DELTA(SUB)], [AIChat(MINDEX)],
1172 7: [UG(BC)], [WGG], [AAG(BC)], [KFS],
1173 7: [KSS], [UAF], [YS(BC)], [PNSF(BC)],
1174 7: [PGMN(BC)], [PFJK], NGDR, USET(BC),
1175 7: OGRIDLOD );
1176 7: ELSE
1177 8: CALL OFFLOAD ( NITER, BC, MINDEX, SUB, GSIZE, BGPDT(BC),
1178 8: [GTKG], [GSTKG], QDP, [AIRFRC(MINDEX)],
1179 8: [DELTA(SUB)], [AIChat(MINDEX)],
1180 8: [UG(BC)], [MGG], [AAG(BC)], [KFS],
1181 8: [KSS], [UAF], [YS(BC)], [PNSF(BC)],
1182 8: [PGMN(BC)], [PFJK], NGDR, USET(BC),
1183 8: OGRIDLOD );
1184 8: ENDIF;
1185 7: $!
1186 6: $!
1187 6: $!
1188 6: $!
Standard MAPOL Sequence Listing — Continued

STAT LEVL
1189  6:1
1190  6:1  IF SYM = 1 THEN
1191  7:1  CALL OFPAEROM (NITER, BC, MINDEX, SUB, GSIZE, GEOMSA,
1192  7:1  [GTK], [GSTK], GQP, [AIRFRC(MINDEX)],
1193  7:1  [DELTA(SUB)], [AICM(MINDEX)],
1194  7:1  [UAG(BC)], OGRIDLOD, OGRIDDSP );
1195  7:1  ELSE
1196  7:1  IF SYM = -1 THEN
1197  8:1  CALL OFPAEROM (NITER, BC, MINDEX, SUB, GSIZE, GEOMSA,
1198  8:1  [GTK], [GSTK], GQP, [AIRFRC(MINDEX)],
1199  8:1  [DELTA(SUB)], [AICM(MINDEX)],
1200  8:1  [UAG(BC)], OGRIDLOD, OGRIDDSP );
1201  8:1
1202  7:1  ENDIF;
1203  7:1  ENDDO;
1204  5:1  ENDF;
1205  4:1  IF BDRSP <> 0 THEN
1206  5:1  CALL OFPOLOAD (NITER, BC, BPOD(BC), PSIZE(BC), ESIZE(BC),
1207  5:1  [PPHLOAD], OGRIDLOD );
1208  5:1  IF BDTR <> 0 OR BMTR <> 0
1209  6:1  CALL OFPOFPC (NITER, BC, 5, 1, GSIZE, ESIZE(BC),
1210  6:1  [PPHLOAD], OGRIDLOD );
1211  6:1  ENDIF;
1212  6:1
1213  6:1
1214  6:1
1215  6:1  IF BDTR <> 0 OR BMTR <> 0
1216  6:1  CALL OFPOFPC (NITER, BC, 6, 2, GSIZE, ESIZE(BC),
1217  6:1  [PPHLOAD], OGRIDLOD );
1218  6:1
1219  6:1
1220  6:1
1221  5:1
1222  4:1
1223  4:1
1224  4:1
1225  4:1
1226  4:1
1227  4:1
1228  4:1
1229  4:1
1230  4:1
1231  4:1
1232  4:1
1233  3:1
1234  3:1
1235  3:1
1236  3:1
1237  3:1
1238  3:1
1239  3:1
1240  3:1
1241  3:1

SELECT ACTIVE CONSTRAINTS

PRINT('LOG=' SENSITIVITY ANALYSIS');
CALL ACTCON (NITER, MAXITER, NRPC, NRPAC, GLBDES, LOCLVAR, [TRANS],
EPS, APPCVIRG, GLCVIRG,
CTL, CTLMIN, CONST, [AMAT], DESHIST, PFLAG, OLOCALDV);
CALL DESPUNCH (BC, KSIZE(BC), NITER );
ENDO;

70
STAT LEVL
1241 3:
1242 3:
1243 4: IF GLBCNVRG OR NITER <= MAXITER THEN
1244 4:
1245 4:
1246 4:
1247 5:
1248 5:
1249 5:
1250 5:
1251 5:
1252 5:
1253 5:
1254 5:
1255 6:
1256 6:
1257 6:
1258 6:
1259 6:
1260 6:
1261 6:
1262 6:
1263 6:
1264 6:
1265 6:
1266 6:
1267 6:
1268 6:
1269 6:
1270 6:
1271 6:
1272 6:
1273 6:
1274 6:
1275 6:
1276 6:
1277 6:
1278 6:
1279 6:
1280 6:
1281 6:
1282 6:
1283 6:
1284 6:
1285 7:
1286 7:
1287 7:
1288 7:
1289 7:
1290 7:
1291 7:
1292 7:

IF GLCMVRG OR NITER > MAXITER THEN
LAST ITERATION OUTPUT
FOR BC = 1 TO NUMOPTBC DO
CALL CFPMROOT ( NITER, BC, NUMOPTBC, LAMBDA, 1 );
CALL OFPDISP ( NUMOPTBC, BC, NITER, GSIZE, BGPDT(BC), ESIZE(BC),
PSIZE(BC), OGRIDDSP, ......... , LAMBDA...., 1 );
CALL OFPEDR ( BC, HSIZE(BC), NITER, 1 );
ENDIF;
IF NOT GLBCNVRG AND NITER <= MAXITER THEN
USE APPROPRIATE RESIZING METHOD
IF NITER >= FSDS AND NITER <= FSDE THEN
CALL FSD ( NDV, NITER, FSDS, FSDE, MPS, OCS, ALPHA, CNVRGLIM, GLBDES, LOCLVAR, [PTRANS], CONST, APPCNVRG, CTL, CTLMIN, DESHIST );
ENDIF;
IF ( NITER >= MPS AND NITER <= MPE ) OR ( NITER >= OCS AND NITER <= OCE ) THEN
USE MATHEMATICAL PROGRAMMING OR OC METHODS
OBTAIN THE SENSITIVITIES OF THE CONSTRAINTS WRT THE DESIGN VARIABLES
CALL MAXDFV ( NITER, NDV, [PINT], [PMAINT], CONST, [AMAT] );
CALL LAMINSNS ( NITER, NDV, C "DES, LOCLVAR, [PTRANS], CONST,
[AMAT] );
SENSITIVITY EVALUATION FOR BOUNDARY CONDITION DEPENDENT CONSTRAINTS!
FOR BC = 1 TO NUMOPTBC DO
CALL ABOUND ( NITER, BC, CONST, ACTBOUND, NAUS, NACSD, [PGA],
PCAS, ACTAERO, ACTDYN, ACTFLUT, NMPC, NSPC, NOMIT, NRSET, NUSR, USET(BC) );
IF ACTBOUND THEN
REESTABLISH THE BASE USET AND PARTITIONING DATA FOR THE BC
IF GDR CHANGED IT
NOTE. THIS LEAVES AN INCOMPATIBILITY BETWEEN USET(BC) AND BGPDT(BC) SINCE THE LATTER IS NOT REGENERATED.
THIS INCOMPATIBILITY WILL NOT AFFECT THE SENSITIVITY ANALYSIS!
AND WILL BE CORRECTED IN THE SUBSEQUENT ANALYSIS

IF NGDR <> 0 THEN
    CALL MKUSST(BC, GSIZEB, [YS(BC)], [TMN(BC)], [PMON(BC)],
               [PMSF(BC)], [PFOA(BC)], [PARL(BC)], USET(BC));
ENDIF;

EVALUATE FREQUENCY CONSTRAINT SENSITIVITIES

IF ACTDYN THEN
    IF NGDR <> 0 THEN
        CALL ROWPART ([PHIG(BC)], [GTMPL], [PGDRG(BC)]);
        CALL FREQSENS (NITER, BC, NDV, GLBDES, CONST, LAMBDA,
                        GMKCT, DKVI, GMKCT, DMVI,
                        [GTMPL], [AMAT]);
    ELSE
        CALL FREQSENS (NITER, BC, NDV, GLBDES, CONST, LAMBDA,
                        GMKCT, DKVI, GMKCT, DMVI,
                        [PHIG(BC)], [AMAT]);
    ENDIF;
ENDIF;

EVALUATE FLUTTER CONSTRAINT SENSITIVITIES

IF ACTFLT THEN
    SUB := 0;
    LOOP := TRUE;
    IF NGDR <> 0 CALL ROWPART ([PHIG(BC)], [GTMPL], [PGDRG(BC)]);
    WHILE LOOP DO
        SUB := SUB + 1;
        IF NGDR <> 0 THEN
            CALL FLUTSENS (NITER, BC, SUB, LOOP, GSIZEB, NDV,
                            GLBDES, CONST, GMKCT, DKVI, GMKCT,
                            GMKCT, LAMBDA, DMVI, LAMBDA,
                            [QHHLFL(BC, SUB)],
                            [MHHFL(BC, SUB)], [BHHFL(BC, SUB)],
                            [KHHFL(BC, SUB)], [GTMPL], [AMAT]);
        ELSE
            CALL FLUTSENS (NITER, BC, SUB, LOOP, GSIZEB, NDV,
                            GLBDES, CONST, GMKCT, DKVI, GMKCT,
                            GMKCT, LAMBDA, DMVI, LAMBDA,
                            [QHHLFL(BC, SUB)],
                            [MHHFL(BC, SUB)], [BHHFL(BC, SUB)],
                            [KHHFL(BC, SUB)], [PHIG(BC)], [AMAT]);
        ENDIF;
    ENDDO:
ENDIF;

EVALUATE ACTIVE DISPLACEMENT DEPENDENT CONSTRAINTS FROM
    THE STATICS DISCIPLINE

IF NAUS > 0 THEN
    EVALUATE SENSITIVITIES OF CONSTRAINTS WRT DISPLACEMENTS FOR STATICS:
CALL NULLMAT ([DFDU], [DPGV]);
IF NACSD > NAUS * NDV THEN
USES GRADIENT METHOD
CALL MAKDFU (NITER, BC, GSIZEB, [SMAT], [GLBSIG],
CONST, [DFDU]);
ELSE
USES VIRTUAL LOAD METHOD
CALL MAKDFU (NITER, BC, GSIZEB, [SMAT], [GLBSIG],
CONST, [DPGV]);
ENDIF;
SOME RELATIVELY SIMPLE CALCULATIONS THAT PRECEDE THE LOOP ON THE DESIGN VARIABLES
IF NGDR <> 0 THEN
CALL PARTN ([UG(BC)],...., [UGA], [PGAS], [PGDRG(BC)]);!
ELSE
CALL COLPART ([UG(BC)],...., [UGA], [PGAS]);!
ENDIF;
OBTAIN THE SENSITIVITIES OF THE DESIGN DEPENDENT LOADS
CALL DOLOAD (NDV, GSIZEB, BC, SMPLOD, DDFLG, [PGAS],[DPI]);!
CALL MAKDVU (NITER, NDV, GLBDES, [UGA], [DKUG],
GMMCT, DKVI);
CALL NULLMAT ([DUG], [DKUG]);
IF NRSET <> 0 THEN
CALL PARTN ([AG(BC)],...., [AGA], [PGAS], [PGDRG(BC)]);!
ELSE
CALL COLPART ([AG(BC)],...., [AGA], [PGAS]);
ENDIF;
CALL MAKDVU (NITER, NDV, GLBDES, [AGA], [DMAG],
GMMCT, DMVI);
[DUG] := [DKUG] + [DMAG];
ELSE
[DUG] := [DKUG];
ENDIF;
ACCOUNT FOR VIRTUAL LOAD METHOD
IF NACSD > NAUS * NDV THEN
USES GRADIENT METHOD
STAT LEVEL
1397 9:
1398 10:
1399 10:
1400 10:
1401 10:
1402 9:
1403 9:
1404 9:
1405 9:
1406 9:
1407 10:
1408 10:
1409 10:
1410 10:
1411 9:
1412 8:
1413 8:
1414 8:
1415 8:
1416 8:
1417 9:
1418 9:
1419 9:
1420 9:
1421 8:
1422 8:
1423 8:
1424 9:
1425 9:
1426 9:
1427 9:
1428 8:
1429 8:
1430 8:
1431 9:
1432 9:
1433 9:
1434 10:
1435 10:
1436 10:
1437 10:
1438 10:
1439 10:
1440 9:
1441 8:
1442 8:
1443 9:
1444 9:
1445 9:
1446 9:
1447 9:
1448 9:

IF DDFLG > 0 THEN
   [DPFV] := [DPVJ] + [DUG];
ELSE
   [DPFV] := [DUG];
ENDIF;
ELSE
   USE VIRTUAL LOAD METHOD
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
   $!
STAT LEVEL
1449 9:
1450 9:
1451 9:
1452 9:
1453 9:
1454 9:
1455 9:
1456 9:
1457 9:
1458 9:
1459 9:
1460 9:
1461 9:
1462 9:
1463 9:
1464 9:
1465 9:
1466 9:
1467 9:
1468 9:
1469 9:
1470 9:
1471 9:
1472 9:
1473 9:
1474 9:
1475 9:
1476 9:
1477 9:
1478 9:
1479 9:
1480 9:
1481 9:
1482 9:
1483 9:
1484 9:
1485 9:
1486 9:
1487 9:
1488 9:
1489 9:
1490 9:
1491 9:
1492 9:
1493 9:
1494 9:
1495 9:
1496 9:
1497 9:
1498 9:
1499 9:
1500 9:

{DULD} := [D(BC)] * [DURD];
CALL ROWNERGE ([DUAD], [DURD], [DULD], [PARL(BC)]);
{DPLV} := [DPLV] * [IFR(BC)] * [DURD];
CALL FBS ([KLLINV(BC)], [DPLV], [DULD]);
CALL YSMERGE ([DUAV], [DULV], [PARL(BC)]);
ELSE
CALL FBS ([KLLINV(BC)], [DPAV], [DUAV]);
ENDIF;

RECOVER TO THE F SET
CALL NULLMAT ([DUFV]);
IF NGDR <> 0 THEN
[DUFV] := [GSUBO(BC)] * [DUAV];
ELSE
IF NOMIT <> 0 THEN
IF NRSET <> 0 THEN
{TMP1} := [DPOV] - [IFM(BC)] * [DUAD];
ELSE
{TMP1} := [DPOV];
ENDIF;
CALL F3S ([KOOINV(BC)], {TMP1}, [UO]);
[UU] := [GSUBO(BC)] * [DUAV] + [UO];
CALL ROWNERGE ([DUFV], [UU], [DUAV], [PFON(BC)]);
ELSE
[DUFV] := [DUAV];
ENDIF;
ENDIF;

REDUCE THE LEFT HAND SIDE MATRIX
IF NMPC <> 0 THEN
CALL GREDUCE ([DFDU], [PFON(BC)], [TMN(BC)], [DFDUN]);
ELSE
[DFDUN] := [DFDU];
ENDIF;

IF NSPC <> 0 THEN
CALL ROWPART ([DFDUN], [DFDUF], [PFON(BC)]);
ELSE
[DFDUF] := [DFDUN];
ENDIF;

ACCOUNT FOR VIRTUAL LOAD METHOD
IF NACSD > NAUS * NDV THEN
USE GRADIENT METHOD
CALL MKAMAT ([AMAT], [DFDUF], [DUFV], PCAS, [PGAS]);
ELSE

75
USE: VIRTUAL LOAD METHOD

CALL MYAMAT ([AMAT], [DFU], [DPDF], [PCAS], [PGAS]);

ENDIF;

END IF ON ACTIVE APPLIED STATIC LOADS

EVALUATE ACTIVE CONSTRAINTS FROM

THE STATIC AEROELASTICITY DISCIPLINE

IF ACTAERO THEN

LOOP := TRUE;

ACTAAG := FALSE;

SUB := 0;

CALL NULLMAT ([DFU]);

WHILE LOOP DO

SUB := SUB + 1;

CALL AROSNSDR (NITER, BC, SUB, LOOP, MINDEX, CONST.

SYM, NGDR. 

[PGDG(BC)], [UAG(BC)], [AAAG(BC)],

ACTUAG, [AGA], [PGAA], [PGUA].

PCAA, [UAGC(BC, SUB), [AAAGC(BC, SUB)],

ACTAEFF, [UAAGC], [AAAGC], [DST]);

IF ACTAEFF THEN

PROCESS PSEUDO DISPLACEMENTS FOR EFFECTIVENESS

CONSTRAINTS

CALL MAKDVU (NITER, NDV, GLBDES, [UAAGC], [DKUG],

GMCT, DMVI); 

IF NRSET <> 0 THEN

CALL MAKDVU (NITER, NDV, [GLBDES, [UAAGC], [DMAG],

GMCT, DMVI); 

[DPGV] := [DKUG] + [DMAG];

CALL MAKDVU (NITER, NDV, GLBDES, [UAAGC], [DMUG],

GMCT, DMVI);

ELSE

[DPGV] := [DMUG];

ENDIF;

REDUCE THE RIGHT HAND SIDES TO THE L SET

CALL NULLMAT ([DPNV], [DMUN]);

IF NMPC <> 0 THEN

CALL GREDUCE ([DPGV], [PGMN(BC)], [TNN(BC)],

[DPNV]);

IF NRSET <> 0 CALL GREDUCE ([DMUG],

[PGMN(BC)], [TNN(BC)], [DMUN]);

ELSE

[DPNV] := [DPGV];

ENDIF;

IF NRSET <> 0 [DMUN] := [DMUG];

ENDIF;

76
Standard MAPOL Sequence Listing — Continued

CALL NULLMAT ([DPFV], [DMUF]);
IF NSPC <> 0 THEN
   CALL NREDUCE ([DPFV], [FNSF(BC)]); IF NRSET <> 0 THEN
   CALL NREDUCE ([DMUN], [FNSF(BC)]); ELSE
   [DPFV] := [DPFV];
   IF NRSET <> 0 [DMUF] := [DMUN];
ENDIF;

CALL NULLMAT ([DPAV], [DMUA]);
IF NRED <> 0 THEN
   [DPAV] := TRANS([GSUBO(BC)]) * [DPFV];
   IF NRSET <> 0 [DMUA] := TRANS([GSUBO(BC)]) * [DMUF]; ELSE
   IF NOMIT <> 0 THEN
      CALL FREduce ([DPFV], [FPGA(BC)]); IF NRSET <> 0 THEN
      Call FREduce ([DMUF], [FPGA(BC)]); IF NRSET <> 0 THEN
      Call FREduce ([DMUA], [FPGA(BC)]); ELSE
      [DPFV] := [DPFV];
      IF NRSET <> 0 [DMUA] := [DMUF];
ENDIF;
ENDIF;
ENDIF;

IF NRSET <> 0 THEN
   CALL ROWPART ([DPAV], [DPRV], [DPLV], [PARL(BC)]);
   CALL ROWPART ([DMUA], [DMUR], [DMUL], [PARL(BC)]);
   CALL GFBS ([RL11(BC,SUB)], [RU11(BC,SUB)],
            [DPLV], [RL11DPL]);
   [DPI] := TRANS([D1(BC)]) * [DMUL] + [DMUR] -
            [R21(BC,SUB)] * [RL11DPL];
   [DRHS] := TRANS([D1(BC)]) * [DPLV] + [DPRV] -
            [R31(BC,SUB)] * [RL11DPL];
ENDIF;

PROCESS ACTIVE CONSTRAINTS FOR SAERO DISCIPLINE

CALL GFBS ([KLL1(BC,SUB)], [KL1(BC,SUB)],
          [DPI], [KL1DVI]);
[DPRV] := [DRHS] - [K21(BC,SUB)] * [DKIV];

CALL DECOMP ([LHSA(BC,SUB)], [LHS1], [LHSU]);
CALL GFBS ([LHS1], [LHSU], [DRHS], [DUV]);
[DUV1] := [DKIV] * [K112(BC,SUB)] * [DU2];

77
Standard MAPOL Sequence Listing — Continued

STAT LEVL
1605 11:
1606 11:
1607 11:
1608 11:
1609 11:
1610 11:
1611 11:
1612 11:
1613 11:
1614 11:
1615 11:
1616 11:
1617 10:
1618 9:
1619 9:
1620 10:
1621 10:
1622 10:
1623 10:
1624 10:
1625 10:
1626 10:
1627 10:
1628 10:
1629 10:
1630 10:
1631 10:
1632 10:
1633 10:
1634 10:
1635 10:
1636 11:
1637 11:
1638 11:
1639 11:
1640 11:
1641 11:
1642 11:
1643 11:
1644 10:
1645 10:
1646 10:
1647 10:
1648 10:
1649 11:
1650 11:
1651 11:
1652 12:
1653 11:
1654 11:
1655 11:
1656 11:

[DUIL] := [R11]DPL1 * [R112(BC, SUB)] * [DU1R] *:

[EFFSENS] := [R112(BC, SUB)] * [DU1] -

CALL AEROEFFS ( NITER, BC, SUB, SYM, NDV, CONST. :=

PCAE, [EFFSENS], [AMAT]);

ELSE

NOTE THAT SAERO W/O SUPPORT IS NOT SUPPORTED

ENDIF;

ENDIF; $ END IF ON ACTAEFF

IF ACTUAG THEN

SENSITIVITIES OF CONSTRAINTS WRT DISPLACEMENTS

FOR SAERO. THE ACTUAG FLAG WILL BE RETURNED

FALSE IF ONLY TRIM PARAMETER CONSTRAINTS ARE ACTIVE

CALL NULLMAT ([DFDU]);

call KAKDFU (NITER, BC, GSIZEB, ISMAT, [GLBSIG],

CONST, [DFDU], ACTUAGG, SUB);

SOME RELATIVELY SIMPLE CALCULATIONS THAT PRECEDE

THE LOOP ON THE DESIGN VARIABLES

CALL KAKDVU (NITER, NDV, GLBDES, [AGA], [DKUG],

GMMCT, DMVI);

CALL NULLMAT ([DPGV]);

IF NRSET <> 0 THEN

CALL KAKDVU (NITER, NDV, GLBDES, [AGA], [DKUG],

GMMCT, DMVI);

[DPGV] := [DKUG] + [DMAG];

CALL KAKDVU (NITER, NDV, GLBDES, [AGA], [DKUG],

GMMCT, DMVI);

ELSE

[DPGV] := [DKUG];

ENDIF;

REduce THE RIGHT HANd SIDEs TO THE L SET

CALL NULLMAT ([DPNV], [DMUN]);

IF NRPC <> 0 THEN

CALL GREDUCE ([DPGV], [PGN(BC)], [TMN(BC)],

[DPNV]);

IF NRSET <> 0 CALL GREDUCE ([DMUG],

[PGN(BC)], [TMN(BC)], [DMUN]);

ELSE

[DPNV] := [DPGV];

IF NRSET <> 0 [DMUN] := [DMUG];

ENDIF;
Standard MAPOL Sequence Listing — Continued

STAT LEVL
1657 10!!$ CALL NULLMAT ( [DPFV], [DMUF] );
1658 10!!$ IF NSPC <> 0 THEN
1659 10!!$ CALL NREDUCE ( [DPFV], [PNVF(BC)], ..., [DPFV] );
1660 11!!$ IF NPSET <> 0 THEN
1661 11!!$ CALL NREDUCE ( [DMUN], [PNVF(BC)], ..., [DMUF] );
1662 12!!$ ELSE
1663 11!!$ [DPFV] := [DPGV];
1664 11!!$ IF NPSET <> 0 THEN [DMUN] := [DMUN];
1665 11!!$ ENDIF;
1666 11!!$ 1667 10!!$ CALL NULLMAT ( [DPAV], [DMUA] );
1668 10!!$ IF NODR <> 0 THEN
1669 10!!$ [DPAV] := TRANS( [GSCF(BC)] ) * [DPFV];
1670 11!!$ IF NPSET <> 0 THEN [DMUA] := TRANS( [GSCF(BC)] ) * [DMUF];
1671 11!!$ ELSE
1672 11!!$ IF NOMIT <> 0 THEN
1673 12!!$ CALL FREDUCE ( [DPFV], [POA(BC)], 1, [KUL(BC,SUB)], [KUL(BC,SUB)], [KUL(BC,SUB)], [GSCF(BC,SUB)], [DPFV], [DPGV], USST(BC) );
1674 12!!$ IF NRSET <> 0 THEN
1675 13!!$ CALL FREDUCE ( [DMUF], [POA(BC)], 1, [KUL(BC,SUB)], [KUL(BC,SUB)], [KUL(BC,SUB)], [GSCF(BC,SUB)], [DMUA], [DMUF], USST(BC) );
1676 13!!$ ELSE
1677 12!!$ [DPAV] := [DPFV];
1678 12!!$ IF NPSET <> 0 THEN [DMUA] := [DMUF];
1679 12!!$ ENDIF;
1680 12!!$ ENDIF;
1681 11!!$ 1682 12!!$ PROCESS ACTIVE CONSTRAINTS FOR SAERO DISCIPLINE
1683 12!!$ 1684 12!!$ CALL GFBS ( [RL11(BC,SUB)], [RL11(BC,SUB)], [DPFV], [R11DPL], [R11DPL] );
1685 12!!$ IF NRSET <> 0 THEN
1686 13!!$ CALL GFBS ( [KUL(BC,SUB)], [KUL(BC,SUB)], [DPFV], [R11DPL], [R11DPL] );
1687 13!!$ ELSE
1688 12!!$ [DRHS] := TRANS( [R11DPL] ) * [DPFV];
1689 12!!$ IF NPSET <> 0 THEN [DMUA] := [DMUF];
1690 12!!$ ENDIF;
1691 11!!$ ENDIF;
1692 11!!$ 1693 11!!$ CALL GFBS ( [RL11(BC,SUB)], [RL11(BC,SUB)], [DPFV], [R11DPL], [R11DPL] );
1694 11!!$ IF NRSET <> 0 THEN
1695 12!!$ CALL GFBS ( [KUL(BC,SUB)], [KUL(BC,SUB)], [DPFV], [R11DPL], [R11DPL] );
1696 12!!$ ELSE
1697 11!!$ [DRHS] := TRANS( [R11DPL] ) * [DPFV];
1698 11!!$ IF NPSET <> 0 THEN [DMUA] := [DMUF];
1699 11!!$ ENDIF;
1700 11!!$ 1701 11!!$ CALL GFBS ( [KL11(BC,SUB)], [KL11(BC,SUB)], [DPFV], [D11DPL], [D11DPL] );
1702 11!!$ IF NRSET <> 0 THEN
1703 12!!$ CALL GFBS ( [KL11(BC,SUB)], [KL11(BC,SUB)], [DPFV], [D11DPL], [D11DPL] );
1704 12!!$ ELSE
1705 11!!$ CALL AEROSENS ( NITER, BC, MINDX, SUB, CONST.
1706 11!!$ SYM, NDV,
1707 11!!$ BGRVT(BC), STABCF, [POA],
1708 11!!$ [LUSA(BC,SUB)], [RUSA(BC,SUB)],
1709 11!!$ $
Standard MAPOL Sequence Listing — Continued

STAT LEVEL
1709 11:
1710 11:$
1711 11:
1712 11:
1713 11:
1714 11:
1715 11:
1716 11:
1717 11:
1718 11:$
1719 11:$
1720 11:$
1721 11:
1722 10:$
1723 10:$
1724 10:$
1725 10:
1726 10:
1727 11:
1728 11:
1729 11: $!
1730 12:
1731 13:
1732 13:
1733 13:
1734 13:
1735 12:
1736 12:
1737 12:
1738 12:
1739 12:
1740 12:
1741 12:
1742 12:
1743 11:
1744 10:
1745 10:
1746 10:$
1747 10:
1748 9:
1749 8:$
1750 8:
1751 9:$
1752 9:$
1753 9:$
1754 9:
1755 9:
1756 10:
1757 10:
1758 10:
1759 10:
1760 10:

NOTE THAT SAERO W/O SUPPORT IS NOT SUPPORTED $!}

ENDIF; $!

RECOVER SENSITIVITIES TO THE F SET $!

CALL NULLMAT ([UAFTMP]);

IF NGCR <> 0 THEN $!

[UAFTMP] := [GAOUBO(BC,SUB)] * [DUAV];

ELSE $!

IF NOMIC <> 0 THEN $!

IF NRSET <> 0 THEN $!

[TMP1] := [DPV] * [POARD(BC, SUB)] * [DDELDV];

ELSE $!

[TMP1] := [DPV];

ENDIF; $!

CALL GBS ([KOOI(BC,SUB)], [KOOO(BC, SUB)], [TMP1], [UOO]);

[UO] := [GAOUBO(BC, SUB)] * [DUAV] * [UOO];

CALL ROWMERGE ([UAFTMP], [UO], [DUAV], [PGAU(BC)];

ELSE $!

[UAFTMP] := [DUAV];

ENDIF; $!

ENDDO; $!

END SUBSCRIPT LOOP $!

IF ACTUAG THEN $!

REDUCE THE LEFT HAND SIDE MATRIX $!

CALL NULLMAT ([DFDUN]);

IF NMPC <> 0 THEN $!

CALL GREduce ([DFDU], [PGMN(BC)], [TMN(BC)], [DFDUN]);

ELSE $!

[DFDUN] := [DFDU];

ENDIF; $!

80
Standard MAPOL Sequence Listing — Continued

STAT EVL
1761 9$ CALL NULLMAT ([DFDUF]);
1762 9$ IF NSPC -> 0 THEN
1763 10$ CALL ROWPART ([DFDUN], [DFDUF], [PNSF(BC)]);
1764 10$ ELSE
1765 10$ [DFDUF] := [DFDUN];
1766 10$ ENDIF;
1767 9$ TAKE MERGED SENSITIVITIES OF DISPLACEMENTS AND
1768 9$ COMPUTE THE AMAT MATRIX TERMS FOR THE SAERO
1769 9$ CONSTRAINTS
1770 9$ CALL MKAMAT ([AMAT], [DFDUF], [DUV], PCAA, [PGAU]);
1771 9$ ENDIF;
1772 9$ ENDIF;
1773 9$ ENDIF;
1774 9$ ENDIF;
1775 9$ ENDIF;
1776 8$ ENDIF;
1777 7$ ENDF;
1778 6$ ENDDO;
1779 5$ CALL OFPGRAD (NITER, NUMOPTBC, [AMAT], GLBDES, CONST, GRADIENT);
1780 5$ IF NITER >= OCS AND NITER <= OCE THEN
1781 6$ PRINT("LOG=('VANGO MODULE')");
1782 6$ CALL VANGO (NITER, NDV, APPCNVRG, MOVLIN, CNVRLIM,
1783 6$ CTL, CTLMIN, NUMOPTBC, GLBDES, CONST, [AMAT],
1784 6$ DESHIST);
1785 6$ ELSE
1786 6$ ENDIF;
1787 6$ IF NITER >= MPS AND NITER <= MPE THEN
1788 7$ PRINT("LOG=('DESIGN MODULE')");
1789 7$ CALL DESIGN (NITER, NDV, APPCNVRG, MOVLIN, CNVRLIM,
1790 7$ CTL, CTLMIN, NUMOPTBC, GLBDES, CONST, [AMAT],
1791 7$ DESHIST);
1792 7$ ENDIF;
1793 7$ ENDIF;
1794 6$ ENDIF;
1795 5$ ENDIF;
1796 5$ ENDIF;
1797 4$ ENDIF;
1798 3$ ENDDO;
1799 2$ ENDF;
1800 1$ BEGIN FINAL ANALYSIS LOOP
1801 1$ IF NBNDCOND > NUMOPTBC THEN
1802 1$ ASSEMBLE THE GLOBAL MATRICES
1803 1$ BEGIN BOUNDARY CONDITION LOOP
1804 1$ ENDIF;
1805 1$ IF NBNDCOND > NUMOPTBC THEN
1806 2$ ASSEMBLE THE GLOBAL MATRICES
1807 2$ BEGIN BOUNDARY CONDITION LOOP
1808 2$ PRINT("LOG=('**********')");
1809 2$ ASSEMBLE THE GLOBAL MATRICES
1810 2$ BEGIN BOUNDARY CONDITION LOOP
1811 2$ EXIT
1812 2$ BEGIN BOUNDARY CONDITION LOOP
STAT LEVL
1813 2!$ PRINT('LOG='"BEGIN FINAL ANALYSIS"');
1814 2! CALL ANALINIT;
1815 2! CALL EMA2 ( NDV, GSIZEB, GLBDES, GMKCT, DKVI, KIGGI,
1816 2! GMMCT, DKVI, (MIGGI I;
1817 2! FOR BC = NUMOPTBC * 1 TO NBNDCOND DO
1818 3! PRINT('LOG=('"BOUNDARY CONDITION ",BC)";
1819 3!$ ESTABLISH THE BASE USET AND PARTITIONING DATA FOR THE BC
1820 3!$ MAKE B.C.-DEPENDENT BGPDT FROM BASE, ADDING THE EXTRA POINTS FOR
1821 3!$ THIS B.C.
1822 3!$ PROCESS MATRICES, TRANSFER FUNCTIONS, AND INITIAL CONDITIONS FOR
1823 3!$ THIS B.C.
1824 3! CALL MKUSET( BC, GSIZEB, [YS(BC)H. [TNONBC), [PGNN(BC)], [PNSF(BC)],
1825 3! [PFOA(BC)], [PARL(BC)], USET(BC)
1826 3! CALL BCBGPDT( BC, GSIZEB, BGPDT(BC), ESIZE(BC)
1827 3! GSIZE GSIZEB; 
1828 3! PSIZE(BC) ESIZE(BC) * GSIZE;
1829 3! CALL BCBULK( BC, PSIZE(BC), BGPDT(BC), JSET(BC)
1830 3;5 CALL BOUND( BC, GSIZE, ESIZE(BC), USET(BC), BLOAD, BMASS, DMODES,
1831 3! BMODES, BSNO, BPVTR, BDYN, BOKSF, BTR, BPRO, BMFR, BGSST, BBLAST, NWPC, NSPC, NOMIT, HRSET, HGDF);
1832 3!$ DETERMINE IF ANY M2GG/K2GG INPUT DATA ARE TO BE ADDED
1833 3!$ CALL NULLMAT([ KGG), [KGG]);
1834 3! IF M2GGFLAG THEN
1835 4! [MGG] := [MGG] + [M2GG];
1836 4! ENDIF;
1837 3! ELSE
1838 4! [MGG] := [KGG];
1839 4! ENDIF;
1840 3! IF K2GGFLAG THEN
1841 4! [KGG] := [KGG] + [K2GG];
1842 4! ELSE
1843 4! [KGG] := [KGG];
1844 4! ENDIF;
1845 3! CALL GPWG ( BC, GPWGGRID, [PGW], OGPWG );
1846 3! IF BLOAD <> O CALL GTLOAD ( BC, GSIZE, BGPDT(BC), GLBDES,
1847 3! SMPLOD, DPTHVI, DPGVRI, PG, OGRIDLOD);
1848 3!$ PARTITION-REDUCTION OF GLOBAL MATRICES
1849 3!$
Standard MAPOL Sequence Listing — Continued

STAT LEVL
1865 3:$
1866 3! IF NBNDCOND > 1 CALL NULLMAT ([KNN], [PN], [MNN], [GTKN], [GSTKN],
1867 4! [UGTKN]);
1868 3!
1869 4:$ IF NMPC <> 0 THEN
1870 4:$ PERFORM MPC REDUCTION
1871 4:$
1872 4! PRINT(‘LOG=’ MPC REDUCTION’);"
1873 4! CALL GREDUCE ([KGG], [PG], [PNNN(BC)], [TNN(BC)], [KNN], [PN]);
1874 4! IF BMASS <> 0 CALL GREDUCE ([KGG], [PNNN(BC)], [TNN(BC)], [MNN]);
1875 4! IF BSAERO <> 0 THEN
1876 5! CALL GREDUCE ([GTKG], [PNNN(BC)], [TNN(BC)], [GTKN]);
1877 5! CALL GREDUCE ([GTKG], [PNNN(BC)], [TNN(BC)], [GSTKN]);
1878 5! ENDIF;
1879 4! IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 THEN
1880 5! CALL GREDUCE ([UGTKG], [PNNN(BC)], [TNN(BC)], [UGTKN]);
1881 4! ELSE
1882 4! NO MPC REDUCTION
1883 4!$
1884 4!$ [KNN] := [KGG];
1885 4! IF BLOAD <> 0 [PN] := [PG];
1886 4! IF BMASS <> 0 [MNN] := [PGG];
1887 4! IF BSAERO <> 0 THEN
1888 5! [GTKN] := [GTKG];
1889 5! [GSTKN] := [GSTKG];
1890 5! ENDIF;
1891 4! IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 THEN
1892 5! [UGTKN] := [UGTKG];
1893 4! ENDIF;
1894 4! IF NBNDCOND > 1 CALL NULLMAT ([KFF], [PF], [MFF], [GTKF], [GSTKF],
1895 4! [UGTKF]);
1896 4! IF NSPC <> 0 THEN
1897 4:$ PERFORM SPC REDUCTION
1898 4:$
1899 4:$
1900 4! PRINT(‘LOG=’ SPC REDUCTION’);"
1901 4! CALL NREDUCE ([KNN], [PN], [PNNS(BC)], [YN(BC)], [KFF], [KFS],
1902 4! [KSS], [PF], [PS]);
1903 4! IF BMASS <> 0 CALL NREDUCE ([KNN], [PNNS(BC)], [MFF]);
1904 4! IF BSAERO <> 0 THEN
1905 5! CALL NREDUCE ([GTKN], [PNNS(BC)], [GTKF]);
1906 5! CALL NREDUCE ([GTKN], [MNSP(BC)], [GSTKF]);
1907 5! ENDIF;
1908 4! IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 THEN
1909 5! CALL NREDUCE ([UGTKN], [PNNS(BC)], [UGTKF]);
1910 4! ELSE
1911 4:$$
1912 4:$$
1913 4:$$
1914 4:$$
1915 4! IF BLOAD <> 0 [PF] := [PN];
1916 4! IF BMASS <> 0 [MFF] := [MNN];
Standard MAPOL Sequence Listing — Continued

STAT LEVL
1917 4: IF BSAERO <> 0 THEN !
1918 5: [GTKP] := [GTKN]; !
1919 5: [GSTKP] := [GSTYN]; !
1920 5: ENDIF; !
1921 4: IF BFLUTR <> 0 OR BGUST <> 0 OR BBLAST <> 0 [UGTKF] := [UGTKN]; !
1922 4: ENDIF; !
1923 3:\$ !
1924 3: IF NBNDCOND > 1 CALL NULLMAT ([KAA], [PA], [KAA], [KAA], [PAA], [UGTKA]); !
1925 3:\$ !
1926 3: IF NGDR <> 0 THEN !
1927 4:\$ !
1928 4:\$ PERFORM THE GENERAL DYNAMIC REDUCTION WHICH IS DISCIPLINE !
1929 4:\$ !
1930 4:\$ INDEPENDENT. THE RESULTING [GSUBO] MATRIX WILL BE USED BY !
1931 4:\$ !
1932 4:\$ PRINT('LOG=(' DYNAMIC REDUCTION')); !
1933 4:\$ !
1934 4:\$ OBTAIN THE OMITTED DOF PARTITION OF KFF AND MFF !
1935 4:\$ !
1936 4: CALL PARTN ([KFF], [KOO], [KOA], [PFOA(BC)]); !
1937 4: CALL PARTN ([MFF], [MOO], [MFOA(BC)]); !
1938 4: ASIZE := GSIZ - NMPC - NSPC - NOMIT; !
1939 4: LSIZE := ASIZE - NRSET; !
1940 4: CALL GDR1 ([KOO], [MOO], [KSOO], [LSOO], [PHIOK], LKSET, LJSET, NEIV, !
1941 4: FMAX, BC, BGPDT(BC), USET(BC), NUMOPTBC, NBNDCOND, !
1942 4: ASIZE, GNORM, BC); !
1943 4:\$ !
1944 4:\$ LKSET MEANING !
1945 4: << 0 APPROX. MODE SHAPES SELECTED !
1946 4: = 0 NO APPROX. MODE SHAPES IN GDR !
1947 4: IF LKSET <> 0 THEN !
1948 5: CALL SDCOMP ([KSOO], [LSOO], USET(BC), SINGOSET); !
1949 5: CALL GDR3 ([LSOO], [MOO], [PHIOK], LKSET, LJSET, !
1950 5: NEIV, FMAX, BC); !
1951 5: ENDIF; !
1952 4: CALL GDR3 ([MOO], [KOA], [MGG], [PHIOK], [KMNB(BC)], [GGO], !
1953 4: [PHIB(BC)], [PNSF(BC)], [PFOA(BC)], [GSUBO(BC)], !
1954 4: BGPDT(BC), USET(BC), !
1955 4: LKSET, LJSET, ASIZE, GNORM, BC); !
1956 4: CALL GDR4 (BC, GSIZE, PSIZE(BC), LKSET, LJSET, NUMOPTBC, NBNDCOND, !
1957 4: [PMMN(BC)], [TMN(BC)], [PNSF(BC)], [PFOA(BC)], !
1958 4: [PARL(BC)], [PDGRG(BC)], [PAJK], [PFJK], BGPDT(BC), !
1959 4: USET(BC)); !
1960 4: ENDIF; !
1961 3:\$ !
1962 3: IF BLOAD <> 0 OR BMODES <> 0 OR BFLUTR <> 0 OR BDYN <> 0 THEN !
1963 4:\$ !
1964 4:\$ REDUCE THE MATRICES WITHOUT AEROELASTIC CORRECTIONS !
1965 4:\$ !
1966 4: IF NGDR <> 0 THEN !
1967 5:\$ !
1968 5:\$ PERFORM THE GENERAL DYNAMIC REDUCTION !
Standard MAPOL Sequence Listing — Continued

```
STAT LEVEL
1969  5:
1970  5:
1971  5:
1972  5:
1973  5:
1974  5:
1975  5:
1976  6:
1977  6:
1978  6:
1979  6:
1980  5:
1981  6:
1982  6:
1983  6:
1984  6:
1985  6:
1986  6:
1987  6:
1988  6:
1989  6:
1990  7:
1991  7:
1992  7:
1993  7:
1994  7:
1995  7:
1996  7:
1997  7:
1998  7:
1999  7:
2000  7:
2001  7:
2002  7:
2003  7:
2004  7:
2005  6:
2006  6:
2007  6:
2008  6:
2009  6:
2010  6:
2011  6:
2012  6:
2013  6:
2014  5:
2015  4:
2016  4:
2017  5:
2018  5:
2019  5:
020  5:

PRINT("LOG="'   SYMMETRIC DYNAMIC REDUCTION'"');

[MAA] := TRANS ( [GSUBO(BC)] ) * [MFF] * [GSUBO(BC)];

[KAA] := TRANS ( [GSUBO(BC)] ) * [KFF] * [GSUBO(BC)];

IF BLOAD <> 0 [PA] := TRANS ( [GSUBO(BC)] ) * [PF];

IF BFLUTR <> 0 OR BHOST <> 0 OR BBLAST <> 0 THEN

[TMP1] := TRANS ( [UGTKF] ) * [GSUBO(BC)];

CALL TRNSPOSE ( [TMP1], [UGTKA] );

ENDIF;

ELSE

IF NOMIT <> 0 THEN

PERFORM THE STATIC REDUCTION

PRINT("LOG="'   STATIC CONDENSATION'"');

CALL FREduce ( [KFF], [PF], [PFOA(BC)], [UOGINV(BC)], [GSUBO(BC)], [KAA], [PA], [PO], USET(BC));

IF BMASS <> 0 THEN

PERFORM GUYAN REDUCTION OF THE MASS MATRIX

CALL PARTN ( [KFF], [MOO], [MOA], [MAABAR], [PFOA(BC)] );

[MAA] := [MAABAR] + TRANS([MOA]) * [GSUBO(BC)];

TRANS([GSUBO(BC)]) * [MOA] +

TRANS([GSUBO(BC)]) * [MOO] * [GSUBO(BC)];

IF NRSET <> 0 THEN

ENCIF;

IF BFLUTR <> 0 OR BHOST <> 0 OR BBLAST <> 0 THEN

CALL ROWPART ( [UGTKF], [UGTKO], [UGTKAB], [PFOA(BC)] );

[TMP1] := TRANS ( [UGTKO] ) * [GSUBO(BC)];

CALL TRNSPOSE ( [TMP1], [TMP2] );

[UGTKA] := [UGTKAB] + [TMP2];

ENDIF:

ELSE

NO F-SET REDUCTION

ENDIF;

ENDIF:

ENDIF;

ELSE

IF NRSET <> 0 THEN

PERFORM THE SUPPORT SET REDUCTION

PRINT("LOG="'   SUPPORT REDUCTION'"');
```
CALL PARTN ([KAA], [KRR], [KLR], [KLL], [PARL(BC)]);  
CALL SDCOMP ([KLL], [KLLINV(BC)], USET(BC), SINGSET);  
CALL FBS ([KLLINV(BC)], [KLR], [D(BC)], -1);  
CALL RBCHECK (BC, USET(BC), BGPDT(BC), [D(BC)], [KLL], [KRR], [KLR]);  
CALCULATE THE REDUCED MASS MATRIX $;  
CALL PARTN ([MAA], [MRRBAR], [MLR], [MLL], [PARL(BC)]);  
[IFR(BC)] := [MLL] * [D(BC)] + [MLR];  
[MRR(BC)] := [MRRBAR] + TRANS ([MLR]) * [D(BC)] + [IFR(BC)];  
[R22] := TRANS ([D(BC)]) * [MLR] + [MRRBAR];  
NO SUPPORT SET REDUCTION $;  
IF BLOAD <> 0 THEN  
PROCESS STATICS WITH INERTIA RELIEF $;  
PRINT("LOG=(' >>DISCIPLINE: STATICS(INERTIA RELIEF))");  
CALL ROWPART ([PA], [PR], [PLBAR], [PARL(BC)]);  
[LNS(BC)] := [MRR(BC)];  
[RHS(BC)] := TRANS([D(BC)]) * [PLBAR] + [PR];  
[AL] := [D(BC)] * [AR];  
CALL ROWMERGE ([AA], [AR], [AL], [PARL(BC)]);  
CALL FBS ([KLLINV(BC)], [RHS(BC)], [UL]);  
CALL YSM([ERG], [UA], [UL], [PARL(BC)]);  
ENDIF;  
IF BMODES <> 0 THEN  
PRINT("LOG=(' >>DISCIPLINE: NORMAL MODES'))");  
CALL REIG (BC, USET(BC), [KAA], [MAA], [MRR(BC)], [D(BC)], LAMBDA, [PHIA], [MII], HSIZE(BC));  
CALL OFFPROOT (BC, NUMOPTBC, LAMBO);  
ENDIF;  
ELSE  
NO SUPPORT SET REDUCTION $;  
ENDIF  
IF BLOAD <> 0 THEN  
PRINT("LOG=(' >>DISCIPLINE: STATICS'))");  
CALL SDCOMP ([KAA], [KLLINV(BC)], USET(BC), SINGASET);  
CALL FBS ([KLLINV(BC)], [PA], [UA]);  
ENDIF;  
IF BMODES <> 0 THEN  
PRINT("LOG=(' >>DISCIPLINE: NORMAL MODES'))");  
CALL REIG (BC, USET(BC), [KAA], [MAA], LAMDO, [PHIA], [MII], HSIZE(BC));  
CALL OFFPROOT (BC, NUMOPTBC, LAMBO);  
ENDIF;  
ENDIF;
STAT LEVL
2073 3$  IF BSAERO <> 0 THEN
2074 3$  PERFORM STATIC AEREOELASTIC ANALYSES
2075 4$  PRINT('LOG=' 'SAERO INITIALIZATION');
2076 4$  CALL TRANSPOSE ([GSTKF], [GSKF]);
2077 4$  LOJP := TRUE;
2078 4$  SUB := 0;
2079 4$  WHILE LOOP DO
2080 5$  SUB := SUB - 1;
2081 5$  CALL SAERODRV (BC, SUB, LOOP, MINDEX, SYM, MACH, QDP, I);
2082 5$  ADJUST KFF MATRIX AND DETERMINE THE RIGID AIR LOADS
2083 5$  IF SYM = 1 [AICS] := [GSTKF]*[TRANS([AICMAT(MINDEX)])]*[GSKF];
2084 5$  ELSE [AICS] := [GSTKF]*[TRANS([AICMAT(MINDEX)])]*[GSKF];
2085 5$  [PAF] := [QDP] * [GSTKF] * [AIRFRC(MINDEX)];
2086 5$  [PAF] := [KFF] - (QDP) * [AICS];
2087 5$  REDUCE THE MATRICES WITH AEREOELASTIC CORRECTIONS
2088 5$  SAVE THE SUBCASE/BC DEPENDENT DATA FOR SENSITIVITY ANALYSIS
2089 5$  IF NGDR <> 0 THEN
2090 6$  PERFORM THE GENERAL DYNAMIC REDUCTION
2091 6$  PRINT('LOG=(' 'SAERO DYNAMIC REDUCTION')*);
2092 6$  [MAAA] := TRANS ([GSUBO(BC)] * ([KAAA] * [GSUBO(BC)]);
2093 6$  [KAAA] := TRANS ([CSUBO(BC)] * [PAF] * [GSUBO(BC)];
2094 6$  ELSE [KAAA] := TRANS ([GSUBO(BC)] * [PAF];
2095 6$  IF NIMIT <> 0 THEN
2096 7$  IF NRSET = 0 AND SUB = 1 AND BLOAD = 0 AND BMODES = 0 AND
2097 7$  IFLUTR = 0 AND BDYN = 0 THEN
2098 7$  Form [KAA] ON SO 'O' CAN BE FORMED
2099 7$  CALL FREDUCE ([KFF], [PFON(BC)], [KOOINV(BC)],
2100 7$  [GSUBO(BC)], [KAA], . . , USET(BC));
2101 7$  ENDIF;
2102 7$  CALL FREDUCE ([KFF], [PAF], [PFON(BC)], BSAERO,
2103 7$  [KGO(BC, SUB)], [KOOU(BC, SUB)], [KAA],
2104 7$  [PAF], [POAO(BC, SUB)], USET(BC));
2105 7$  ENDIF;
STAT LEVL
2125 7! IF BMASS <> 0 THEN
2126 8!$ PERFORM GUYAN REDUCTION OF THE MASS MATRIX
2127 8!$ !
2128 8!$ CALL PARTN1(MFF), [MOO]. [MCA], [MAABAR], [PMF]
2129 8!$ [PF0A(BC)]] !
2130 8! [MAABAR] := [MAABAR] + TRANS([MOA]) * [GASUBO(BC,SUB)] * [MOA] +
2131 8! [GASUBO(BC,SUB)] * [MOA] *
2132 8! TRANS([GASUBO(BC,SUB)] * [MOA]) +
2133 8! [GASUBO(BC,SUB)]
2134 8! IF NRSET <> 0 THEN
2135 8! IF NRSET <> 0 AND SUB = 0 AND BLOAD = 0 AND
2136 9! iPERFORM THE SUPPORT SET REDUCTION
2137 8! ENDIF;
2138 7! ELSE
2139 7! [IF SUB = 1 AND BLOAD = 0 AND BFLTR = 0 AND BDYN = 0 THEN
2140 7! FORM [KAA] ON FIRST PASS SO [D] CAN BE FORMED
2141 7! [KAA] := [KFF];
2142 8! ENDIF;
2143 7! [KAAA] := [KAPF];
2144 7! [KAA] := [MFF];
2145 7! [PAA] := [PAF];
2146 7! ENDIF;
2147 6! ENDIF;
2148 5! IF NRSET <> 0 THEN
2149 5! PRINT("LOG:: SAERO SUPPORT REDUCTION");
2150 5! IF SUB = 1 AND BLOAD = 0 AND BMODES = 0 AND BFLTR = 0 AND
2151 5! BDYN = 0 THEN
2152 5! [D] WAS NOT COMPUTED FOR NON-SAERO DISCIPLINES SO
2153 5! NEED TO COMPUTE IT NOW
2154 5! CALL PARTN (L [KAA], [KRR], [KLR], [KLL], [PARL(BC)] )
2155 5! CALL RSCOMP([KLL], [KLLINV(BC)], USET(BC), SINGLESET)
2156 5! CALL RBCHECK (BC, USET(BC), BFLTR(BC), [DBC], [KLL],
2157 5! [KRR], [KRR]);
2158 5! ENDIF;
2159 5! IF SUB = 0 AND BLOAD = 0 AND
2160 5! CALL PARTN1([MAAA], [MRRBAR], [MLR], [MLL], [PARL(BC)]]);
Standard MAPOL Sequence Listing — Continued

2177 6] [R13(BC, SUB)] := [MLL] * [DBC] * [MLR];
2178 6] [R33] := [MRBAR] + TRANS ([MLR]) * [DBC] +
2179 6] TRANS ([DBC]) + [R13(BC, SUB)];
2180 6] [R22] := TRANS ([DBC]) * [MLR] + [MRBAR];
2181 6] CALL TRANSPOSE ([R13(BC, SUB)], [R21(BC, SUB)]); $!
2182 6]$ PROCESS STEADY AEROELASTIC DISCIPLINE $!
2183 6]$ PRINT("LOG("");$!
2184 6] CALL PARTN ([KAA], [KARR], [R12(BC, SUB)], [KARL], [R11], [PARL(BC)];$!
2185 6] [R32(BC, SUB)] := TRANS([DBC]) * [R12(BC, SUB)] + [KARR];$!
2186 6] [R31(BC, SUB)] := TRANS([DBC]) * [R11] + [KARL];$!
2187 6] CALL DECOMP ([K11], [K111(BC, SUB)], [K111(BC, SUB)])$!
2188 6] CALL ROWPART ([PAR1], [PARBAR], [PAL], [PARL(BC)]);$!
2189 6] CALL GFBS ([R11(BC, SUB)], [R11(BC, SUB)], [PAL]),$!
2190 6] [R11PAL(BC, SUB)], -1);$!
2191 6] [PRIGID] := [PARBAR] + TRANS([DBC]) * [PAL];$!
2192 6] [P3] := [R21(BC, SUB)] * [R11PAL(BC, SUB)];$!
2193 6] [P2] := [PRIGID] + [R31(BC, SUB)] * [R11PAL(BC, SUB)];$!
2194 6] CALL GFBS ([R11(BC, SUB)], [R11(BC, SUB)], [PAL]),$!
2195 6] [R1112(BC, SUB)], -1);$!
2196 6] [K12(BC, SUB)] := [R21(BC, SUB)] * [R1113(BC, SUB)];$!
2197 6] [K21(BC, SUB)] := [R32(BC, SUB)] + [R31(BC, SUB)] * [R1112(BC, SUB)];$!
2198 6] [K22] := [R33] + [R31(BC, SUB)] * [R1113(BC, SUB)];$!
2199 6] CALL DECOMP ([K11], [K111(BC, SUB)], [K111(BC, SUB)]);$!
2201 6] [R1112(BC, SUB)], -1);$!
2203 6] [K1113(BC, SUB)], -1);$!
2204 6] [K11] := [K22] + [R21(BC, SUB)] * [R1113(BC, SUB)];$!
2205 6] [K12(BC, SUB)] := [R21(BC, SUB)] * [R1113(BC, SUB)];$!
2206 6] [K21(BC, SUB)] := [R32(BC, SUB)] + [R31(BC, SUB)] * [R1112(BC, SUB)];$!
2207 6] [K22] := [R33] + [R31(BC, SUB)] * [R1113(BC, SUB)];$!
2208 6]$ CALL DECOMP ([K11], [K111(BC, SUB)], [K111(BC, SUB)]);$!
2210 6] [P3], $!
2211 6] [PAR(BC, SUB)];$!
2213 6] [K1112(BC, SUB)], -1);$!
2214 6] [LHS(BC, SUB)] := [K22] + [K21(BC, SUB)] * [K1112(BC, SUB)];$!
2215 6] [RHS(BC, SUB)] := [P2] - [K21(BC, SUB)] * [PAR(BC, SUB)];$!
2216 6] CALL SAERO (BC, MINDX, SUB, SYM, DDP, STANC);$!
2217 6] BOPOT(BC), [LHS(BC, SUB)], [RHS(BC, SUB)], [AAR],$!
2218 6] [DELTA(SUB)], [PRIGID], [R33]);$!
2219 6] [AAL] := [DBC] * [AAR];$!
2220 6] CALL ROWMERGE ([AAR|SUB]), [AAR], [AAL], [PARL(BC)];$!
2221 6] [UAR] := [K112(BC, SUB)] * [AAR] + [PAR(BC, SUB)] *$!
2222 6] [DELTA(SUB)];$!
2223 6] [UAL] := [R1112(BC, SUB)] * [UAR] + [R1113(BC, SUB)] * [AAR];$!
2224 6] [P22(BC, SUB)] * [DELTA(SUB)];$!
2225 6] CALL ROWMERGE ([UAR|SUB]), [UAR], [UAL], [PARL(BC)];$!
2226 6] IF NOKIT <> 0 (PAO|SUB) := [PAOAR(BC, SUB)] * [DELTA(SUB)];$!
2227 6] ELSE$!
2228 6]$
STAT LEVL
2229 6: $ NO SUPPORT SET REDUCTION $  
2230 6: $ $  
2231 6: $ $  
2232 6: $ PROCESS STEADY AEROELASTIC DISCIPLINE $  
2233 6: $ $  
2234 6: $ PRINT("LOG=" >>>DISCIPLINE: STEADY AERO");  
2235 6: $ ENDIF;  
2236 5: $ ENDDO;  
2237 4: $ ENDF;  
2238 3: $ $  
2239 3: $ PERFORM DYNAMIC ANALYSES -- NOTE THAT THESE ARE INDEPENDENT  
2240 3: $ OF THE SUPPORT SET  
2241 3: $ $  
2242 3: IF BDYN <> 0 THEN  
2243 4: IF BFLUTR <> 0 THEN  
2244 5: PRINT("LOG=" >>>DISCIPLINE: FLUTTER");  
2245 5: SUB := 0;  
2246 5: LOOP := TRUE;  
2247 5: WHILE LOOP DO  
2248 6: SUB := SUB + 1;  
2249 6: CALL FLUTDRY (BC SUB, LOOP);  
2250 6: CALL FLUTQHN (BC SUB, ESIZE(BC), PSIZE(BC), [QKAI],  
2251 6: [UGTKAI], [PHIA], USET(BC),  
2252 6: [TMNI(BC)], [GSUBO(BC)]; NGDR, AECOMPU, GEOMUA,  
2253 6: [PHIRH], [QRHFRC(BC, SUB)], OADRDP);  
2254 6: CALL FLUTMA (BC, SUB, ESIZE(BC), PSIZE(BC), BQPDF(BC),  
2255 6: USET(BC), [MAA], [KA], [TMNI(BC)], [GSUBO(BC)],  
2256 6: NGDR, LAMBDA, [PHIA], [WHERE(BC, SUB)],  
2257 6: [BHHFRC(BC, SUB)], [KHIFRC(BC, SUB)];  
2258 6: CALL FLUTTRAN (BC, SUB, ESIZE(BC), [PHN],  
2259 6: [QHNN], [QHJL], [PHIKH], [QHFL], [QHKR],  
2260 6: CGSUBO(BCHj), NGDR, LAMBDA, [PHIA], [HOD], [BDD], [KDDT],  
2261 6: ENDDO;  
2262 5: ENDF;  
2263 4: IF BDDSP <> 0 THEN  
2264 5: IF BMTR <> 0 OR BDT <> 0 THEN  
2265 6: PRINT("LOG=" >>>DISCIPLINE: TRANSIENT RESPONSE");  
2266 6: ENDF;  
2267 5: IF BMTR <> 0 OR BDFR <> 0 THEN  
2268 6: PRINT("LOG=" >>>DISCIPLINE: FREQ RESPONSE");  
2269 6: ENDF;  
2270 5: CALL QMHLEN (BC, ESIZE(BC), [QKAI], [QKJL], [UGTKAI], [PHIA],  
2271 5: [PHIRH], [QHJL], [QRHFRC(BC, SUB)],  
2272 5: [RAA], [TMNI(BC)], [GSUBO(BC)]; NGDR,  
2273 5: LAMBDA, [PHIA], [MD], [BED], [KDT], [KDDF],  
2274 5: [KHFF], [KHH], [KHHFRC], [KHIFRC]);  
2275 5: CALL DYNLOAD (BC, ESIZE(BC), PSIZE(BC), SMPL, BPDF(BC),  
2276 5: USET(BC), [TMNI(BC)], [GSUBO(BC)],  
2277 5: NGDR, [PHIA], [QKJL], [KDT], [PDF],  
2278 5: [PTGLOAD], [PFLLOAD], [PFLOAD], [PFMLoad]);  
2279 5: CALL DYNRSP (BC, ESIZE(BC), [NDD], [BED], [KDT], [KDDF],  

90
BEGIN: adding the generation of dynamic reduction recovery operations

IF BBLAST <> 0 THEN
  CALL COLPART (PHIA), (PHIE), (MPART);
  CALL ROWPART (GENK), (GENQ), (QKE), (QEE), (MPART);
  CALL PARTN (GENQ), (QRR), (QRE), (QEE), (MPART);
  CALL PARTN (GENQ), (QRR), (QRE), (QEE), (MPART);
  IF NBNDCOND > 1 CALL NULMAY (AF), (PHIF);
  IF NDRD <> 0 THEN
    DATA RECOVERY WITH GDR

    APPEND THE GDR-GENERATED DOFS TO THE F-SET

    PRINT('LOG='DYNAMIC REDUCTION RECOVERY')*);
Standard MAPOL Sequence Listing — Continued

STAT LEVEL

2333 6!
2334 6!
2335 6!
2336 5!
2337 4!
2338 5!
2339 6!
2340 6!
2341 6!
2342 6:
2343 6:
2344 6:
2345 6:
2346 6!
2347 6!
2348 7!
2349 7!
2350 7!
2351 7!
2352 7!
2353 6:
2354 5:
2355 4!
2356 5!
2357 5!
2358 5!
2359 5!
2360 4!
2361 5!
2362 5!
2363 5!
2364 5!
2365 4!
2366 5!
2367 5!
2368 5!
2369 5!
2370 4!
2371 4!
2372 5:
2373 5:
2374 5:
2375 5!
2376 5!
2377 6!
2378 6!
2379 6!
2380 7!
2381 6!
2382 5!
2383 6!
2384 7!

IF BSAERO <> 0 THEN
FOR S = 1 TO SUB DO

IF BSAERO <> 0 THEN
FOR S = 1 TO SUB DO

IF BSAERO <> 0 THEN
FOR S = 1 TO SUB DO

ELSE
IF NOMIT <> 0 THEN

ELSE
IF NOMIT <> 0 THEN

DATA RECOVERY WITH STATIC CONDENSATION

DATA RECOVERY WITH STATIC CONDENSATION

PRINT('LOC=('" STATIC CONDENSATION RECOVERY"')

PRINT('LOC=('" STATIC CONDENSATION RECOVERY"')

IF BLOAD <> 0 THEN

IF BLOAD <> 0 THEN

IF BLOAD <> 0 THEN

CALL RECOVA ( [AA], [UJ], [PA], [PF])

CALL RECOVA ( [AA], [UJ], [PA], [PF])

CALL RECOVA ( [AA], [UJ], [PA], [PF])

ENDIF;

ENDIF;

ENDIF;

ENDIF;

ENDIF;

ENDIF;

ENDIF;

ENDIF;

ENDIF;

ENDIF;

ENDIF;

IF BLOAD <> 0 THEN

IF BLOAD <> 0 THEN

IF BLOAD <> 0 THEN

PRINT('LOC=('" STATIC CONDENSATION RECOVERY"')

PRINT('LOC=('" STATIC CONDENSATION RECOVERY"')

PRINT('LOC=('" STATIC CONDENSATION RECOVERY"')

IF BLOAD <> 0 THEN

IF BLOAD <> 0 THEN

IF BLOAD <> 0 THEN

CALL RECOVA ( [AA], [UJ], [PA], [PF])

CALL RECOVA ( [AA], [UJ], [PA], [PF])

CALL RECOVA ( [AA], [UJ], [PA], [PF])

ENDIF;

ENDIF;

ENDIF;

ENDIF;

ENDIF;

ENDIF;

ENDIF;

ENDIF;
STAT LEVL 2385 7!
2386 7!
2387 7!
2388 7!
2389 7!
MERGE THE CURRENT SUBCASE DEPENDENT RESULTS INTO A SINGLE $1
M matrix of response quantities for further recovery $1
2390 7!
2391 7!
CALL SAEROMRG ( BC, S, [UAF], [UAFTMP] ) $1
2392 7!
IF NRSET <> 0 THEN $1
2393 7!
CALL RECOVA ( [AAC(S)], [GASUBO(BC,S)],...... $1
2394 8!
[PPOA(BC)], [UAFTMP]); $
2395 8!
CALL SAEROMRG ( BC, S, [AAC], [UAFTMP] ) $1
2396 8!
ENDIF; $1
2397 8!
ENDIF; 
2398 7!
ENDDO; 
2399 6!
IF BMODES <> 0 THEN 
2400 5!
[PHA] := [GSUBO(BC)] * [PHIA]; 
2401 6!
CALL ROWMERGE ( [PHIF], [PHIO], [PHIA], [PPOA(BC)] ); 
2402 6!
ENDIF; 
2403 6!
IF BDTR <> 0 OR BMTR <> 0 THEN 
2404 5!
CALL RECOVA ( [UTRANA], [GSUBO(BC)],...... $1
2405 6!
[PPOA(BC)], [UTRANF] ); 
2406 6!
ENDIF; 
2407 6!
IF BDFR <> 0 OR BMFR <> 0 THEN 
2408 5!
CALL RECOVA ( [UFREQA], [GSUBO(BC)],...... 
2409 6!
[PPOA(BC)], [UFREQF] ); 
2410 6!
ENDIF; 
2411 6!
ELSE 
2412 5!
ENDIF; 
2413 5!
DATA RECOVERY WITHOUT F-SET REDUCTION $1
2414 5!
2415 5!
IF BLOAD <> 0 THEN $1
2416 5!
[UF] := [UA]; $1
2417 6!
IF NRSET <> 0 [AF] := [AA];  
2418 6!
ENDIF; 
2419 6!
IF NSAOER <> 0 THEN 
2420 5!
FOR S = 1 TO SUB DO 
2421 6!
MERGE THE CURRENT SUBCASE DEPENDENT RESULTS INTO A SINGLE 
M $1
MATRIX OF RESPONSE QUANTITIES FOR FURTHER RECOVERY 
2422 7!
2423 7!
2424 7!
2425 7!
2426 7!
2427 7!
2428 7!
ENDIF; $1
2429 6!
IF BMODES <> 0 [PHIF] := [PHIA]; 
2430 5!
IF BDTR <> 0 OR BMTR <> 0 [UTRANA] := [UTRANA]; 
2431 5!
IF BDFR <> 0 OR BMFR <> 0 [UFREQA] := [UFREQA];  
2432 5!
ENDIF; 
2433 5!
ENDIF; 
2434 4!
ENDIF; 
2435 3!
IF NBNOCOND > 1 CALL NULLMAT ( [UN], [AN], [PHIN] ); 
2436 3!
STAT LEVEL
2417 3:
2418 4:
2419 4:
2420 4:
2421 4:
2422 4:
2423 5:
2424 5:
2425 5:
2426 5:
2427 5:
2428 5:
2429 5:
2430 5:
2431 5:
2432 5:
2433 5:
2434 5:
2435 5:
2436 5:
2437 5:
2438 4:
2439 4:
2440 4:
2441 4:
2442 4:
2443 5:
2444 5:
2445 5:
2446 5:
2447 5:
2448 5:
2449 5:
2450 4:
2451 5:
2452 5:
2453 5:
2454 4:
2455 5:
2456 5:
2457 5:
2458 6:
2459 6:
2460 6:
2461 6:
2462 5:
2463 4:
2464 5:
2465 5:
2466 4:
2467 4:
2468 5:
2469 4:
2470 5:
2471 5:
2472 5:
2473 4:
2474 4:
2475 5:
2476 5:
2477 5:
2478 5:
2479 4:
2480 4:
2481 4:
2482 4:
2483 4:
2484 5:
2485 5:
2486 5:
2487 4:
2488 5:

DATA RECOVERY WITH SPC-REDUCTION
$!

PRINT('LOG=(' SPC RECOVERY')$!

IF NSPC <> 0 THEN

DATA RECOVERY WITH SPC-REDUCTION
$!

IF BLOAD <> 0 THEN

PRINT('LOG=(' SPC RECOVERY')$!

IF NSPC <> 0 THEN

DATA RECOVERY WITH SPC-REDUCTION
$!

IF BLOAD <> 0 THEN

PRINT('LOG=(' SPC RECOVERY')$!

IF NSPC <> 0 THEN

DATA RECOVERY WITH SPC-REDUCTION
$!

IF BLOAD <> 0 THEN

PRINT('LOG=(' SPC RECOVERY')$!

IF NSPC <> 0 THEN

DATA RECOVERY WITH SPC-REDUCTION
$!

IF BLOAD <> 0 THEN

PRINT('LOG=(' SPC RECOVERY')$!

IF NSPC <> 0 THEN

DATA RECOVERY WITH SPC-REDUCTION
$!

IF BLOAD <> 0 THEN

PRINT('LOG=(' SPC RECOVERY')$!

IF NSPC <> 0 THEN

DATA RECOVERY WITH SPC-REDUCTION
$!

IF BLOAD <> 0 THEN

PRINT('LOG=(' SPC RECOVERY')$!

IF NSPC <> 0 THEN

DATA RECOVERY WITH SPC-REDUCTION
$!

IF BLOAD <> 0 THEN

PRINT('LOG=(' SPC RECOVERY')$!

IF NSPC <> 0 THEN

DATA RECOVERY WITH SPC-REDUCTION
$!

IF BLOAD <> 0 THEN

PRINT('LOG=(' SPC RECOVERY')$!

IF NSPC <> 0 THEN

DATA RECOVERY WITH SPC-REDUCTION
$!

IF BLOAD <> 0 THEN

PRINT('LOG=(' SPC RECOVERY')$!
STAT LEVEL
2469 5: IF NRSET <> 0 [AAN] := [AAF];
2470 5: ENDIF;
2471 4: IF BMODES <> 0 [PHIN] := [PHIT];
2472 4: IF BOFR <> 0 OR BNTR <> 0 [UTRANN] := [UTRANA];
2473 4: IF BDFK <> 0 OR BMR <> 0 [UFREQ] := [UFREQA];
2474 4: ENDIF;
2475 3: IF N Também = 1 CALL NULLMA: {UAG(BC), [AG(BC)], [UAG(BC)], [AAG(BC)]};
2476 4: IF BMODES > 0 THEN
2477 4: ENDIF;
2478 3: IF NRSET <> 0 THEN
2479 4: IF BMODES <> 0 THEN
2480 5: IF NRSET <> 0 THEN
2481 6: IF NRSET <> 0 THEN
2482 7: IF NRSET <> 0 THEN
2483 8: ELSE
2484 9: DATA RECOVERY WITHOUT MPC-REDUCTION
2485 5: IF NRSET <> 0 THEN
2486 6: IF NRSET <> 0 THEN
2487 7: IF NRSET <> 0 THEN
2488 8: ELSE
2489 9: DATA RECOVERY WITHOUT MPC-REDUCTION
2490 5: IF NRSET <> 0 THEN
2491 6: IF NRSET <> 0 THEN
2492 7: IF NRSET <> 0 THEN
2493 8: ELSE
2494 9: DATA RECOVERY WITHOUT MPC-REDUCTION
2495 5: IF NRSET <> 0 THEN
2496 6: IF NRSET <> 0 THEN
2497 7: IF NRSET <> 0 THEN
2498 8: ELSE
2499 9: DATA RECOVERY WITHOUT MPC-REDUCTION
STAT LVL
2541 5:
  2542 5:
  2543 4:
  2544 4:
  2545 4:
  2546 4:
2547 3:
  2548 3:
  2549 3:
  2550 3:
  2551 3:
  2552 3:
  2553 3:
  2554 3:
  2555 3:
  2556 3:
  2557 3:
  2558 3:
  2559 4:
  2560 4:
  2561 4:
  2562 5:
  2563 5:
  2564 5:
  2565 5:
  2566 5:
  2567 5:
  2568 6:
  2569 6:
  2570 6:
  2571 6:
  2572 6:
  2573 6:
  2574 6:
  2575 6:
  2576 6:
  2577 7:
  2578 7:
  2579 7:
  2580 7:
  2581 7:
  2582 7:
  2583 7:
  2584 7:
  2585 7:
  2586 7:
  2587 8:
  2588 8:
  2589 8:
  2590 8:
  2591 8:
  2592 8:

RECOVER PHYSICAL BLAST: EXPLINE DISPLACEMENTS
2548 3:
2549 3:
2550 3:
2551 3:
2554 3:
2555 3:
2556 3:
2557 3:
2558 3:
2559 4:
2560 4:
2561 4:
2562 5:
2563 5:
2564 5:
2565 5:
2566 5:
2567 5:
2568 6:
2569 6:
2570 6:
2571 6:
2572 6:
2573 6:
2574 6:
2575 6:
2576 6:
2577 7:
2578 7:
2579 7:
2580 7:
2581 7:
2582 7:
2583 7:
2584 7:
2585 7:
2586 7:
2587 8:
2588 8:
2589 8:
2590 8:
2591 8:
2592 8:

PRINT("LOG": "OUTPUT PROCESSING": ";
2555 3:
2556 3:
2557 3:
2558 3:
2559 4:
2560 4:
2561 4:
2562 5:
2563 5:
2564 5:
2565 5:
2566 5:
2567 5:
2568 6:
2569 6:
2570 6:
2571 6:
2572 6:
2573 6:
2574 6:
2575 6:
2576 6:
2577 7:
2578 7:
2579 7:
2580 7:
2581 7:
2582 7:
2583 7:
2584 7:
2585 7:
2586 7:
2587 8:
2588 8:
2589 8:
2590 8:
2591 8:
2592 8:

RECOVER STATIC AEROELASTIC LOADS DATA
2557 4:
2558 4:
2559 4:
2560 4:
2561 4:
2562 5:
2563 5:
2564 5:
2565 5:
2566 5:
2567 5:
2568 6:
2569 6:
2570 6:
2571 6:
2572 6:
2573 6:
2574 6:
2575 6:
2576 6:
2577 7:
2578 7:
2579 7:
2580 7:
2581 7:
2582 7:
2583 7:
2584 7:
2585 7:
2586 7:
2587 8:
2588 8:
2589 8:
2590 8:
2591 8:
2592 8:

RECOVERY PHYSICAL BLAST: EXPLINE DISPLACEMENTS
2548 3:
2549 3:
2550 3:
2551 3:
2554 3:
2555 3:
2556 3:
2557 3:
2558 3:
2559 4:
2560 4:
2561 4:
2562 5:
2563 5:
2564 5:
2565 5:
2566 5:
2567 5:
2568 6:
2569 6:
2570 6:
2571 6:
2572 6:
2573 6:
2574 6:
2575 6:
2576 6:
2577 7:
2578 7:
2579 7:
2580 7:
2581 7:
2582 7:
2583 7:
2584 7:
2585 7:
2586 7:
2587 8:
2588 8:
2589 8:
2590 8:
2591 8:
2592 8:

RECOVERY PHYSICAL BLAST: EXPLINE DISPLACEMENTS
2548 3:
2549 3:
2550 3:
2551 3:
2554 3:
2555 3:
2556 3:
2557 3:
2558 3:
2559 4:
2560 4:
2561 4:
2562 5:
2563 5:
2564 5:
2565 5:
2566 5:
2567 5:
2568 6:
2569 6:
2570 6:
2571 6:
2572 6:
2573 6:
2574 6:
2575 6:
2576 6:
2577 7:
2578 7:
2579 7:
2580 7:
2581 7:
2582 7:
2583 7:
2584 7:
2585 7:
2586 7:
2587 8:
2588 8:
2589 8:
2590 8:
2591 8:
2592 8:

RECOVERY PHYSICAL BLAST: EXPLINE DISPLACEMENTS
2548 3:
2549 3:
2550 3:
2551 3:
2554 3:
2555 3:
2556 3:
2557 3:
2558 3:
2559 4:
2560 4:
2561 4:
2562 5:
2563 5:
2564 5:
2565 5:
2566 5:
2567 5:
2568 6:
2569 6:
2570 6:
2571 6:
2572 6:
2573 6:
2574 6:
2575 6:
2576 6:
2577 7:
2578 7:
2579 7:
2580 7:
2581 7:
2582 7:
2583 7:
2584 7:
2585 7:
2586 7:
2587 8:
2588 8:
2589 8:
2590 8:
2591 8:
2592 8:
Standard MAPOL Sequence Listing — Continued

STAT LEVL
2591 6: [DELTA(SUB)], [AIMAT(MINDEX)],
2592 6: [UAG(BC)], [OAGRLOAD, OAGRDDSP];
2593 6: ELSE
2594 6: IF SYM = -1 THEN
2595 7: CALL OFFAEROM (NITER, BC, MINDEX, SUB, GSIZE, GEOMSA,
2596 7: [GTKG], [GSTRG], QDF, [AIRFRC(MINDEX)],
2597 7: [DELTA(SUB)], [AIMAT(MINDEX)],
2598 7: (UAG(BC)], OAGRLOAD, OAGRDDSP).
2599 7: ENDIF;
2600 7: ENDIF;
2601 7: ENVIF;
2602 7: ELSE
2603 5: ENDOC;
2604 4: ENDIF;
2605 3: IF BDRSP <> 0 THEN
2606 4: CALL OFFLOAD (BC, BGPDT(BC), PSIZE(BC), ESIZE(BC), [PHIG(BC)],
2607 4: [PTLOAD], [PTNLLOAD], [PFNLLOAD], OGRIDLOD);
2608 4: IF BDRS <> 0 OR BMTR <> 0
2609 5: CALL OFFGCF (0, BC, 5, 1, GSIZE, ESIZE(BC),
2610 5: [PHIG(BC)], [PHGL]. [PTGL], [PFFLOAD], [PFHLOAD], OGRIDLOD);
2611 5: BGPDT(BC), OGRIDLOD);
2612 5: IF BDRS <> 0 OR BMTR <> 0
2613 5: CALL OFFGCF (0, BC, 6, 2, GSIZE, ESIZE(BC),
2614 5: [PHIG(BC)], [PHGL]. [PTGL], [PFFLOAD], [PFHLOAD],
2615 5: BGPDT(BC), OGRIDLOD);
2616 5: ENDIF;
2617 4: CALL OFFLOAD (NUMOPTBC, BC, GSIZE, BGPDT(BC), PSIZE(BC),
2618 4: [PHIG(BC)], [PHGL]. [PTGL], [PFFLOAD], [PFHLOAD], OGRIDLOD);
2619 4: CALL OFPSCF (0, BC, 6, 2, GSIZE, ESIZE(BC),
2620 4: [PHIG(BC)], [PHGL]. [PTGL], [PFFLOAD], [PFHLOAD], OGRIDLOD);
2621 4: CALL EUR (NUMOPTBC, BC, NZV, GSIZE, EDSM, EODISC,
2622 4: GLBDES, LCLVAR, [PFTRANS],
2623 4: [UAG(BC)], [UAGR(BC)], [UAGRLOAD], [UAGRDDSP]);
2624 4: ENDIF;
2625 3: END;
3. THE SOLUTION CONTROL PACKET

The solution control packet provides the means by which the user selects the optimization and analysis tasks to be performed by the ASTROS system, their order of execution and the engineering data related to each. The solution control commands are analogous in purpose to the NASTRAN Case Control commands but they are very different in form and subtly different in interpretation. Understanding the differences between ASTROS and NASTRAN in the area of solution control is fundamental in understanding multidisciplinary optimization in the ASTROS system because the solution control command structure follows directly from the ASTROS capability to perform multidisciplinary analyses in a single run. It is critical that the user clearly understand the subtleties of solution control syntax and hierarchies. This section, therefore, augments the presentation of the solution control mechanics with a discussion of the design considerations that are embodied in the solution control commands. The detailed definition of all solution control commands follows at the end of the chapter.

In ASTROS, the solution control is very closely linked to the structure of the standard MAPOL sequence. It may be advantageous for the beginning user to read the standard MAPOL sequence discussion in the preceding section and to study the Theoretical Manual discussion of multidisciplinary optimization before reading the remainder of this section.

The solution control packet is initiated with the keyword SOLUTION which follows the DEBUG and MAPOL packets (if present) in the input data stream. The packet is terminated when the BULK DATA packet, or the end of the input stream, is encountered. The data are composed of solution control statements which can begin in any column and can extend over multiple physical records. Each statement is formed from a combination of keywords separated by blanks or commas as indicated in the detailed syntactical descriptions at the end of the chapter. Further, each command keyword can be abbreviated by the first four (or more) characters of the keyword. The solution control packet follows a prescribed hierarchy with the following levels:

<table>
<thead>
<tr>
<th>INITIAL LEVEL (Level 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE OF BOUNDARY CONDITION (Level 2)</td>
</tr>
<tr>
<td>BOUNDARY CONDITION(S) (Level 3)</td>
</tr>
<tr>
<td>DISCIPLINE(S) (Level 4)</td>
</tr>
</tbody>
</table>

Each of these levels is discussed in the following sections and compared and contrasted to their NASTRAN counterparts. In addition to these hierarchical commands, there are commands for output processing that can occur at several levels in the hierarchy. This section presents the available commands and output quantities, but the reader is referred to Chapter 5 of this document for the in-depth presentation of ASTROS output processing.
The hierarchical nature of solution control means that, if the user enters a command at one level in the hierarchy, it remains in effect at all subsequent levels at or below the current one unless overridden. If it is overridden at the same level, that overwrites the original command. If, on the other hand, the command is overridden at a lower level, it only supercedes the original command for the duration of that level and lower levels. Solution Control reverts to use the higher level default after the lower level has been left. Table 14 describes how the commands move from one level to the next and the defaults that they use in each.

### Table 14. Levels of Solution Control

<table>
<thead>
<tr>
<th>CURRENT LEVEL IS:</th>
<th>INCREASING LEVELS</th>
<th>DECREASING LEVELS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IF COMMAND IS:</td>
<td>USE DEFAULTS FROM:</td>
</tr>
<tr>
<td>LEVEL 1 (Initial)</td>
<td>ANALYZE</td>
<td>LEVEL 1</td>
</tr>
<tr>
<td>LEVEL 2</td>
<td>BOUNDARY</td>
<td>LEVEL 2</td>
</tr>
<tr>
<td>LEVEL 3</td>
<td>Discipline commands (e.g., STATICS)</td>
<td>LEVEL 3</td>
</tr>
<tr>
<td>LEVEL 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The user must be aware of these hierarchies especially when requesting output at higher levels. It is possible to get print requests by default where they are not expected if one is not careful with the solution control hierarchy. Another common problem is to place an output request on the wrong side of a level-incrementing solution command thus placing a command at a higher level than expected. Consider the following two examples:

<table>
<thead>
<tr>
<th>EXAMPLE 1</th>
<th>EXAMPLE 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPTIMIZE</td>
<td>OPTIMIZE</td>
</tr>
<tr>
<td>BOUNDARY SPC=1</td>
<td>BOUNDARY SPC=1</td>
</tr>
<tr>
<td>LABEL = CASE 1</td>
<td>STATICS (MECH=10)</td>
</tr>
<tr>
<td>STATICS (MECH=10)</td>
<td>LABEL = CASE 1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>LABEL = CASE 2</td>
<td>STATICS (MECH=20)</td>
</tr>
<tr>
<td>STATICS (MECH=20)</td>
<td>LABEL = CASE 2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>LABEL = CASE 3</td>
<td>STATICS (MECH=30)</td>
</tr>
<tr>
<td>STATICS (MECH=30)</td>
<td>LABEL = CASE 3</td>
</tr>
<tr>
<td>END</td>
<td>END</td>
</tr>
</tbody>
</table>

In example 1, there are three discipline commands, STATICS, and three LABEL commands, one for each discipline. The indenture in the example helps to explain the results of these commands. The first STATICS case will be labelled CASE 2, because the LABEL command appears at LEVEL 4 with the STATICS (MECH=10) command. Similarly, the second STATICS case will be labelled CASE 3. Finally, the third STATICS case will be labelled CASE 1 because that particular LABEL command appeared at LEVEL 3 prior to STATICS (MECH=10). Example 2 illustrates the probable intent of the user. Here, the LABEL commands are placed below the STATICS command. As a result, the LABELs match the cases.

3.1. TYPE OF BOUNDARY CONDITION

ASTROS has been designed primarily to be an automated design tool, but it can also perform analyses without doing any design. This is reflected in the division of the solution control packet into two subpackets, either of which is optional. The first, or OPTIMIZE, subpacket defines the boundary condition(s) and discipline(s) which will generate design constraints to be used in the redesign task. In defining an optimization boundary condition, the user either implicitly or explicitly specifies that constraints be applied to certain (discipline dependent) response quantities. ASTROS then considers the complete set of constraints from all disciplines in all optimization boundary conditions in the redesign task. The second, ANALYZE, subpacket defines analyses that are to be performed on the possibly redesigned structure. The ANALYZE subpacket is intended to provide the designer with the means to obtain additional output that is not desired during the optimization phase or to perform additional analyses which were not performed in the design task. It can also be used to perform analyses on structures that are not to be designed at all. The form of the solution control packet is then:
If optimization is being performed, the OPTIMIZE subpacket must precede the ANALYZE subpacket. Any number of boundary conditions and/or disciplines can be performed in either subpacket.

3.2. BOUNDARY CONDITION

Each analysis discipline requires a set of physical boundary conditions and, in the case of unrestrained structures, a set of fictitious supports. These are defined in ASTROS in a manner very similar to that in NASTRAN; namely, through the definition of multipoint constraints (MPC), single point constraints (SPC) and support points (SUPORT). Unlike NASTRAN, however, ASTROS requires a more rigorous definition of a boundary condition. The reason for this is that the user must ensure that the system matrices at each stage of matrix reduction up to the analysis set are uniquely defined by the boundary condition specification. At or below the analysis set, certain disciplines allow looping over families of direct matrix input, damping options, transfer functions, etc. For example, if the user intends to perform a normal modes analysis, a modal transient analysis and a modal flutter analysis in the same boundary condition, ASTROS requires that the modal representation of the system under analysis be the same for each discipline in the boundary condition. This requirement, which is necessary to efficiently perform multidisciplinary analysis, adds a number of additional parameters to the boundary condition definition beyond the MPC, SPC and SUPORT definitions. They include definitions to perform matrix reductions (available in NASTRAN through Bulk Data but not always selectable in the Case Control Deck) as well as selection of additional point degrees of freedom. In NASTRAN, these data are either implicitly selected through the rigid format selection and/or bulk data or are a "discipline option" in the case control deck. While the boundary condition definition in ASTROS appears to be very complex, it is relatively simple if one realizes that the fundamental purpose of the BOUNDARY command is to uniquely specify the system level matrices and the matrix reductions that should be performed on them.

There is one level of boundary condition specification which is not treated in the BOUNDARY command. It deals with geometry options which play a restrictive role in multi-disciplinary analysis, especially for aerodynamic disciplines. The symmetry options are often limited by the nature of the structural and/or aerodynamic models that are defined in the bulk data packet. For example, if the structural model is a half model only, the user cannot specify that asymmetric structural boundary conditions be analyzed. As a more common example, the user might want to perform an asymmetric aeroelastic analysis with a structural half model. Unfortunately, this is not possible in ASTROS. Whenever possible, the implicit (model-defined) boundary condition specifications that existed in the NASTRAN bulk data definitions and in the interface between bulk data and solution control have been replaced with solution control dependent options. There are, however, still limitations imposed through
the interactions between the model and the solution control on combining symmetric/antisymmetric and asymmetric boundary conditions within a single run. The ten boundary condition specifications in ASTROS are shown in the following table:

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPC</td>
<td>Selects single point constraints defining DOF's with fixed or prescribed motion.</td>
</tr>
<tr>
<td>MPC</td>
<td>Selects multi-point constraints defining dependency relations among specific DOF's.</td>
</tr>
<tr>
<td>REDUCE</td>
<td>Defines the DOF's to be retained after a Guyan reduction.</td>
</tr>
<tr>
<td>SUPPORT</td>
<td>Defines DOF's to provide support conditions for free-free modal extraction, inertial relief and aeroelastic analyses.</td>
</tr>
<tr>
<td>METHOD</td>
<td>Specifies an EIGR bulk data entry which gives eigenvalue extraction data if an eigenanalysis is to be performed.</td>
</tr>
<tr>
<td>DYNRED</td>
<td>Invokes dynamic reduction.</td>
</tr>
<tr>
<td>INERTIA</td>
<td>Specifies a JSET bulk data set for dynamic reduction.</td>
</tr>
<tr>
<td>ESET</td>
<td>Specifies the extra point DOF's to be included in dynamic response analyses.</td>
</tr>
<tr>
<td>K2GG</td>
<td>Specifies the name of the direct stiffness matrix input in the structural set (g-set) to be included in ALL analyses.</td>
</tr>
<tr>
<td>M2GG</td>
<td>Specifies the name of the direct mass matrix input in the structural set (g-set) to be included in ALL analyses.</td>
</tr>
</tbody>
</table>

A boundary condition is defined by the BOUNDARY request and one or more of these further specifications, each of which points to bulk data entries.

As enumerated above, the specification of METHOD and ESET at this level in the hierarchy is in recognition of the fact that a number of the disciplines could require different sets of data for the associated items and it is desirable to group operations with one set of items together. This does, by definition, create a restriction that only one eigenanalysis and only one size of p-size matrices can be accommodated per boundary condition. Examples of boundary definitions are:

```
BOUNDARY SPC = 100
BOUNDARY MPC = 10, SPC = 100
BOUNDARY SPC = 10, MPC = 20, REDUCE = 30, SUPPORT = 40
BOUNDARY SPC = 10, REDUCE = 20, METHOD = 100
BOUNDARY SPC = 1, K2GG = STIFF, M2GG = MASS
BOUNDARY SPC = 4, DYNRED = 2, INERTIA = 4
```

Note that all desired specifications are listed and that their order of appearance is not important. At least one option is required.
Several boundary conditions may appear within a given subpacket. For example:

```
ANALYZE
  BOUNDARY SPC = 10
  STATICS (MECH=5)
  ...
  ...
  BOUNDARY SPC = 20, REDUCE = 30, METHOD = 1111
  STATICS (THERM=10)
  ...
  ...
  MODES
  ...
  ...
END
```

In this case, a `STATICS` analysis is performed using the first boundary condition followed by a `STATICS` and modes analysis for the second boundary condition. Note that unlike NASTRAN, the sets of points to be retained in the Guyan reduction and used for the support definition are selected.

The appearance of a `BOUNDARY` command leads to expensive matrix partitioning and decomposition operations. Therefore, some thought should be expended to avoid unnecessary computer resource use. For example, suppose an ASTROS execution was directed to perform static analyses with two boundaries: `SPC=10` and `SPC=20`, and a dynamic analysis with two boundaries: `SPC=10` and `SPC=100`. The direct solution sequence could be:

```
ANALYZE
  BOUNDARY SPC = 10
  STATICS (MECH=10)
  BOUNDARY SPC = 20
  STATICS (MECH=20)
  ...
  ...
  BOUNDARY SPC = 10, METHOD = 30
  MODES
  ...
  ...
  ...
  BOUNDARY SPC = 100, METHOD = 40
  MODES
  ...
  ...
END
```
This sequence would cause four separate partitionings of the system level matrices. On the other hand, the sequence:

```plaintext
ANALYZE
BOUNDARY SPC = 10, METHOD=30
  STATICS (MECH=10)
    ...
  MODES
  ...
BOUNDARY SPC = 20
  STATICS (MECH=20)
    ...
  ...
BOUNDARY SPC = 100, METHOD = 40
  MODES
    ...
  ...
END
```

eliminates one of the four partitioning operations.

### 3.3. DISCIPLINES

A number of types of analyses, or disciplines, can be performed during a given `ANALYZE` or `OPTIMIZE` boundary condition. In fact, it is this multidisciplinary capability that makes the ASTROS code viable in a preliminary design context. The preceding sections have already alluded to the fact that each of these disciplines has an associated set of commands:

```plaintext
ANALYZE
BOUNDARY SPC = 30
  DISCIPLINE 1
    ...
  ...
  DISCIPLINE 2
    ...
  ...
END
```
A suite of eight disciplines are available in ASTROS as shown in Table 15. Of these options, TRANSIENT, FREQUENCY, BLAST and NPSAERO do not generate any design constraints and so are not useful in OPTIMIZE boundary conditions. Should the user wish to see output from these disciplines during the optimization, however, they are supported in the OPTIMIZE subpacket with the exception of NPSAERO. Since NPSAERO does not require a structural model of any kind, it is restricted to the ANALYZE subpacket. As discussed in Chapter 2, its solution is performed in the ASTROS preface phase.

The standard MAPOL sequence contains almost no restrictions on the combination of disciplines and subcases in a boundary condition. SAERO disciplines, for example, require multiple symmetry, Mach number and dynamic pressure dependent correction matrices. The standard algorithm automatically re-sorts the user's input subcases to solve the maximum number of right hand sides for a given aeroelastic correction matrix. The results are then returned to the user's order with no limitations imposed. Similarly, the flutter discipline loops over a set of direct dynamic input matrices to accommodate multiple closed loop systems using a single set of structural matrices. The only limits are those of symmetry discussed earlier in which the structural and aerodynamic symmetries should be the same for all subcases in a boundary condition and the restriction to a single transient and a single frequency response per boundary condition.

<table>
<thead>
<tr>
<th>DISCIPLINE</th>
<th>DESCRIPTION</th>
<th>DISCIPLINE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATICS</td>
<td>Static structural analysis</td>
<td>TRANSIENT</td>
<td>Transient response analysis</td>
</tr>
<tr>
<td>MODES</td>
<td>Normal modes of vibration</td>
<td>FREQUENCY</td>
<td>Frequency response analysis</td>
</tr>
<tr>
<td>SAERO</td>
<td>Steady-state aeroelastic analysis</td>
<td>BLAST</td>
<td>Transient response to a nuclear blast.</td>
</tr>
<tr>
<td>FLUTTER</td>
<td>Aeroelastic stability analysis</td>
<td>NPSAERO</td>
<td>Nonplanar rigid static aerodynamic analysis.</td>
</tr>
</tbody>
</table>
3.3.1. DISCIPLINE OPTIONS

Each of the disciplines requires further options to completely define the execution process. These options point to set IDs in the bulk data packet that define engineering data. For example, the STATICS discipline requires that loads information be supplied. This is implemented in ASTROS by a parenthetical "phrase" attached to the STATICS discipline:

```
SOLUTION
  OPTIMIZE STRATEGY=FSD
  ...
  ...
  STATICS (MECH=10)
  ...
END
```

In this case, bulk data applied load entries with a set ID of 10 are used to construct a mechanical load vector in a STATICS analysis. In general, the discipline commands have the form:

```
<disc> <type> [(<option> = <n>, <option> = <n>)]
```

The discipline options that are available are:

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>MECHANICAL</td>
<td>Specify load set IDs for the STATICS discipline.</td>
</tr>
<tr>
<td>GRAVITY</td>
<td></td>
</tr>
<tr>
<td>THERMAL</td>
<td></td>
</tr>
<tr>
<td>TRIM</td>
<td>Specifies a TRIM bulk data entry which gives flight condition information for the SAERO and BLAST disciplines.</td>
</tr>
<tr>
<td>DCON DCONSTRAINT</td>
<td>Specifies the set IDs of constraint bulk data entries that apply for the given discipline.</td>
</tr>
<tr>
<td>STRESS</td>
<td>Specifies the set IDs of stress constraint bulk data entries that apply for the given STATICS or SAERO discipline.</td>
</tr>
<tr>
<td>STRESSCONSTRAINT</td>
<td></td>
</tr>
<tr>
<td>STRAIN</td>
<td>Specifies the set IDs of strain constraint bulk data entries that apply for the given STATICS or SAERO discipline.</td>
</tr>
<tr>
<td>STRAINCONSTRAINT</td>
<td></td>
</tr>
<tr>
<td>DLOAD</td>
<td>Specifies applied loads for the TRANSIENT and FREQUENCY disciplines.</td>
</tr>
<tr>
<td>TSTEP</td>
<td>Specifies the time step for the TRANSIENT and BLAST disciplines as well as for the discrete form of the GUST discipline.</td>
</tr>
<tr>
<td>FSTEP</td>
<td>Specifies the frequencies for the FREQUENCY and the harmonic form of the GUST discipline.</td>
</tr>
<tr>
<td>IC</td>
<td>Specifies the initial conditions that are to be used in the direct method for the TRANSIENT discipline.</td>
</tr>
<tr>
<td>FFT</td>
<td>Specifies that the Fast Fourier technique is to be used in the TRANSIENT or GUST disciplines.</td>
</tr>
<tr>
<td>DLCOND</td>
<td>Specifies parameters for the BLAST discipline.</td>
</tr>
<tr>
<td>FLCOND</td>
<td>Specifies parameters for the FLUTTER discipline.</td>
</tr>
<tr>
<td>OPTION</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>CONTROL</td>
<td>Specifies the name of a control surface modifier matrix for flutter analysis.</td>
</tr>
<tr>
<td>GUST</td>
<td>Specifies that a gust analysis is to be performed for the accompanying transient or frequency discipline.</td>
</tr>
<tr>
<td>K2PP</td>
<td>Specifies an input stiffness matrix on the physical degrees of freedom for FREQUENCY, TRANSIENT and FLUTTER disciplines.</td>
</tr>
<tr>
<td>M2PP</td>
<td>Specifies an input mass matrix on the physical degrees of freedom for FREQUENCY, TRANSIENT and FLUTTER disciplines.</td>
</tr>
<tr>
<td>B2PP</td>
<td>Specifies an input damping matrix on the physical degrees of freedom for FREQUENCY, TRANSIENT and FLUTTER disciplines.</td>
</tr>
<tr>
<td>TFL</td>
<td>Specifies transfer functions that are to be included in FREQUENCY, TRANSIENT and FLUTTER disciplines.</td>
</tr>
<tr>
<td>DAMPING</td>
<td>Specifies structural or viscous damping to be used in FREQUENCY, TRANSIENT and FLUTTER disciplines.</td>
</tr>
</tbody>
</table>

The discipline types are:

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIRECT</td>
<td>Specifies that the direct method is to be used in the TRANSIENT or FREQUENCY disciplines.</td>
</tr>
<tr>
<td>MODAL</td>
<td>Specifies that the modal method is to be used in the TRANSIENT or FREQUENCY disciplines.</td>
</tr>
<tr>
<td>SYMMETRIC</td>
<td>Specifies that the SAERO or NPSAERO subcase is to use aerodynamics derived with symmetric conditions about the Y=0 plane.</td>
</tr>
<tr>
<td>ANTISYMMETRIC</td>
<td>Specifies that the SAERO or NPSAERO subcase is to use aerodynamics derived with antisymmetric conditions about the Y=0 plane.</td>
</tr>
</tbody>
</table>

Table 16 presents a matrix that defines options and types available for each of the disciplines. In addition, disciplines requiring particular boundary condition specifications are noted; for example, modal disciplines require a METHOD specification on the BOUNDARY command. The following subsections present each discipline in turn to more explicitly define the discipline options. Most importantly, these subsections present the definition of a "subcase" of the discipline as it is defined in the ASTROS system and present the response quantities that can be constrained in the optimization task.

3.3.2. STATICS Discipline Options

One or more of the MECHANICAL, GRAVITY or THERMAL load specifications must be called out as a discipline option for STATICS. Each STATICS discipline constitutes one subcase (one load vector) so specifying a combination of load types will generate a linear combination of the selected loads. A reference to the LOAD bulk data entry as a MECHANICAL load can also be used to obtain linear load combinations. If the STATICS discipline appears in the OPTIMIZE subpacket, the DCONSTRAINT option can be used to refer to DCONDSP bulk data entries to apply displacement constraints. Stress constraints defined on DCONFT, DCONFTM, DCONFTP, DCONTW, DCONTWM, DCONTWP, DCONVM, DCONVWM, DCONVMP are selected.
Table 16. Summary of Discipline Options

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>DISCIPLINE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>STAT</td>
</tr>
<tr>
<td>MECH</td>
<td>×</td>
</tr>
<tr>
<td>GRAV</td>
<td>×</td>
</tr>
<tr>
<td>THERM</td>
<td></td>
</tr>
<tr>
<td>TRIM</td>
<td></td>
</tr>
<tr>
<td>DCONSTRAINT</td>
<td>×</td>
</tr>
<tr>
<td>STRESS</td>
<td>×</td>
</tr>
<tr>
<td>STRAIN</td>
<td>×</td>
</tr>
<tr>
<td>DLOAD</td>
<td></td>
</tr>
<tr>
<td>TSTEP</td>
<td></td>
</tr>
<tr>
<td>FSTEP</td>
<td></td>
</tr>
<tr>
<td>IC</td>
<td></td>
</tr>
<tr>
<td>FFT</td>
<td></td>
</tr>
<tr>
<td>DIRECT</td>
<td></td>
</tr>
<tr>
<td>MODAL</td>
<td></td>
</tr>
<tr>
<td>BLCOND</td>
<td></td>
</tr>
<tr>
<td>FLCOND</td>
<td></td>
</tr>
<tr>
<td>GUST</td>
<td></td>
</tr>
<tr>
<td>K2PP</td>
<td></td>
</tr>
<tr>
<td>M2PP</td>
<td></td>
</tr>
<tr>
<td>B2PP</td>
<td></td>
</tr>
<tr>
<td>TFL</td>
<td></td>
</tr>
<tr>
<td>DAMPING</td>
<td></td>
</tr>
<tr>
<td>SYMMETRIC</td>
<td></td>
</tr>
<tr>
<td>ANTSYMMETRIC</td>
<td></td>
</tr>
</tbody>
</table>

Notes:
- Required Commands: ●
- Optional Commands: ○
by the STRESSCONSTRAINT option. Strain constraints defined on DCONEP, DCONEPM and DCONEPF are selected by the STRAINCONSTRAINT option. All DCONxxx bulk data entries, such as DCONTHK, that do not have SETID fields will be applied to the model in combination with set selectable constraints to make up the set of design constraints.

### 3.3.3. Modes Discipline Options

**MOMES** is completely defined for analysis by the METHOD boundary specification, which refers to an EIGR bulk data entry selecting the eigenvalue extraction method. If, however, the modal analysis is performed in the OPTIMIZE subpacket, the DCONSTRAINT option can be used to apply frequency constraints through the DCONFRQ bulk data entry. Note that more than one frequency can be constrained and that more than one constraint can be placed on the same modal frequency. The user is warned against defining the frequency constraints in such a way as to specify an excluded range of frequencies for a mode; for example, requiring that a modal frequency be below 10 Hz OR above 20 Hz. ASTROS treats all applied constraints as Boolean AND statements so the above example would be interpreted by ASTROS as an inconsistent requirement that the frequency be both above 20 Hz and below 10 Hz. All DCONxxx bulk data entries, such as DCONTHK, that do not have SETID fields will be applied to the model in combination with set selectable constraints to make up the set of design constraints.

In ASTROS, each eigenvector is considered to be a separate subcase. It is important to note in this case that more than one subcase is represented by a single solution control discipline statement. In output requests, therefore, the subcases for which output is desired must be explicitly selected. This is presented in greater detail in section 3.4 and in Chapter 5.

### 3.3.4. SAERO Discipline Options

The **SAERO** discipline must have a TRIM condition and symmetry type specified in the solution control. The symmetry default is **SYMMETRIC**. For analysis, this selection completes the specification of the discipline with each TRIM condition generating one subcase. In the OPTIMIZE subpacket, the DCONSTRAINT option can be used to select a number of different constraint types which depend on the type of TRIM analysis selected. In general the DCONSTRAINT can refer to DCONDSF bulk data entries for displacement constraints, DCONCLA for lift effectiveness constraints, DCONALE for aileron effectiveness constraints, DCONSCF for stability coefficient constraints and DCONTRM for constraints on trim parameters. The **SAERO** discipline always generates a static displacement field to which any static constraint may be applied. Stress constraints defined on DCONFT, DCONFTM, DCONFTP, DCONTW, DCONTWM, DCONTWP, DCONVM, DCONVMM, DCONVMP are selected by the STRESSCONSTRAINT option. Strain constraints defined on DCONEP, DCONEPM and DCONEPF are selected by the STRAINCONSTRAINT option. All DCONxxx bulk data entries, such as DCONTHK, that do not have SETID fields will be applied to the model in combination with set selectable constraints to make up the set of design constraints.

### 3.3.5. Flutter Discipline Options

The **FLUTTER** discipline must have a flight condition specified in the solution control through the FLCOND option. In addition, the K2PP, B2PP, M2PP, TFL and DAMPING options may be used with or without an ESET Boundary Condition option to impose a case-by-case set of additional inputs/degrees-of-freedom for modelling control systems, etc. For analysis, this selection completes the specification of the discipline with each FLCOND condition generating up to one "subcase" (consisting of up to one flutter...
eigenvector) for each Mach number and density ratio if flutter occurs. In the OPTIMIZE subpacket, the DCONSTRAINT option can be used to select DCONPLT bulk data entries to place a required damping limit on each of the roots extracted in the flutter analysis. The actual flutter root and eigenvector cannot be obtained in the OPTIMIZE subpacket.

3.3.6. TRANSIENT Discipline Options

The TRANSIENT discipline must have time step and load information specified in the solution control through the TSTEP and DLOAD options. This discipline has no associated constraints and, while it is fully supported in the OPTIMIZE subpacket, it will not generate data for use in the re-design task. There are many additional options which can be selected in transient analysis. These are 1) initial conditions, which can be selected through the IC option for DIRECT transient analyses; 2) Fast Fourier Transform techniques, which are selected with the FFT option; and 3) discrete gust loads, which are applied using the GUST option. In each case, the solution control option points to a bulk data entry having the same name. In addition, the K2PP, B2PP, M2PP, TFL and DAMPING options may be used with or without an ESET Boundary Condition option to impose a case-by-case set of additional inputs/degrees-of-freedom for modelling control systems, etc.

In ASTROS, each time step for which output is saved is considered to be a separate subcase. It is important to note that, like the MODES discipline, more than one subcase is represented by a single solution control discipline statement. In output requests, therefore, the subcases for which output is desired must be explicitly selected. This is presented in greater detail in section 3.4 and in Chapter 5.

3.3.7. FREQUENCY Discipline Options

The FREQUENCY discipline is very similar to the TRANSIENT discipline presented in the preceding subsection. Frequency step and load information are specified in the solution control through the FSTEP and DLOAD options. This discipline has no associated constraints and, while it is fully supported in the OPTIMIZE subpacket, it will not generate data for use in the re-design task. There are two additional options which can be selected in frequency response analysis. These are 1) Fast Fourier Transform techniques, which are selected with the FFT option; and 2) harmonic gust loads, which are applied using the GUST option. In each case, the solution control option points to a bulk data entry having the same name. In addition, the K2PP, B2PP, M2PP, TFL and DAMPING options may be used with or without an ESET Boundary Condition option to impose a case-by-case set of additional inputs/degrees-of-freedom for modelling control systems, etc.

In ASTROS, each frequency step for which output is saved is considered to be a separate subcase. It is important to note that, like the MODES discipline, more than one subcase is represented by a single solution control discipline statement. In output requests, therefore, the subcases for which output is desired must be explicitly selected. This is presented in greater detail in Subsection 3.4 and in Chapter 5.

3.3.8. BLAST Discipline Options

The BLAST discipline's solution control is a limited subset of those in the TRANSIENT discipline. It must have time steps specified in the solution control through the TSTEP option but, unlike the normal transient response, DLOAD is not used to generate the load. Instead, the BLCOND option is used to select a blast condition defined on a BLAST bulk data entry. As with the other dynamic response disciplines, there
are no associated design constraints. These two options along with the analysis type (DIRECT or MODAL) completely specify the BLAST analysis. The K2PP, B2PP, M2PP, TFL and DAMPING options are not available.

In ASTROS, each time step for which output is saved is considered to be a separate subcase. It is important to note that, like the MODES discipline, more than one subcase is represented by a single solution control discipline statement. In output requests, therefore, the subcases for which output is desired must be explicitly selected. This is presented in greater detail in Subsection 3.4 and in Chapter 5.

3.3.9. NPSAERO Discipline Options

The NPSAERO discipline must have a TRIM condition and symmetry type specified in the solution control and is only available in the ANALYZE subpacket. The symmetry default is SYMMETRIC. This selection completes the specification of the discipline with each TRIM condition generating one subcase. The results are the rigid aerodynamic loads on the aerodynamic model. No structural model is needed and flexibility effects are not included in the analysis.

3.4. OUTPUT REQUESTS

Most analysis disciplines in ASTROS have response quantities (displacements, stresses, strains, etc.) computed at either grid points, structural elements or aerodynamic elements. The user can select that these results be written to the print (output) file through the PRINT command and its associated options or written to a punch file through the PUNCH command. In addition, there are a number of solution control commands that can be used to label the output. This subsection documents the PRINT and PUNCH commands and the labeling commands and discusses their use. The PRINT and PUNCH commands are identical in form and interpretation, so the PRINT command will be used to represent both commands in the following discussion. There are also many features and utilities available to the user to obtain output through modifications to the executive MAPOL sequence. These include direct use of MAPOL utilities, modification of print parameters in functional module calling sequences and user written procedures or modules. These output capabilities and a more complete discussion of the output processing (PRINT and PUNCH) capabilities of the ASTROS system is presented in Chapter 5 of this manual.

The PRINT and PUNCH commands have a number of options which can be separated into three groups: subset options, response quantity options and form options. The subset options select a set of subcases and/or design iterations to which the PRINT command applies while the remaining options select the actual data quantities that are desired; (e.g. stresses, strains, and displacements) and the form in which complex quantities are to be printed. The output selection can appear at any level of the solution control hierarchy and will apply at that level until it is overridden. When more than one discipline is covered by a print request at the boundary level, ASTROS will consider only the relevant print requests for each discipline. For example, if STATICS and flutter are performed, the STATICS discipline will ignore any ROOTS requests and the flutter discipline will ignore any STRESS requests.
3.4.1. Subset Options

As alluded to in the preceding subsections, some disciplines have more than one subcase per solution control statement. Others, like STATICS and SAERO have a separate solution control statement for each subcase. In all cases, disciplines within the OPTIMIZE subpacket may be analyzed at one or more design iterations. When one subcase is defined per statement, the user is free to modify the print requests from subcase to subcase; for example:

```
ANALYZE
  BOUNDARY SPC = 10
  STATICS ( MECH = 10 )
    PRINT STRESS = ALL, DISP = 100
  STATICS ( MECH = 20, GRAV = 100 )
    PRINT DISP = ALL
```

specifies that stresses for all elements and displacements for nodes listed in set 100 be printed for the first subcase (mechanical loads with set identification 10) and only the displacements be printed for the next load condition. When the discipline generates more than one subcase, however, the user must specify the subcases to which the PRINT request applies. For example:

```
ANALYZE
  BOUNDARY SPC=10, METHOD=1000
  MODES
    PRINT (MODES=ALL) DISP=100
```

selects the displacements (eigenvectors) for all the computed mode shapes be printed. If the MODES=ALL selection were not included in the PRINT statement, the user would get no output at all. The user is cautioned that the output processing in ASTROS is designed to limit output to those quantities that are explicitly selected and, therefore, the default for subcase option MODES is that no modes are selected. Whenever multiple subcases are generated by a discipline, as in the case of MODES, TRANSIENT, FREQUENCY and BLAST, a subcase selection option is required on the PRINT command in order to get any output.

If the discipline appears in the OPTIMIZE subpacket, the user may request that the output appear only at certain iterations. For example:

```
OPTIMIZE
  BOUNDARY SPC=10, METHOD=10
  MODES (DCON=1000)
  PRINT (ITER=10, MODES=ALL) DISP=100
```

selects the displacements (eigenvectors) for all the computed mode shapes be printed at the iterations given in ITERLIST 10. Unlike the other subset selectors, the default for ITER is ALL. Omission of the ITER selector therefore implies that the quantity will be printed at every iteration. This default is a consequence of compatibility with early versions of ASTROS in which there was no ITERATION selection at all.

The subset selections can be specified at two levels as parenthetical phrases attached to the print or punch statement. At the higher level, the subset options generate defaults for the entire print or punch statement. For example:
requests that all stresses and strains at all time steps for the iterations in ITERLIST 10 be printed. In
addition, the subset options can be attached to the individual quantity options to override the print
default. For example:

\[
\text{PRINT (ITER=10, TIME=ALL) STRESS=ALL, STRAIN=ALL}
\]

overrides the \text{TIME=ALL} default for the strain output. At both levels, the defaults are \text{NONE} for \text{TIME},
\text{FREQ} and \text{MODE} and \text{ALL} for \text{ITER}.

The subset options in ASTROS are:

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>FREQUENCY</td>
<td>Selects the frequency steps of frequency response disciplines at which output is desired by referencing a FREQLIST bulk data entry.</td>
</tr>
<tr>
<td>ITERATION</td>
<td>Selects the design iterations at which output is desired by referencing a ITERLIST bulk data entry.</td>
</tr>
<tr>
<td>MODE</td>
<td>Selects the eigenvectors of a normal modes discipline at which output is desired by referencing a MODELIST bulk data entry.</td>
</tr>
<tr>
<td>TIME</td>
<td>Selects the time steps of transient response and blast analyses at which output is desired by referencing a TIMELIST bulk data entry.</td>
</tr>
</tbody>
</table>

### 3.4.2. Response Quantity Options

ASTROS is able to compute a number of response quantities for each discipline type. Each
discipline type generates a different set of quantities so that the quantity selected by a particular
keyword can sometimes change from one discipline to another. In addition, the available quantities are a
sometimes a function of the boundary condition type. For example, the flutter mode shape is not available
as an output from a flutter analysis performed in the \text{OPTIMIZE} subpacket. This subsection will present
the available quantities, the \text{PRINT} options which select them and the limitations (if any) on their
availability.

The following table summarizes the available \text{PRINT} and \text{PUNCH} response quantity options:

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCELERATION</td>
<td>Selects accelerations at nodal points.</td>
</tr>
<tr>
<td>AIRDISPLACEMENT</td>
<td>Selects displacements on aerodynamic boxes.</td>
</tr>
<tr>
<td>CGRAIIENT</td>
<td>Selects gradients of active constraints.</td>
</tr>
<tr>
<td>DCONSTRAINT</td>
<td>Selects active constraints at each iteration.</td>
</tr>
<tr>
<td>DISPLACEMENTS</td>
<td>Selects displacements at nodal points.</td>
</tr>
<tr>
<td>ENERGY</td>
<td>Selects strain energy at structural elements.</td>
</tr>
<tr>
<td>OPTION</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FORCE</td>
<td>Selects element forces at structural elements.</td>
</tr>
<tr>
<td>GDESIGN</td>
<td>Selects global design variables.</td>
</tr>
<tr>
<td>GFORCE</td>
<td>Selects grid point forces at nodal points.</td>
</tr>
<tr>
<td>GPWG</td>
<td>Selects print of grid point weight summary.</td>
</tr>
<tr>
<td>KSNS</td>
<td>Selects stiffness sensitivities at design variables.</td>
</tr>
<tr>
<td>LDESIGN</td>
<td>Selects local design variables.</td>
</tr>
<tr>
<td>LOAD</td>
<td>Selects applied loads at nodal points.</td>
</tr>
<tr>
<td>MASS</td>
<td>Selects mass matrix at nodal points.</td>
</tr>
<tr>
<td>MODEL</td>
<td>Selects Bulk Data at current design point. (PUNCH only)</td>
</tr>
<tr>
<td>MSNS</td>
<td>Selects mass sensitivities at design variables.</td>
</tr>
<tr>
<td>OGRADIENT</td>
<td>Selects gradient of the objective function.</td>
</tr>
<tr>
<td>QHH</td>
<td>Selects QHH generalized unsteady aerodynamic forces at modes.</td>
</tr>
<tr>
<td>QHJ</td>
<td>Selects QHJ generalized unsteady aerodynamic forces at modes.</td>
</tr>
<tr>
<td>ROOT</td>
<td>Selects flutter and normal modes roots (eigenvalues).</td>
</tr>
<tr>
<td>SPCFORCE</td>
<td>Selects forces of single point constraint at nodal points.</td>
</tr>
<tr>
<td>STIFFNESS</td>
<td>Selects stiffness matrix at nodal points.</td>
</tr>
<tr>
<td>STRAIN</td>
<td>Selects strains at structural elements.</td>
</tr>
<tr>
<td>STRESS</td>
<td>Selects stresses at structural elements.</td>
</tr>
<tr>
<td>TPRESSURE</td>
<td>Selects trim pressures at aerodynamic boxes.</td>
</tr>
<tr>
<td>TRIM</td>
<td>Selects trim and stability coefficients for steady aeroelastic analyses.</td>
</tr>
<tr>
<td>VELOCITY</td>
<td>Selects velocities at nodal points.</td>
</tr>
</tbody>
</table>

As in NASTRAN, stresses, strains and element forces are computed in the element coordinate system at predetermined or user selected points in the element. Nodal quantities are computed in the global coordinate system. CGRA, DCON, GDES, KSNS, MODEL, MSNS, OGRA and HIST are only applicable in the OPTIMIZE subpacket above the first BOUNDARY (since these requests transcend all analyses). The DISP option for flutter analyses is only applicable in the ANALYZE subpacket. Other options are available independent of the boundary condition type. Table 17 presents a matrix of response quantity options for each discipline type, showing the applicability of each option. Any requests for quantities that do not apply to the particular discipline will be ignored by the output processor without warning.
Table 17. Response Quantities by Discipline

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESIGN</th>
<th>STAT</th>
<th>MODE</th>
<th>SAERO</th>
<th>FLUT</th>
<th>TRANS</th>
<th>FREQ</th>
<th>BLAS</th>
<th>NPSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCEL</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>AIRDISP</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CGRADIENT</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCONSTRATNT</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DISP</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ENERGY</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FORCE</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GDDESIGN</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GFFORCE</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>GPWG</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>KSNS</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDESIGN</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOAD</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MASS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MODEL</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSNS</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OGRADIENT</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QGH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QHJ</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROOT</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPFORCE</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STIFFNESS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STRAIN</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STRESS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TPRESSURE</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRIM</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VELO</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Most options can be ALL, NONE or an integer value which selects bulk data entry sets listing the items for which the response quantity is desired. For example, the STRESS option points to the ELEM LIST bulk data entity which lists the elements for which stresses are desired. The NONE option is used to override a default established through a print or punch request at a higher level in the hierarchy. The ASTROS output philosophy is similar to that of NASTRAN in that it is assumed that mistakes in the output requests should not terminate execution. If, for example, the requested structural element does not exist in the model, the output request will be ignored without any warning to the user. Other output request errors in ASTROS are treated in a similar manner, occasionally generating a warning message, but more typically resulting in no visible indication that the request was in error. Therefore the user can, in most cases, request output that does not apply to the discipline, for entities (nodes or elements) which do not exist and/or for subcases that are not defined without causing termination of the execution.

3.4.3. Form Options

For complex response quantities, the form option is provided to select either RECTANGULAR or POLAR form. Rectangular form gives the cartesian components of the quantity in the rectangular complex plane in which the first number represents the real component and the second number the imaginary component. Polar form gives the components in polar coordinates in which the first number represents the radial distance from the origin (the magnitude) and the second represents the angular displacement from the real coordinate axis (the phase angle). The phase angle is computed in degrees.

The form can be specified at two levels as parenthetical phrases attached to the print or punch statement. At the higher level, the form option generates a default for the entire print or punch statement. For example:

```
PRINT (POLAR) STRESS=ALL, STRAIN=ALL
```

requests that polar form be used for both stress and strain response quantities. In addition, the form option can be attached to the individual quantity options to override the print default. For example:

```
PRINT (POLAR) STRESS=ALL, STRAIN(RECT)=ALL
```

overrides the polar default for the strain output. At both levels, the default form is rectangular and any polar requests for real output quantities are ignored.

3.4.4. Labeling Options

Labeling of printed output is performed through the use of three optional commands identical in form to their NASTRAN counterparts:

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE</td>
<td>A title header that will appear as the first line on each page of output.</td>
</tr>
<tr>
<td>SUBTITLE</td>
<td>A secondary header that will appear on the second line of each page of output.</td>
</tr>
<tr>
<td>LABEL</td>
<td>A tertiary header that is typically used to identify subcase (discipline level) output.</td>
</tr>
</tbody>
</table>
Each of these commands can appear at any level in the solution control hierarchy and will be applied until superseded.

3.5. SOLUTION CONTROL COMMANDS

The ASTROS Solution Control Commands are described in this section.
Solution Control Command:  $ 

Description: Allows commentary data to be placed in the Solution Control packet.

Hierarchy Level: Various

Format:
$ THIS IS A COMMENT
Solution Control Command: **ANALYZE**

**Description:** The first command in the ANALYZE subpacket

**Hierarchy Level:** Type of run

**Format:**

**ANALYZE**
Solution Control Command:  **BLAST**

**Description:** Invokes the blast discipline

**Hierarchy Level:** Discipline

**Format and Examples:**

BLAST type (BLCOND = i, TSTEP = j)

BLAST MODAL (BLCOND = 20, "STEP = 30")

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>DIRECT or MODAL. Selects the solution approach. Default = DIRECT.</td>
</tr>
<tr>
<td>i</td>
<td>Set identification of a BLAST bulk data entry used to specify nuclear blast parameters.</td>
</tr>
<tr>
<td>j</td>
<td>Set identification of TSTEP bulk data entries used to specify time step data for the blast response analysis.</td>
</tr>
</tbody>
</table>

**Remarks:**

1. BLAST must not appear in the OPTIMIZE subpacket.
Solution Control Command: BOUNDARY

Description: Specifies the displacement sets and related data used in a particular boundary condition.

Hierarchy Level: Boundary condition

Format and Examples:

BOUNDARY MPC = i, SPC = j, REDUCE = k, SUPPORT = l, METHOD = m, DYNRED = n, INERTIA = o, ESET = p, K2GG = q, M2GG = r

BOUNDARY SPC = 6
BOUNDARY SPC = 10, REDUCE = 20, SUPPORT = 30
BOUNDARY SPC = 12, DYNRED = 100, INERTIA = 100, K2GG = FUSSTIFF

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>Set identification of a multipoint constraint set. Invokes MPC and MPCADD bulk data entries. (Integer&gt;0)</td>
</tr>
<tr>
<td>j</td>
<td>Set identification of a single point constraint set. Invokes SPC, SPC1 and SPCADD bulk data entries. (Integer&gt;0)</td>
</tr>
<tr>
<td>k</td>
<td>Set identification of a static condensation set. Invokes ASET, ASET1, OMIT and OMIT1 bulk data entries. (Integer&gt;0)</td>
</tr>
<tr>
<td>l</td>
<td>Set identification of the free body support. Invokes SUPPORT bulk data entries. (Integer&gt;0)</td>
</tr>
<tr>
<td>m</td>
<td>Set identification of the NIGR bulk data entry to be used. (Integer&gt;0)</td>
</tr>
<tr>
<td>n</td>
<td>Selects the dynamic reduction parameters from the DYNRED bulk data entry (Integer&gt;0)</td>
</tr>
<tr>
<td>o</td>
<td>Selects the JSBT1 bulk data entries identifying inertia relief degrees of freedom for performing dynamic reduction (Integer&gt;0)</td>
</tr>
<tr>
<td>p</td>
<td>Set identification of the extra degrees of freedom for the boundary condition. Invokes EPOINT bulk data entries. (Integer&gt;0)</td>
</tr>
<tr>
<td>q</td>
<td>Selects the direct input stiffness matrix in the g-set. This matrix will be added to KGG for this boundary condition. Refers to a DMI or DMIG Bulk Data entry.</td>
</tr>
<tr>
<td>r</td>
<td>Selects the direct input mass matrix in the g-set. This matrix will be added to MGG for this boundary condition. Refers to a DMI or DMIG Bulk Data entry.</td>
</tr>
</tbody>
</table>

Remarks:

1. Note that the REDUCE and ESET set specifications are innovative relative to NASTRAN.
2. The bulk data entries will not be used in ASTROS unless selected in Solution Control.
3. None of the options are required but at least one must appear.
4. K2GG and M2GG affect the system stiffness and mass matrices, respectively, for all disciplines within the boundary condition.
5. K2GG and M2GG names will typically refer to DMI or DMIG entries but may refer to any data base matrix entity of the proper dimension.
Solution Control Command: END

Description: Indicates the end of a subpacket.

Hierarchy Level: End

Format:
END

Remarks:
1. The ANALYZE and OPTIMIZE subpackets each require an END command.
Solution Control Command: FLUTTER

Description: Invokes the flutter analysis discipline

Hierarchy Level: Discipline

Format and Examples:

```
FLUTTER (FLCOND = i, DCONSTRAINT = j, CONTROL = k, K2PP = l, M2PP = m, B2PP = n, TFL = o, DAMPING = p )
```

```
FLUTTER (FLCOND = 100 )
```

```
FLUTTER (FLCOND=100, CONTROL=AILERON, K2PP=AILM, M2PP=MAIL, TFL=5)
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>Set identification of a FLUTTER bulk data entry that provides flutter parameters.</td>
</tr>
<tr>
<td>j</td>
<td>Set identification of a DCONFLT bulk data entry that defines flutter constraint conditions.</td>
</tr>
<tr>
<td>k</td>
<td>Selects the input matrix for splining the extra points to the aerodynamic model. Refers to a DMI bulk data entry.</td>
</tr>
<tr>
<td>l</td>
<td>Selects the direct input stiffness matrix. Refers to a DMI or DMIG bulk data entry.</td>
</tr>
<tr>
<td>m</td>
<td>Selects the direct input mass matrix. Refers to a DMI or DMIG bulk data entry.</td>
</tr>
<tr>
<td>n</td>
<td>Selects the direct input damping matrix. Refers to a DMI or DMIG bulk data entry.</td>
</tr>
<tr>
<td>o</td>
<td>Selects the transfer function set to be added to the input matrices. Refers to TF bulk data entries.</td>
</tr>
<tr>
<td>p</td>
<td>Set identification of VSDAMP and/or TABDMP bulk data entries that define damping data.</td>
</tr>
</tbody>
</table>

Remarks:

1. The FLCOND option is required, all others are optional.
2. M2PP, B2PP and K2PP and CONTROL names will typically refer to DMI and DMIG entries, but may refer to any existing database entity of the proper dimension.
3. The use of the CONTROL matrix requires that extra points be defined in the boundary condition.
Solution Control Command: FREQUENCY

Description: Invokes the frequency response analysis discipline

Hierarchy Level: Discipline

Format and Examples:

FREQUENCY type (DLOAD = i, FSTEP = j, GUST = k, K2PP = l, M2PP = m, B2PP = n, TFL = o, DAMPING = p)

FREQUENCY DIRECT (DLOAD = 10, FSTEP = 20)

FREQUENCY MODAL (DLOAD=100, FSTEP=30, K2PP=KFreq, M2PP=MFreq, TFL=5)

FREQUENCY DIRECT (DLOAD = 100, FSTEP = 20, GUST = 55)

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>Selects the solution approach from DIRECT or MODAL.</td>
</tr>
<tr>
<td>i</td>
<td>Set identification of a DLOAD bulk data entry.</td>
</tr>
<tr>
<td>j</td>
<td>Set identification of frequency bulk data entries (FREQ, FREQ1, or FREQ2) that define the frequency steps for the analysis.</td>
</tr>
<tr>
<td>k</td>
<td>Set identification of a GUST bulk data entry which defines the gust parameters.</td>
</tr>
<tr>
<td>l</td>
<td>Selects the direct input stiffness matrix. Refers to a DMI or DMIG bulk data entry.</td>
</tr>
<tr>
<td>m</td>
<td>Selects the direct input mass matrix. Refers to a DMI or DMIG bulk data entry.</td>
</tr>
<tr>
<td>n</td>
<td>Selects the direct input damping matrix. Refers to a DMI or DMIG bulk data entry.</td>
</tr>
<tr>
<td>o</td>
<td>Selects the transfer function set to be added to the input matrices. Refers to TF bulk data entries.</td>
</tr>
<tr>
<td>p</td>
<td>Set identification of VSDAMP and/or TABDMP bulk data entries that define damping data.</td>
</tr>
</tbody>
</table>

Remarks:

1. The FREQUENCY discipline does not generate design constraints for optimization.
2. type, DLOAD and FSTEP are required.
3. No more than one FREQUENCY analysis can be done in a single boundary condition.
4. M2PP, B2PP and K2PP names will typically refer to DMI and DMIG entries, but may refer to any existing database entity of the proper dimension.
Solution Control Command: LABEL

Description: Provides identifying information on subcase output.

Hierarchy Level: Label information

Format and Example:
LABEL = n
LABEL = SYMMETRIC MANEUVER LOAD

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>Any descriptive message that the user wishes to use to distinguish output.</td>
</tr>
</tbody>
</table>

Remarks:
1. LABEL information is used until it is superseded.
2. The LABEL command is optional.
3. Labels are limited to no more than 72 characters.
Solution Control Command: **MODES**

**Description:** Selects the Normal Modes discipline.

**Hierarchy Level:** Discipline

**Format and Examples:**

- `MODES (DCONS = n)`
- `MODES`
- `MODES (DCONS = 10)`

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>Set identification of DCONFREQ bulk data entries which define frequency constraints for the optimization task.</td>
</tr>
</tbody>
</table>

**Remarks:**

1. Only one modal analysis can be performed in a boundary condition using the **EIGR** bulk data entry selected on the **BOUNDARY** command.
Solution Control Command: NPSAERO

Description: Invokes the nonplanar static aerodynamics discipline

Hierarchy Level: Discipline

Format and Examples:

NPSAERO [ symtype ] ( TRIM = m )

NPSAERO ( TRIM = 60 )

NPSAERO ANTISYMMETRIC ( TRIM = 70 )

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>symtype</td>
<td>Selects the symmetry type for the subcase from SYMMETRIC or ANTISYMMETRIC. (Default is SYMMETRIC)</td>
</tr>
<tr>
<td>m</td>
<td>Set identification of a TRIM bulk data entry which provides flight condition information.</td>
</tr>
</tbody>
</table>

Remarks:

1. TRIM is required.
2. NPSAERO analyses may be freely combined with all other ASTROS disciplines.

WARNING

This feature is not working correctly
Solution Control Command: OPTIMIZE

Description: Invokes the ASTROS design capability

Hierarchy Level: Type of boundary condition

Format and Examples:

```
OPTIMIZE STRATEGY = ((m1,niter1),(m2,niter2),(m3,niter3)), MAXITER = n,
                    MOVILIM = o, WINDOW = p, OCMOVILIM = q, ALPHA = r,
                    CNVRGLIM = s, NRFAC = t, EPS = u

OPTIMIZE
OPTIMIZE MAXITER = 10, NRFAC = 0.6, EPS = -.05, MOVILIM = 1.3
OPTIMIZE STRATEGY = FSD, ALPHA = 0.8, MAXFSD = 10
OPTIMIZE STRATEGY = (FSD,3), ALPHA = 0.8, MAXFSD = 10
OPTIMIZE STRATEGY = ((FSD,3),(OC), ALPHA = 0.8, MAXFSD = 10
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1, m2, m3</td>
<td>The strategy to be used in optimization. One of MP for math programming methods, OC for generalized optimality criteria or FSD for fully stressed design. The order of input on the strategy command is the order that will be used. Each strategy MP, OC or FSD may only appear once. Default for m1=MP. (Only MP methods will be used)</td>
</tr>
<tr>
<td>niter1</td>
<td>The number of iterations for m1, m2 and m3 respectively. The default for each is to use the last named method for those iterations remaining up to MAXITER. If MAXITER is less than the sum of specified iterations, ASTROS will warn the user but stop at MAXITER iterations.</td>
</tr>
<tr>
<td>niter2, niter3</td>
<td></td>
</tr>
<tr>
<td>n</td>
<td>The maximum number of iterations to be performed using MP, FSD or OC. Default = 15.</td>
</tr>
<tr>
<td>o</td>
<td>The move limit applied to local design variables in MP. The local variable after each redesign will lie between t/MOVLIM and t*MOVLIM where t is the initial value. Default = 2.0, must be greater than 1.0.</td>
</tr>
</tbody>
</table>
The window around zero in which the MOVILIM bound is overridden to allow the local variable to change sign. If WINDOW=0.0, the local variable may not change sign. If WINDOW is nonzero, the half width of a band around zero, EPS is computed

\[ \text{EPS} = \frac{\text{WINDOW}}{100} \times \max(\text{ABS(TMIN)}, \text{ABS(TMAX)}) \]

If the local variable falls within the band, the new minimum or maximum for the current iteration is changed to lie on the other side of zero from the local variable. The bandwidth EPS is a percentage of the larger of TMAX or TMIN where WINDOW specifies the percentage. Default = 0.0, must be greater than or equal to 0.0.

The move limit applied to local design variables in OC. The local variable after each redesign will lie between \( \frac{t}{\text{MOVILIM}} \) and \( t \times \text{MOVILIM} \) where \( t \) is the initial value. Default = 10000.0, must be greater than 1.0. This default implies no move limits.

Exponential move limit for FSD. Numbers less than 1.0 result in a smaller move with smoother convergence. Ignored if STRAT=MP. Default = 0.90, must be greater than 0.0.

Convergence limit specifying the maximum percentage change in the objective function that can be considered converged. Default = 0.50, must be greater than 0.0.

Constraint retention factor for MP and OC methods. The number of active constraints will be at least NRFAC times the number of design variables. Default = 3.0.

Constraint retention parameter in which all constraints having a value greater than EPS will be considered active. Default = -0.10

Remarks:
1. None of the options are required
2. MAXITER and CNVRGLIM are global parameters that apply to the MP, FSD and OC strategies.
3. MOVILIM and WINDOW control the move limits for MP. WINDOW is only useful for LOCAL design variables that need to cross between positive and negative values.
4. NRFAC and EPS control the constraint deletion algorithm for MP and OC, both values are always applied.
Solution Control Command: PRINT

Description: Specifies the required output file processing for the print file

Hierarchy Level: Various

Format and Examples:

PRINT (Form, FREQ=a, ITER=b, MODE=c, TIME=d)
ACCE (Form, FREQ=a, ITER=b, TIME=d) = e,
AIRD (Form, ITER=b, MODE=c, TIME=d) = f,
CGRA (ITER=b) = h,
DCON (ITER=b) = i,
DISP (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = j,
ENER (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = k,
FORC (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = l,
GDES (ITER=b) = m,
GPFO (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = n,
GPWG (ITER=b) = n1,
KSNS (ITER=b) = o,
LDES (ITER=b) = p,
LOAD (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = q,
MASS (ITER=b) = r,
MSNS (ITER=b) = t,
OGRA (ITER=b) = u,
QHH (ITER=b, MODE=c) = x,
QIJ (ITER=b, MODE=c) = y,
ROOT (Form, ITER=b, MODE=c) = z,
SPCP (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = aa,
STIF (ITER=b) = ab,
STRA (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = ac,
STRE (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = ad,
TPRE (ITER=b) = ae,
VELO (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = af
TRIM

PRINT DISP = ALL
PRINT (RECT, MODE=10, ITER=20) DISP(ITER-LAST) = 6, ENERGY(POLA) = 10
PRINT (MODE=NONE)

<table>
<thead>
<tr>
<th>Options</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form</td>
<td>RECT or POLA requests output in RECTangular or POLAr format (See Remarks 1 and 2).</td>
</tr>
<tr>
<td>a</td>
<td>Set identification of a FREQLIST bulk data entry that is used to request the frequencies at which output is to be printed (See Remark 2).</td>
</tr>
<tr>
<td>b</td>
<td>Set identification of an ITERLIST bulk data entry that is used to request the optimization iterations at which output is to be printed (See Remark 2).</td>
</tr>
<tr>
<td>c</td>
<td>Set identification of a MODELIST bulk data entry that is used to request the modes at which output is to be printed (See Remark 2).</td>
</tr>
</tbody>
</table>
Set identification of a TIMELIST bulk data entry that is used to request the times at which output is to be printed (See Remark 2).

Set identification of a GRIDLIST bulk data entry that is used to request the grid points at which accelerations are to be printed.

Set identification of an ELEMLIST bulk data entry that is used to request the aerodynamic box elements at which displacements for the aerodynamic model are to be printed.

Set identification of an DCONLIST bulk data entry that is used to request the subset of active constraints for which gradients are to be printed (See Remark 2).

Set identification of an DCONLIST bulk data entry that is used to request the subset of active constraints which are to be printed (See Remark 2).

Set identification of a GRIDLIST bulk data entry that is used to request the grid points at which displacements are to be printed.

Set identification of an ELEMELIST bulk data entry that is used to request the elements for which strains are to be printed.

Set identification of an DCONLIST bulk data entry that is used to request the subset of active constraints for which strains are to be printed.

Set identification of an DCONLIST bulk data entry that is used to request the subset of active constraints which are to be printed.

Set identification of a GRIDLIST bulk data entry that is used to request the grid points at which strains are to be printed.

Set identification of an ELEMELIST bulk data entry that is used to request the elements for which strain energies are to be printed.

Set identification of an ELEMELIST bulk data entry that is used to request the elements for which forces are to be printed.

Set identification of a GRIDLIST bulk data entry that is used to request the grid points at which grid point forces are to be printed.

Either ALL or NONE depending on whether the GPWG is to be computed/printed. If a GPWG entry is in the Bulk Data file, it will be used by the algorithm.

Set identification of an LDVLIST and/or a GDVLIST bulk data entry that is used to request the design variables for which stiffness sensitivities are to be printed.

Set identification of an LDVLIST bulk data entry that is used to request the local design variable IDs for which local design variables are to be printed.

Set identification of a GRIDLIST bulk data entry that is used to request the grid points at which applied loads are to be printed.

Set identification of a GRIDLIST bulk data entry that is used to request the grid points degrees of freedom for which the mass matrix is to be printed.

Set identification of an LDVLIST and/or a GDVLIST bulk data entry that is used to request the design variables for which mass sensitivities are to be printed.

Set identification of a GDVLIST bulk data entry that is used to request the design variables for which objective function gradients are to be printed.

Set identification of an ELEMELIST bulk data entry that is used to request the aerodynamic elements for which QHH is to be printed.

Set identification of an ELEMELIST bulk data entry that is used to request the aerodynamic elements for which QHJ is to be printed.

Set identification of an MODELIST bulk data entry that is used to request the modes for which flutter and normal modes eigenvalue results are to be printed.

Set identification of a GRIDLIST bulk data entry that is used to request the grid points at which SPC forces are to be printed.

Set identification of a GRIDLIST bulk data entry that is used to request the grid points degrees of freedom for which the stiffness matrix is to be printed.
Set identification of an ELEMLIST bulk data entry that is used to request the elements at which strains are to be printed.

Set identification of an ELEMLIST bulk data entry that is used to request the elements for which stresses are to be printed.

Set identification of an ELEMLIST bulk data entry that is used to request the aerodynamic elements for which the pressure coefficients at aeroelastic trim are to be printed.

Set identification of a GRIDLIST bulk data entry that is used to request the grid points at which velocities are to be printed.

Remarks:

1. Form is an optional parameter for printing complex data. RECTangular data outputs complex data with real and imaginary components while POLAr outputs complex data using magnitude and phase.

2. If used with the PRINT command, all data that are not otherwise specified use the requested Form, FREQ, ITER, MODE, and TIME, if applicable for that type of data. If used with an option, Form, FREQ, ITER, MODE, and TIME override the global request. Options a through f can be either ALL, NONE, or a positive integer, and additionally, option b (ITER) can be LAST, and options h (CGRA) and i (DCON) can be ACTIVE. ALL requests all values. NONE turns off a request from a previous hierarchy while an integer value refers to a bulk data entry. LAST requests that output be printed for only the final value in a list. For example, ITER=LAST selects output for the final iteration in an optimization. ACTIVE selects the active constraints.

3. HIST and TRIM are toggles. If they are present, the specified data are printed. TRIM indicates that stability derivative data associated with an aeroelastic trim are to be printed. HIST indicates that the design iteration history summary is to be printed.

4. Aerodynamic macro elements are selected indirectly. A macro element is chosen by selecting one or more aerodynamic box elements contained within the macro element.
Solution Control Command: PUNCH

Description: Specifies the required output file processing for the punch file

Hierarchy Level: Various

Format and Examples:

PUNCH
(Form, FREQ=a, ITER=b, MODE=c, TIME=d)
ACCE (Form, FREQ=a, ITER=b, TIME=d) = e,
AIRD (Form, ITER=b, MODE=c, TIME=d) = f,
CGRA (ITER=b) = h,
DCON (ITER=b) = i,
DISP (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = j,
ENR (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = k,
FORC (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = l,
GDES (ITER=b) = m,
GPFO (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = n,
GPWG (ITER=b) = n1,
KSNS (ITER=b) = o,
LDES (ITER=b) = p,
LOAD (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = q,
MASS (ITER=b) = r,
MODEL (ITER=b) = ah
MSNS (ITER=b) = t,
OGRA (ITER=b) = u,
QH (ITER=b, MODE=c) = x,
QJ (ITER=b, MODE=c) = y,
ROOT (Form, ITER=b, MODE=c) = z,
SPCP (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = aa,
STIP (ITER=b) = ab,
STRA (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = ac,
STRE (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = ad,
TPRE (ITER=b) = ae,
VELO (Form, FREQ=a, ITER=b, MODE=c, TIME=d) = af
TRIM

PUNCH DISP = ALL

PUNCH (RECT, MODE=10, ITER=20) DISP(ITER=LAST) = 6, ENERGY(POLA) = 10
PUNCH (MODE=NONE)

Options

<table>
<thead>
<tr>
<th>Form</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rect or POLA requests output in RECTangular or POLAr format (See Remarks 1 and 2).</td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>Set identification of a FREQLIST bulk data entry that is used to request the frequencies at which output is to be punched (See Remark 2).</td>
</tr>
<tr>
<td>b</td>
<td>Set identification of an ITERLIST bulk data entry that is used to request the optimization iterations at which output is to be punched (See Remark 2).</td>
</tr>
<tr>
<td>c</td>
<td>Set identification of a MODELIST bulk data entry that is used to request the modes at which output is to be punched (See Remark 2).</td>
</tr>
</tbody>
</table>
Set identification of a TIMELIST bulk data entry that is used to request the times at which output is to be punched (See Remark 2).

Set identification of a GRIDLIST bulk data entry that is used to request the grid points at which accelerations are to be punched.

Set identification of an ELEMLIST bulk data entry that is used to request the aerodynamic box elements at which displacements for the aerodynamic model are to be punched.

Set identification of a DCONLIST bulk data entry that is used to request the the subset of active constraints for which gradients are to be punched (See Remark 2).

Set identification of an DCONLIST bulk data entry that is used to request the the subset of active constraints which are to be punched (See Remark 2).

Set identification of a GRIDLIST bulk data entry that is used to request the grid points at which displacements are to be punched.

Set identification of an ELEMLIST bulk data entry that is used to request the elements for which strain energies are to be punched.

Set identification of an ELEMLIST bulk data entry that is used to request the elements for which forces are to be punched.

Set identification of a GDVLIST bulk data entry that is used to request the global design variable IDs for which global design variables are to be punched.

Set identification of a GRIDLIST bulk data entry that is used to request the grid points at which grid point forces are to be punched.

Either ALL or NONE depending on whether the GPWG is to be computed/punched. If a GPWG entry is in the Bulk Data file, it will be used by the algorithm.

Set identification of an LDVLIST and/or a GDVLIST bulk data entry that is used to request the design variables for which stiffness sensitivities are to be punched.

Set identification of an LDVLIST bulk data entry that is used to request the local design variable IDs for which local design variables are to be punched.

Set identification of a GRIDLIST bulk data entry that is used to request the grid points at which applied loads are to be punched.

Set identification of a GRIDLIST bulk data entry that is used to request the grid points degrees of freedom for which the mass matrix is to be punched.

Set identification of an LDVLIST and/or a GDVLIST bulk data entry that is used to request the design variables for which mass sensitivities are to be punched.

Set identification of a GDVLIST bulk data entry that is used to request the design variables for which objective function gradients are to be punched.

Set identification of an ELEMLIST bulk data entry that is used to request the aerodynamic elements for which QHH is to be punched.

Set identification of an ELEMLIST bulk data entry that is used to request the aerodynamic elements for which QHJ is to be punched.

Set identification of a MODELIST bulk data entry that is used to request the modes for which flutter and normal modes eigenvalue results are to be punched.

Set identification of a GRIDLIST bulk data entry that is used to request the grid points at which SPC forces are to be punched.

Set identification of a GRIDLIST bulk data entry that is used to request the grid points degrees of freedom for which the stiffness matrix is to be punched.
ac  Set identification of an ELEMLIST bulk data entry that is used to request the elements at
which strains are to be punched.
ad  Set identification of an ELEMLIST bulk data entry that is used to request the elements for
which stresses are to be punched.
ae  Set identification of an ELEMLIST bulk data entry that is used to request the aerodynamic
elements for which the pressure coefficients at aeroelastic trim are to be punched.
af  Set identification of a GRIDLIST bulk data entry that is used to request the grid points at
which velocities are to be punched.
ag  Specifies the iterations at which the design model will be punched. May be ALL,
NONE, LAST, or the set identification of an ITERLIST bulk data entry which specifies
the iterations at which to punch the model.
ah  Specifies the portion of the model which will be punched. May be ALL or NONE.
(Note: an integer value is accepted and treated as ALL)

Remarks:
1. Form is an optional parameter for printing complex data. REctangular data outputs complex
data with real and imaginary components while POLar outputs complex data using magni-
tude and phase.
2. If used with the PRINT command, all data that are not otherwise specified use the requested
Form, FREQ, ITER, MODE, and TIME, if applicable for that type of data. If used with an option,
Form, FREQ, ITER, MODE, and TIME override the global request. Options a through af can be
either ALL, NONE, or a positive integer, and additionally, option b (ITER) can be LAST, and
options h (CGRA) and i (DCON) can be ACTIVE. ALL requests all values. NONE turns off a
request from a previous hierarchy while an integer value refers to a bulk data entry. LAST
requests that output be printed for only the final value in a list. For example, ITER=LAST
selects output for the final iteration in an optimization. ACTIVE selects the active constraints.
3. HIST and TRIM are toggles. If they are present, the specified data are punched. TRIM indicates
that stability derivative data associated with an aeroelastic trim are to be punched. HIST
indicates that the design iteration history summary is to be punched.
4. Aerodynamic macro elements are selected indirectly. A macro element is chosen by selecting
one or more aerodynamic box elements contained within the macro element.
Solution Control Command: **SAERO**

**Description:** Invokes the static aerodynamics discipline

**Hierarchy Level:** Discipline

**Format and Examples:**

```
SAERO [ symtype ] ( TRIM = k , DCON = 0, STRESS = m, STRAIN = n )
SAERO ( TRIM = 60 )
SAERO ANTISYMMETRIC ( TRIM = 70 , STRESS = 100 )
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>symtype</td>
<td>Selects the symmetry type for the subcase from SYMMETRIC or ANTISYMMETRIC. (Default is SYMMETRIC)</td>
</tr>
<tr>
<td>k</td>
<td>Set identification of a TRIM bulk data entry which provides flight condition information.</td>
</tr>
<tr>
<td>m</td>
<td>Set identification for stress constraints as defined by DCONVM, DCONVM, DCONVMP, DCONTW, DCONTWM, or DCONTWF bulk data entries.</td>
</tr>
<tr>
<td>n</td>
<td>Set identification for strain constraints as defined by D ConeP, D ConePM, D ConePP, D ConFT, D ConFTM, or D ConFTP bulk data entries.</td>
</tr>
<tr>
<td>o</td>
<td>Set identification for displacement constraints as defined by D ConDSP, D ConTRM, D ConCLA, D ConALE, or D ConSCF bulk data entries.</td>
</tr>
</tbody>
</table>

**Remarks:**

1. **TRIM** is required. Both symtype and the CONSTRAINT section are optional.
2. **SAERO** disciplines may be freely combined with other ASTROS disciplines.
3. For compatibility, the alternate form of constraint specification shown below is also allowed. Its use is, however, discouraged.

```
SAERO [ symtype ] ( TRIM = k ), CONSTRAINT(STRESS=m, STRAIN=n, GENERAL=o)
```
Solution Control Command:  SOLUTION

Description:  The first command in the solution control packet.

Hierarchy Level:  Beginning of solution

Format:
SOLUTION

Remarks:
1. One SOLUTION command must always appear as the first command of the solution control packet.
Solution Control Command: **STATICS**

Description: Invokes the statics analysis discipline

Hierarchy level: Discipline

Format and Examples:

STATICS (MECH = i, THERMAL = j, GRAVITY = k, DCON = o, STRESS = m, STRAIN = n)
STATICS (MECH = 10)
STATICS (MECH = 4, THERMAL = 6, DCON = 10)

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>Set identification for external loads as defined by LOAD, PLOAD, FORCE, FORCE1, MOMENT, and MOMENT1 bulk data entries.</td>
</tr>
<tr>
<td>j</td>
<td>Set identification for temperatures defined by TEMP or TEMPD bulk data entries.</td>
</tr>
<tr>
<td>k</td>
<td>Set identification of GRAV bulk data entries which define gravity forces.</td>
</tr>
<tr>
<td>m</td>
<td>Set identification for stress constraints defined by DCONVM, DCONVM1, DCONVMP, DCONTW, DCONTW1, or DCONTW2 bulk data entries.</td>
</tr>
<tr>
<td>n</td>
<td>Set identification for strain constraints defined by DCONEP, DCONEP1, DCONEP2, DCONFT, DCONFT1, or DCONFT2 bulk data entries.</td>
</tr>
<tr>
<td>o</td>
<td>Set identification of DCONDSP bulk data entries which define displacement constraints.</td>
</tr>
</tbody>
</table>

Remarks:

1. The sum of all the loads forms a single right hand side for a statics analysis.
2. At least one of the load types must be present. The CONSTRAINT section is optional.
3. Gravity forces may be included indirectly if referenced by the LOAD bulk data entry.
4. For compatibility, the alternate form of constraint specification shown below is also allowed. Its use is, however, discouraged.

```
STATICS (MECH = i, THERMAL = j, GRAVITY = k),
CONSTRANT (STRESS=m, STRAIN=n, GENERAL=o)
```
Solution Control Command: SUBTITLE

Description: Defines a subtitle which will appear in the output.

Hierarchy Level: Label information

Format and Example:

SUBTITLE = n
SUBTITLE = SUPersonic Design Condition

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>Any descriptive information can be inserted here</td>
</tr>
</tbody>
</table>

Remarks:

1. SUBTITLE information is used until it is superseded.
2. The SUBTITLE command is optional.
3. Subtitles are limited to 72 characters.
Solution Control Command: TITLE

Description: Defines a title which will appear in the output.

Hierarchy Level: Label information

Format and Examples:

TITLE = n
TITLE = DESIGN OF A FORWARD SWEPT WING MODEL

Option | Meaning
--- | ---
n | Any descriptive information can be included here

Remarks:

1. TITLE information is used until it is superseded.
2. The TITLE command is optional.
3. Titles are limited to no more than 72 characters.
**Solution Control Command:**  TRANSIENT

**Description:** Invokes the transient analysis discipline

**Hierarchy Level:** Discipline

**Format and Examples:**

TRANSIENT type (DLOAD = i, TSTEP = j, FFT = k, IC = l, GUST = m,
K2PP = n, M2PP = o, B2PP = p, TFL = q, DAMPING = r)

TRANSIENT MODAL ( DLOAD = 10, TSTEP = 20 )

TRANSIENT DIRECT (DLOAD=100, TSTEP=30, K2PP=KFREQ, M2PP=MFREQ, IC=45, TFL=5)

TRANSIENT MODAL ( DLOAD = 100, TSTEP = 20 , FFT = 999,GUST = 55)

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>Selects the solution approach from DIRECT or MODAL.</td>
</tr>
<tr>
<td>i</td>
<td>Set identification of a DLOAD bulk data entry.</td>
</tr>
<tr>
<td>j</td>
<td>Set identification of TSTEP bulk data entries which provide the time step information for the analysis.</td>
</tr>
<tr>
<td>k</td>
<td>Set identification of an FFT bulk data entry which provides parameters to use the Fast Fourier Transform methods in performing the transient analysis.</td>
</tr>
<tr>
<td>l</td>
<td>Set identification of IC bulk data entries which define the initial conditions.</td>
</tr>
<tr>
<td>m</td>
<td>Set identification of a GUST bulk data entry which defines the gust parameters.</td>
</tr>
<tr>
<td>n</td>
<td>Selects the direct input stiffness matrix. Refers to a DMI or DMIG bulk data entry.</td>
</tr>
<tr>
<td>o</td>
<td>Selects the direct input mass matrix. Refers to a DMI or DMIG bulk data entry.</td>
</tr>
<tr>
<td>p</td>
<td>Selects the direct input damping matrix. Refers to a DMI or DMIG bulk data entry.</td>
</tr>
<tr>
<td>q</td>
<td>Selects the transfer function set to be added to the input matrices. Refers to TF bulk data entries.</td>
</tr>
<tr>
<td>r</td>
<td>Set identification of VSDAMP and/or TABDMP bulk data entries that define damping data.</td>
</tr>
</tbody>
</table>

**Remarks:**

1. The TRANSIENT discipline does not generate design constraints for optimization.
2. *type*, DLOAD and TSTEP are required.
3. If GUST is present, FFT must also be used.
4. Initial conditions, IC, are only valid for DIRECT analyses. IC cannot be used with GUST or FFT.
5. No more than one TRANSIENT analysis can be done in a single boundary condition.
6. M2PP, B2PP and K2PP names will typically refer to DMI and DMIG entries, but may refer to any existing database entity of the proper dimension.
4. THE BULK DATA PACKET

The bulk data packet provides the ASTROS system with the engineering data needed to perform the specific tasks requested by the user. It contains the model geometries for the structural model, the aerodynamic model(s) and the design model as well as the pool of data from which the solution control requests are made. Finally, specialized information required by the analysis disciplines (e.g., Mach number and reduced frequency pairs for unsteady aerodynamic analyses) is also provided to the system through the bulk data packet. The basic input item is the bulk data entry which is directly analogous to the NAStRAN bulk data card. In fact, NAStRAN compatible formats were chosen for the ASTROS bulk data entries whenever possible because modern structures are often analyzed using large NAStRAN finite element models having tens of thousands of lines of bulk data. Further, these large models are usually prepared using software designed specifically to generate NAStRAN models. Thus, by utilizing NAStRAN bulk data structures where possible and by using NAStRAN's bulk data style for the additional engineering data, ASTROS is highly compatible with existing NAStRAN models and with current finite element model generation methods.

Just as in NAStRAN, the bulk data packet begins with the keyword BEGIN BULK (which may be abbreviated BEGIN) and is terminated by the optional keyword ENDDATA or by the end of the input stream. The intervening bulk data entries can appear in any order. An alphabetically sorted listing of the bulk data input will be echoed to the output file unless suppressed by the user through the BEGIN BULK command line options.

All the input entries are interpreted by IPP through templates that are defined as part of the system generation task. The templates provide for basic error checking, establish defaults and direct the placement of the raw data onto the database. The use of templates allows additional entries to be added to the system very simply without software changes. The definition of the templates and the means of adding new entries are documented in the Programmer's Manual. In addition, the complete listing of ASTROS bulk data templates is included in the output summary generated by the SYSGEN system generation utility during the creation of the system database files.

On restart with a bulk data packet in the input stream, the IPP module will append the new data onto the data from the previous run(s). There is no provision for deleting existing bulk data except through MAPOL sequence modifications or direct interaction using the ICE program (Reference 7). This restart feature, while limited, can be useful in many instances; e.g. when additional analysis disciplines are desired or when different output requests are desired. The remainder of this section presents the structure of the bulk data entry for ASTROS and discusses some features of the IPP module that are useful to the general user. ASTROS bulk data entries have been carefully designed to be NAStRAN compatible, so the NAStRAN User's Manual (Reference 2) has provided much of the information in the following discussion as well as having directed the design of the IPP software. The reader is also referred to the ASTROS Programmer's Manual for more information on the IPP module and for information on
the addition of new bulk data entries. An additional resource is the MSC/NASTRAN Primer of Reference 3.

4.1. BULK DATA ECHO OPTIONS

There are special options on the BEGIN BULK command which allow the user to control the echoing of the Bulk Data. The format of this command is:

```
BEGIN BULK [ ECHO NOECHO ] [ PRINT PUNCH BOTH ] [ SORT NOSORT ]
```

The following table describes the actions which are performed for the various options.

<table>
<thead>
<tr>
<th>ECHO</th>
<th>FILE</th>
<th>ORDER</th>
<th>ACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECHO</td>
<td>PRINT</td>
<td>SORT</td>
<td>Sorted echo to the output file.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UNSORT</td>
<td>Unsorted echo to the output file.</td>
</tr>
<tr>
<td>PUNCH</td>
<td>SORT</td>
<td></td>
<td>Sorted echo to punch file.</td>
</tr>
<tr>
<td></td>
<td>UNSORT</td>
<td></td>
<td>Unsorted echo to punch file.</td>
</tr>
<tr>
<td>BOTH</td>
<td>SORT</td>
<td></td>
<td>Sorted echo to both output and punch files.</td>
</tr>
<tr>
<td></td>
<td>UNSORT</td>
<td></td>
<td>Unsorted echo to both output and punch files.</td>
</tr>
<tr>
<td>NOECHO</td>
<td></td>
<td></td>
<td>No echo to either output or punch file.</td>
</tr>
</tbody>
</table>

4.2. FORMAT OF THE BULK DATA ENTRY

Each bulk data entry consists of a required parent line followed by a number of optional continuation lines. Therefore, a single bulk data entry resides on one or more lines. The basic bulk data line has one mnemonic field of eight characters followed by either eight data fields of eight characters or by four data fields of 16 characters and terminates with an eight character continuation field as shown in Figure 4. The data field size (either eight or 16 characters) is determined by the presence of the optional large field marker in the first mnemonic field of each bulk data line. The parent line begins with a character mnemonic identifying the entry followed by 4 or 8 data fields and ending with a continuation field. The continuation lines are identical except that the leading mnemonic field contains a continuation label which is used to link it to its parent line. This structure is identical to that in NASTRAN. One important exception to NASTRAN compatibility is that ASTROS requires that the continuation lines follow continuously from the parent line although the bulk data entries themselves can be in any order. Random placement of continuations in NASTRAN is an artifact from using physical cards that were punched with the bulk data. If the card deck were dropped, the resulting random order still had to be interpretable by the code. This feature no longer needs to be supported in light of modern computer storage methods but NASTRAN compatibility dictated that similar continuation labeling be used.
A continuation line is defined for a bulk data entry that requires more than eight (or four large) data fields. The last field of the parent line is used in conjunction with the first field of the continuation line as an identifier. The parent continuation field can contain any alphanumeric entry while the first field of the continuation line contains a plus (+) as a continuation character in column 1 followed by the last 7 characters from the parent continuation label. For the parent line, the large field marker is an asterisk (*) following the name of the entry which signifies that large data fields are to be used. For continuation lines, the asterisk used as the continuation character plays the role of the large field marker as shown below. Each bulk data line must be either all narrow field or all large field, although separate lines of a single bulk data entry can have different field widths simply by using the proper field marker. This means that the same bulk data entry in wide and narrow formats are functionally identical with no need for separate templates. Unlike NASTRAN, the continuation mnemonics need not be unique among all the bulk data entries in the bulk data packet since there is no provision for randomly sorted continuations.

The input on a bulk data line can either be in fixed format, in which each item must reside within the field to which it belongs, or in free format, in which fields are separated by commas and can be positioned anywhere to the left of the column in which the fixed field would normally start. Free format input is indicated by the appearance of a comma in the first 10 characters of the input line. ASTROS requires that each line (not each bulk data entry) be either all fixed or all free format and that each free format field be separated by a comma. The NASTRAN use of a blank character as a field separator is not.

<table>
<thead>
<tr>
<th>Small Field Entry with a Small Field Continuation</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Small Field Entry with a Large Field Continuation</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Large Field Entry with both Large and Small Field Continuations</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME*</td>
</tr>
<tr>
<td>+EF</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Large Field Entry with a Large Field Continuation</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
</tr>
</tbody>
</table>

Figure 4. Bulk Data Entry Formats
supported. When free format input is used, the continuation lines can reside on the same physical line of input with the continuation labels either included or not as in the following equivalent examples:

<table>
<thead>
<tr>
<th>MKAERO1, 1, 0.3, 0.5,, ABC, +BC, 0.01, 0.05, 0.1, 0.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>MKAERO1, 1, 0.3, 0.5,, 0.01, 0.05, 0.1, 0.2</td>
</tr>
</tbody>
</table>

In the latter case, ASTROS will automatically generate the missing continuation mnemonics. Care must be taken, however, that the first two data fields of the continuation line be non-blank. If not, there is an ambiguity as to whether the first continuation field constitutes a continuation label or a data field. This ambiguity causes the IFP to terminate execution with an error indicating that there is a missing continuation line. Free format input in which the parent and continuation lines are broken into separate physical lines or which explicitly include the continuation mnemonics do not suffer this limitation. Free format input is further restricted in that the break between physical lines, if needed, must occur at a break in the logical line, that is, the split must occur between the ending continuation field on the current logical line and the continuation field of the next logical line. This means, for the preceding example, that the first example entry could be broken into two lines between the ABC and +BC fields but nowhere else. When an entry is broken into multiple physical lines, the continuation mnemonics must be supplied. Obviously, fixed format input requires continuation mnemonics for any bulk data entries having continuation lines.

4.3. DATA FIELD FORMATS

The interior fields of a bulk data line can contain either integer data, real data, character data or certain combinations (e.g. either integer or real data). The template for each entry defines which types of data are acceptable in each field. Each data item is limited to the number of characters that fit in the length of the field. For narrow width fields no more than eight characters can be used in the data item. Unlike NASTRAN, any extra characters will spill to the next field and will result in IFP errors, there is no provision for rounding real data to fit the field size.

In order to be considered valid, the data item must first satisfy the data type requirement as specified on the template. Real numbers, including zero, must contain a decimal point, although there are a number of formats supported. For example, the real number 3.1 may be encoded as shown or as 3.1E0, +3.1D00, 0.31E1, or 3.1+. Unlike NASTRAN, however, there cannot be embedded blanks anywhere in the real number and a D edit descriptor is treated as a single precision number until actually loaded to a double precision relational attribute. Blank fields that do not have other defaults specified on the template, will be interpreted as blank characters, an integer zero or a real zero as required. Integer values must be formed from the ten decimal digits with an optional leading plus or minus sign. Character data consist of any combination of alphanumeric characters including any digits, decimal points, etc., with no restriction that the first character be alphabetic.

4.4. ERROR CHECKING IN THE INPUT FILE PROCESSOR

As mentioned in the preceding subsection, the IFP module performs basic error checking to ensure that the input data is of the correct type. In addition, the templates provide for error checks that enable the IFP to check that the data satisfy particular requirements. For example, the IFP can be
directed to require that a particular value be greater than zero or be one of a finite number of selections. At its most complex, the bulk data processor checks to ensure specific relationships among data on a single bulk data entry. It is important to understand, however, that no error checks occur in the IFP to ensure that references to, and interrelationships among, multiple bulk data entries are satisfied. These more complex checks occur in subsequent engineering modules. A complete description of the available template error checks and the mechanism provided to add additional error checks is presented in the Programmer's Manual. The reader may find it helpful to study this documentation since the bulk data packet and the bulk data entries are closely linked to the software in both the SYSGEN utility and the IFP module.

4.5. BULK DATA ENTRY SUMMARY

This section contains a summary of all the bulk data entries in the ASTROS system separated into logically related groups. The groups are composed of either model definition entries, subcase definition entries or general list entries. This is followed by a detailed description of each of the entries listed in this section. Section 4.6 discusses the differences between NASTRAN and ASTROS for those entries that have been changed or are completely different than in NASTRAN but that use the same mnemonic and serve a similar purpose. Entries indicated by * are unchanged from NASTRAN.

4.5.1. Aerodynamic Load Transfer

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTACH</td>
<td>Rigid load transfer definition.</td>
</tr>
<tr>
<td>SET1</td>
<td>A structural grid point list for spline interpolation or a mode list for omitting normal modes in flutter analysis.</td>
</tr>
<tr>
<td>SET2</td>
<td>Structural grid point list in term of aerodynamic macroelements.</td>
</tr>
<tr>
<td>SPLINE1</td>
<td>Surface spline definition for out-of-plane motion.</td>
</tr>
<tr>
<td>SPLINE2</td>
<td>Beam spline definition for interpolating panels and bodies.</td>
</tr>
</tbody>
</table>

4.5.2. Applied Dynamic Loads

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLAGS</td>
<td>Time and phase lag definition for a spatial load.</td>
</tr>
<tr>
<td>DLOAD</td>
<td>Linear combination of dynamic load sets.</td>
</tr>
<tr>
<td>DLONLY</td>
<td>Direct definition of dynamic spatial load.</td>
</tr>
<tr>
<td>GUST</td>
<td>Stationary vertical gust definition.</td>
</tr>
<tr>
<td>RLOAD1</td>
<td>Frequency dependent dynamic load definition.</td>
</tr>
<tr>
<td>RLOAD2</td>
<td>Frequency dependent dynamic load definition.</td>
</tr>
<tr>
<td>TABLED1</td>
<td>Tabular function definition for dynamic load generation.</td>
</tr>
<tr>
<td>TLOAD1</td>
<td>Time dependent dynamic load definition.</td>
</tr>
<tr>
<td>TLOAD2</td>
<td>Time dependent dynamic load definition.</td>
</tr>
</tbody>
</table>

4.5.3. Applied Static Loads

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORCE</td>
<td>Definition of a concentrated load at a grid point.</td>
</tr>
<tr>
<td>FORCE1</td>
<td>Definition of a concentrated load at a grid point.</td>
</tr>
<tr>
<td>GRAV</td>
<td>Definition of an acceleration vector for gravity loads.</td>
</tr>
</tbody>
</table>

147
LOAD*  Definition of linear load combinations.
MOMENT* Definition of a moment at a grid point.
MOMENT1* Definition of a moment at a grid point.
PLOAD* Definition of a pressure load over an area.
TEMP* Definition of a temperature at a structural node.
TEMPD* Definition of default nodal temperatures.

4.5.4. Boundary Condition Constraints

ASET* Analysis set definition.
ASET1* Analysis set definition.
DYNRED Dynamic reduction parameters.
JSET Inertia relief mode shape parameter definition.
JSET1 Inertia relief mode shape parameter definition.
MPC* Multipoint constraint definition.
MPCADD* Definition of combinations of MPC sets.
OMIT Omit set definition.
OMIT1 Omit set definition.
RBAR Rigid bar element
RBE1 Rigid body element
RBE2 Rigid body element
RBE3 Rigid body element
RROD Rigid rod element
SPC* Single point constraint/enforced displacement definition.
SPC1* Single point constraint definition.
SPCADD* Definition of combinations of SPC sets.
SUPORT Definition of coordinates for determinate reactions.

4.5.5. Design Constraints

DCONALE Aileron effectiveness constraint definition.
DCONCLA Lift effectiveness constraint definition.
DCONDSP Displacement constraint definition.
DCONEP Principal strain constraint definition.
DCONEPM Principal strain constraint definition.
DCONEPP Principal strain constraint definition.
DCONFLT Flutter constraint definition.
DCONFRQ Modal frequency constraint definition.
DCONFT Fiber/transverse strain constraint definition.
DCONFTM Fiber/transverse strain constraint definition.
DCONFTP Fiber/transverse strain constraint definition.
4.5.6. Design Variables, Linking and Optimization Parameters

- **DESELM** Unique physical design variable definition.
- **DESVARP** Linked physical design variable definition.
- **DESVARS** Linked shape function design variable definition.
- **ELIST** Element list for physical linking.
- **MPPARM** Mathematical programming default parameter override.
- **OCPARM** Optimality criteria default parameter override.
- **PLIST** Physical design variable linking definition.
- **SHAPE** Definition of element linking factors to define a shape variable.

4.5.7. Geometry

- **CORD1C** Cylindrical coordinate system definition.
- **CORD1R** Rectangular coordinate system definition.
- **CORD1S** Spherical coordinate system definition.
- **CORD2C** Cylindrical coordinate system definition.
- **CORD2R** Rectangular coordinate system definition.
- **CORD2S** Spherical coordinate system definition.
- **EPOINT** Extra point definition for dynamics.
- **GRDSSET** Default parameters for fields on the GRID entry.
- **GRID** Grid point location and coordinate system selection.
- **SPOINT** Scalar point definition.

4.5.8. Material Properties

- **MAT1** Isotropic elastic properties definition.
- **MAT2** Two-dimensional anisotropic properties definition.
- **MAT8** Orthotropic properties definition.
- **MAT9** Anisotropic properties definition for isoparametric hexahedral elements.
4.5.9. **Miscellaneous Inputs**

- ` Comment: Commentary data.
- ` CONVERT: Conversion factor definitions.
- ` DMI: Direct matrix input.
- ` DMIG: Direct matrix input at structural nodes.
- ` MFORM: Mass matrix form (LUMPED or COUPLED).
- ` SAVE: List of database entities not to be purged.
- ` SEQGP: Structural set resequencing definition.

4.5.10. **Output Selection Lists**

- ` DCONLIST: A list of design constraints for which constraint value and/or gradient output are desired.
- ` ELEMLIST: List of elements for element dependent output.
- ` FREQLIST: List of frequency steps for which output is desired.
- ` GDVLIST: List of global design variables for which output is desired.
- ` GPWGT: Definition of the location to perform grid point weight generation.
- ` GRIDLIST: List of nodes for nodal dependent output.
- ` ITERLIST: List of iteration steps for which output is desired.
- ` LDVLIST: List of local design variables for which output is desired.
- ` MODELIST: List of normal modes for which output is desired.
- ` TIMELIST: List of time steps for which output is desired.

4.5.11. **Steady Aerodynamics**

- ` AEROS: Reference parameters.
- ` AEFACT: List of real parameters.
- ` AESURF: Aerodynamic control surface definition.
- ` AIRFOIL: Airfoil property definition.
- ` AXSTA: Body axial station parameter definition.
- ` BODY: Body configuration definition.
- ` CAERO6: Macroelement (panel) definition.
- ` CONEFTS: Definition of static aerodynamic control effectiveness.
- ` CONLINK: Definition of linked control surfaces.
- ` PAERO6: Body parameter definition.

4.5.12. **Structural Element Connection**

- ` BAROR: Definition of default parameters for the CBAR bar element.
- ` CBAR: Prismatic beam element.
- ` CELAS1: Scalar elastic spring element.
- ` CELAS2: Scalar elastic spring element.
CI HEX 1
Linear isoparametric hexahedral element.
CI HEX 2*
Quadratic isoparametric hexahedral element.
CI HEX 3*
Cubic isoparametric hexahedral element.
CMASS 1 Scalar mass element.
CMASS 2 Scalar mass element.
CONM 1* Direct 6 x 6 mass matrix definition at a structural node.
CONM 2 Concentrated mass at a structural node.
CONROD Rod element.
CQDMEM 1 Isoparametric quadrilateral membrane element.
CQUAD 4 Isoparametric quadrilateral element with bending and membrane stiffness.
CROD Rod element.
CSHEAR Shear panel.
CTRIA 3 Isoparametric triangular element with bending and membrane stiffness.
CTRMEM Constant strain triangular membrane element.

4.5.13. Structural Element Properties

PBAR Prismatic beam element.
PCOMP Composite laminate definition for CQDMEM 1, CQUAD 4, CTRIA 3, and CTRMEM elements.
PCOMP 1 Composite laminate definition for CQUAD 4 and CTRIA 3 elements.
PCOMP 2 Composite laminate definition for CQUAD 4 and CTRIA 3 elements.
PELAS Scalar elastic spring element.
PHEX* Linear, quadratic and cubic isoparametric hexahedral element.
PMASS Scalar mass element.
PQDMEM 1 Isoparametric quadrilateral membrane element.
PROD Rod element.
PSHEAR Shear panel.
PSHELL Definition of shell element properties for CQUAD 4 and CTRIA 3 elements.
PTRMEM Constant strain triangular membrane element.

4.5.14. Unsteady Aerodynamics

AERO Reference parameters.
CAERO 1* Aerodynamic macroelement (panel) definition.
CAERO 2* Body configuration definition.
CONEFF Definition of flutter aerodynamic control effectiveness.
FLFACT* Parameter definition for flutter analysis.
MKAERO 1 Table of symmetries, Mach numbers, and reduced frequencies.
MKAERO 2 Table of symmetries, Mach numbers, and reduced frequencies.
4.5.15. **Discipline Dependent Problem Control**

The following bulk data entries are the controlling entries referenced by Solution Control in selecting specific disciplines and subcases. In each case, many of these inputs can appear in the bulk data packet with the particular input to be used for the subcase referenced in the Solution Control Packet.

- **BLAST**: Parameters for nuclear blast analyses.
- **FLUTTER**: Basic parameters for flutter analyses.
- **TRIM**: Flight condition for steady aeroelastic trim analyses.
- **EIGC**: Complex eigenvalue extraction parameters.
- **EIGR**: Real eigenvalue extraction parameters.
- **FFT**: Fast Fourier Transform parameter definition.
- **FREQ**: Frequency step definition for frequency response.
- **FREQ1**: Frequency step definition for frequency response.
- **FREQ2**: Frequency step definition for frequency response.
- **IC**: Initial condition definition for direct transient response (same as NASTRAN TIC entry).
- **TABDMP1**: Modal damping table for modal dynamic response.
- **TF**: Dynamic transfer function definition.
- **TSTEP**: Time step definition for transient response.
- **VSDAMP**: Definition of viscous damping based on equivalent structural damping.

### 4.6. **DIFFERENCES BETWEEN ASTROS AND NASTRAN BULK DATA**

Some of the bulk data entries listed in the preceding Section do not exist in the MSC/NASTRAN or the COSMIC/NASTRAN versions that guided the definition of the bulk data entries. Some of them do exist in other NASTRAN systems, however; the DYNRED, JSET, JSSET1, PCOMP1, and PCOMP2 entries are examples. Others take the place of the NASTRAN PARAM entry which was felt to have been overused to the point where it had lost all utility. Examples of these inputs are the CONVERT, MFORM and VSDAMP entries. The steady aeroelastic model is completely new to ASTROS since NASTRAN uses the same modeling for both steady and unsteady analysis. Also, it was felt that the NASTRAN mechanism for defining dynamic loads was needlessly complicated. Working from the NASTRAN inputs, a simpler, but equally general set of entries was developed. This resulted in the generation of a number of new entries and the modification of others. The definition of the design variables, design variable linking and the design constraints is, of course, completely new for ASTROS.

The majority of the changed entries have been modified to accommodate the design task. In these cases, the bulk data entry is often identical to the NASTRAN version for use in analysis with optional additional fields to specify the design data. The element connectivity and property entries are all examples of this type of change in that additional field(s) have been added to specify the maximum and minimum allowable physical design variable value if shape function design variable linking is used. In
cases where data from NASTRAN preprocessors are used, there are no changes required unless shape function linking is desired.

A more subtle set of changes was required to perform multidisciplinary analysis. In NASTRAN, as was mentioned in the discussion of the Solution Control packet, many parameters were specified as part of the model definition or discipline specification because the code was limited to performing a single analysis of the given discipline. In order to remove these artificial restrictions, these data have been moved to the proper discipline's subcase definition. Examples of this form of modification are the addition of symmetry options to the **MKAERO1**, **GUST**, **FLUTTER**, and **TRIM** entries and the removal of subcase dependent data from the **AERO** entry. Further, the rigid elements, **ASETi**, **OMITi** and **EPOINT** entries were modified to include a set identification number to enable multiple boundary conditions and multiple control systems to be analyzed simultaneously.

The last type of modification came about because of the nature of the ASTROS database management system. These were limited to the **DM1** and **DMIG** entries for direct loading of database entities. The NASTRAN inputs were not compatible with the ASTROS database and so had to be modified. In fact, these entries, while having the same name as a NASTRAN entry, are completely new entries for ASTROS. A minor additional modification to the input definitions was made for the **TABDmPi** entry to make it more compact and to remove the spurious **ENDT** table termination symbol. In ASTROS, all tabular input entries are terminated when no more data appears and require no specific declaration of the table end.

While a seemingly large number of bulk data entries have been changed relative to their NASTRAN counterparts, in fact only a few have been changed in such a way that the NASTRAN version will not work in analysis. By far, the majority of the modeling bulk data entries are completely unchanged except for certain design variable linking options. In unsteady and steady aerodynamic disciplines care must be taken to account for the subcase dependencies that NASTRAN defined implicitly or with **PARAM** entries. Finally, the use of **ASET** and **OMIT** entries will cause minor problems in that ASTROS requires a set identification for these entries. While this latter restriction can require some effort to fix, the gain in capability simply required that the bulk data entry be modified.

The most serious potential problem using NASTRAN models in ASTROS is that the set of bulk data entries is more limited in ASTROS than in NASTRAN. The ASTROS system has been developed primarily as a multidisciplinary preliminary design tool and does not yet contain the wide range of option produced by a mature code like NASTRAN. The many NASTRAN input entries supporting these options, therefore, have not been defined to the ASTROS system because they are not supported by any ASTROS code. Thus, there will be instances where a NASTRAN input deck will have to be modified to remove these entries which serve no purpose in ASTROS. The majority of these bulk data entries deal with unsupported elements, plotting options, output options, etc., which are not felt to present a major problem. More important is the support for NASTRAN's model definitions, most of which have already been adopted by ASTROS.

### 4.7. BULK DATA DESCRIPTIONS

This Section contains a complete description of each of the ASTROS Bulk Data entries.
**Input Data Entry:**  $Comment

**Description:**  For user convenience in inserting commentary material into the unsorted echo of the input Bulk Data Deck. The $ entry is otherwise ignored by the program. These entries will not appear in a sorted echo.

**Format and Example:**

```
1 2 3 4 5 6 7 8 9 10
$ Followed by any legitimate characters in columns 2-80

$ THIS (*,",$$)--/
```

**Remarks:**

1. The comment entry may also be used in the Solution Control packet.
Input Data Entry: **AEPACT**  
Aerodynamic Lists

**Description:** Used to specify lists of real numbers for aeroelastic analysis.

**Format and Example:**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>AEPACT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AEPACT</td>
<td>97</td>
<td>.3</td>
<td>.7</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Field** | **Contents**
--- | ---
SID | Set identification number (Unique Integer > 0).
Di | Number (Real).

**Remarks:**

1. These factors must be selected by an **AIRFOIL, AXSTA, CAEROi or PAEROi** data entry.
2. Embedded blank fields are forbidden.
3. If used to specify division points, note that there is one more division point than the number of divisions.
Input Data Entry: AERO Aerodynamic Physical Data

Description: Gives basic aerodynamic parameters for unsteady aerodynamic disciplines.

Format and Example:

<p>| | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AERO</td>
<td>ACSID</td>
<td>REFC</td>
<td>RHOREF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>300.0</td>
<td>1.1E-7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field | Contents
--- | ---
ACSID | Aerodynamic coordinate system identification (Integer ≥ 0 or Blank). See Remark 2.
REFC | Reference length (for reduced frequency) (Real ≥ 0).
RHOREF | Reference density (Real ≥ 0).

Remarks:
1. This entry is required for unsteady aerodynamic disciplines. Only one AERO entry is allowed.
2. The ACSID must be a rectangular coordinate system. Flow is in the positive x-direction. If blank, the basic coordinate system is used.
**Input Data Entry:** AEROS Static Aero Physical Data

**Description:**
Gives basic parameters for static aeroelasticity.

**Format and Example:**

<table>
<thead>
<tr>
<th>AEROS</th>
<th>ACSID</th>
<th>RCSID</th>
<th>REFC</th>
<th>REFB</th>
<th>REFS</th>
<th>GREF</th>
<th>REFD</th>
<th>REFL</th>
</tr>
</thead>
<tbody>
<tr>
<td>AEROS</td>
<td>10</td>
<td>20</td>
<td>10.</td>
<td>100.</td>
<td>1000.</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field Contents**

- **ACSID**: Aerodynamic coordinate system identification (Integer > 0) or blank. See Remark 2.
- **RCSID**: Reference coordinate system identification for rigid body motions (Integer > 0) or blank.
- **REFC**: Reference chord length (Real > 0.0) (D = 1.0)
- **REFB**: Reference span (Real > 0.0) (D = 1.0)
- **REFS**: Reference wing area (Real > 0.0) (D = 1.0)
- **GREF**: Reference grid point for stability derivative calculations (Integer > 0).
- **REFD**: Fuselage reference diameter (Real > 0) or blank (D = 1.0)
- **REFL**: Fuselage reference length (Real > 0) or blank (D = 1.0)

**Remarks:**

1. This entry is required for static aeroelasticity problems. Only one AEROS entry is allowed.
2. The ACSID must be a rectangular coordinate system. Flow is in the positive x-direction.
3. The RCSID must be a rectangular coordinate system. All degrees of freedom defining trim variables will be defined in this coordinate system.
Input Data Entry  

**AESURF**  
Aerodynamic Control Surface

**Description:** Specifies an Aerodynamic Control Surface.

**Format and Example:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>AESURF</td>
<td>LABEL</td>
<td>TYPE</td>
<td>ACID</td>
<td>CID</td>
<td>FBOXID</td>
<td>LBOXID</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

AESURF  ELEV  SYM  6000  6010  6030

**Field**  
**Contents**

**LABEL**  Unique alphanumeric string of up to eight characters used to identify the control surface

**TYPE**  Surface type (Character) (Remark 2)

- **SYM** symmetric surface
- **ANTISYM** antisymmetric surface
- **ASYM** Asymmetric surface

**ACID**  Identification number of the aircraft component (CAERO6) on which the surface lies. (Integer > 0)

**CID**  Identification number of a rectangular coordinate system whose y-axis defines the hinge line of the control surface. (Integer > 0 or blank)

**FBOXID**  First aero box on the control surface relative to ACID. (Integer > 0)

**LBOXID**  Last aero box on the control surface relative to ACID. (Integer > 0)

**Remarks:**

1. The LABEL is arbitrary, but all labels must be unique.

2. The asymmetric surface, TYPE=ASYM, is not currently available. Pitch controllers are TYPE=SYM while yaw and roll controllers are TYPE=ANTISYM.

3. The aerodynamic box numbering scheme is illustrated on the CAERO1 Bulk Data entry.
**Input Data Entry**  \[\text{AIRFOIL}\]  \[\text{Airfoil Definition}\]

**Description:** Defines airfoil properties for the static aerodynamic model.

**Format and Example:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIRFOIL</td>
<td>ACID</td>
<td>CMPNT</td>
<td>CP</td>
<td>CHORD</td>
<td>USO/THK</td>
<td>LSO</td>
<td>CAM</td>
<td>RADIUS</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>X1</td>
<td>Y1</td>
<td>Z1</td>
<td>X12</td>
<td>IPANEL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field**  
- **ACID**: Associated aircraft component identification number referenced by a matching CAERO6 bulk data entry. (Integer > 0)
- **CMPNT**: Type of aircraft component selected from WING, FIN, and CANARD. (Text) (See Remark 3)
- **CP**: Coordinate system for airfoil. (Integer > 0, or blank) (See Remark 4)
- **CHORD**: Identification number of an AEFACT data entry containing a list of division points (in terms of percent chord) at which airfoil thickness and camber data are specified. (Integer > 0)
- **USO/THK**: Identification number of an AEFACT data entry defining either the upper surface ordinates in percent chord if LSO is not blank, or the half thicknesses about the camber ordinates if CAM is not blank. (Integer > 0, or blank) (See Remark 3)
- **LSO**: Identification number of an AEFACT data entry defining the lower surface ordinates in percent chord. Must be used in conjunction with USO. (Integer > 0, or blank) (See Remark 3)
- **CAM**: Identification number of an AEFACT data entry defining the mean line (camber line) ordinates in percent chord. (Integer) (See Remark 3)
- **RADIUS**: Radius of leading edge in percent chord. (Real ≥ 0.0)
- **X1, Y1, Z1**: Location of the airfoil leading edge in coordinate system CP. (Real, Y1 ≥ 0.0)
- **X12**: Airfoil chord length in x-axis coordinate of system CP. (Real > 0 or blank)
- **IPANEL**: Identification number of an AEFACT data entry containing a list of chord wise cuts in percent chord for wing paneling. (Integer > 0, or blank)

**Remarks:**

1. If the **RADIUS** field is blank, a round leading edge of radius zero is used.
2. **IPANEL** is optional and is used when different chord-wise cuts on each end of the panel are desired.
For **WING** components, the options for **USO**, **LSO**, **THK** and **CAM** are:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Blank</td>
<td>Default flat plat airfoil generated automatically</td>
</tr>
<tr>
<td><strong>USO</strong> alone</td>
<td>Lower and upper surface ordinates of airfoil are defined with effectively <strong>LSO</strong> = <strong>USO</strong> internally generated</td>
</tr>
<tr>
<td><strong>USO/LSO</strong></td>
<td>Lower and upper surface ordinates of airfoil are defined; <strong>CAM</strong> must be blank</td>
</tr>
<tr>
<td><strong>THK/CAM</strong></td>
<td>Half thicknesses about the camber line are defined. <strong>LSO</strong> must be blank</td>
</tr>
<tr>
<td><strong>USO/LSO/CAM</strong></td>
<td><strong>Illegal</strong> over-specification of data</td>
</tr>
<tr>
<td><strong>LSO/CAM</strong></td>
<td><strong>Illegal</strong>, must use <strong>THK</strong> field for half thickness</td>
</tr>
<tr>
<td><strong>CAM</strong> alone</td>
<td><strong>Illegal</strong> under-specification of data</td>
</tr>
</tbody>
</table>

For **CANARD** components, the options are as above except that camber is not allowed so **CAM** must be blank.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Blank</td>
<td>Default flat plat airfoil generated automatically</td>
</tr>
<tr>
<td><strong>USO</strong> alone</td>
<td>Lower and upper surface ordinates of airfoil are defined with effectively <strong>LSO</strong> = <strong>USO</strong> internally generated</td>
</tr>
<tr>
<td><strong>USO/LSO</strong></td>
<td>Lower and upper surface ordinates of airfoil are defined; <strong>CAM</strong> must be blank</td>
</tr>
<tr>
<td><strong>THK/CAM</strong></td>
<td><strong>Illegal</strong> specification, <strong>CAM</strong> must be blank</td>
</tr>
<tr>
<td><strong>USO/LSO/CAM</strong></td>
<td><strong>Illegal</strong> over-specification of data</td>
</tr>
<tr>
<td><strong>LSO/CAM</strong></td>
<td><strong>Illegal</strong>, <strong>CAM</strong> must be blank</td>
</tr>
<tr>
<td><strong>CAM</strong> alone</td>
<td><strong>Illegal</strong> under-specification of data and <strong>CAM</strong> must be blank for <strong>CANARD</strong></td>
</tr>
</tbody>
</table>

For **FIN** components, the options are very limited: only symmetric airfoils are allowed and they must be entered as an upper surface ordinate (the lower surface ordinates are then defaulted).

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Blank</td>
<td>Default flat plat airfoil generated automatically</td>
</tr>
<tr>
<td><strong>USO</strong> alone</td>
<td>Lower and upper surface ordinates of airfoil are defined with effectively <strong>LSO</strong> = <strong>USO</strong> internally generated. <strong>Only Legal Nonblank Fin Option</strong></td>
</tr>
</tbody>
</table>

4. The basic coordinate system must be used (CP blank). This field exists to allow the addition of user defined coordinate systems in the future.
**Input Data Entry:** ASET  

**Description:** Defines degrees of freedom that the user desires to place in the analysis set. Used to define the number of independent degrees of freedom.

**Format and Examples:**

<table>
<thead>
<tr>
<th>ASET</th>
<th>SETID</th>
<th>ID</th>
<th>C</th>
<th>ID</th>
<th>C</th>
<th>ID</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>16</td>
<td>2</td>
<td>23</td>
<td>3516</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field** | **Contents**
--- | ---
SETID | The set identification number of the REDUCE set. (Integer > 0)
ID | Grid or scalar point identification number (Integer > 0)
C | Component number, zero or blank for scalar points, any unique combinations of the digits 1 through 6 for grid points.

**Remarks:**

1. When ASET and/or ASET1 entries are present, all degrees of freedom not otherwise constrained will be placed on the o-set. The o-set is a mutually exclusive set. Degrees of freedom may not be specified on other entries that define mutually exclusive sets.

2. ASET entries must be selected in Solution Control (REDUCE=SETID) to be used.
Input Data Entry: **ASET1** Selected Coordinates for the a-set, Alternate Form

**Description:** Defines degrees of freedom that the user desires to place in the analysis set. Used to define the number of independent degrees of freedom.

**Format and Examples:**

```
<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASET1</td>
<td>SETID</td>
<td>C</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

```
| ASET1 | 345 | 2 | 1 | 3 | 10 | 9 | 6 | 15 | ABC |
| +bc | 7 | 8 | &nbsp; |
```

**Alternate Form:**

```
<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASET1</td>
<td>SETI</td>
<td>C</td>
<td>ID1</td>
<td>&quot;THRU&quot;</td>
<td>ID2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**Field**

| **SETID** | The REDUCE set identification number (Integer > 0) |
| **C** | Component number (any unique combination of the digits 1 through 6 with no embedded blanks) when point identification numbers are grid points; must be null or zero if point identification numbers are scalar points. |
| **G, ID1, ID2** | Grid or scalar point identification numbers (Integer > 0, ID2 > ID1) |

**Remarks:**

1. When ASET and/or ASET1 entries are present, all degrees of freedom not otherwise constrained will be placed in the o-set. The o-set is a mutually exclusive set. Degrees of freedom may not be specified on other entries that define mutually exclusive sets.

2. If the alternate form is used, all points in the sequence ID1 through ID2 are required to exist.

3. ASET1 entries must be selected in Solution Control (REDUCE=SETID) to be used.
Input Data Entry: ATTACH

Description: Defines the aerodynamic control points to be attached to a reference grid for load transfer.

Format and Example:

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTACH</td>
<td>EID</td>
<td>MACROID</td>
<td>BOX1</td>
<td>BOX2</td>
<td>RGRID</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| ATTACH | 100 | 111 | 111 | 118 | 1 |

Field Contents

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>EID</td>
<td>Element identification number (Integer &gt; 0)</td>
</tr>
<tr>
<td>MACROID</td>
<td>Element identification of a CAERO1 or PAERO1 element which contains the specified aerodynamic control points (Integer &gt; 0)</td>
</tr>
<tr>
<td>BOX1, BOX2</td>
<td>Starting and final box whose force is to be transferred to the referenced grid (Integer &gt; 0, BOX2 &gt; BOX1)</td>
</tr>
<tr>
<td>RGRID</td>
<td>Grid point identification of reference grid point (Integer &gt; 0)</td>
</tr>
</tbody>
</table>

Remarks:

1. The EID is used only for error messages.
2. This entry applies to both the steady and unsteady aerodynamic models.
3. The attached aerodynamic boxes are selected as shown below:
**Input Data Entry: AXSTA**

**Description:** Defines body axial station parameters. There is one AXSTA for each axial station at which the surface points are defined.

**Format and Examples:**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AXSTA</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BCID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XSTA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CBOD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ABOD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LYRAD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LZRAD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

<table>
<thead>
<tr>
<th>BCID</th>
<th>Body component identification number (Integer &gt; 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>XSTA</td>
<td>Value of the x-ordinate of the body station (Real)</td>
</tr>
<tr>
<td>CBOD</td>
<td>Value of the z-ordinate of the center line at this station. This defines the body camber (Real).</td>
</tr>
<tr>
<td>ABOD</td>
<td>Cross sectional area of the body at this station (Real ≥ 0.0).</td>
</tr>
<tr>
<td>LYRAD, LZRAD</td>
<td>Identification number of an AEPACT data entry containing a list of the y-ordinates (z-ordinates) of the body section. (Integer ≥ 0.0)</td>
</tr>
</tbody>
</table>

**Remarks:**

1. If ABOD is present, the body is assumed to be circular and the radial ordinates are computed at NRAD (cf. the BODY bulk data entry) equal intervals. No LYRAD and LZRAD data are allowed when ABOD is present.

2. If ABOD is blank, LYRAD and LZRAD data must be present.

3. For Pods, CBOD, LYRAD and LZRAD data are not permitted.

4. For the fuselage, XSTA is actual x location; for pods, XSTA is relative to the XLOC value given on the BODY bulk data entry.
Input Data Entry: BAROR  Simple Beam (BAR) Orientation Default Values

Description: Defines default values for fields 3 and 6 - 8 of the CBAR entry.

Format and Examples:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>PID</th>
<th></th>
<th>X1, GO</th>
<th>X2</th>
<th>X3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>BAROR</td>
<td>PID</td>
<td>X1, GO</td>
<td>X2</td>
<td>X3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAROR</td>
<td>39</td>
<td>0.6</td>
<td>2.9</td>
<td>-5.87</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field  Contents
PID  Identification number of FBAR property entry (Integer > 0 or blank)
Xi   Vector components measured in displacement coordinate system at GA to determine (with the vector from end A to end B) the orientation of the element coordinate system for the bar element (Real or blank)
GO  Grid point identification number (Integer > 0)

Remarks:
1. The contents of fields on this entry will be assumed for any CBAR entry whose corresponding fields are blank.
2. Only one BAROR entry may appear in the user’s Bulk Data Deck.
Input Data Entry: **BLAST**  
Nuclear Blast Parameters

**Description:** Defines basic parameters needed for nuclear blast response analysis.

**Format and Example:**

```
<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLAST</td>
<td>BLID</td>
<td>ALTITUDE</td>
<td>VELOCITY</td>
<td>WKT</td>
<td>BALT</td>
<td>MACH</td>
<td>DELTAX</td>
<td>DELTAY</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>KGRD</td>
<td>HGRD</td>
<td>SYMXZ</td>
<td>SYMXY</td>
<td>TRSURF</td>
<td>NZ</td>
<td>CONT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>TMIN</td>
<td>Tmax</td>
<td>NTIME</td>
<td>BMIN</td>
<td>BMAX</td>
<td>NBETA</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

```
BLAST 100 10000. 1000.0 1.E5 15000. 0.8  
*BC 1 ELEV 6.0
```

**Field**  
**Contents**

- **BLID:** Set identification number referenced by Solution Control  
  (Integer > 0)
- **ALTITUDE:** Aircraft altitude (Real ≥ 0.0)
- **VELOCITY:** Aircraft velocity (Real ≥ 0.0)
- **WKT:** Weapon yield (Real ≥ 0.0)
- **BALT:** Blast altitude (Real)
- **MACH:** Mach number (Real 0.0 ≤ Mach ≤ 1.0)
- **DELTAX:** X distance from aircraft to blast point (Real)
- **DELTAY:** Y distance from aircraft to blast point (Real)
- **KGRD:** Key denoting presence of the ground (Integer)  
  - = 0, no ground
  - = 1, include the ground
- **HGRD:** Height of ground level (Real)
- **SYMXXZ:** Symmetry flag for blast analysis about xz plane (Integer)
- **SYMXY:** Symmetry flag for blast analysis about xy plane (Integer)
- **TRSURF:** Label of an A祀surf entry used as the trim surface (Character)
- **NZ:** Load factor for the trim calculation (Real)
- **TMIN:** Minimum time used in the definition of polynomial curve fitting (Real > 0.0, or blank, default = 0.10)
- **TMAX:** Maximum time used in the definition of polynomial curve fitting (Real > 0.0, or blank, default = 20.0)
- **NTIME:** Number of time steps (Integer > 0, or blank, default = 20)
- **BMIN:** Minimum Beta value used in the definition of polynomial curve fitting (Real > 0.0, or blank, default = 0.375)
BMAX  Maximum Beta value used in the definition of polynomial curve fitting (Real > 0.0, or blank, default = 10.0)

NBETA  Number of Beta values (Integer > 0, or blank, default = 7)

Remarks:
1. In order to be used, the BLID must be referenced by Solution Control.
2. The second continuation entry is not required.
3. The default values of TMIN, TMAX, NTIME, BMIN, BM and NBETA should be adequate.
4. The symmetry flags refer to the options selected on the MAKERO1 entries.
Input Data Entry: BODY

Description: Defines body configuration parameters for steady aeroelasticity.

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>BODY</td>
<td>BCID</td>
<td>CMPNT</td>
<td>CP</td>
<td>NRAD</td>
<td>XLOC</td>
<td>YLOC</td>
<td>ZLOC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field | Contents
---|---
BCID | Body component identification number (Integer > 0)
CMPNT | Component type (FUSEL for the fuselage and POD for a pod)
CP | Coordinate system of the geometry input (Integer ≥ 0, or blank)
NRAD | Number of equal radial cuts used to define the body (Integer ≥ 0, or blank)
XLOC, YLOC, ZLOC | Ordinates of the nose of the pod in the CP coordinate system (Real)

Remarks:

1. NRAD is input if equally spaced radial cuts are desired. Arbitrary radial cuts are specified using the AXSTA and AEFACT data entries.

2. The geometry given with the XLOC, YLOC, ZLOC entries is used only with POD components.
Input Data Entry: CAERO1 Aerodynamic Panel Element Connection

Description: Defines an aerodynamic macroelement (panel) in terms of two leading edge locations and side chords. This is used for Doublet-Lattice theory.

Format and Example:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAERO1</td>
<td>EID</td>
<td>PID</td>
<td>CP</td>
<td>NSPAN</td>
<td>NCHORD</td>
<td>LSPAN</td>
<td>LCHORD</td>
<td>IGID</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>+BC</td>
<td>X1</td>
<td>Y1</td>
<td>Z1</td>
<td>X12</td>
<td>X4</td>
<td>Y4</td>
<td>24</td>
<td>X43</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CAERO1 1000 1 3 2 1 ABC
+BC 0.0

Field Contents

**EID** Element identification number (Integer > 0)

**PID** Identification number of property entry (Integer > 0, or blank). Used to specify associated bodies

**CP** Coordinate system for locating points 1 and 4 (Integer ≥ 0 or blank)

**NSPAN** Number of span-wise boxes; if a positive value is given NSPAN, equal divisions are assumed; if zero or blank, a list of division points is given at LSPAN (Integer ≥ 0 or blank)

**NCHORD** Number of chord-wise boxes; if a positive value is given NCHORD, equal divisions are assumed; if zero of blank, a list of division points is given at LCHORD (Integer > 0 or blank)

**LSFAN** Identification number of an AEFACT data entry containing a list of division points for span-wise boxes. Used only if NSPAN is zero or blank (Integer ≥ 0 or blank)

**LCHORD** Identification number of an AEFACT data entry containing a list of division points for chord-wise boxes. Used only if NCHORD is zero or blank (Integer ≥ 0 or blank)

**IGID** Interference group identification (aerodynamic elements with different IGIDs are uncoupled) (Integer > 0)

**X1, Y1, Z1; X4, Y4, Z4; X12; X43** Location of points 1 and 4, in coordinate system CP (Real)

Remarks:

1. The boxes are numbered sequentially, beginning with EID.
2. The continuation entry is required.
3. The number of division points is one greater than the number of boxes. Thus, if NSPAN = 3, 3 division points are 0.0, 0.333, 0.667, 1.000. If the user supplies division points, the first and last points need not be 0 and 1 (in which the corners to the panel would not be at the reference points).
4. A triangular element is formed if X12 or X43 = 0.0.
5. The element coordinate system (right-handed) is shown in the sketch below.
Input Data Entry: **CAERO2**  Unsteady Aerodynamic Body Connection

Description: Defines an aerodynamic body for Doublet-Lattice aerodynamics.

Format and Examples:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAERO2</td>
<td>EID</td>
<td>PID</td>
<td>CP</td>
<td>NSB</td>
<td>NINT</td>
<td>LSB</td>
<td>LINT</td>
<td>IGID</td>
<td>CONT</td>
</tr>
<tr>
<td>+BC</td>
<td>X1</td>
<td>Y1</td>
<td>Z1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| CAERO2 | 1500 | 2 | 100 | 4 | 99 | 1 | ABC |
| +BC | -1.0 | 100 | -30 | 175 | |

**Field** | **Contents**
---|---
EID | Element identification number (Integer > 0)
PID | Property identification number (Integer > 0)
CP | Coordinate system for locating point 1 (Integer ≥ 0, or blank)
NSB | Grid point identification number of connection points (Integer > 0)
NINT | Number of interference elements; if a positive number is given, NSB equal divisions are assumed; if zero of blank, see LSB for a list of divisions (Integer ≥ 0, or blank)
LSB | Identification number of an AEFACT data entry for slender body division points; used only if NSB is zero or blank (Integer ≥ 0, or blank)
LINT | Identification number of an AEFACT data entry containing a list of division points for interference elements; used only if NINT is zero or blank (Integer ≥ 0, or blank)
IGID | Interference group identification (aerodynamic elements with different IGID's are uncoupled) (Integer > 0)
X1,Y1,Z1 | Location of points 1 and 4, in coordinate system CP (Real)
X12 | Edge chord lengths (in aerodynamic coordinate system) (Real ≥ 0, and not both zero)

**Remarks:**

1. Point 1 is the leading point of the body.
2. All **CAERO1** (panels) and **CAERO2** (bodies) in the same group (IGID) will have aerodynamic interaction.
3. At least one interference element is required for each aerodynamic body specified by this entry.
4. Element identification numbers on the aerodynamic bodies must have the following sequence:
   (A) Panels first
   (B) Z bodies (see **PAERO2** orientation flag)
   (C) ZY bodies
   (D) Y bodies
Input Data Entry: CAERO6

Description: Defines an aerodynamic macroelement (panel) for USSAERO.

Format and Example:

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACID</td>
<td>Component identification number (Integer &gt; 0)</td>
</tr>
<tr>
<td>CMPNT</td>
<td>Aircraft component (Text)</td>
</tr>
<tr>
<td>CP</td>
<td>Coordinate system (Integer ≥ 0, or blank) (See Remark 5)</td>
</tr>
<tr>
<td>IGRP</td>
<td>Group number for this component (Integer &gt; 0)</td>
</tr>
<tr>
<td>LCHORD</td>
<td>Identification number of an ABFACT Bulk Data entry containing a list of division points in percent chord for chord-wise boxes for the aerodynamic surface. If LCHORD is zero, the chord-wise divisions are identified by the IPANEL entry on the AIRFOIL Bulk Data entry (Integer ≥ 0 or blank)</td>
</tr>
<tr>
<td>LSPAN</td>
<td>Identification number of an ABFACT Bulk Data entry containing a list of division points for spanwise boxes. For WINGS and CANARDS use the y (lateral) dimensional coordinates of the stations, and for FINS, use the z (vertical) dimensional coordinates. If LSPAN is zero or blank, the y/z locations from the AIRFOIL Bulk Data entries for the component ACID are used (Integer ≥ 0 or blank)</td>
</tr>
</tbody>
</table>

Remarks:

1. Allowable components are WING, FIN, and CANARD.
2. The IGRP field allows related components to be processed together for interference effects; e.g., one group could be a wing/body/tail combination while a second group could be a pod/fin combination.
3. Note that the chord-wise cuts are in percent while the span-wise cuts require physical coordinates. For span-wise cuts, y-coordinates are input for wings and canards while z-coordinates are input for fins.
4. Only the right half-plane can be modeled in USSAERO. As such, all y-coordinates specified by LSPAN must be positive.
5. The basic coordinate system must be used (CP blank). This field exists for the addition of a user defined coordinate system in the future.
Input Data Entry: CBAR  Simple Beam Element Connection

Description: Defines a simple beam element (BAR) of the structural model.

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBAR</td>
<td>EID</td>
<td>PID</td>
<td>GA</td>
<td>GB</td>
<td>X1, GO</td>
<td>X2</td>
<td>X3</td>
<td>TMAX</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>PA</td>
<td>PB</td>
<td>W1A</td>
<td>W2A</td>
<td>W3A</td>
<td>W1B</td>
<td>W2B</td>
<td>W3B</td>
<td></td>
</tr>
</tbody>
</table>

CBAR 2 39 7 3 13 +23 123

Field Contents

EID  Unique element identification number (Integer > 0).

PID  Identification number of a PBAR property entry (Default is EID unless BAROR entry has nonzero entry in Field 3) (Integer > 0)

GA, GB  Grid point identification numbers of connection points (Integer > 0).

Xi  Components of vector \(\{v\}\), at end A, measured at end A, parallel to the components of the displacement coordinate system for \(\text{GA}\), to determine (with the vector from end A to end B) the orientation of the element coordinate system for the BAR element (Real)

GO  Grid point identification number to optionally supply \(\text{Xi}\) (Integer > 0). Direction of orientation vector is \(\text{GA}\) to \(\text{GO}\)

TMAX  Maximum allowable cross-sectional area in design (Real > 0.0, or blank). Default = 10^4

PA, PB  Pin flags for bar ends A and B, respectively (up to 5 of the unique digits 1 through 6 anywhere in the fields with no embedded blanks; Integer > 0 or blank). Used to remove connections between the grid point and selected degrees of freedom of the bar. The degrees of freedom are defined in the element’s coordinate system. The bar must have stiffness associated with the pin flag. For example, if PA=4 is specified, the PBAR entry must have a value for J, the torsional stiffness.

W1A, W2A, W3A  Components of offset vectors wa and wb, respectively, in displacement coordinate systems at points \(\text{GA}\) and \(\text{GB}\), respectively (Real or blank).

W1B, W2B, W3B

Remarks:

1. The element coordinate system is shown in the following figure:
2. If there are no pin flags or offsets, the continuation entry may be omitted.

3. The THMAX value is used only for shape function design variable linking.

4. See the BAROR entry for default options for Fields 3 and 6 through 8.
Input Data Entry:  CELAS1    Scalar Spring Connection

Description:  Defines a scalar spring element of the structural model

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CELAS1</td>
<td>EID</td>
<td>PID</td>
<td>G1</td>
<td>C1</td>
<td>G2</td>
<td>C2</td>
<td>TMAX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CELAS1</td>
<td>2</td>
<td>6</td>
<td>8</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field | Contents
---|---
EID | Element identification number (Integer > 0)
PID | Identification number of a PELAS property entry (Default is EID) (Integer > 0)
Gi | Geometric grid point identification number (Integer > 0)
Ci | Component number (6 ≥ Integer ≥ 0)
TMAX | Maximum value for design (Real, Default = 1.0E4)

Remarks:
1. Scalar points may be used for G1 and/or G2 in which case the corresponding C1 and/or C2 must be zero or blank. Zero or blank may be used to indicate a grounded terminal G1 or G2 with a corresponding blank or zero C1 or C2. A grounded terminal is a point whose displacement is constrained to zero.
2. The two connection points (G1, C1) and (G2, C2) must be distinct.
3. TMAX is ignored unless the element is designed using shape function linking.
Input Data Entry: **CELAS2** Scalar Spring Property and Connection

Description: Defines a scalar spring element of the structural model without reference to a property entry.

Format and Example:

```
1 2 3 4 5 6 7 8 9 10
```

```
CELAS2  EID  K  G1  C1  G2  C2  GE  S  CONT
CONT  TMIN  TMAX
```

Field | Contents
--- | ---
EID | Element identification number (Integer > 0)
K | The value of the scalar spring (Real > 0.0)
G1 | Geometric grid point identification number (Integer ≥ 0)
C1 | Component number (6 ≥ Integer ≥ 0)
GE | Damping coefficient (Real ≥ 0.0)
S | Stress coefficient (Real ≥ 0.0)
TMIN, TMAX | Minimum and maximum values for design (Real)

Remarks:
1. Scalar points may be used for G1 and/or G2 in which case the corresponding C1 and/or C2 must be zero or blank. Zero or blank may be used to indicate a grounded terminal G1 or G2 with a corresponding blank or zero C1 or C2. A grounded terminal is a point whose displacement is constrained to zero.
2. This single entry completely defines the element since no material or geometric properties are required.
3. The two connection points (G1, C1) and (G2, C2) must be distinct.
4. The TMIN and TMAX values are ignored unless shape function design variable linking is used.
Input Data Entry: CIHEX1 Linear Isoparametric Hexahedron Element Connection

Description: Defines a linear isoparametric hexahedron element of the structural model.

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIHEX1</td>
<td>EID</td>
<td>PID</td>
<td>G1</td>
<td>G2</td>
<td>G3</td>
<td>G4</td>
<td>G5</td>
<td>G6</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>G7</td>
<td>G8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIHEX1</td>
<td>137</td>
<td>5</td>
<td>3</td>
<td>8</td>
<td>5</td>
<td>4</td>
<td>9</td>
<td>14</td>
<td>ABC</td>
</tr>
<tr>
<td>+BC</td>
<td>88</td>
<td>602</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

EID Element identification number (Integer > 0).

PID Identification number of a PHX property entry (Integer > 0). (Default is EID)

Gi Grid point identification numbers of connection points (Integer > 0, G1 ≠ G2 ≠...G8).

Remarks:

1. Grid points G1, G2, G3, and G4 must be given in counterclockwise order about one quadrilateral, with G1 and G5 along the same edge as shown in the figure below:

![Diagram of grid points](image)

2. There is no nonstructural mass.

3. The quadrilateral faces need not be planar.

4. Stresses are given in the basic coordinate system.

5. The continuation is required.

6. No physical property in this element can be used as a local design variable for automated design.
Input Data Entry:  CIHEX2  Quadratic Isoparametric Hexahedron Element Connection

Description:  Defines a quadratic isoparametric hexahedron element of the structural model.

Format and Example:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIHEX1</td>
<td>EID</td>
<td>PID</td>
<td>G1</td>
<td>G2</td>
<td>G3</td>
<td>G4</td>
<td>G5</td>
<td>G6</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>G7</td>
<td>G8</td>
<td>G9</td>
<td>G10</td>
<td>G11</td>
<td>G12</td>
<td>G13</td>
<td>G14</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>G15</td>
<td>G16</td>
<td>G17</td>
<td>G18</td>
<td>G19</td>
<td>G20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CIHEX1</td>
<td>110</td>
<td>7</td>
<td>3</td>
<td>8</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>9</td>
<td>ABC</td>
<td></td>
</tr>
<tr>
<td>+BC</td>
<td>5</td>
<td>4</td>
<td>16</td>
<td>19</td>
<td>20</td>
<td>17</td>
<td>23</td>
<td>27</td>
<td>DEF</td>
<td></td>
</tr>
<tr>
<td>+EF</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>28</td>
<td>25</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>EID</td>
<td>Element identification number (Integer &gt; 0)</td>
</tr>
<tr>
<td>PID</td>
<td>Identification number of a PIHEX property entry (Integer &gt; 0) (Default is BID)</td>
</tr>
<tr>
<td>Gi</td>
<td>Grid point identification numbers of connection points (Integer &gt; 0, G1 ≠ G2 ≠ ... ≠ G20).</td>
</tr>
</tbody>
</table>

Remarks:

1. Grid points G1,...,G8 must be given in counterclockwise order about one quadrilateral face when viewed from inside the element. G9,...,G12 and G13,...,G20 must also be in a counterclockwise direction with G1, G9 and G13 along the same edge as shown in the figure below:

![Diagram of a hexahedron element with grid points](image)

2. There is no nonstructural mass.
3. The quadrilateral faces need not be planar.
4. Stresses are given in the basic coordinate system.
5. The continuations are required.
6. No physical property in this element can be used as a local design variable for automated design.
Input Data Entry: CIHEX3 Cubic Isoparametric Hexahedron Element Connection

Description: Defines a cubic isoparametric hexahedron element of the structural model.

Format and Example:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIHEX1</td>
<td>EID</td>
<td>PID</td>
<td>G1</td>
<td>G2</td>
<td>G3</td>
<td>G4</td>
<td>G5</td>
<td>G6</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>G7</td>
<td>G8</td>
<td>G9</td>
<td>G10</td>
<td>G11</td>
<td>G12</td>
<td>G13</td>
<td>G14</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>G15</td>
<td>G16</td>
<td>G17</td>
<td>G18</td>
<td>G19</td>
<td>G20</td>
<td>G21</td>
<td>G22</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>G23</td>
<td>G24</td>
<td>G25</td>
<td>G26</td>
<td>G27</td>
<td>G28</td>
<td>G29</td>
<td>G30</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>G31</td>
<td>G32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field | Contents
---|---
EID | Element identification number (Integer > 0).
PID | Identification number of a PIHEX property entry (Integer > 0) (Default is EID).
Gi | Grid point identification number of connection points (Integer > 0, G1 ≠ G2 ≠ ... ≠ G32).
Remarks:
1. Grid points G1,...,G12 must be given in counterclockwise order about one quadrilateral face when viewed from inside the element. G13,...,G16, G17,...,G20, and G21,...,G32 must also be in a counterclockwise direction with G1, G13, G17, and G21 along the same edge.
2. There is no nonstructural mass.
3. The quadrilateral faces need not be planar.
4. Stresses are given in the basic coordinate system.
5. The continuations are required.
6. No physical property in this element can be used as a local design for automated design.
Input Data Entry:  **CMASS1**  Scalar Mass Connection

**Description:** Defines a scalar mass element of the structural model.

**Format and Example:**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMASS1</td>
<td>EID</td>
<td>PID</td>
<td>G1</td>
<td>C1</td>
<td>G2</td>
<td>C2</td>
<td>TMAX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMASS1</td>
<td>32</td>
<td>6</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field**  **Contents**

- **EID**  Element identification number (Integer > 0)
- **PID**  Identification number of a **PMASS** property entry (Default is **EID**) (Integer > 0)
- **Gi**  Geometric grid point identification number (Integer > 0)
- **Ci**  Component number (6 ≥ Integer ≥ 0)
- **TMAX**  The maximum mass value allowed in design (Real, Default = $10^4$)

**Remarks:**

1. Scalar points may be used for **Gi** and/or **G2** in which case the corresponding **Ci** and/or **C2** must be zero or blank. Zero or blank may be used to indicate a grounded terminal **G1** or **G2** with a corresponding blank or zero **Ci** or **C2**. A grounded terminal is a point whose displacement is constrained to zero.

2. The two connection points (**G1, C1**) and (**G2, C2**), must be distinct. Except in unusual circumstances, one of them will be a grounded terminal with blank entries for **G** and **C**.

3. The **TMAX** value is used only for shape function design variable linking.
Input Data Entry: **CMASS2** Scalar Mass Property and Connection

**Description:** Defines a scalar mass element of the structural model without reference to a property entry.

**Format and Example:**

```
  1 2 3 4 5 6 7 8 9 10
CMASS2 EID  M  G1  C1  G2  C2  TMIN  TMAX
CMASS2  32  9.25  6  1
```

**Field Contents**

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>EID</td>
<td>Element identification number (Integer &gt; 0)</td>
</tr>
<tr>
<td>M</td>
<td>The value of the scalar mass (Real)</td>
</tr>
<tr>
<td>G1, C1</td>
<td>Geometric grid point identification number (Integer &gt; 0)</td>
</tr>
<tr>
<td>G2, C2</td>
<td>Component number (Integer ≥ 0)</td>
</tr>
<tr>
<td>TMIN, TMAX</td>
<td>The minimum and maximum mass values in design (Real)</td>
</tr>
</tbody>
</table>

**Remarks:**

1. Scalar points may be used for G1 and/or G2 in which case the corresponding C1 and/or C2 must be zero or blank. Zero or blank may be used to indicate a grounded terminal G1 or G2 with a corresponding blank or zero C1 or C2. A grounded terminal is a point whose displacement is constrained to zero.

2. This single entry completely defines the element since no material or geometric properties are required.

3. The two connection points (G1, C1) and (G2, C2), must be distinct. Except in unusual circumstances, one of them will be a grounded terminal with blank entries for G and C.

4. The TMIN and TMAX values are used only for shape function design variable linking.
Input Data Entry: CONEFF Flutter aerodynamic control effectiveness data

Description: Defines adjustment factors of control surface effectiveness values for use in flutter analysis.

Format and Example:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONEFF</td>
<td>EFFID</td>
<td>EFF</td>
<td>MODE</td>
<td>MACROID</td>
<td>BOX1</td>
<td>BOX2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONEFF</td>
<td>10</td>
<td>0.60</td>
<td>6</td>
<td>1001</td>
<td>1007</td>
<td>1021</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

- **EFFID**: Effectiveness identification number (Integer > 0)
- **EFF**: Effectiveness value (Real)
- **MODE**: Structural mode to which the effectiveness is to be applied (Integer > 0)
- **MACROID**: Aerodynamic component (macroelement) on which the control surface lies
- **BOX1, BOX2**: First and last box whose effectiveness is to be altered (Integer > 0, BOX2 > BOX1)

Remarks:

1. The **EFFID** is referenced by the FLUTTER bulk data entry.
2. The **EFFID** need not be unique.
3. The pressures for the referenced mode and all the referenced boxes will be modified by the **EFF** parameter. For example, **EFF = 0.60** indicates a 40 percent reduction in the effectiveness for the affected boxes.
4. Refer to the SPLINE1 bulk data entry for the interpretation of **BOX1** and **BOX2**.
Input Data Entry: CONEFFS Static aerodynamic control effectiveness data

Description: Defines adjustment factors for control surface effectiveness values for use in static aeroelastic analysis and nonplanar aerodynamic analysis.

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONEFFS</td>
<td>EFFID</td>
<td>LABEL1</td>
<td>EFF1</td>
<td>LABEL2</td>
<td>EFF2</td>
<td>LABEL3</td>
<td>EFF3</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>LABEL4</td>
<td>EFF4</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| CONEFFS | 10 | AIL1 | 0.65 | INBORD | 0.55 |     |     |     |

Field                  Contents
---                    ---
EFFID                  A unique identification number identifying the set
LABELi                 A unique alphanumeric string of up to eight characters to identify a control surface defined by an ABSURF entry
EFFi                   Effectiveness value for the associated surface (Real)

Remarks:

1. The set identification number is referenced by the TRIM bulk data entry.

2. All aerodynamic forces created by the control surface will be reduced to the reference amount. For example, EFF1 = 0.70 indicates a 30 percent reduction in the forces.
Input Data Entry: **CONLINK**  Linked Control Surfaces

Description: Causes control surfaces to vary in a prescribed fashion relative to one another.

Format and Example:

```
1 2 3 4 5 6 7 8 9 10
CONLINK LABEL LABEL1 VAL1 LABEL2 VAL2 LABEL3 VAL3 CONT
CONT LABEL4 VAL4 -etc-
CONLINK ROLL1 AIL 1.0 LEFLAP 1.0
```

Field | Contents
--- | ---
LABEL | A unique alphanumeric string of up to eight characters to identify the control surface that is composed of other control surfaces.
LABELi | A unique alphanumeric string of up to eight characters to identify a control surface defined by an AZSURF entry
VALi | Participation factor (Real)

Remarks:

1. All of the LABEL surfaces must be of the same TYPE, e.g. SYM. See the AZSURF entry for additional information.
2. An arbitrary number of entries are allowed.
3. The CONLINK entry may not reference the LABEL of another CONLINK entry.
Input Data Entry: **CONM1**  Concentrated Mass Element Connection, General Form

**Description:** Defines a 6 x 6 symmetric matrix at a geometric grid point of the structural model.

**Format and Example:**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONM1</td>
<td>EID</td>
<td>G</td>
<td>CID</td>
<td>M11</td>
<td>M21</td>
<td>M31</td>
<td>M32</td>
<td>CONT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>M33</td>
<td>M41</td>
<td>M42</td>
<td>M43</td>
<td>M44</td>
<td>M51</td>
<td>M52</td>
<td>M53</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>M54</td>
<td>M55</td>
<td>M61</td>
<td>M62</td>
<td>M63</td>
<td>M64</td>
<td>M65</td>
<td>M66</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| CONM1 | 2 | 22 | 2 | 2.9 | 6.3 | +1 |     |     | +1 |
| -1    | 4.8 | 28.6 |     |     |     | +2 |
| +2    | 28.6 |     |     |     |     | 28.6 |

**Field** | **Contents**
---|---
EID | Element identification number (Integer > 0).
G | Grid point identification number (Integer > 0).
CID | Coordinate system identification number for the mass matrix (Integer ≥ 0 or blank).
Mij | Mass matrix values (Real).

**Remarks:**

1. For a less general means of defining concentrated mass at grid points, see **CONM2**.
2. No physical property in this element can be used as a local design variable for automated design.
Input Data Entry: CONM2 Concentrated Mass Element Connection, Rigid Body Form

Description: Defines a concentrated mass at a grid point of the structural model.

Format and Example:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONM2</td>
<td>EID</td>
<td>G</td>
<td>CID</td>
<td>M</td>
<td>X1</td>
<td>X2</td>
<td>X3</td>
<td>CONT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>I11</td>
<td>I21</td>
<td>I22</td>
<td>I31</td>
<td>I32</td>
<td>I33</td>
<td>TMIN</td>
<td>TMAX</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONM2</td>
<td>2</td>
<td>15</td>
<td>6</td>
<td>49.7</td>
<td></td>
<td></td>
<td></td>
<td>123</td>
</tr>
<tr>
<td>+23</td>
<td>16.2</td>
<td>16.2</td>
<td>7.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

EID  Element identification number (Integer > 0).

G  Grid point identification number (Integer > 0).

CID  Coordinate system identification number (Integer ≥ -1). A CID of –1 (integer) allows the user to input \( x_1 \) as the center of gravity location in the basic coordinate system. A CID of 0 implies the basic coordinate system.

M  Mass value (Real).

\( x_i \)  Offset distances from the grid point to the center of gravity of the mass in the coordinate system defined in Field 4, unless CID = –1, in which case \( x_i \) are the coordinates of the center of gravity of the mass in the basic coordinate system (Real).

\( I_{ij} \)  Mass moments of inertia measured at the mass c.g., in coordinate system defined by Field 4 (Real). If CID = –1, the basic coordinate system is implied.

TMIN, TMAX  The minimum and maximum mass values for design (Real)

Remarks:

1. The continuation entry may be omitted.

2. If CID = –1, offsets are internally computed as the difference between the grid point location and \( x_i \). The grid point locations may be defined in a nonbasic coordinate system. In this case, the values of \( I_{ij} \) must be in a coordinate system that parallels the basic coordinate system.
3. The form of the inertia matrix about its c.g. is taken as:

\[
M = \begin{bmatrix}
M_{11} & M_{12} & M_{13} \\
M_{12} & M_{22} & M_{23} \\
M_{13} & M_{23} & M_{33}
\end{bmatrix}
\]

where \( M = \int p \, dv \)

\[
I_{11} = \int p \left(x_2^2 + x_3^2\right) \, dv
\]

\[
I_{22} = \int p \left(x_1^2 + x_3^2\right) \, dv
\]

\[
I_{33} = \int p \left(x_1^2 + x_2^2\right) \, dv
\]

\[
I_{21} = \int p \, x_1 \, x_2 \, dv
\]

\[
I_{31} = \int p \, x_1 \, x_3 \, dv
\]

\[
I_{32} = \int p \, x_2 \, x_3 \, dv
\]

and \( x_1, x_2, x_3 \) are components of distance from the c.g. in the coordinate system defined in Field 4. The negative signs for the off-diagonal terms are supplied by the program. A warning message is issued if the inertia matrix is non-positive definite, as this may cause fatal errors in dynamic analysis modules.

4. For design, the mass moments of inertia must be zero.

5. The \( T_{MIN} \) and \( T_{MAX} \) values are used only for shape function design variable linking.
Input Data Entry: CONROD  Rod Element Property and Connection

Description: Defines a rod element of the structural model without reference to a property entry.

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONROD</td>
<td>EID</td>
<td>G1</td>
<td>G2</td>
<td>MID</td>
<td>A</td>
<td>J</td>
<td>C</td>
<td>NSM</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>TMIN</td>
<td>TMAX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

- **EID**: Element identification number (Integer > 0).
- **G1, G2**: Grid point identification numbers of connection points (Integer > 0).
- **MID**: Material identification number (Integer > 0).
- **A**: Area of rod (Real ≥ 0.0).
- **J**: Torsional constant (Real ≥ 0.0).
- **C**: Coefficient for torsional stress determination (Real).
- **NSM**: Nonstructural mass per unit length (Real).
- **TMIN, TMAX**: Minimum and maximum allowable cross-sectional areas in design (Real > 0.0, or blank).

Remarks:

1. For structural problems, CONROD entries may only reference MAT1 material entries.
2. The continuation entry is optional.
3. TMAX and TMIN are ignored unless element is linked to global design variable through a SHAPE entry.
**Input Data Entry:**  
CONVERT

**Description:** Defines conversion factors for various physical quantities.

**Format and Example:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONVERT</td>
<td>QUANT1</td>
<td>FACTOR</td>
<td>QUANT2</td>
<td>FACTOR</td>
<td>QUANT3</td>
<td>FACTOR</td>
<td>QUANT4</td>
<td>FACTOR</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>QUANT</td>
<td>FACTOR</td>
<td>QUANT</td>
<td>FACTOR</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONVERT</td>
<td>MASS</td>
<td>0.00259</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field** | **Contents**
--- | ---
QUANT | A character string identifying the physical quantity to be converted
= MASS, or VELOCITY
FACTOR | The conversion factor (Real \( \neq 0.0 \))

**Remarks:**

1. Any number of valid quantity-factor combinations can be entered on a single entry.
2. Only MASS and VELOCITY are currently valid quantity entries.
3. Input mass values will be multiplied by the input factor. Input velocities will be multiplied by the factor.
**Input Data Entry:**  
CORD1C  
Cylindrical Coordinate System Definition, Form 1.

**Description:**  
Defines a cylindrical coordinate system by reference to three grid points. These points must be defined in coordinate systems whose definition does not involve the coordinate system being defined. The first point is the origin, the second lies on the z-axis, and the third lies in the plane of the azimuthal origin.

**Format and Example:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORD1C</td>
<td>CID</td>
<td>G1</td>
<td>G2</td>
<td>G3</td>
<td>CID</td>
<td>G1</td>
<td>G2</td>
<td>G3</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>CID</td>
<td>Coordinate system identification number (Integer &gt; 0)</td>
</tr>
<tr>
<td>G1</td>
<td>Grid point identification number (Integer &gt; 0; G1 ≠ G2 ≠ G3).</td>
</tr>
</tbody>
</table>

**Remarks:**

1. Coordinate system identification numbers on all CORD1R, CORD1C, CORD1S, CORD2R, CORD2C, and CORD2S entries must be unique.
2. The three points G1, G2, and G3 must be noncollinear.
3. The location of a grid point (P in the sketch) in this coordinate system is given by \((R, \theta, Z)\) where \(\theta\) is measured in degrees.
4. The displacement coordinate directions at P are dependent on the location of P as shown above by \((u_r, u_\theta, u_z)\).
5. Points on the z-axis may not have their displacement directions defined in this coordinate system since an ambiguity results.
6. One or two coordinate systems may be defined on a single entry.
Input Data Entry: CORD1R Rectangular Coordinate System Definition, Form 1.

Description: Defines a rectangular coordinate system by reference to three grid points. These points must be defined in coordinate systems whose definition does not involve the coordinate systems defined. The first point is the origin, the second lies on the z-axis, and the third lies in the x-z plane.

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORD1R</td>
<td>CID</td>
<td>G1</td>
<td>G2</td>
<td>G3</td>
<td>CID</td>
<td>G1</td>
<td>G2</td>
<td>G3</td>
<td></td>
</tr>
</tbody>
</table>

| CORD1R | 3 | 16 | 32 | 19 |    |    |    |

Field Contents

<table>
<thead>
<tr>
<th>CID</th>
<th>Coordinate system identification number (Integer &gt; 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gi</td>
<td>Grid point identification number (Integer &gt; 0; G1 ≠ G2 ≠ G3).</td>
</tr>
</tbody>
</table>

Remarks:
1. Coordinate system identification numbers on all CORD1R, CORD1C, CORD1S, CORD2R, CORD2C, and CORD2S entries must be unique.
2. The three points G1, G2, and G3 must be noncollinear.
3. The location of a grid point (P in the sketch) in this coordinate system is given by (X, Y, Z).
4. The displacement coordinate directions at P are shown above by (ux, uy, uz).
5. One or two coordinate systems may be defined on a single entry.
Input Data Entry: CORD1S  Spherical Coordinate System Definition, Form 1.

Description: Defines a spherical coordinate system by reference to three grid points. These points must be defined in coordinate systems whose definition does not involve the coordinate systems defined. The first point is the origin, the second lies on the z-axis, and the third lies in the plane of the azimuthal origin.

Format and Examples:

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CORD1S</td>
<td>CID</td>
<td>G1</td>
<td>G2</td>
<td>G3</td>
<td>CID</td>
<td>G1</td>
<td>G2</td>
<td>G3</td>
</tr>
</tbody>
</table>

Field Contents

CID  Coordinate system identification number (Integer > 0)

G1  Grid point identification number (Integer > 0; G1 ≠ G2 ≠ G3).

Remarks:

1. Coordinate system identification numbers on all CORD1R, CORD1C, CORD1S, CORD2R, CORD2C, and CORD2S entries must be unique.
2. The three points G1, G2, and G3 must be noncollinear.
3. The location of a grid point (P in the sketch) in this coordinate system is given by (R, θ, φ) where θ and φ are measured in degrees.
4. The displacement coordinate directions at P are dependent on the locations of P as shown above by (u_r, u_φ, u_θ).
5. Points in the polar axis may not have their displacement direction defined in this coordinate system since an ambiguity results.
6. One or two coordinate systems may be defined on a single entry.
**Input Data Entry:** CORD2C  Cylindrical Coordinate System Definition, Form 2.

**Description:** Defines a cylindrical coordinate system by reference to the coordinates of three grid points. The first point defines the origin. The second point defines the direction of the z-axis. The third lies in the plane of the azimuthal origin. The reference coordinate system must be independently defined.

**Format and Example:**

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>CID</td>
<td>Coordinate system identification number (Integer &gt; 0)</td>
</tr>
<tr>
<td>RID</td>
<td>Reference to a coordinate system which is defined independently of new coordinate system (Integer ≥ 0, or blank)</td>
</tr>
<tr>
<td>A1, B1, C1</td>
<td>Coordinates of three points in coordinate system, defined in Field 3 (Real)</td>
</tr>
</tbody>
</table>

**Remarks:**

1. Continuation entry must be present.
2. The three points (A1, A2, A3), (B1, B2, B3), (C1, C2, C3) must be unique and noncollinear. Noncollinearity is checked by the geometry processor.
3. Coordinate system identification numbers on all CORD1R, CORD1C, CORD1S, CORD2R, CORD2C, and CORD2S entries must all be unique.
4. An RID of zero references the basic ordinate system.

5. The location of a grid point (P in the sketch) in this coordinate is given by \((R, \theta, Z)\) where \(\theta\) is measured in degrees.

6. The displacement coordinate directions at P are dependent on the location of P as shown above by \((u_r, u_\theta, u_z)\).

7. Points on the z-axis may not have their displacement direction defined in this coordinate system since an ambiguity results.
Input Data Entry: CORD2R Rectangular Coordinate System Definition, Form 2.

Description: Defines a rectangular coordinate system by reference to coordinates of three points. The first point defines the origin. The second defines the direction of the z-axis. The third point defines a vector which, with the z-axis, defines the x-z plane. The reference coordinate system must be independently defined.

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORD2R</td>
<td>CID</td>
<td>RID</td>
<td>A1</td>
<td>A2</td>
<td>A3</td>
<td>B1</td>
<td>B2</td>
<td>B3</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>C1</td>
<td>C2</td>
<td>C3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CORD2R 3 17 -2.9 1.0 0.0 3.6 0.0 1.0 123
+23 5.2 1.0 -2.9

Field Contents

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>CID</td>
<td>Coordinate system identification number (Integer &gt; 0)</td>
</tr>
<tr>
<td>RID</td>
<td>Reference to a coordinate system which is defined independently of new coordinate system (Integer ≥ 0, or blank)</td>
</tr>
<tr>
<td>Ai, Bi, Ci</td>
<td>Coordinates of three points in coordinate system defined in Field 3 (Real)</td>
</tr>
</tbody>
</table>

Remarks:

1. The continuation entry must be present.

2. The three points (A1, A2, A3), (B1, B2, B3), (C1, C2, C3) must be unique and noncollinear. Noncollinearity is checked by the geometry processor.

3. Coordinate system identification numbers on all CORD1R, CORD1C, CORD1S, CORD2R, CORD2C, and CORD2S entries must all be unique.

4. An RID of zero references the basic coordinate system.

5. The location of a grid point (P in the sketch) in this coordinate system is given by (X, Y, Z)

6. The displacement coordinate directions at P are shown by (ux, uy, uz)
Input Data Entry: CORD2S  Spherical Coordinate System Definition, Form 2.

Description: Defines a spherical coordinate system by reference to the coordinates of three points. The first point defines the origin. The second point defines the direction of the z-axis. The third lies in the plane of the azimuthal origin. The reference coordinate system must be independently defined.

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORD2S</td>
<td>CID</td>
<td>RID</td>
<td>A1</td>
<td>A2</td>
<td>A3</td>
<td>B1</td>
<td>B2</td>
<td>B3</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>C1</td>
<td>C2</td>
<td>C3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CORD2S 3 17 -2.9 1.0 0.0 3.6 0.0 1.0 123
+23 5.2 1.0 -2.9

Field Contents

CID  Coordinate system identification number (Integer > 0)
RID  Reference to a coordinate system which is defined independently of the new coordinate system (Integer ≥ 0, or blank)
Ai,Bi,Ci Coordinates of three points in coordinate system defined in Field 3 (Real)

Remarks:
1. The continuation entry must be present.
2. The three points (A1, A2, A3), (B1, B2, B3), (C1, C2, C3) must be unique and noncollinear.
3. Coordinate system identification numbers on all CORD1R, CORD1C, CORD1S, CORD2R, CORD2C and CORD2S entries must all be unique.
4. An RID of zero references the basic coordinate system.
5. The location of a grid point (P in the sketch) in this coordinate system is given by \((R, \theta, \varphi)\) where \(\theta\) and \(\varphi\) are measured in degrees.

6. The displacement coordinate directions at P are shown above by \((u_r, u_\theta, u_\varphi)\).

7. Points on the polar axis may not have their displacement directions defined in this coordinate system since an ambiguity results.
Input Data Entry: CQDMEM1 Isoparametric Quadrilateral Element Connection

Description: Defines the isoparametric quadrilateral membrane element.

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CQDMEM1</td>
<td>EID</td>
<td>PID</td>
<td>G1</td>
<td>G2</td>
<td>G3</td>
<td>G4</td>
<td>TH</td>
<td>TMAX</td>
<td></td>
</tr>
<tr>
<td>CQDMEM1</td>
<td>72</td>
<td>13</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>29.2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

- **EID**: Element identification number (Integer > 0).
- **PID**: Identification number of a PQDMEM1 or PCOMP property entry (Default is EID) (Integer > 0).
- **Gi**: Grid point identification numbers of connection points (Integer > 0)
- **TH**: Material property orientation angle. If TH is real, the sketch below gives the sign convention for TH. If TH is an integer, the material x-axis is along the x-axis of coordinate system identified by the integer.
- **TMAX**: Maximum allowable element thickness in design (Real > 0.0 or blank). (Default=10^4)

Remarks:

1. Grid points G1 through G4 must be ordered consecutively around the perimeter of the element as shown in the figure below.

2. All interior angles must be less than 180°.

3. TMAX is ignored unless element is linked to global design variable by a SHAPE entry.
Input Data Entry:  CQUAD4 Quadrilateral Element Connection

Description:  Quadrilateral plate element (QUAD4) of the structural model. This is an isoparametric membrane-bending element.

Format and Example:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CQUAD4</td>
<td>EID</td>
<td>PID</td>
<td>G1</td>
<td>G2</td>
<td>G3</td>
<td>G4</td>
<td>TM</td>
<td>ZOFF</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td></td>
<td></td>
<td>T1</td>
<td>T2</td>
<td>T3</td>
<td>T4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CQUAD4</td>
<td>101</td>
<td>17</td>
<td>1001</td>
<td>1005</td>
<td>1010</td>
<td>1024</td>
<td>45.0</td>
<td>0.01</td>
<td>ABC</td>
<td></td>
</tr>
<tr>
<td>+BC</td>
<td></td>
<td></td>
<td>0.03</td>
<td>0.125</td>
<td>0.05</td>
<td>0.04</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

- EID: Element identification number (Integer > 0)
- PID: Identification number of a PSHELL or PCOMP1 entry (Default is EID) (Integer > 0).
- Gi: Grid point identification numbers of connection points (Integer > 0).
- ZOFF: Offset of the element reference plane from the plane of grid points. A positive value means the +ze direction. (Real or blank, see Remark 2 for default).
- TM: Material property orientation specification (Real or blank; or 0 ≤ Integer < 1,000,000). If Real or blank, specifies the material property orientation angle in degrees. If Integer, the orientation of the material x-axis is along the projection onto the plane of the element of the x-axis of the coordinate system specified by the integer value.
- TMAX: Maximum allowable element thickness in design (Real > 0.0).
- Ti: Membrane thickness of element at grid points Gi (Real or blank, see Remark 3 for default).

Remarks:

1. The QUAD4 geometry, coordinate systems and numbering are shown in the figure below:
2. The material coordinate system (TM) and the offset (ZOFP) may also be provided on the PSHELL entry. The property data will be used if the corresponding field on the CQUAD4 entry is blank. The element reference plane is located at the mid-thickness of the element parallel to the element mean plane.

3. The Ti are optional, if not supplied they will be set to the value of T specified on the PSHELL entry. In such cases, the continuation entry is not required.

4. TMAX is ignored unless the element is linked to the global design variables by a SHAPE entry.
**Input Data Entry:**  CROD  Rod Element Connection

**Description:**  Defines a tension-compression-torsion element (ROD) of the structural model.

**Format and Examples:**

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>EID</td>
<td>Element identification number (Integer &gt; 0).</td>
</tr>
<tr>
<td>PID</td>
<td>Identification number of a PROD property entry (Default is EID) (Integer &gt; 0).</td>
</tr>
<tr>
<td>G1</td>
<td>Grid point identification numbers of connection points (Integer &gt; 0)</td>
</tr>
<tr>
<td>TMAX</td>
<td>Maximum allowable rod area in design (Real &gt; 0.0 or blank)</td>
</tr>
</tbody>
</table>

**Remarks:**

1. See CONROD for alternative method of rod definition.
2. Only one ROD element may be defined on a single entry.
3. TMAX is ignored unless the element is linked to global design variables by a SHAPE entry.
Input Data Entry: **CSHEAR**  
Shear Panel Element Connection

**Description:** Defines a shear panel element (SHEAR) of the structural model.

**Format and Example:**

```
   1   2   3   4   5   6   7   8   9   10
CSHEAR  EID  PID  G1  G2  G3  G4  TMAX
CSHEAR  3   6   1   5   3   7
```

**Field** | **Contents**
---|---
EID | Element identification number (Integer > 0).
PID | Identification number of a PSHEAR property entry (Default is PID) (Integer > 0).
G\(i\) | Grid point identification numbers of connection points (Integer > 0)
TMAX | Maximum allowable thickness in design (Real > 0.0, or blank).

**Remarks:**
1. Grid points \(G_1\) through \(G_4\) must be ordered consecutively around the perimeter of the element.
2. All interior angles must be less than 180°.
3. \(T_{MAX}\) is ignored unless element is linked to global design variable by a SHAPE entry.
**Input Data Entry:**  
**CTRIA3** Triangular Element Connection

**Description:** Defines a triangular shell element (TRIA3) of the structural model.

**Format and Example:**

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CTRIA3</td>
<td>EID</td>
<td>PID</td>
<td>G1</td>
<td>G2</td>
<td>G3</td>
<td>TM</td>
<td>ZOFF</td>
<td>abc</td>
</tr>
<tr>
<td>+BC</td>
<td>TMAX</td>
<td>T2</td>
<td>T3</td>
<td>T4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| CTRIA3 | 101 | 17 | 1001 | 1005 | 1010 | 45.0 | 0.01 | ABC |
| +BC | 0.03 | 0.125 | 0.05 |

**Field**

<table>
<thead>
<tr>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EID</strong></td>
</tr>
<tr>
<td><strong>PID</strong></td>
</tr>
<tr>
<td><strong>G1</strong></td>
</tr>
<tr>
<td><strong>ZOFF</strong></td>
</tr>
<tr>
<td><strong>TM</strong></td>
</tr>
<tr>
<td><strong>TMAX</strong></td>
</tr>
<tr>
<td><strong>Ti</strong></td>
</tr>
</tbody>
</table>

**Remarks:**

1. The TRIA3 geometry, coordinate systems and numbering are shown in the figure below:

![Diagram of TRIA3 element](https://via.placeholder.com/150)

2. The material coordinate system (TM) and the offset (ZOFF) may also be provided on the PSHELL entry. The property data will be used if the corresponding field on the CTRIA3 entry is blank. The element reference plane is located at the mid-thickness of the element parallel to the element mean plane.

205
3. The $T_1$ are optional, if not supplied they will be set to the value of $T$ specified on the PHELL entry. In such cases, the continuation entry is not required.

4. $T_{\text{MAX}}$ is ignored unless the element is linked to the global design variables by a SHAPE entry.
Input Data Entry:  CTRMEM

Description:  Defines a triangular membrane element.

Format and Examples:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTRMEM</td>
<td>100</td>
<td>500</td>
<td>1</td>
<td>7</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field          | Contents
---             | ---
EID            | Element identification number (Integer > 0).
PID            | Identification of PTRMEM or PCOMP entry (Integer > 0) Default = EID.
G1             | Grid point identifications of connection points (Integer > 0).
THETA          | Material orientation angle (Real) or 0 < Integer < 1,000,000. If integer, then material x-axis lies along x-axis of coordinate system identified by the integer.
TMAX           | Maximum allowable thickness in design. (Real ≥ 0., Default = 10^4)

Remarks:

1. The TMAX value is used only for shape function design variable linking.
Input Data Entry: DCONALE

Description: Defines an aileron effectiveness constraint of the form:

\[ AE \leq \text{AEREQ} \] (upper bound) or \[ AE \leq \text{AEREQ} \] (lower bound)

where,

\[ AE = \frac{-C_{a_e}}{C_{a_e}} \]

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>DCONALE</td>
<td>SID</td>
<td>LABEL</td>
<td>CTYPE</td>
<td>AEREQ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCONALE</td>
<td>25</td>
<td>OUTBAIL</td>
<td>LOWER</td>
<td>0.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>Aerodynamic set identification for the imposed constraint (Integer &gt; 0)</td>
</tr>
<tr>
<td>LABEL</td>
<td>A unique alphanumeric string of up to eight characters to identify the AESURF or CONLINK control surface</td>
</tr>
<tr>
<td>CTYPE</td>
<td>Constraint type: either UPPER for upper bound or LOWER for lower bound (Character. Default = LCNER)</td>
</tr>
<tr>
<td>AEREQ</td>
<td>Required aileron effectiveness (Real ≠ 0.0)</td>
</tr>
</tbody>
</table>

Remarks:

1. This constraint will only be applied if selected by the Solution Control discipline option DCON-SID and if an antisymmetric aeroelastic trim analysis is being performed.

2. A LOWER bound constraint excludes all values to the left of AEREQ on a real number line, while an UPPER bound constraint excludes all values to the right, irrespective of the sign of AEREQ.

3. The effectiveness in roll of multiple control surfaces may be specified using multiple DCONALE entries with one constraint generated for each LABEL/CTYPE combination.
Input Data Entry: DCONCLA

Description: Defines a flexible lift curve slope constraint of the form:

\[ \text{CLA} \leq \text{CLAREQ} \text{ or } \text{CLA} \geq \text{CLAREQ} \]

where,

\[ \text{CLA} = \frac{(C_{\omega l})}{(C_{\omega r})} \]

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONCLA</td>
<td>SID</td>
<td>CTYPE</td>
<td>CLAREQ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

- **SID**: Aerodynamic set identification for the imposed constraint (Integer > 0)
- **CTYPE**: Constraint type: either **UPPER** for upper bound or **LOWER** for lower bound (Text, Default = **LOWER**)
- **CLAREQ**: Required flexible-to-rigid lift curve slope (Real \( \neq 0.0 \))

Remarks:

1. Displacement constraints are selected in Solution Control with the discipline option: **DCON=SID**

2. A **LOWER** bound constraint excludes all values to the left of **CLAREQ** on a real number line, while an **UPPER** bound constraint excludes all values to the right, irrespective of the sign of **CLAREQ**.
Input Data Entry: DCONDSP

Description: Defines a deflection constraint of the form:

\[ \sum_j A_j u_j \leq \delta \text{ (UPPER bound) or } \sum_j A_j u_j \geq \delta \text{ (LOWER bound)} \]

Format and Example:

<table>
<thead>
<tr>
<th>DCONDSP</th>
<th>CTSET</th>
<th>DCID</th>
<th>CTYPE</th>
<th>DALL</th>
<th>LABEL</th>
<th>G</th>
<th>C</th>
<th>A</th>
<th>CONT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONT</td>
<td>G</td>
<td>C</td>
<td>A</td>
<td>G</td>
<td>C</td>
<td>A</td>
<td></td>
<td></td>
<td>-etc-</td>
</tr>
<tr>
<td>DCONDSP</td>
<td>1</td>
<td>10</td>
<td>LOWER</td>
<td>-2.3</td>
<td>TIP</td>
<td>32</td>
<td>3</td>
<td>2.0</td>
<td>ABC</td>
</tr>
<tr>
<td>+BC</td>
<td>7</td>
<td>3</td>
<td>-4.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

- **CTSET**: Constraint set identification number (Integer > 0)
- **DCID**: Constraint identification number (Integer > 0)
- **CTYPE**: Constraint type, either **UPPER** or **LOWER** bound (Text, Default = **UPPER**)
- **DALL**: Allowable displacement (Real)
- **LABEL**: User specified label to identify constraint (Text)
- **G**: Grid identification (Integer > 0)
- **C**: Component number—any one of digits 1 through 6
- **A**: Real coefficient (Real \( \neq 0.0 \))

Remarks:

1. Displacement constraints are selected in Solution Control with the discipline option:

   \[ \text{DCON} = \text{CTSET} \]

   The **CTSET** is the constraint set identification number and **DCID** is an arbitrary constraint identifier supplied by the user. All **DCONDSP** that share the same **CTSET** and **DCID** will form one constraint equation.

2. Both upper and lower bounds on the deflections can be specified by this entry. For example, if constraints of the form \(|u| \leq 2.0\) are to be imposed, one **DCONDSP** entry would use **CTYPE** = **UPPER**, **DALL** = 2.0, **G** = 32, **C** = 3, **A** = 1.0 while a second entry would use **CTYPE** = **LOWER**, **DALL** = -2.0, **G** = 32, **C** = 3, **A** = 1.0.

3. Twist constraints can be specified by differencing two displacements while **LOWER** bounds can be expressed as a weighted sum of three displacements.

4. Any number of continuation entries are permitted.

5. A **LOWER** bound constraint excludes all values to the left of **DALL** on a real number line, while an **UPPER** bound constraint excludes all values to the right, irrespective of the sign of **DALL**.

210
Input Data Entry  DCONEP  Principal Strain Constraint Definition

Description: Defines a principal strain constraint by specifying the identification numbers of constrained elements.

Format and Example:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONEP</td>
<td>SID</td>
<td>ST</td>
<td>SC</td>
<td>SS</td>
<td>ETYPE</td>
<td>LAYRNUM</td>
<td>EID1</td>
<td>EID2</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CON</td>
<td></td>
<td>EID3</td>
<td>EID4</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| DCONEP | 100 | 1.-2 | -1.-2 | 1.-2 | BAR | 101 | 102 | ABC |
| +BC | 107 | 108 | 142 |

Alternate Form:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONEP</td>
<td>SID</td>
<td>ST</td>
<td>SC</td>
<td>SS</td>
<td>ETYPE</td>
<td>LAYRNUM</td>
<td>EID1</td>
<td>THRU</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CON</td>
<td></td>
<td>EID2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| DCONEP | 100 | 1.-2 | -1.-2 | 1.-2 | BAR | 101 | 102 | ABC |
| +BC | 107 | 108 | 142 |

Field  Contents

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>Strain constraint set identification (Integer &gt; 0)</td>
</tr>
<tr>
<td>ST</td>
<td>Principal strain limit in tension (Real &gt; 0.0)</td>
</tr>
<tr>
<td>SS</td>
<td>Principal strain limit in compression (Real, Default = ST)</td>
</tr>
<tr>
<td>ETYPE</td>
<td>Element type (Text)</td>
</tr>
<tr>
<td>LAYRNUM</td>
<td>Layer number of a composite element (Integer &gt; 0 or Blank)</td>
</tr>
<tr>
<td>EID1</td>
<td>Element identification numbers (Integer &gt; 0)</td>
</tr>
</tbody>
</table>

Remarks:

1. Strain constraints are selected in Solution Control with the discipline option:

   \[ \text{STRAIN} = \text{sid} \]

2. ETYPE may be selected from BAR, Q425, QUAD, SHEAR, TRIA3, or TRUKH.

3. If the alternate form is used, EID2 must be greater than or equal to EID1. Elements in the range which do not exist are ignored.

4. The shear strain limit, SS, is used only with the SHEAR element.

5. The strain limit for compression, SC, is always treated as a negative value regardless of the sign of the input value.

6. LAYRNUM is only used if the element is composed of a composite material defined with PCOMP Bulk Data entries.
Input Data Entry  DCONEMP  Principal Strain Constraint Definition

Description:  Defines a principal strain constraint by specifying material identification numbers.

Format and Example:

<table>
<thead>
<tr>
<th>DCONEMP</th>
<th>SID</th>
<th>ST</th>
<th>SC</th>
<th>SS</th>
<th>MID1</th>
<th>MID2</th>
<th>MID3</th>
<th>MID4</th>
<th>CONT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONT</td>
<td></td>
<td>MID5</td>
<td>MID6</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DCONEPM 100 1.2 1.2 1.2 8888 9999 1 99 ABC
+BC 111 123

Alternate Form:

<table>
<thead>
<tr>
<th>DCONEMP</th>
<th>SID</th>
<th>ST</th>
<th>SC</th>
<th>SS</th>
<th>MAT1</th>
<th>THRU</th>
<th>MAT2</th>
</tr>
</thead>
</table>

Field  Contents
SID  Strain constraint set identification (Integer > 0)
ST  Principal strain limit in tension (Real > 0.0)
SC  Principal strain limit in compression (Real, Default = ST)
SS  Principal strain limit in shear (Real > 0.0)
MID1  Material identification numbers (Integer > 0)

Remarks:
1. Strain constraints are selected in Solution Control with the discipline option:
   \[ \text{STRAIN} = \text{sid} \]
2. If the alternate form is used, MID2 must be greater than or equal to MID1. Material properties in the range which do not exist are ignored.
3. The shear strain limit, SS, is used only with the SHEAR element.
4. The strain limit for compression, SC, is always treated as a negative value regardless of the sign of the input value.
**Input Data Entry**

**DCONEPP**  Principal Strain Constraint Definition

**Description:** Defines a principal strain constraint by specifying element property identification numbers.

**Format and Example:**

```
<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONEPP</td>
<td>SID</td>
<td>ST</td>
<td>SC</td>
<td>SS</td>
<td>PTYPE</td>
<td>LAYRNUM</td>
<td>PID1</td>
<td>PID2</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>PID3</td>
<td>PID4</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

```
DCONEPP 100 1.-2 -1.-2 1.-2 PBAR 100 200 ABC
+BC 300 400 500
```

**Alternate Form:**

```
<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONEPP</td>
<td>SID</td>
<td>ST</td>
<td>SC</td>
<td>SS</td>
<td>PTYPE</td>
<td>LAYRNUM</td>
<td>PID1</td>
<td>THRU</td>
<td>ABC</td>
</tr>
<tr>
<td>+BC</td>
<td>PID2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**Field Contents**

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>Strain constraint set identification (Integer &gt; 0)</td>
</tr>
<tr>
<td>ST</td>
<td>Principal strain limit in tension (Real &gt; 0.0)</td>
</tr>
<tr>
<td>SC</td>
<td>Principal strain limit in compression (Real, Default = ST)</td>
</tr>
<tr>
<td>SS</td>
<td>Principal strain limit in shear (Real &gt; 0.0)</td>
</tr>
<tr>
<td>PTYPE</td>
<td>Property type (Text)</td>
</tr>
<tr>
<td>LAYRNUM</td>
<td>Layer number of a composite element (Integer &gt; 0 or Blank)</td>
</tr>
<tr>
<td>PID1</td>
<td>Property identification numbers (Integer &gt; 0)</td>
</tr>
</tbody>
</table>

**Remarks:**

1. Strain constraints are selected in Solution Control with the discipline option: `STRAIN=sid`

2. `PTYPE` may be selected from `PBAR`, `PCOMP`, `PCOMP1`, `PCOMP2`, `PODMEM1`, `PROD`, `PSHEAR`, `PSHELL`, or `PODMEM`.

3. If the alternate form is used, `PID2` must be greater than or equal to `PID1`. Property identification numbers in the range which do not exist are ignored.

4. The shear strain limit, `SS`, is used only with the `SHEAR` element.

5. The strain limit for compression, `SC`, is always treated as a negative value regardless of the sign of the input value.

6. `LAYRNUM` is only used if the element is composed of a composite material defined with `PCOMP` Bulk Data entries.
**Input Data Entry:** DCONFLT Flutter Constraint Definition

**Description:** Defines a flutter constraint in the form of a table:

\[
\frac{Y - Y_{REF}}{GFAC} \leq 0.0
\]

**Format and Example:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONFLT</td>
<td>SID</td>
<td>VTYPE</td>
<td>GFAC</td>
<td>V1</td>
<td>GAM1</td>
<td>V2</td>
<td>GAM2</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>V3</td>
<td>GAM3</td>
<td>V4</td>
<td>GAM4</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCONFLT</td>
<td>100</td>
<td>EQUIV</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>35.</td>
<td>0.05</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field Contents**

- **SID**: Constraint set identification, the constraints are referenced by the design constraint id in Solution Control (Integer > 0)
- **VTYPE**: Nature of the velocity referred to in the table. Either TRUE for true velocity or EQUIV for equivalent air speed. Default = TRUE.
- **GFAC**: Constraint scaling factor (Real > 0.0, Default = 0.10)
- **Vi**: Velocity value (Real ≥ 0.0)
- **GAMi**: Required damping value (Real)

**Remarks:**

1. Flutter constraints are selected in Solution Control with the discipline option:
   
   FLCOND=SID

2. A negative value of GAMi refers to a stable system.

3. The Vi must be in either ascending or descending order.

4. Linear interpolation is used to determine GAMA for a given velocity.

5. At least two pairs must be entered.

6. Jumps between two points (Vi = Vi+1) are allowed, but not at the end points. If the jump point is used, the average of the two GAMi will be returned.
Input Data Entry:  DCONFQ

Description:  Defines a frequency constraint of the form:

\[ f \leq f_{\text{all}} \text{ or } f \geq f_{\text{all}} \]

Format and Example:

```
1 2 3 4 5 6 7 8 9 10
DCONFQ SID MODE CTYPE FRQALL
```

```
DCONFQ 3 1 LOWER 6.0
```

Field          Contents

- **SID**: Constraint set identification (Integer > 0)
- **MODE**: Modal number of the frequency to be constrained (Integer > 0)
- **CTYPE**: Constraint type: either **UPPER** for upper bound or **LOWER** for lower bound (Text, Default = LOWER)
- **FRQALL**: Frequency constraint (in Hz.).

Remarks:

1. More than one constraint can be placed on a mode allowing specification of pseudo-equality constraints.
**Input Data Entry**  
**DCONFT**  
**Fiber/Transverse Strain Constraint Definition**

**Description:** Defines fiber/transverse strain constraints for composite elements by specifying the identification numbers of constrained elements.

**Format and Example:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONFT</td>
<td>SID</td>
<td>EFT</td>
<td>EFC</td>
<td>ETT</td>
<td>ETC</td>
<td>ETYPE</td>
<td>LAYRNUM</td>
<td>EID1</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>EID2</td>
<td>EID3</td>
<td>etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| DCONFT | 100 | 1.2 | -1.2 | 1.3 | -1.3 | QUAD4 | 1   | 101  | ABC |
| +BC    | 102 | 110 |       |     |     |       |     |      |     |

**Alternate Form:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONFT</td>
<td>SID</td>
<td>EFT</td>
<td>EFC</td>
<td>ETT</td>
<td>ETC</td>
<td>ETYPE</td>
<td>LAYRNUM</td>
<td>EID1</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>THRU</td>
<td>EID2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field**  
**Contents**

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>Strain constraint set identification (Integer &gt; 0)</td>
</tr>
<tr>
<td>EFT</td>
<td>Tensile strain limit in the fiber direction (Real &gt; 0.0)</td>
</tr>
<tr>
<td>EFC</td>
<td>Compressive strain limit in the fiber direction (Real, Default = EFT)</td>
</tr>
<tr>
<td>ETT</td>
<td>Tensile strain limit in the transverse direction (Real &gt; 0.0)</td>
</tr>
<tr>
<td>ETC</td>
<td>Compressive strain limit in the transverse direction (Real, Default = ETT)</td>
</tr>
<tr>
<td>ETYPE</td>
<td>Element type (Text)</td>
</tr>
<tr>
<td>LAYRNUM</td>
<td>The layer number of a composite element (Integer &gt; 0, or blank)</td>
</tr>
<tr>
<td>EID1</td>
<td>Element identification numbers (Integer &gt; 0)</td>
</tr>
</tbody>
</table>

**Remarks:**

1. Strain constraints are selected in Solution Control with the discipline option:  
   
   ```
   STRAIN=sid
   ```

2. Fiber/transverse strain constraints may only be applied to elements defined using composite materials.

3. **ETYPE** may be selected from **QUMEM1, QUAD4, TRIA3, or T3MEM.**

4. If the alternate form is used, **EID2** must be greater than or equal to **EID1.** Elements in the range which do not exist are ignored.

4. The strain limits for compression, **EFC** and **ETC,** are always treated as negative values regardless of the signs of the input values.
**Input Data Entry**

**DCONFTM**  
Fiber/Transverse Strain Constraint Definition

**Description:** Defines fiber/transverse strain constraints for composite elements by specifying material identification numbers.

**Format and Example:**

<table>
<thead>
<tr>
<th>DCONFTM</th>
<th>SID</th>
<th>EFT</th>
<th>EFC</th>
<th>ETT</th>
<th>ETC</th>
<th>MID1</th>
<th>MID2</th>
<th>MID3</th>
<th>CONT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONT</td>
<td>MID4</td>
<td>MID5</td>
<td>etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DCONFTM 100 1.2 1.2 1.3 -1.3 11 16 101 ABC

**Alternate Form:**

<table>
<thead>
<tr>
<th>DCONFTM</th>
<th>SID</th>
<th>EFT</th>
<th>EFC</th>
<th>ETT</th>
<th>ETC</th>
<th>MID1</th>
<th>THRU</th>
<th>MID2</th>
</tr>
</thead>
<tbody>
<tr>
<td>+BC</td>
<td>19</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field Contents**

- **SID**: Strain constraint set identification (Integer > 0)
- **EFT**: Tensile strain limit in the fiber direction (Real > 0.0).
- **EFC**: Compressive strain limit in the fiber direction (Real, Default = EFT).
- **ETT**: Tensile strain limit in the transverse direction (Real > 0.0).
- **ETC**: Compressive strain limit in the transverse direction (Real, Default = ETT).
- **MIDi**: Material identification numbers (Integer > 0).

**Remarks:**

1. Strain constraints are selected in Solution Control with the discipline option:

   `STRAIN=sid`

2. Fiber/transverse strain constraints may only be applied to elements defined using composite materials.

3. If the alternate form is used, MID2 must be greater than or equal to MID1. Material properties in the range which do not exist are ignored.

4. The strain limits for compression, EFC and ETC, are always treated as negative values regardless of the signs of the input values.
Input Data Entry  
DCONFTP  
Fiber/Transverse Strain Constraint Definition

Description:  Defines fiber/transverse strain constraints for composite elements by specifying property identification numbers.

Format and Example:

1 2 3 4 5 6 7 8 9 10
DCONFTP SID EFT EFC ETT ETC PTYPE LAYRNUM PID1 CONT
CONT PID2 PID3 -etc-

DCONFTP 100 1.-2 1.-2 2.-3 3.-3 PCOMP 2 100 CONT
CONT 110 120

Alternate Form:

1 2 3 4 5 7 8 9 10
DCONFTP SID EFT EFC ETT ETC PTYPE LAYRNUM PID1 CONT
CONT THRU PID2

Field  
Contents
SID  Strain constraint set identification (Integer > 0).
EFT  Tensile strain limit in the fiber direction (Real > 0.0)
EFC  Compressive strain limit in the fiber direction (Real, Default = EFT)
ETT  Tensile strain limit in the transverse direction (Real > 0.0).
ETC  Compressive strain limit in the transverse direction (Real, Default = ETT)
PTYPE  Property type (Text)
LAYRNUM  The layer number of a composite element (Integer > 0 or blank)
PIDi  Property identification numbers (Integer > 0)

Remarks:
1. Strain constraints are selected in Solution Control with the discipline option:
   STRAIN=sid

2. PTYPE may be selected from PCOMP, PCOMP1, and PCOMP2.
3. Fiber/transverse strain constraints may only be applied to elements defined using composite materials.
4. If the alternate form is used, PID2 must be greater than or equal to PID1. Properties in the range which do not exist are ignored.
5. The strain limits for compression, EFC and ETC, are always treated as negative values regardless of the signs of the input values.
Input Data Entry:  DCONLAM  Composite laminate composition constraint.

Description:  Defines a constraint on the relative thickness of a ply that is part of a laminate. The constraint is of the form:

\[
\frac{\%_{req}}{100} - \frac{t_{ply}}{t_{lam}} \leq 0 \quad \text{(lower bound)}
\]

\[
\frac{t_{ply}}{t_{lam}} - \frac{\%_{req}}{100} \leq 0 \quad \text{(upper bound)}
\]

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONLAM</td>
<td>CTYPE</td>
<td>%REQ</td>
<td>PLYNUM</td>
<td>PLYSET</td>
<td>LAM</td>
<td>SID</td>
<td>SID</td>
<td>SID</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>SID</td>
<td>SID</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| DCONLAM | UPPER | 40.0 | 100 | ALL | 1000 | 1001 |

Field                  | Contents
-----------------------|-------------------------------------------------------------
CTYPE                  | Constraint type: either UPPER for upper bound or LOWER for lower bound. (Text, Default = UPPER)
%REQ                   | Minimum (lower bound) or maximum (upper bound) PERCENTAGE (0.0 to 100.0) of the total laminate thickness that is to be made up of the ply thickness. (see Remark 2) (Real > 0.0)
PLYNUM                 | Single ply number (numbered in the order used on the PCOMP entries) that constitutes the ply thickness. Only one of PLYNUM or PLYSET may be used. (Integer > 0 or blank)
PLYSET                 | Set identification number of one or more PLYLIST bulk data entries naming a set of plies whose summed thicknesses constitute the ply thickness in the constraint. Only one of PLYNUM or PLYSET may be used. (Integer > 0 or blank)
LAM                    | The character string ALL or the set identification number of one or more PLYLIST entries naming a set of plies whose summed thicknesses constitute the laminate thickness in the constraint. If ALL, the laminate is defined to be all the layers on the PCOMP entries that define the laminate thickness. (Character = ALL or Integer > 0, Default = ALL)
SID                    | Set identification of one or more ELEMSET entries that define the set of composite elements to which this composition constraint will be applied. (Integer > 0 or blank)

Remarks:

1. One and only one of either PLYNUM or PLYSET must be given.
2. The definition of ply and laminate thickness can vary from entry to entry. If PLYNUM is used to define \( t_{ply} \) that one layer constitutes a ply; otherwise \( t_{ply} \) is the sum of the layer thicknesses of all the layers listed in PLYSET.
Similar rules are applied for tlam. If ALL is used, every layer of the element is used to compute tlam (including undesigned layers—see Remark 3); otherwise the summed thicknesses of the layers specified by the PLYLIST set will be used. As a result, there is no real distinction between a ply thickness and a laminate thickness. Typically, the ply will be a subset of the layers that define the laminate, but that is not a requirement.

3. If this constraint is applied to a composite element with undesigned layers, these layers may be freely included in the layer(s) composing the ply and/or the layer(s) composing the laminate. The only restriction is that at least one layer in the ply must be a local design variable and at least one layer in the laminate must be a local design variable.
Input Data Entry: **DCONLNM**  Composite laminate minimum gauge constraint.

**Description:** Defines a lower bound constraint on the total thickness of all or part of the layers of a composite element. The constraint is of the form:

\[
1.0 - \frac{t_{\text{lam}}}{t_{\text{min}}} \leq 0
\]

**Format and Example:**

```
1 2 3 4 5 6 7 8 9 10
DCONLN  MINTHK  LAM  SID  SID  SID  SID  SID  SID  SID  CONT
CONT    SID  SID  -etc-
```

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINTHK</td>
<td>Minimum laminate thickness. (Real &gt; 0.0, Default = 10^{-4})</td>
</tr>
<tr>
<td>LAM</td>
<td>The character string <strong>ALL</strong> or the set identification number of one or more <strong>PLYLIST</strong> entries naming a set of plies whose <em>summed</em> thicknesses constitute the laminate thickness in the constraint. If <strong>ALL</strong>, the laminate is defined to be all the layers on the <strong>PCOMP</strong>s of the elements selected by <strong>SID</strong>. (Character = <strong>ALL</strong> or Integer &gt; 0, Default = <strong>ALL</strong>)</td>
</tr>
<tr>
<td>SID</td>
<td>Set identification of one or more <strong>ELEM</strong> entries that define the set of composite elements to which this composition constraint will be applied. (Integer &gt; 0 or blank)</td>
</tr>
</tbody>
</table>

**Remarks:**

1. Because of the generality of the definition of the laminate, there is no real distinction between the **DCONLNM** and the **DCONPMN** constraints. Only the defaults are different to allow simple definitions of the common laminate in **DCONLNM** (**ALL**) or ply (**PLYNUM**) in **DCONPMN**.

2. The definition of laminate thickness can vary from entry to entry. If **ALL** is used, every layer of the element is used to compute \( t_{\text{lam}} \) (including undesigned layers—see Remark 3); otherwise the *summed* thicknesses of the layers specified by the **PLYLIST** set will be used.

3. If this constraint is applied to a composite element with undesigned layers, these layers may be freely included in the layer(s) composing the ply and/or the layer(s) composing the laminate. The only restriction is that at least one layer in the laminate must be a local design variable.

4. If the laminate is composed of a single layer, this constraint becomes redundant with the **TMIN** entered on the **PCOMP** field (for shape function linking) or the **VMIN** entered on the **DESELM** or **DESVARP** entry (for physical linking). In this case, the most critical limit will be determined from among all sources (**DCONPMN**, **DCONLNM**, **TMIN/VMIN**) and will be used to update the local variable side constraint. The **DCONxxx** entry will then be automatically removed since it will no longer be necessary. A summary of this action will be echoed to the print file.
Input Data Entry: DCONPMN Composite element ply minimum gauge constraint.

Description: Defines a lower bound constraint on the total thickness of all or part of the layers of a composite element. The constraint is of the form:

\[
1.0 - \frac{t_{\text{ply}}}{t_{\text{min}}} \leq 0
\]

Format and Example:

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINTHK</td>
<td>Minimum ply thickness. (Real &gt; 0.0, Default = 10^-4)</td>
</tr>
<tr>
<td>PLYNUM</td>
<td>Single ply number (numbered in the order used on the PCOMP1) that constitutes the ply thickness. Only one of PLYNUM or PLYSET may be used. (Integer &gt; 0 or blank)</td>
</tr>
<tr>
<td>PLYSET</td>
<td>Set identification number of one or more PLYLIST bulk data entries naming a set of plies whose summed thicknesses constitute the ply thickness in the constraint. Only one of PLYNUM or PLYSET may be used. (Integer &gt; 0 or blank)</td>
</tr>
<tr>
<td>SID</td>
<td>Set identification of one or more ELEMIST entries that define the set of composite elements to which this composition constraint will be applied. (Integer &gt; 0 or blank)</td>
</tr>
</tbody>
</table>

Remarks:

1. One and only one of either PLYNUM or PLYSET must be given.

2. Because of the generality of the definition of the ply, there is no real distinction between the DCONLMN and the DCONPMN constraints. Only the defaults are different to allow simple definitions of the common laminate in DCONLMN (ALL) or ply (PLYNUM) in DCONPMN.

3. The definition of ply thickness can vary from entry to entry. If PLYNUM is used to define \( t_{\text{ply}} \), that one layer constitutes a ply; otherwise \( t_{\text{ply}} \) is the sum of the layer thicknesses of all the layers listed in PLYSET.

4. If this constraint is applied to a composite element with undesigned layers, these layers may be freely included in the layer(s) composing the ply. The only restriction is that at least one layer in the ply must be a local design variable.

5. If the ply is composed of a single layer, this constraint becomes redundant with the TMIN entered on the PCOMP1 field (for shape function linking) or the VMIN entered on the DESEIM or DESVARP entry (for physical linking). In this case, the most critical limit will be determined from among all sources (DCONPMN, DCONLMN, TMIN/VMIN) and will be used to update the local variable side constraint. The DCONxxx entry will then be automatically removed since it will no longer be necessary. A summary of this action will be echoed to the print file.
Input Data Entry  DCONLIST  Design Constraint List

Description:  Defines a list of design constraints for which constraint value output and/or constraint gradient output are desired.

Format and Example:

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>Set identification number (Integer &gt; 0)</td>
</tr>
<tr>
<td>TYPE</td>
<td>The design constraint type. One of the following:</td>
</tr>
<tr>
<td>FREQ</td>
<td>frequency</td>
</tr>
<tr>
<td>FLUT</td>
<td>flutter</td>
</tr>
<tr>
<td>DISP</td>
<td>displacement</td>
</tr>
<tr>
<td>VMISES</td>
<td>Von Mises</td>
</tr>
<tr>
<td>TSAIWU</td>
<td>Tsai-Wu</td>
</tr>
<tr>
<td>STRAIN</td>
<td>strain</td>
</tr>
<tr>
<td>THICK</td>
<td>thickness</td>
</tr>
<tr>
<td>END</td>
<td>aeroelastic effectiveness</td>
</tr>
<tr>
<td>SCF</td>
<td>stability coefficient</td>
</tr>
<tr>
<td>TRIM</td>
<td>trim</td>
</tr>
<tr>
<td>ALL</td>
<td>all of the above</td>
</tr>
<tr>
<td>OTHER</td>
<td>all EXCEPT the above</td>
</tr>
<tr>
<td>NRFAC</td>
<td>Constraint retention factor for math programming methods. At least NRFAC * (number of design variables) constraints will be considered active. (Real &gt; 0.0, Default = 3.0)</td>
</tr>
<tr>
<td>EPS</td>
<td>Constraint retention parameter in which all constraints having a value greater than EPS will be considered active. (Real, Default = -0.1)</td>
</tr>
</tbody>
</table>

Remarks:

1. NRFAC and EPS control the number of constraints that are selected for print and punch output. For constraint gradients, only those considered active by the global constraint screening algorithm (NRFAC and EPS from the OPTIMIZE command in Solution control) are available to be selected.

2. More than one DCONLIST with the same set identification number may be used to select subsets of different constraint types.
**Input Data Entry**  
DCONSCF  
Stability Derivative Constraint

**Description:** Defines a constraint on the flexible stability derivative at the reference grid point associated with the force or moment due to a trim parameter or control surface deflection of the form:

\[
\frac{\partial C_F}{\partial \theta_{trim}} \leq \frac{\partial C_F}{\partial \theta_{trim}} \leq \frac{\partial C_F}{\partial \theta_{trim}}
\]

**Format and Example:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONSCF</td>
<td>SETID</td>
<td>ACCLAB</td>
<td>PRMLAB</td>
<td>CTYEF</td>
<td>PRMREQ</td>
<td>UNITS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCONSCF</td>
<td>999</td>
<td>PACCEL</td>
<td>AILERON</td>
<td>LOWER</td>
<td>1.0</td>
<td>RADIANS</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field**  
**Contents**

- **SETID**  
  Set identification number referenced by the DCONSTRAINT Solution Control option of the SAERO command. (Integer > 0)

- **ACCLAB**  
  Alphanumeric string identifying the aerodynamic force or moment by naming the corresponding structural acceleration in a manner consistent with the TRIM entry. See Remarks 2 and 4.

- **PRMLAB**  
  Alphanumeric string identifying a constrained control surface or aeroelastic trim parameter (e.g. ALPHA or PRATE). See Remarks 3 and 4.

- **CTYEF**  
  Constraint type; either UPPER, for upper bound, or LOWER for lower bound. (Character, default=UPPER)

- **PRMREQ**  
  Bound for the stability coefficient. For units, see Remarks 5 and 6. (Real)

- **UNITS**  
  Units for the stability coefficient. Either RADIANS or DEGREES. See Remark 6. (Real, Default=DEGREES)

**Remarks:**

1. The DCONSCF entry is selected in Solution Control with the DCONSTRAINT=SETID option of the SAERO command.

2. The ACCLAB may refer to any of the TRIM Bulk Data entry trim parameters that are structural accelerations. Valid trim parameters are NX, NY, NZ, PACCEL, QACCEL, and RACCEL.

3. The PRMLAB may refer to AESURF or CONLINK control surfaces or to any of the TRIM entry parameters except the structural accelerations. Valid selections are: PRATE, QRATE, RRATE, ALPHA, BETA, THRCAM and any control surface label. Invalid trim parameters are: NX, NY, NZ, PACCEL, QACCEL and RACCEL.

4. Any combination of forces or moments and trim parameters/control surfaces may be used on this entry provided they have the same symmetry as the associated TRIM entry. Furthermore, to apply the constraint to the flexible derivative, the degree of freedom corresponding to the force or moment must be supported in the boundary condition. For example, to constrain the pitching moment, QACCEL, due to angle of attack, ALPHA, the y-rotation of the support point must be on the SUPPORT entry for the boundary condition in which the TRIM is analyzed.
5. The stability derivatives are nondimensional quantities derived from the flexible forces and moments due to "unit" parameters. The constraint is applied to the nondimensional derivative at the user-defined reference point. To assist the defining PRMREQ, the following normalizations are used in ASTROS:

<table>
<thead>
<tr>
<th>Symmetric Derivatives</th>
<th>Control Surfaces</th>
<th>Stability Coeff = ( F/(QDP*S) )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Forces</td>
<td>Stability coeff = ( F<em>2</em>VO/(QDP<em>S</em>C) )</td>
</tr>
<tr>
<td></td>
<td>Rates</td>
<td>&quot;unit&quot; rate = unit dimensional rate * ( C/2*VO )</td>
</tr>
<tr>
<td></td>
<td>Control Surfaces</td>
<td>Stability coeff = ( M/(QDP<em>S</em>C) )</td>
</tr>
<tr>
<td></td>
<td>Rates</td>
<td>Stability coeff = ( M<em>4</em>VO/(QDP<em>S</em>C*2) )</td>
</tr>
<tr>
<td></td>
<td>&quot;unit&quot; rate = unit dimensional rate * ( C/2*VO )</td>
<td></td>
</tr>
<tr>
<td>Antisymmetric Derivatives</td>
<td>Control Surfaces</td>
<td>Stability coeff = ( F/(QDP*S) )</td>
</tr>
<tr>
<td></td>
<td>Forces</td>
<td>Stability coeff = ( F<em>2</em>VO/(QDP<em>S</em>B) )</td>
</tr>
<tr>
<td></td>
<td>Rates</td>
<td>&quot;unit&quot; rate = unit dimensional rate * ( C/2*VO )</td>
</tr>
<tr>
<td></td>
<td>Control Surfaces</td>
<td>Stability coeff = ( M/(QDP<em>S</em>B) )</td>
</tr>
<tr>
<td></td>
<td>Rates</td>
<td>Stability coeff = ( M<em>4</em>VO/(QDP<em>S</em>B*2) )</td>
</tr>
<tr>
<td></td>
<td>&quot;unit&quot; rate = unit dimensional rate * ( B/2*VO )</td>
<td></td>
</tr>
</tbody>
</table>

\( F \) and \( M \) are the dimensional flexible forces and moments for the full vehicle; \( S, C, \) and \( B \) are the non-dimensional factors from the AEROS Bulk Data entry (the inputs are assumed to be for the full vehicle); and \( QDP \) and \( VO \) are defined on the TRIM Bulk Data entry.

6. **RADIANS** or **DEGREES** refer to the units of the unit control surface deflection or unit rate. **RADIANS** imply the value due to a unit RAD or RAD/S while **DEGREES** imply the value due to a unit DEG or DEG/S. THRCAM has no valid angular unit, hence the **UNITS** field is ignored.

7. A **LOWER** bound constraint excludes all values to the left of **PRMREQ** on a real number line, while an **UPPER** bound excludes all values to the right, irrespective of the sign of **PRMREQ**.
Input Data Entry: DCONTH2 Thickness constraints on layers of composite elements

Description: Defines a set of layers for a list of elements linked using SHAPE entries for which the ply thickness constraints are to be retained on all design iterations.

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONTH2</td>
<td>ETYP</td>
<td>LAYN</td>
<td>LAYRLST</td>
<td>EID</td>
<td>EID</td>
<td>EID</td>
<td>EID</td>
<td>EID</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>EID</td>
<td>EID</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONTHK</td>
<td>ETYP</td>
<td>LAYN</td>
<td>LAYRLST</td>
<td>EID</td>
<td>EID</td>
<td>EID</td>
<td>EID</td>
<td>EID</td>
<td>EID</td>
</tr>
<tr>
<td>QUAD4</td>
<td>100</td>
<td>101</td>
<td>200</td>
<td>205</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Alternate Form:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONTH2</td>
<td>ETYP</td>
<td>LAYN</td>
<td>LAYRLST</td>
<td>EID</td>
<td>THRU</td>
<td>EID</td>
<td>EID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field | Contents

ETYPE Character input identifying the element type. One of the following:
QUAD4 TRIA3
QDMEM1 TRMEM
LAYN Layer number of the layer(s) to be retained. The given layer will be retained for each element in the list of elements (Integer > 0 or blank, See Remark 1)
LAYRLST Set identification number of a PLYLIST bulk data entry naming a set of plies to be retained as active for each element. (Integer > 0 or blank, See Remark 1)
EID Element identification number (Integer > 0 or blank)

Remarks:

1. One and only one of either LAYN or LAYRLST must be given. Noncomposite elements must be called out on DCONTHK entries.

2. The purpose of this bulk data list is to ensure that adequate physical move limits are retained in optimization with shape function design variable linking without requiring retention of all move limits. For problems with large numbers of local variables using shape functions, the move limits often cause too many minimum thickness constraints (see Remark 3) to be retained in the optimization task. Using this bulk data entry or its noncomposite counterpart DCONTHK to name "critical" minimum gauge constraints (see Remark 4) will cause only the named elements' thickness constraints to be computed and retained. All layers of composite elements named on DCONTHK will be retained.

NOTE that all elements' thickness constraints will always be computed irrespective of the DCONTHK entries, but may be deleted in the constraint deletion.

3. The global design variable in shape function linking is non-physical and no reasonable restriction for a global variable move limit (side constraint) can be defined. Therefore, constraints on the local design variables controlled by shape functions are generated by ASTROS to ensure that the design is reasonable (i.e., nonnegative thicknesses).
4. The DCONTR2 entry should select a minimum number of elements linked to shape functions that will enable the optimizer to select physically reasonable designs without retaining all the minimum thickness constraints (potentially a very large number). Typically, this means $N+1$ elements spread over the range of the shape function (e.g. span or chord) where $N$ is the order of the shape ($N=0$, UNIFORM: $N=1$, LINEAR, etc.).
Input Data Entry: DCONTHK  Thickness constraints on elements

Description: Defines a list of elements linked using SHAPE entries for which thickness constraints are to be retained on all design iterations.

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONTHK</td>
<td>ETYPE</td>
<td>EID</td>
<td>EID</td>
<td>EID</td>
<td>EID</td>
<td>EID</td>
<td>EID</td>
<td>EID</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>EID</td>
<td>EID</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Alternate Form:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONTHK</td>
<td>ETYPE</td>
<td>EID</td>
<td>THRU</td>
<td>EID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETYPE</td>
<td>Character input identifying the element type. One of the following: BAR, QUAD4, CONM2, ROD, ELAS, SHEAR, MASS, TRIA3, QDMEM1, TRMEM</td>
</tr>
<tr>
<td>EID</td>
<td>Element identification number (Integer &gt; 0 or blank)</td>
</tr>
</tbody>
</table>

Remarks:

1. The purpose of this bulk data list is to ensure that adequate physical move limits are retained in optimization with shape function design variable linking without requiring retention of all move limits. For problems with large numbers of local variables using shape functions, the move limits often cause too many minimum thickness constraints (see Remark 2) to be retained in the optimization task. Using this bulk data entry OR its composite counterpart DCONTH2 to name "critical" minimum gauge constraints (see Remark 3) will cause only the named elements' thickness constraints to be computed and retained. All layers of composite elements named on DCONTHK will be retained. NOTE that all elements' thickness constraints will always be computed irrespective of the DCONTHK entries, but may be deleted in the constraint deletion.

2. The global design variable in shape function linking is non-physical and no reasonable restriction for a global variable move limit (side constraint) can be defined. Therefore, constraints on the local design variables controlled by shape functions are generated by ASTROS to ensure that the design is reasonable (ie, nonnegative thicknesses).

3. The DCONTHK entry should select a minimum number of elements linked to shape functions that will enable the optimizer to select physically reasonable designs without retaining all the minimum thickness constraints (potentially a very large number). Typically, this means N+1 elements spread over the range of the shape function (e.g. span or chord) where N is the order of the shape (N=0, UNIFORM: N=1, LINEAR, etc.). Use DCONTH2 for composite elements in which linking across layers may allow certain layers to be omitted from the retention set.
Input Data Entry  DCONTRM  Aeroelastic Trim Parameter Constraint

Description:  Defines a trim parameter constraint of the form:

\[ \delta_{\text{trim}} \leq \delta_{\text{trim}_{\text{Req}}} \quad \text{or} \quad \delta_{\text{trim}} \geq \delta_{\text{trim}_{\text{Req}}} \]

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONTRM</td>
<td>SETID</td>
<td>PRMLAB</td>
<td>CTYPE</td>
<td>PRMREQ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCONTRM</td>
<td>100</td>
<td>AILERON</td>
<td>UPPER</td>
<td>25.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field  Contents

<table>
<thead>
<tr>
<th>SETID</th>
<th>Set identification number referenced by the DCONSTRAINT Solution Control command. (Integer &gt; 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRMLAB</td>
<td>Alphanumeric string identifying a constrained control surface or aeroelastic trim parameter (e.g. ALPHA or PRATE). (See Remark 2.)</td>
</tr>
<tr>
<td>CTYPE</td>
<td>Constraint type; either UPPER, for upper bound, or LOWER for lower bound. (Character, Default = UPPER)</td>
</tr>
<tr>
<td>PRMREQ</td>
<td>Bound for the trim parameter. For units, see Remark 3. (Real)</td>
</tr>
</tbody>
</table>

Remarks:

1. The DCONTRM entry is selected in Solution Control with the DCONSTRAINT=SETID option of the SAERO command.

2. The PRMLAB may refer to AESURF or CONLINK control surfaces or to any of the TRIM entry parameters, NX, NY, NZ, PACCEL, QACCEL, PACCEL, PRATE, QRATE, RRATE, ALPHA, or BETA. The only requirement is that the constrained control surface must be declared on the TRIM entry. The user will be warned if trim parameters not on the TRIM entry are constrained (since these parameters are fixed, they are design invariant).

3. The units for control surface deflections are degrees. For rates, the units should be radians/sec. For linear accelerations NX, NY, NZ, the units should be consistent, (length/sec/sec) or, if a CONVERT,MASS entry was used, should be dimensionless. Angular accelerations should be in radians/sec/sec.

4. A LOWER bound constraint excludes all values to the left of PRMREQ on a real number line, while an UPPER bound excludes all values to the right, irrespective of the sign of PRMREQ.
Input Data Entry  DCONTW  Tsai-Wu Stress Constraint Definition

Description: Defines Tsai-Wu stress constraints by specifying the identification numbers of constrained elements.

Format and Example:

<table>
<thead>
<tr>
<th>SID</th>
<th>XT</th>
<th>XC</th>
<th>YT</th>
<th>YC</th>
<th>SS</th>
<th>F12</th>
<th>ETYP</th>
<th>CONT</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONTW</td>
<td>LAYRNUM</td>
<td>EID1</td>
<td>EID2</td>
<td>EID3</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DCONTW  100  1.6+6  -1.6+6  1.4+4  -1.4+4  1.5+3  QUAD4  ABC

+BC  1  102  106  110

Alternate Form:

<table>
<thead>
<tr>
<th>SID</th>
<th>XT</th>
<th>XC</th>
<th>YT</th>
<th>YC</th>
<th>SS</th>
<th>F12</th>
<th>ETYP</th>
<th>CONT</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONTW</td>
<td>LAYRNUM</td>
<td>EID1</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DCONTW  100  1.6+6  -1.6+6  1.4+4  -1.4+4  1.5+3  QUAD4  ABC

+BC  1  102  106  110

Field Contents

SID  Stress constraint set identification (Integer > 0)
XT  Tensile stress limit in the longitudinal direction (Real > 0.0)
XC  Compressive stress limit in the longitudinal direction (Real, Default = XT)
YT  Tensile stress limit in the transverse direction (Real > 0.0)
YC  Compressive stress limit in the transverse direction (Real, Default = YT)
SS  Shear stress limit for in-plane stress (Real > 0.0)
F12  Tsai-Wu interaction term (Real)
ETYPE  Element type (Text)
LAYRNUM  The layer number of a composite element (Integer > 0 or blank)
EID1  Element identification numbers (Integer > 0)

Remarks:

1. Stress constraints are selected in Solution Control with the discipline option:

   STRESS=sid

2. ETYP must be selected from Q4MEM1, QUAD4, TRIA3, or TRMEM.

3. If the alternate form is used, EID2 must be greater than or equal to EID1. Elements in the range which do not exist are ignored.

4. The strain limits for compression, XC and YC, are always treated as negative values regardless of the sign of the input values.

5. LAYRNUM is only used if the element is composed of a composite material defined with PCOMB Bulk Data entries.
**Input Data Entry**  
**DCONTWM**  
Tsai-Wu Stress Constraint Definition

**Description:** Defines Tsai-Wu stress constraints by specifying material identification numbers

**Format and Example:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONTWM</td>
<td>SID</td>
<td>XT</td>
<td>XC</td>
<td>YT</td>
<td>YC</td>
<td>SS</td>
<td>F12</td>
<td>MID1</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>MID2</td>
<td>MID3</td>
<td>MID4</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONTWM</td>
<td>100</td>
<td>1.+6</td>
<td>-1.+6</td>
<td>1.+4</td>
<td>-1.+4</td>
<td>1.5+3</td>
<td>101</td>
<td>ABC</td>
<td></td>
</tr>
<tr>
<td>+BC</td>
<td>102</td>
<td>200</td>
<td>310</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Alternate Form:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONTWM</td>
<td>SID</td>
<td>XT</td>
<td>XC</td>
<td>YT</td>
<td>YC</td>
<td>SS</td>
<td>F12</td>
<td>MID1</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>THRU</td>
<td>MID2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field Contents**

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>Stress constraint set identification (Integer &gt; 0)</td>
</tr>
<tr>
<td>XT</td>
<td>Tensile stress limit in the longitudinal direction (Real &gt; 0.0)</td>
</tr>
<tr>
<td>XC</td>
<td>Compressive stress limit in the longitudinal direction (Real, Default = XT)</td>
</tr>
<tr>
<td>YT</td>
<td>Tensile stress limit in the transverse direction (Real &gt; 0.0)</td>
</tr>
<tr>
<td>YC</td>
<td>Compressive stress limit in the transverse direction (Real, Default = YT)</td>
</tr>
<tr>
<td>SS</td>
<td>Shear stress limit for in-plane stress (Real &gt; 0.0)</td>
</tr>
<tr>
<td>F12</td>
<td>Tsai-Wu interaction term (Real)</td>
</tr>
<tr>
<td>MID1</td>
<td>Material identification numbers (Integer &gt; 0)</td>
</tr>
</tbody>
</table>

**Remarks:**

1. Stress constraints are selected in Solution Control with the discipline option:

   `STRESS=sid`

2. If the alternate form is used, `MID2` must be greater than or equal to `MID1`. Materials in the range which do not exist are ignored.

3. The stress limits for compression, `XC` and `YC`, are always treated as negative values regardless of the sign of the input values.
**Input Data Entry**  
**DCONTWP**  
**Tsai-Wu Stress Constraint Definition**

**Description:** Defines Tsai-Wu stress constraints by specifying element property identification numbers

**Format and Example:**

```
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONTWP</td>
<td>SID</td>
<td>XT</td>
<td>XC</td>
<td>YT</td>
<td>YC</td>
<td>SS</td>
<td>F12</td>
<td>PTYPE</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>LAYRNUM</td>
<td>PID1</td>
<td>PID2</td>
<td>PID3</td>
<td>etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>1.06</td>
<td>-1.06</td>
<td>1.04</td>
<td>-1.04</td>
<td>1.53</td>
<td>PCOMP</td>
<td>ABC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+BC</td>
<td>100</td>
<td>200</td>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**Alternate Form:**

```
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONTWP</td>
<td>SID</td>
<td>XT</td>
<td>XC</td>
<td>YT</td>
<td>YC</td>
<td>SS</td>
<td>F12</td>
<td>PTYPE</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>LAYRNUM</td>
<td>PID1</td>
<td>THRU</td>
<td>PID2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**Field Contents**

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>Stress constraint set identification (Integer &gt; 0)</td>
</tr>
<tr>
<td>XT</td>
<td>Tensile stress limit in the longitudinal direction (Real &gt; 0.0)</td>
</tr>
<tr>
<td>XC</td>
<td>Compressive stress limit in the longitudinal direction (Real, Default = XT)</td>
</tr>
<tr>
<td>YT</td>
<td>Tensile stress limit in the transverse direction (Real &gt; 0.0)</td>
</tr>
<tr>
<td>YC</td>
<td>Compressive stress limit in the transverse direction (Real, Default = YT)</td>
</tr>
<tr>
<td>SS</td>
<td>Shear stress limit for in-plane stress (Real &gt; 0.0)</td>
</tr>
<tr>
<td>F12</td>
<td>Tsai-Wu interaction term (Real)</td>
</tr>
<tr>
<td>PTYPE</td>
<td>Property type (Text)</td>
</tr>
<tr>
<td>LAYRNUM</td>
<td>The layer number of a composite element (Integer &gt; 0 or blank)</td>
</tr>
<tr>
<td>PID1</td>
<td>Property identification numbers (Integer &gt; 0)</td>
</tr>
</tbody>
</table>

**Remarks:**
1. Stress constraints are selected in Solution Control with the discipline option:
   ```
   STRESS=sid
   ```
2. **PTYPE** may be selected from `PCOMP, PCOMP1, PCOMP2, PQMEM1, PSHELL, or PTRMEM`.
3. If the alternate form is used, PID2 must be greater than or equal to PID1. Properties in the range which do not exist are ignored.
4. The stress limits for compression, XC and YC, are always treated as negative values regardless of the sign of the input values.
5. **LAYRNUM** is only used if the element is composed of a composite material defined with `PCOMP` Bulk Data entries.
**Input Data Entry**

**DCONVM**

Von-Mises Stress Constraint Definition

**Description:** Defines a Von-Mises stress constraint by specifying the identification numbers of constrained elements

**Format and Example:**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONVM</td>
<td>SID</td>
<td>ST</td>
<td>SC</td>
<td>SS</td>
<td>ETYPE</td>
<td>LAYRNUM</td>
<td>EID1</td>
<td>EID2</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>EID3</td>
<td>EID4</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DCONVM

100 1. +6 -1. +6 1. +4 BAR 101 102 ABC

+BC 107 108 142

**Alternate Form:**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONVM</td>
<td>SID</td>
<td>ST</td>
<td>SC</td>
<td>SS</td>
<td>ETYPE</td>
<td>LAYRNUM</td>
<td>EID1</td>
<td>THRU</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>EID2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field**

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>Stress constraint set identification (Integer &gt; 0)</td>
</tr>
<tr>
<td>ST</td>
<td>Tensile stress limit (Real &gt; 0.0 or blank)</td>
</tr>
<tr>
<td>SC</td>
<td>Compressive stress limit (Real, Default = ST).</td>
</tr>
<tr>
<td>SS</td>
<td>Shear stress limit (Real &gt; 0.0 or blank)</td>
</tr>
<tr>
<td>ETYPE</td>
<td>Element type (Text)</td>
</tr>
<tr>
<td>LAYRNUM</td>
<td>The layer number of a composite element (Integer &gt; 0 or blank)</td>
</tr>
<tr>
<td>EIDi</td>
<td>Element identification numbers (Integer &gt; 0)</td>
</tr>
</tbody>
</table>

**Remarks:**

1. Stress constraints are selected in Solution Control with the discipline option:

   STRESS=sid

2. ETYPE may be selected from BAR, QDMEM1, QUAD4, ROD, SHEAR, TRIA3, or TRMEM.

3. If the alternate form is used, EID2 must be greater than or equal to EID1. Elements in the range which do not exist are ignored.

4. The stress limit for compression, SC, is always treated as a negative value regardless of the sign of the input value.

5. LAYRNUM is only used if the element is composed of a composite material defined with PCOMP Bulk Data entries.
**Input Data Entry**  
**DCONVMM**  
**Von-Mises Stress Constraint Definition**

**Description:** Defines a Von-Mises stress constraint by specifying material identification numbers.

**Format and Example:**

```
<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONVMM</td>
<td>SID</td>
<td>ST</td>
<td>SC</td>
<td>SS</td>
<td>MID1</td>
<td>MID2</td>
<td>MID3</td>
<td>MID4</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>MID5</td>
<td>MID6</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

```
| DCONVMM | 100 | 1.+6 | -1.+6 | 1.+4 | 101 | 201 | 301 | 401 | ABC |
| +BC | 501 | 601 | 701 |
```

**Alternate Form:**

```
<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCONVMM</td>
<td>SID</td>
<td>ST</td>
<td>SC</td>
<td>SS</td>
<td>MID1</td>
<td>THRU</td>
<td>MID2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**Field Contents**

- **SID**: Stress constraint set identification (Integer > 0)
- **ST**: Tensile stress limit (Real > 0.0 or blank)
- **SC**: Compressive stress limit (Real, Default = ST)
- **SS**: Shear stress limit (Real > 0.0 or blank)
- **MIDI**: Material identification numbers (Integer > 0)

**Remarks:**

1. Stress constraints are selected in Solution Control with the discipline option:

   **STRESS=sid**

2. If the alternate form is used, MID2 must be greater than or equal to MID1. Materials in the range which do not exist are ignored.

3. The stress limit for compression, SC, is always treated as a negative value regardless of the sign of the input value.
Input Data Entry  DCONVMP  Von-Mises Stress Constraint Definition

Description: Defines a Von-Mises stress constraint by specifying property identification numbers.

Format and Example:

<table>
<thead>
<tr>
<th>DCONVMP</th>
<th>SID</th>
<th>ST</th>
<th>SC</th>
<th>SS</th>
<th>PTYPE</th>
<th>LAYRNUM</th>
<th>PID1</th>
<th>PID2</th>
<th>CONT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONT</td>
<td>PID3</td>
<td>PID4</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Alternate Form:

<table>
<thead>
<tr>
<th>DCONVMP</th>
<th>SID</th>
<th>ST</th>
<th>SC</th>
<th>SS</th>
<th>PTYPE</th>
<th>LAYRNUM</th>
<th>PID1</th>
<th>PID2</th>
<th>THRU</th>
<th>CONT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONT</td>
<td>PID2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

- **SID**: Stress constraint set identification (Integer > 0).
- **ST**: Tensile stress limit (Real > 0.0 or blank)
- **SC**: Compressive stress limit (Real, Default = ST)
- **SS**: Shear stress limit (Real > 0.0 or blank)
- **PTYPE**: Property type (Text)
- **LAYRNUM**: The layer number of a composite element (Integer > 0 or blank)
- **PID1**: Property identification numbers (Integer > 0)

Remarks:

1. Stress constraints are selected in Solution Control with the discipline option:

   \[ \text{STRESS} = \text{sid} \]

2. **PTYPE** may be selected from **PBAR**, **PCOMP**, **PCOMP1**, **PCOMP2**, **PCOMP3**, **PROD**, **PSHEAR**, **PSHELL**, or **PTRMM**.

3. If the alternate form is used, **PID2** must be greater than or equal to **PID1**. Properties in the range which do not exist are ignored.

4. The stress limit for compression, **SC**, is always treated as a negative value regardless of the sign of the input value.

5. **LAYRNUM** is only used if the element is composed of a composite material defined with **PCOMP** Bulk Data entries.
**Input Data Entry:** DESELM

**Description:** Designates design variable properties when the design variable is uniquely associated with a single finite element.

**Format and Example:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESELM</td>
<td>DVID</td>
<td>EID</td>
<td>ETYPE</td>
<td>VMIN</td>
<td>VMAX</td>
<td>VINIT</td>
<td>LAYERNUM</td>
<td>LABEL</td>
<td></td>
</tr>
<tr>
<td>DESELM</td>
<td>5</td>
<td>10</td>
<td>CROD</td>
<td>0.01</td>
<td>10.0</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field Contents**

- **DVID**: Design variable identification (Integer > 0)
- **EID**: Element identification (Integer > 0)
- **ETYPE**: Element type
- **VMIN**: Minimum allowable value of the design variable (Real ≥ 0.0) (Default = 0.01)
- **VMAX**: Maximum allowable value of the design variable (Real ≥ 0.0) (Default = 1000.0)
- **VINIT**: Initial value of the design variable (Real, VMIN ≤ VINIT ≤ VMAX) (Default is VMIN)
- **LAYERNUM**: The layer number of a composite element to be designed (Integer > 0, or blank)
- **LABEL**: Optional user-supplied label to define the design variable (text)

**Remarks:**

1. Valid ETYPES are CROD, CONROD, CBAR, CSHEAR, CTRIA3, CTRME1, CQDMEM1, CQUAD4, CYLAS1, CELAS2, CMASS1, CMASS2 and CONM2.

2. The initial element thickness or area used in the structural analysis is derived from the VINIT value and the property value on the associated property entry:

   \[ t_{\text{init}} = \text{VINIT} \times \text{property\_value} \]

   Similarly, the minimum and maximum values are the VMIN and VMAX values of the element property are derived from:

   \[ t_{\text{min}} = \text{VMIN} \times \text{property\_value} \]

   \[ t_{\text{max}} = \text{VMAX} \times \text{property\_value} \]

3. DVID must be unique among all DESELM, DESVARP and DESVARS entries.
Input Data Entry: DESVARP

Description: Designates physically linked global design variable properties

Format and Example:

<table>
<thead>
<tr>
<th>DESVARP</th>
<th>DVID</th>
<th>LINKID</th>
<th>VMIN</th>
<th>VMAX</th>
<th>VINIT</th>
<th>LAYRNUN</th>
<th>LAYRLST</th>
<th>LABEL</th>
<th>NBSTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESVARP</td>
<td>1</td>
<td></td>
<td>0.01</td>
<td>2.0</td>
<td>1.0</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field | Contents
--- | ---
DVID | Design variable identification (Integer > 0)
LINKID | link identification number referring to ELIST and/or PLIST entries (Integer > 0, or blank) (Default = DVID)
VMIN | Minimum allowable value of the design variable (Real > 0.0) (Default = 0.001)
VMAX | Maximum allowable value of the design variable (Real ≥ 0.0) (Default = 1000.0)
VINIT | Initial value of the design variable (Real, VMIN ≤ VINIT ≤ VMAX) (Default is VMIN)
LAYRNUN | Layer number if referencing a single layer of composite element(s) (Integer > 0 or blank)
LAYRLST | Set identification number of PLYLIST entries specifying a set of composite layers to be linked (Integer > 0 or blank)
LABEL | Optional user supplied label to define the design variable (Text)

Remarks:
1. The elements linked to the DESVARP are specified using either a PLIST or an ELIST data entry.
2. The initial element thickness or area used in the structural analysis is derived from the VINIT value and the property value on the associated property entry:

   \[ t_{\text{init}} = \text{VINIT} \times \text{property\_value} \]

   Similarly, the minimum and maximum values are the VMIN and VMAX values of the element property are derived from:

   \[ t_{\text{min}} = \text{VMIN} \times \text{property\_value} \]
   \[ t_{\text{max}} = \text{VMAX} \times \text{property\_value} \]
3. LAYRNUN and LAYRLST are mutually exclusive.
4. Noncomposite elements may be linked to composite layers by including them in the ELIST and/or PLIST sets.
Input Data Entry: DESVARS

Description: Designates shape function linked global design variable properties.

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESVARS</td>
<td>DVID</td>
<td>SHAPEID</td>
<td>VMIN</td>
<td>VMAX</td>
<td>VINIT</td>
<td>LAYERNUM</td>
<td>LAYRLST</td>
<td>LABEL</td>
<td></td>
</tr>
<tr>
<td>DESVARS</td>
<td>1</td>
<td></td>
<td>0.01</td>
<td>2.0</td>
<td>1.0</td>
<td>13</td>
<td></td>
<td>INBDTOPE</td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

- **DVID**: Design variable identification (Integer > 0)
- **SHAPEID**: Identification number of SHAPE bulk data entries defining the shape function (Integer > 0, or blank) (Default = DVID)
- **VMIN**: Minimum allowable value of the design variable (Real) (Default = -10^20)
- **VMAX**: Maximum allowable value of the design variable (Real) (Default = -10^20)
- **VINIT**: Initial value of the design variable (Real, VMIN ≤ VINIT ≤ VMAX) (Default is VMIN)
- **LAYERNUM**: Layer number if referencing a single layer of composite element(s) (Integer > 0 or blank)
- **LAYRLST**: Set identification of PLYLIST entries specifying a set of composite layers to be linked (Integer > 0 or blank)
- **LABEL**: Optional user supplied label to define the design variable (Text)

Remarks:

1. The elements linked to the DESVARS are specified using either a PLIST or an ELIST data entries.
2. The initial local variables are computed from:

   \[ \{ \mathbf{t}_{\text{INIT}} \} = [\mathbf{P}] \{ \mathbf{VINIT} \} \]

   The minimum and maximum values for the local variables are taken from the TMIN and TMAX values on the property and connectivity entries, respectively.
3. LAYERNUM and LAYRLST are mutually exclusive.
4. Noncomposite elements may be linked to composite layers by including them in the referenced SHAPE set.
Input Data Entry: **DLARGS**

**Description:** This entry is used in conjunction with **RLOAD1, RLOAD2, TLOAD1** and **TLOAD2** data entries and defines time lags and phase lags as well as the set identification of the static load.

**Format and Example:**

```
<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLARGS</td>
<td>SID</td>
<td>LID</td>
<td>TAU</td>
<td>PHASE</td>
<td>LID</td>
<td>TAU</td>
<td>PHASE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLARGS</td>
<td>5</td>
<td>21</td>
<td>0.04</td>
<td>20.0</td>
<td>10</td>
<td>0.0</td>
<td>45.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**Field** | **Contents**
--- | ---
SID | Identification number of **DLARGS** set (Integer > 0)
LID | Identification number of time (or frequency) independent applied load (Integer > 0)
TAU | Time delay for the designated load set (Real)
PHASE | Phase lag (in degrees) for the designated load set (Real)

**Remarks:**
1. One or two dynamic load sets may be defined on a single entry.
2. Refer to **RLOAD1, RLOAD2, TLOAD1** or **TLOAD2** entries for formulas which define the manner in which **TAU** and **PHASE** are used.
3. The phase parameter is used only in conjunction with **RLOAD1** and **RLOAD2** data entries.
4. The LID set can refer to statically applied loads as well as to additional dynamic loads input on **DLONLY** entries.
5. **TAU** and **PHASE** can be defaulted to zero, but LID must not be zero.
**Input Data Entry:** DLOAD

**Description:** Defines a dynamic loading condition for frequency response or transient response problems as a linear combination of load sets defined using RLOAD1 or RLOAD2 entries (for frequency response) or TLOAD1 or TLOAD2 entries (for transient response).

**Format and Example:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLOAD</td>
<td>SID</td>
<td>S</td>
<td>S1</td>
<td>L1</td>
<td>S2</td>
<td>L2</td>
<td>S3</td>
<td>L3</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>S4</td>
<td>L4</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DLOAD 17 1.0 2.0 6 -2.0 7 2.0 8 +A

**Field** | **Contents**
--- | ---
SID | Load set identification number (Integer > 0)
S | Scale factor (Real ≠ 0.0)
S1 | Scale Factors (Real ≠ 0.0)
L1 | Load set identification numbers defined via bulk data entries enumerated above (Integer > 0)

**Remarks:**
1. The load vector being defined by this entry is given by
   \[ [P] = S \sum S_1 P_1 \]
2. The L1 must be unique.
3. SID must be unique from all L1.
4. TLOAD1 and TLOAD2 loads may be combined only through the use of the DLOAD entry.
5. RLOAD1 and RLOAD2 loads may be combined only through the use of the DLOAD entry.
6. SID must be unique for all TLOAD1, TLOAD2, RLOAD1 and RLOAD2 entries.
7. Linear load sets must be selected by a solution control command (DLOAD = SID).
**Input Data Entry: ** DLONLY

**Description:** This entry is used in conjunction with the RLOAD1, RLOAD2, TLOAD1 and TLOAD2 entries and defines the point where the dynamic load is to be applied with the scale factor A.

**Format and Example:**

<p>| | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>DLONLY</td>
<td>SID</td>
<td>P</td>
<td>C</td>
<td>A</td>
<td>P</td>
<td>C</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLONLY</td>
<td>3</td>
<td>6</td>
<td>2</td>
<td>8.2</td>
<td>15</td>
<td>1</td>
<td>10.1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field**  

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>Identification number of DLONLY set (Integer &gt; 0)</td>
</tr>
<tr>
<td>P</td>
<td>Grid, extra point or scalar point identification number (Integer &gt; 0)</td>
</tr>
<tr>
<td>C</td>
<td>Component number (1 through 6 for grid point; blank or 0 for extra points or scalar points)</td>
</tr>
<tr>
<td>A</td>
<td>Load factor A for the designated coordinate (Real)</td>
</tr>
</tbody>
</table>

**Remarks:**

1. One or two load factors may be defined on a single entry.
2. Refer to RLOAD1, RLOAD2, TLOAD1 or TLOAD2 entries for the formulas which define the load factor A.
3. Component numbers refer to global coordinates.
4. The SID field is referred to as the DLAGS entry.
5. The scale factor, A, applied to any grid/component will be the sum of all A_i for that degree of freedom on all DLONLY entries with the same SID.
Input Data Entry: DMI  Direct Matrix Input

Description: Used to input matrix data base entities directly. Generates a real or complex matrix of the form:

\[ A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix} \]

where the elements \( A_{ij} \) may be real or complex.

Format and Example:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>C1</td>
<td>R1</td>
<td>A(R1, C1)</td>
<td>C2</td>
<td>R2</td>
<td>A(R2, C2)</td>
<td>A(R1+1, C2)</td>
<td>C3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>R1</td>
<td>A(R1, C3)</td>
<td></td>
<td>C4</td>
<td>R2</td>
<td>A(R2, C4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DMI</th>
<th>TEST</th>
<th>RDP</th>
<th>REC</th>
<th>3</th>
<th>4</th>
<th></th>
<th></th>
<th>ABC</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>+BC</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3.0</td>
<td>4.0</td>
<td>4</td>
<td>DEF</td>
</tr>
<tr>
<td>+EF</td>
<td>1</td>
<td>5.0</td>
<td>4</td>
<td>3</td>
<td></td>
<td>6.5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>Any valid data base entity name (Text 8)</td>
</tr>
<tr>
<td>PREC</td>
<td>The precision of the matrix entity to be loaded. Any one of the following character strings: RSP, CSP, RDP, CDP</td>
</tr>
<tr>
<td>FORM</td>
<td>The form of the matrix entity to be loaded. Any one of the following REC, SYM, DIAG, IDENT, SQUARE</td>
</tr>
<tr>
<td>M</td>
<td>The number of rows in the matrix (Integer &gt; 0)</td>
</tr>
<tr>
<td>N</td>
<td>The number of columns in the matrix (Integer &gt; 0)</td>
</tr>
<tr>
<td>C1</td>
<td>The column number of the column being loaded (Integer)</td>
</tr>
<tr>
<td>R1</td>
<td>The row number of the first row in the string being loaded (Integer)</td>
</tr>
<tr>
<td>A(R1, C1)</td>
<td>Matrix terms (Real)</td>
</tr>
</tbody>
</table>

Remarks:

1. If the named entity exists, it will be flushed and reloaded. If the entity does not exist, it will be created.

2. Column and row identifiers \((C_i, R_i)\) must always appear together although they can appear in any two contiguous fields.

3. Columns must be loaded in increasing column number order. If more than one string is to be loaded for a particular column, the \( C_i \) field must contain the same value as in the previous string. Strings must be loaded in increasing row order without overlap. Complex matrices require two real values for each matrix term. These can be split across physical entry boundaries.
**Input Data Entry:**  

**DMIG**  

**Direct Matrix Input a Grid Points**  

**Description:**  
Defines structure-related direct input matrices with terms located by external grid/component values.

**Format and Example:**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMIG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>NAME</td>
<td>PREC</td>
<td>FORM</td>
<td>CROW</td>
<td>Xij</td>
<td>Yij</td>
<td>CONT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>GCOL</td>
<td>CCOL</td>
<td>GROW</td>
<td>CROW</td>
<td>Xij</td>
<td>Yij</td>
<td>CONT</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DMIG</th>
<th>TEST</th>
<th>RDP</th>
<th>REC</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DMIG</td>
<td>1001</td>
<td>4</td>
<td>2001</td>
<td>2</td>
<td>1.25+5</td>
<td></td>
<td>ABC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+EF</td>
<td>1001</td>
<td>4</td>
<td>3001</td>
<td>3</td>
<td>2.67+4</td>
<td>-etc-</td>
<td>DEF</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field**  

<table>
<thead>
<tr>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NAME</strong></td>
</tr>
<tr>
<td><strong>PREC</strong></td>
</tr>
<tr>
<td><strong>FORM</strong></td>
</tr>
<tr>
<td><strong>GCOL</strong></td>
</tr>
<tr>
<td><strong>CCOL</strong></td>
</tr>
<tr>
<td><strong>GROW</strong></td>
</tr>
<tr>
<td><strong>CROW</strong></td>
</tr>
<tr>
<td><strong>Xij, Yij</strong></td>
</tr>
</tbody>
</table>

**Remarks:**

1. If the named entity exists, it will be flushed and reloaded. If the entity does not exist, it will be created.
2. The number of rows and columns will be either p-set size or g-set size depending on whether the named entity is requested by K2PP, M2PP, B2PP or K2GG, M2GG, etc.
3. Each non-null term in the matrix requires a continuation entry. The column index and row index values can appear any number of times on a logical entry but a fatal error will occur if the same term is entered more than once.
4. The matrix terms can be entered in any order.
5. The **TRIANG** input **FORM** implies that only the upper or lower triangular portion of the symmetric matrix is input. ASTROS will automatically expand the input across the diagonal.
Input Data Entry:  **DYNRED**  
Dynamic Reduction Data

Description:  Defines dynamic reduction control data.

Format and Example:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DYNRED</strong></td>
<td>SID</td>
<td>FMAX</td>
<td>NVEC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DYNRED</strong></td>
<td>1</td>
<td>50.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field | Contents
---|---
SID | Set identification number (Integer > 0)
FMAX | Highest frequency of interest (Hertz) (Real > 0 or blank)
NVEC | Number of generalized coordinates desired (Integer > 0 or blank)

Remarks:

1. Dynamic reduction data must be requested in the Solution Control packet with:
   
   **DYNRED=SID**

2. The user should select either an FMAX, or both the FMAX and NVEC fields. FMAX should not be greater than necessary for the specific dynamic analysis. NVEC, if specified, should be significantly less than the size of the f-set to realize any computational cost savings. NVEC will limit dynamic reduction to using NVEC flexible vectors.

3. Dynamic reduction transforms the motions of the f-set to the motions of the user defined A-set plus motions of generalized coordinates created in the process. The generalized coordinates represent overall structure displacements which are approximate normal mode shapes. The generalized coordinates are identified by SCALAR points that are automatically generated. The SCALAR point identification numbers begin with 1 greater than the highest user GRID, SCALAR, or EXTRA point identification number.
Input Data Entry:  EIGC  Complex Eigenvalue Extraction Data.

Description:  Specifies complex eigensolution control data.

Format and Example:

<table>
<thead>
<tr>
<th>EIGC</th>
<th>SID</th>
<th>METHOD</th>
<th>NORM</th>
<th>G</th>
<th>C</th>
<th>E</th>
<th>abc</th>
</tr>
</thead>
<tbody>
<tr>
<td>+bc</td>
<td>PA1</td>
<td>QA1</td>
<td>PB1</td>
<td>QB1</td>
<td>W1</td>
<td>NE1</td>
<td>ND1</td>
</tr>
<tr>
<td>+ef</td>
<td>PA2</td>
<td>QA2</td>
<td>PB2</td>
<td>QB2</td>
<td>W2</td>
<td>NE2</td>
<td>ND2</td>
</tr>
</tbody>
</table>

Field Contents

SID  Set identification number (Unique integer > 0)

METHOD  Method of complex eigenvalue extraction, one of the strings 'INV' or 'HESS'

  INV - Inverse power method

  HESS - Upper Hessenberg method

NORM  Method for normalizing eigenvectors, one of the strings 'MAX' or 'POINT'

  MAX - Normalize to a unit value for the real part and a zero value for the imaginary part, the component having the largest magnitude.

  POINT - Normalize to unit value of the component G, C (defaults to 'MAX' if point is not defined)

G  Grid or scalar point identification number (Required if and only if NORM = POINT (Integer > 0))

C  Component number (Required if and only if NORM = POINT and G is a geometric grid point) (0 < Integer < 6)

E  Convergence test (Real, Default = 10^-6)

PA1, QA1, PB1, QB1  Two complex points defining a line in the complex plane (Real)

W  Width of region in complex plane (Real > 0)

NE1  Estimated number of roots in each region (Integer > 0)

ND1  Desired number of roots in each region (Default is 3*NE1) (Integer > 0)
Remarks:

1. Each continuation entry defines a rectangular search region. Any number of regions may be used and they may overlap. Roots in overlapping regions will not be extracted more than once.

2. The complex eigenvalue extraction data entries must be selected in Solution Control (METHOD=SID).

3. The units of P, Q, and \( w \) are radians per unit time.

4. At least one continuation entry is required.

5. For the Upper Hessenberg method, \( nd1 \) controls the number of vectors computed. Only one continuation entry is considered and the \((P, Q)\) pairs, along with the parameters \( W1 \) and \( NE1 \) are ignored. All eigenvalues are computed for this method.

6. If \((P, Q)\) pairs and parameters \( W1 \) and \( NE1 \) are provided, and insufficient memory exists for the Upper Hessenberg method, ASTROS will switch to the Inverse power method.

7. A pair \((P, Q)\) defines a complex eigenvalue. From this pair the following may be computed:

   \[ f_N = \text{undamped frequency} = \frac{1}{2\pi} \sqrt{P^2 + Q^2} \]

   \[ \xi = \text{damping coefficient} = \frac{P}{\sqrt{P^2 + Q^2}} \]

   \[ f_D = \text{damped frequency} = f_N \sqrt{1 - \xi^2} \]

   for lightly damped systems, \( Q \) is a measure of the radian frequency and \( P \) is a measure of the damping.

8. Parameter \( Wi \) should be kept greater than 5 percent of the segment length \( Ai \) to \( Bi \) for relatively efficient processing.

9. The \( SID \) may be referenced directly in the CEIG module, or it may be entered using the \texttt{METHOD} option on the BOUNDARY command.
Input Data Entry: EIGR Real Eigenvalue Extraction Data.

Description: Defines data needed to perform real eigenvalue extraction.

Format and Example:

<table>
<thead>
<tr>
<th>EIGR</th>
<th>SID</th>
<th>METHOD</th>
<th>F1</th>
<th>F2</th>
<th>NE</th>
<th>ND</th>
<th>E</th>
<th>CONT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONT</td>
<td>NORM</td>
<td>G</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

EIGR 13 INV 1.9 15.6 12 CONT

Field Contents

SID: Set identification number (Unique integer > 0)

METHOD: Method of eigenvalue extraction, one of the character values, INV, SINV, GIV, or MGIV
  INV - Inverse power method
  SINV - Inverse power method with sturm sequence checks
  GIV - Given's method
  MGIV - Modified Given's method

EIGR - Frequency range of interest Frequency range of interest (Real > 0.0). If ND is not blank, eigenvectors are found whose natural frequencies lie in the range between F1 and F2.

NE: Estimate of number of roots in range (Required)

ND: Desired number of roots. Desired number of eigenvectors. (Integer > 0). If ND is blank or aero, the number of eigenvectors is determined from Φ1 and Φ2. (Default = 0)

E: Convergence test (Real, Default = 10^-6)

NORM: Method for normalizing eigenvectors, one of the character values, MASS, MAX, or POINT
  MASS - Normalize to unit value of the generalized mass (Default)
  MAX - Normalize to unit value of the largest component in the analysis set
  POINT - Normalize to unit value of the component defined by G, C (defaults to "MAX" if point is not defined)

G: Grid or scalar point identification number (Required only if NORM = "POINT") (Integer > 0)

C: Component number (One of the integers 1-6) (Required only if NORM = "POINT" and G is a geometric grid point)

Remarks:
1. Real eigenvalue extraction data sets must be selected in Solution Control (Method = SID) to be used.
2. The units of F1 and F2 are cycles per unit time.
3. The continuation entry is not required. MAX normalization is then used.

4. If METHOD = "GIV", all eigenvalues are found.

5. If METHOD = "GIV", the mass matrix for the analysis set must be positive definite. Singularities or near-singularities of the mechanism type in the mass matrix will produce poor numerical stability for the GIV method.

6. If NORM=MAX, components that are not in the analysis set may have values larger than unity.

7. If NORM=POINT, the selected component should be in the analysis set. The program uses MAX when it is not in the analysis set.

8. The desired number of roots (ND) includes all roots previously found, such as rigid body modes determined with the use of the SUPORT entry.

9. The "SINV" method will actually be used for requested "INV" method.
Input Data Entry: ELEMLIST

Description: Defines a list of elements for which outputs are desired.

Format and Example:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELEM</td>
<td>SID</td>
<td>ETYPE</td>
<td>EID1</td>
<td>EID2</td>
<td>EID3</td>
<td>EID4</td>
<td>EID5</td>
<td>EID6</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Alternate Form:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELEM</td>
<td>SID</td>
<td>ETYPE</td>
<td>EID1</td>
<td>THRU</td>
<td>EID2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

<table>
<thead>
<tr>
<th>SID</th>
<th>Set identification number referenced by Solution Control. (Integer &gt; 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETYPE</td>
<td>Character input identifying the element type. One of the following:</td>
</tr>
<tr>
<td></td>
<td>BAR</td>
</tr>
<tr>
<td></td>
<td>CONM2</td>
</tr>
<tr>
<td></td>
<td>ELAS</td>
</tr>
<tr>
<td></td>
<td>IHEX1</td>
</tr>
<tr>
<td></td>
<td>IHEX2</td>
</tr>
<tr>
<td></td>
<td>IHEX3</td>
</tr>
<tr>
<td></td>
<td>MASS</td>
</tr>
</tbody>
</table>

| EID1 | Element identification number (Integer > 0 or blank) |

Remarks:

1. In order to be used, the SID must be referenced by Solution Control.
2. If the alternate form is used, EID2 must be greater than or equal to EID1.
3. Nonexistent elements may be referenced and will result in no error message.
4. Any number of continuations is allowed.
Input Data Entry: **ELIST**

Description: Defines elements associated with a design variable.

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELIST</td>
<td>LINKID</td>
<td>ETYP</td>
<td>EID1</td>
<td>EID2</td>
<td>EID3</td>
<td>EID4</td>
<td>EID5</td>
<td>EID6</td>
<td>CONT</td>
</tr>
<tr>
<td>EID7</td>
<td>EID8</td>
<td>EID9</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ELIST</td>
<td>6</td>
<td>CROD</td>
<td>12</td>
<td>14</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Alternate form:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELIST</td>
<td>LINKID</td>
<td>ETYP</td>
<td>EID1</td>
<td>THRU</td>
<td>EID2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field | Contents
--- | ---
LINKID | Element list identifier (Integer > 0)
ETYP | Element type associated with this list (e.g., CROD)
EID1, EID2, EID3 | Element entry identifications (Integer > 0, or blank)

Remarks:

1. Allowable ETYPES are: CROD, CONROD, CSHEAR, CQDMEM1, CQUAD4, CTRIA3, CTMEN, CBAR, CELAS1, CELAS2, CONM2, CMASS1 and CMASS2.
2. If the alternate form is used, EID2 must be greater than or equal to EID1.
3. All elements listed as ELIST entries for a particular LINKID, will be designed by (linked to) that design variable referencing the ELIST LINKID.
Input Data Entry: **EPOINT** Extra Point List

**Description:** Defines extra points of the structural model for use in dynamics problems.

**Format and Example:**

```
1 2 3 4 5 6 7 8 9 10
EPOINT SETID ID1 ID2 ID3 ID4 ID5 ID6 ID7 CONT
CONT ID8 ID9 -etc-
EPOINT 1000 3 18 1 4 16 2
```

**Alternate Form:**

```
1 2 3 4 5 6 7 8 9 10
EPOINT SETIC ID1 THRU ID2
```

**Field** | **Contents**
--- | ---
SETID | Extra point sets identification numbers. (Integer > 0)
IDi | Extra point identification number (Integer > 0)

**Remarks:**

1. The extra point set identification is selected on the **BOUNDARY** entry. All extra points defined with this **SETID** will be used in dynamic analyses in the boundary condition.

2. All extra point identification numbers must be unique with respect to all other structural and scalar points.

3. This entry is used to define coordinates used in transfer function definitions (see **TF** entry) and Direct Matrix input.

4. If the alternate form is used, ID2 must be greater than or equal to ID1.
Input Data Entry: FFT

Description: Defines parameters for controlling the Fast Fourier Transformation (FFT) during time domain response analysis.

Format and Example:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFT</td>
<td>SID</td>
<td>TIME</td>
<td>NT</td>
<td>RDELTF</td>
<td>RF</td>
<td>FRIM</td>
<td>OTYPE</td>
<td>FLIM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

|   | FFT | 3 | 20. | 1024 | 1.0 |     |     |     |    |    |

Field Contents

SID | FFT set identification number (Integer > 0)
TIME | Length of time period to be analyzed (Real > 0.0)
NT | Number of time points to be used for the FFT (Integer ≥ 2)
RDELTF | Ratio of incremental frequency (del F) to 1 / T. See remarks 4 and 6. (Default = 1.0, Real > 0.0)
RF | Ratio of total frequency duration (F) to NT / 2*T. See remarks 5 and 6. (Default = 1.0, Real > 0.0)
FRIM | Frequency response interpolation method. Character string LINEAR or CUBIC. Default is LINEAR.
OTYPE | Type of response to be output. Character string TIME, FREQ or BOTH. Default is TIME.
FLIM | Frequency load interpolation method. Character string LINEAR or CUBIC. Default is LINEAR.

Remarks:

1. SID must be selected by a FFT option on a TRANSIENT command in solution control.
2. TIME is the period for periodic dynamic loads defined in the time domain. For non-periodic loads, T is the total time duration of the excitation plus any quiet portion desired for response decay. T may be larger than the time duration defined by TLOAD1 or TLOAD2 data, in which case the forcing function will be automatically set to zero for the additional time.
3. NT should be a power of 2; i.e., NT = 2**m, m = 1,2,...; or NT = 2, 4, 8,... If NT is not a power of 2, it will be automatically set to the next highest power of 2 value.
4. The incremental frequency, Δ F, required by the FFT algorithm, is 1 / T. The value of Δ F may be adjusted by the user with the RDELTF factor. However, the most accurate results are normally obtained with the default case of RDELTF = 1.0.
5. The frequency duration required by the FFT algorithm is F = NT / 2*T. This is the frequency duration used when the default value of RF = 1.0 is used. If RF < 1.0, the response between RF and 1.0 is set to zero when using the inverse Fourier transform to compute time domain responses.
6. The frequency list used in the frequency response calculations is generated using a constant incremental frequency of del F = RDELTF * Δ F, and the total frequency duration is F = RF * F.
Input Data Entry: FLFACT Aerodynamic Physical Data

Description: Used to specify density ratios, velocity lists, Mach numbers, and reduced frequencies for FLUTTER analysis.

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLFACT</td>
<td>SID</td>
<td>F1</td>
<td>F2</td>
<td>F3</td>
<td>F4</td>
<td>F5</td>
<td>F6</td>
<td>F7</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>F8</td>
<td>F9</td>
<td>-etc-</td>
<td> </td>
<td> </td>
<td> </td>
<td> </td>
<td> </td>
<td> </td>
</tr>
<tr>
<td>FLFACT</td>
<td>97</td>
<td>.3</td>
<td>.7</td>
<td>3.5</td>
<td> </td>
<td> </td>
<td> </td>
<td> </td>
<td> </td>
</tr>
</tbody>
</table>

Field Contents

SID Set identification number (Integer > 0).
Fi Aerodynamic factor (Real).

Remarks:

1. Only the factors selected by a FLUTTER data entry will be used.
2. Embedded blank fields are forbidden.
3. Parameters must be listed in the order in which they are to be used within the looping of FLUTTER analysis.
4. All FLFACT entries having the same SETID will be treated as a single set.
**Input Data Entry:** FLUTTER Aerodynamic FLUTTER Data

**Description:** Defines data needed to perform FLUTTER analysis.

**Format and Example:**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLUTTER</td>
<td>SID</td>
<td>METHOD</td>
<td>DENS</td>
<td>MACH</td>
<td>VEL</td>
<td>MLIST</td>
<td>KLIST</td>
<td>EFFID</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>SYMXZ</td>
<td>SYMXY</td>
<td>EPS</td>
<td>CURFIT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| FLUTTER | 19 | PK | 119 | 0.85 | 319 | 10 |   |   |   |    |

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>Set identification number (Integer &gt; 0)</td>
</tr>
<tr>
<td>METHOD</td>
<td>FLUTTER analysis method, &quot;PK&quot; for PK-method (Text)</td>
</tr>
<tr>
<td>DENS</td>
<td>Identification number of an FLFACT set specifying density ratios to be used in FLUTTER analysis (Integer &gt; 0).</td>
</tr>
<tr>
<td>MACH</td>
<td>Mach number to be used in the FLUTTER analysis (Real ≥ 0.0)</td>
</tr>
<tr>
<td>VEL</td>
<td>Identification number of an FLFACT set specifying velocities to be used in the FLUTTER analysis. (Integer &gt; 0).</td>
</tr>
<tr>
<td>MLIST</td>
<td>Identification number of a SET1 set specifying a list of normal modes to be omitted from the FLUTTER analysis (Integer &gt; 0, or blank).</td>
</tr>
<tr>
<td>KLIST</td>
<td>Identification number of an FLFACT set specifying a list of hard point reduced frequencies for the given Mach number for use in the FLUTTER analysis (See Remark 4) (Integer ≥ 0, or blank)</td>
</tr>
<tr>
<td>EFFID</td>
<td>Identification number of a CONEFFF set specifying control surface effectiveness values (See Remark 5) (Integer ≥ 0, or blank)</td>
</tr>
<tr>
<td>SYMXZ, SYMXY</td>
<td>Symmetry flags associated with the aerodynamics (See Remark 6) (Integer)</td>
</tr>
<tr>
<td>EPS</td>
<td>Convergence parameter for FLUTTER eigenvalue (Real, Default = 10⁻⁵)</td>
</tr>
<tr>
<td>CURFIT</td>
<td>Type of curve fit to be used in the PK FLUTTER analysis. One of LINEAR, QUAD, CUBIC, or ORIG (See Remarks 7, 8, and 9) (Text, Default = LINEAR)</td>
</tr>
</tbody>
</table>

**Remarks:**

1. The FLUTTER data entry must be selected in the Solution Control packet. Only those Mach numbers and symmetries selected in Solution will be processed in the UNSTEADY aerodynamic preface.
2. The density is given by \( \rho \times \rho_{ref} \), where \( \rho_{ref} \) is the reference value given on the AERO Bulk Data entry. and \( \rho \) is the density ratio from the FLFACT entry.
3. If the MLIST is blank or zero, all computed normal modes will be retained in the FLUTTER analysis.
4. If the KLIST is blank or zero, all "hard point" k values (those on the MKAEROi entries) associated with the Mach number/symmetries on the FLUTTER entry will be used in the interpolation of the aerodynamics. Specifying a subset may be used to improve the ORIG interpolation. Those hard point k values nearest in value to those listed on the FLFACT will be used. No duplicate hard point k's will be used and no errors will be printed.

5. If the EFFID is blank or zero, no effectiveness corrections will be made.

6. The symmetry flags are used to select the appropriate unsteady aerodynamic matrices generated from the list on the MKAEROi entries.

7. The LINEAR, QUAD, and CUBIC fits are separate first, second and third order, respectively, fits of the real and complex terms of the generalized aerodynamic matrix at each hard point k. Only the closest 2, 3 or 4, respectively, k's are utilized for each fit and LINEAR fitting is used off the ends of the hard point KLIST. The program automatically reduces the order of the fit if too few points are available for the higher order fit (e.g., CUBIC becomes QUAD if only 3 k's are used in the KLIST) (Refer to the Version 9.0 Release Notes for more information).

8. The ORIGinal fit (documented in the Theoretical Manual) is a cubic fit over all the hard point k's. Its use is not recommended since it tends to experience numerical problems for any but small k ranges and small numbers of k's.

9. For all fitting options, the generalized aerodynamic matrices are normalized by the hard point k value before fitting, as documented on the Theoretical Manual.
**Input Data Entry:** FORCE Static Load

**Description:** Defines a static load at a grid point by specifying a vector.

**Format and Example:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORCE</td>
<td>SID</td>
<td>G</td>
<td>CID</td>
<td>F</td>
<td>N1</td>
<td>N2</td>
<td>N3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FORCE</td>
<td>2</td>
<td>5</td>
<td>6</td>
<td>2.9</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field** | **Contents**
--- | ---
SID | Load set identification number (Integer > 0)
G | Grid point identification number (Integer > 0)
CID | Coordinate system identification number (Integer ≥ 0, or blank) (Default = 0)
F | Scale factor (Real)
N1 | Components of a vector measured in the coordinate system defined by CID (Real; must have at least one nonzero component)

**Remarks:**

1. The static load applied to grid point G is given by

   \[
   \mathbf{f} = F \mathbf{n}
   \]

   where \( \mathbf{n} \) is the vector defined in Fields 6, 7 and 8.

2. A CID of zero references the basic coordinate system.
Input Data Entry: FORCE1 Static Load, Alternate Form 1

Description: Used to define a static load by specification of a value and two grid points which determine the direction.

Format and Example:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>SID</td>
<td>G</td>
<td>F</td>
<td>G1</td>
<td>G2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FORCE1</td>
<td>6</td>
<td>13</td>
<td>-2.93</td>
<td>16</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

SID Load set identification number (Integer > 0)
G Grid point identification number (Integer > 0)
F Value of load (Real)
G1 Grid point identification numbers (Integer > 0; G1 ≠ G2)

Remarks:
1. The direction of the force is determined by the vector from G1 to G2.
Input Data Entry: \textbf{FREQ}

\textbf{Description:} Defines a set of frequencies to be used in the solution of frequency response problems.

\textbf{Format and Example:}

\begin{tabular}{|c|c|c|c|c|c|c|c|c|c|}
\hline
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
\hline
FREQ & SID & F1 & F2 & F3 & F4 & F5 & F6 & F7 & CONT \\
\hline
CONT & F8 & F9 & \textit{-etc-} & & & & & CONT & \\
\hline
FREQ & 3 & 2.98 & 3.05 & 17.9 & 21.3 & 25.6 & 28.8 & 31.2 & ABC \\
+BC & 29.2 & 22.4 & 19.3 & & & & & & \\
\hline
\end{tabular}

\textbf{Field} \hspace{2cm} \textbf{Contents}

\begin{itemize}
\item \textbf{SID} \hspace{1cm} Frequency set identification number (Integer \(> 0\))
\item \textbf{Fi} \hspace{1cm} Frequency value (Real \(\geq 0.0\))
\end{itemize}

\textbf{Remarks:}

1. The units for the frequencies are cycles per unit time.
2. Frequency sets must be selected by the Solution Control (\texttt{FSTEP=SID}) to be used.
3. All \texttt{FREQ}, \texttt{FREQ1} and \texttt{FREQ2} entries with the same frequency set identification numbers will be used. Duplicate frequencies will be ignored. \(f_N\) and \(f_{N-1}\) are considered duplicated if

\[ |f_N - f_{N-1}| < 10^{-6} \ast (f_{MAX} - f_{MIN}) \]
Input Data Entry: FREQ1

Description: Defines a set of frequencies to be used in the solution of frequency response problems by specification of a starting frequency, frequency increment, and number of increments desired.

Format and Example:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>FREQ1</td>
<td>SID</td>
<td>F1</td>
<td>DF</td>
<td>NDF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FREQ1</td>
<td>6</td>
<td>2.9</td>
<td>0.5</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

<table>
<thead>
<tr>
<th>SID</th>
<th>Frequency set identification number (Integer &gt; 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>First frequency in set (Real ≥ 0.0)</td>
</tr>
<tr>
<td>DF</td>
<td>Frequency increment (Real &gt; 0.0)</td>
</tr>
<tr>
<td>NDF</td>
<td>Number of frequency increments (Integer &gt; 0)</td>
</tr>
</tbody>
</table>

Remarks:

1. The units for the frequency F1 and the frequency increment DF are cycles per unit time.
2. The frequencies defined by this entry are given by:

\[ f_i = F1 + (i-1) \times DF, \quad i = 1, NDF + 1 \]

3. Frequency sets must be selected by the Solution Control (FSTEP=SID) to be used.
4. All FREQ, FREQ1 and FREQ2 entries with the same frequency set identification numbers will be used. Duplicate frequencies will be ignored. FN and FN-1 are considered duplicated if

\[ |f_N - f_{N-1}| < 10^{-5} \times (f_{MAX} - f_{MIN}) \]

259
**Input Data Entry:**  
**FREQ2**

**Description:** Defines a set of frequencies to be used in the solution of frequency response problems by specification of a starting frequency, final frequency, and number of logarithmic increments desired.

**Format and Example:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>FREQ2</td>
<td>SID</td>
<td>F1</td>
<td>F2</td>
<td>NF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FREQ2</td>
<td>6</td>
<td>1.0</td>
<td>8.0</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field** | **Contents**
--- | ---
SID | Frequency set identification number (Integer > 0)
F1 | First frequency (Real > 0.0)
F2 | Last frequency (Real > 0.0; F2 > F1)
NF | Number of logarithmic intervals (Integer > 0)

**Remarks:**

1. The units for the frequencies F1 and F2 are cycles per unit time.
2. The frequencies defined by this entry are given by:
   \[
   f_i = F_1 \cdot e^{i-1} \cdot d
   \]
   where,
   \[
   d = \left( \frac{1}{NF} \right) \ln \left( \frac{F_2}{F_1} \right)
   \]
   For the example shown, the list of frequencies will be 1.0, 1.4142, 2.0, 2.8284, 4.0, 5.6569 and 8.0 cycles per unit time.
3. Frequency sets must be selected by the Solution Control (FSTEP=SID) to be used.
4. All FREQ, FREQ1 and FREQ2 entries with the same frequency set identification numbers will be used. Duplicate frequencies will be ignored. FN and FN-1 are considered duplicated if:
   \[
   |F_N - F_{N-1}| < 10^{-6} \cdot (f_{\text{MAX}} - f_{\text{MIN}})
   \]
Input Data Entry: **FREQLIST**

**Description:** Defines a list of frequencies for which outputs are defined.

**Format and Example:**

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FREQLIST</td>
<td>SID</td>
<td>FREQ1</td>
<td>FREQ2</td>
<td>FREQ3</td>
<td>FREQ4</td>
<td>FREQ5</td>
<td>FREQ6</td>
<td>FREQ7</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>FREQ8</td>
<td>FREQ9</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| FREQLIST | 100 | 10.0 | 20.0 | 50.0 | 100.0 |   |   |   |   |

**Field** | **Contents**
---|---
SID | Set identification number referenced by Solution Control (Integer > 0)
FREQ_i | Frequency (in Hertz) at which outputs are desired (Real)

**Remarks:**

1. In order to be used, the SID must be referenced by Solution Control.
2. The nearest frequency to FREQ, either above or below, which was used in the Frequency Response analysis will be used to satisfy the output requests.
3. Any number of continuations is allowed.
**Input Data Entry**  
**GDVLIST**  
Global Design Variable List

**Description:**  
Defines a list of global design variables for which outputs are desired.

**Format and Example:**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDVLIST</td>
<td>SID</td>
<td>GDVID1</td>
<td>GDVID2</td>
<td>GDVID3</td>
<td>GDVID4</td>
<td>GDVID5</td>
<td>GDVID6</td>
<td>GDVID7</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>GDVID8</td>
<td>GDVID9</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Alternate Form**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDVLIST</td>
<td>SID</td>
<td>GDVID1</td>
<td>THRU</td>
<td>GDVID2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field**  
**SID**  
Set identification number referenced by Solution Control (Integer > 0)

**GDVID**  
Global design variable identification number (Integer > 0 or blank)

**Remarks:**

1. In order to be used, the **SID** must be referenced by Solution Control.

2. If the alternate form is used, **GDVID2** must be greater than or equal to **GDVID1**.

3. Nonexistent global design variables may be referenced and will result in no error message.

4. Any number of continuations is allowed, except when using the alternate form, which allows no continuations.
Input Data Entry  GPWG  Weight Generator Data

Description: Contains definition of the location about which to perform grid point weight generation

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPWG</td>
<td>GID/XO</td>
<td>Y0</td>
<td>Z0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

GPWG 10

Field |

<table>
<thead>
<tr>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>GID</td>
</tr>
<tr>
<td>X0</td>
</tr>
<tr>
<td>Y0</td>
</tr>
<tr>
<td>Z0</td>
</tr>
</tbody>
</table>

Remarks:

1. Either a grid point identification number or the basic x, y, z components of the reference point may be given.

2. If no GPWG data entry exists, the grid point weight generation will be computed about the origin of the basic coordinate system.

3. If more than one GPWG entry exists, the first one appearing in the sorted bulk data echo will be used.
**Input Data Entry:** GRAV Gravity Vector

**Description:** Used to define gravity vectors for use in determining gravity loading for the structural model.

**Format and Example:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRAV</td>
<td>SID</td>
<td>CID</td>
<td>G</td>
<td>N1</td>
<td>N2</td>
<td>N3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| GRAV | 1 | 3 | 32.2 | 0.0 | 0.0 | -1.0 |   |   |    |

**Field** | **Contents**
--- | ---
SID | Set identification number (Integer > 0)
CID | Coordinate system identification number (Integer ≥ 0)
G | Gravity vector scale factor (Real ≠ 0.0)
N1 | Gravity vector components (Real; at least one nonzero component)

**Remarks:**
1. The gravity vector is defined by \( \mathbf{g} = G \mathbf{N} \). The direction of \( \mathbf{g} \) is the direction of free fall.
2. A CID of zero references the basic coordinate system.
3. Gravity loads may be combined with "simple loads" (e.g., **FORCE, MOMENT**) by specification on a **LOAD** entry or by **GRAV** = SID. Gravity loads with the same SID as simple load entries will not be used unless referenced by one of these methods.
4. Load sets must be selected in Solution Control to be used.
5. The units of \( G \) should be length/sec² in consistent length units.
Input Data Entry: **GRDSET**  
**Grid Point Default**

**Description:** Defines default options for Fields 3, 7, and 8 of all GRID entries.

**Format and Example:**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRDSET</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRDSET</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>32</td>
<td>3456</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field** | **Contents**
---|---
CP | Identification number of coordinate system in which the location of the grid point is defined (Integer \( \geq 0 \))
CD | Identification number of coordinate system in which the displacements are measured at grid point (Integer \( \geq 0 \))
PS | Permanent single-point constraints associated with grid point (any of the digits 1 through 6 with no embedded blanks) (Integer \( \geq 0 \))

**Remarks:**

1. The contents of Fields 3, 7, or 8 of this entry are assumed for the corresponding fields of any GRID entry whose Field 3, 7, and 8 are blank. If any of these fields on the GRID entry are blank, the default option defined by this entry occurs for that field. If no permanent single-point constraints are desired or one of the coordinate systems is basic, the default may be overridden on the GRID entry by making one of theFields 3, 7, or 8 zero (rather than blank). Only one **GRDSET** entry may appear in the user's Bulk Data packet.

2. The primary purpose if this entry is to minimize the burden of preparing data for problems with a large amount of repetition (e.g., two-dimensional pinned-joint problems).

3. At least one of the entries CP, CD, or PS must be nonzero.
Input Data Entry: GRID  

Description: Defines the location of a geometric grid point of the structural model, the directions of its displacement, and its permanent single-point constraints.

Format and Example:

<table>
<thead>
<tr>
<th>ID</th>
<th>CP</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>CD</th>
<th>PS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>GRID</td>
<td>ID</td>
<td>CP</td>
<td>X1</td>
<td>X2</td>
<td>X3</td>
<td>CD</td>
</tr>
<tr>
<td>GR</td>
<td>2</td>
<td>3</td>
<td>1.0</td>
<td>2.0</td>
<td>3.0</td>
<td>315</td>
</tr>
</tbody>
</table>

Field Contents

ID  
Grid point identification number (Integer > 0)

CP  
Identification number of coordinate system in which the location of the grid point is defined (Integer > 0 or blank)

Xi  
Location of the grid point in coordinate system CP (Real)

CD  
Identification number of coordinate system in which displacements, degrees of freedom, constraints, and solution vectors are defined at the grid point (Integer > 0 or blank)

PS  
Permanent single-point constraints associated with grid point (any of the digits 1-6 with no embedded blanks) (Integer > 0 or blank)

Remarks:

1. All grid point identification numbers must be unique with respect to all other structural and scalar points.

2. The meaning of $x_1$, $x_2$, and $x_3$ depend on the type of coordinate system CP, as follows:

<table>
<thead>
<tr>
<th>TYPE</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rectangular</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
<tr>
<td>Cylindrical</td>
<td>R</td>
<td>$\theta$ (deg)</td>
<td>Z</td>
</tr>
<tr>
<td>Spherical</td>
<td>R</td>
<td>$\theta$ (deg)</td>
<td>$\phi$ (deg)</td>
</tr>
</tbody>
</table>

Also see CORDij entry descriptions.

3. The collection of all coordinate systems defined on all GRID entries is called the Global Coordinate System. All degrees-of-freedom, constraints, and solution vectors are expressed in the Global Coordinate System.
Input Data Entry:  GRIDLIST

Description:  Defines a list of points at which outputs are desired.

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRIDLIST</td>
<td>SID</td>
<td>GID1</td>
<td>GID2</td>
<td>GID3</td>
<td>GID4</td>
<td>GID5</td>
<td>GID6</td>
<td>GID7</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>GID8</td>
<td>GID9</td>
<td>etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRIDLIST</td>
<td>100</td>
<td>1001</td>
<td>1010</td>
<td>1020</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Alternate Form:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRIDLIST</td>
<td>SID</td>
<td>GID1</td>
<td>THRU</td>
<td>GID2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field  Contents

<table>
<thead>
<tr>
<th>SID</th>
<th>Set identification number referenced by Solution Control (Integer &gt; 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GID1</td>
<td>Grid, scalar or extra point id at which outputs are desired (Integer &gt; 0)</td>
</tr>
</tbody>
</table>

Remarks:
1. In order to be used, the SID must be referenced by Solution Control.
2. If the alternate form is used, GID2 must be greater than or equal to GID1.
3. Nonexistent points may be referenced and will result in no error message.
4. Any number of continuations is allowed.
Input Data Entry:  GUST  Aerodynamic Gust Load Description

Description:  Defines a stationary vertical gust for use in aeroelastic analysis.

Format and Example:

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>GUST</td>
<td>SID</td>
<td>GLOAD</td>
<td>WG</td>
<td>XO</td>
<td>V</td>
<td>QDP</td>
<td>MACH</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>SYMXZ</td>
<td>SYMXY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>GUST</td>
<td>133</td>
<td>61</td>
<td>1.0</td>
<td>0.0</td>
<td>1.4</td>
<td>13.5</td>
<td>0.9</td>
<td>ABC</td>
</tr>
<tr>
<td>+BC</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field contents

SID  Gust set identification number (Integer > 0)
GLOAD  The SID of a TLOAD or RLOAD data entry which defines the time or frequency dependence (Integer > 0)
WG  Scale factor (gust velocity/forward velocity) for gust velocity (Real > 0)
XO  Location of reference plane in aerodynamic coordinates (Real)
V  Velocity of vehicle (Real > 0.0)
QDP  Dynamic pressure (Real > 0.0)
MACH  Mach number (Real > 0.0)
SYMXZ, SYMXY  Symmetry flags (Integer)

Remarks:

1. The GUST entry is selected as a discipline option for FREQUENCY or TRANSIENT in Solution Control.
2. The gust angle is in the +z direction of the aerodynamic coordinate system. The value is,

\[ WG = T \left| t - \frac{x - xo}{v} \right| \]

where T is the tabular function.
3. The symmetry flags will be used to select the appropriate unsteady aerodynamic matrices from the list of m-k pairs for each symmetry option given on the MKAEROi entries.
Input Data Entry:  IC  Transient Initial Condition

Description:  Defines values for the initial conditions of coordinates used in transient analysis. Both displacement and velocity values may be specified at independent coordinates of the structural model.

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC</td>
<td>SID</td>
<td>G</td>
<td>C</td>
<td>UO</td>
<td>VO</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IC</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>5.0</td>
<td>-6.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field  Contents

SID:  Set identification number (Integer > 0)

G:  Grid or scalar or extra point identification number (Integer > 0)

C:  Component number (blank or zero for scalar or extra points, any one of the digits 1 through 6 for a grid point)

UO:  Initial displacement value (Real)

VO:  Initial velocity value (Real)

Remarks:

1. Transient initial condition sets must be selected in the Solution Control (IC=SID) to be used.

2. If no IC set is selected, all initial conditions are assumed zero.

3. Initial conditions for coordinates not specified on IC entries will be assumed zero.

4. Initial conditions may be used only in direct formulation. In a modal formulation the initial conditions are all zero.
Input Data Entry ITERLIST Iteration List

Description: Defines a list of iteration steps for which outputs are desired.

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITERLIST SID ITER ITER ITER ITER ITER ITER ITER CONT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONT ITER ITER -etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Alternate Form:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITERLIST SID .ITER THRU ITER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

<table>
<thead>
<tr>
<th>SID</th>
<th>Set identification number referenced by Solution Control. (Integer &gt; 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITER</td>
<td>Iteration step number. (Integer &gt; 0 or blank)</td>
</tr>
</tbody>
</table>

Remarks:

1. In order to be used, the SID must be referenced by Solution Control.

2. Nonexistent iteration steps may be referenced and will result in no error message.

3. Any number of continuations is allowed, except when using the alternate form, which allows no continuations.
Input Data Entry: JSET
Select Coordinates for the j-set

Description: Defines coordinates (degrees of freedom) that the user desires to use in the computation of inertia relief mode shape in Dynamic Reduction.

Format and Example:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSET</td>
<td>SETID</td>
<td>ID</td>
<td>C</td>
<td>ID</td>
<td>C</td>
<td>ID</td>
<td>C</td>
<td>CONT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>ID</td>
<td>C</td>
<td>ID</td>
<td>C</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

SETID: The set identification number of the INERTIA set. (Integer > 0)

ID: Grid or scalar point identification number (Integer > 0).

C: Component number, zero or blank for scalar points, any unique combination of the digits 1 through 6 for grid points. (Integer)

Remarks:
1. Coordinates specified on this entry form members of a mutually exclusive set. They may not be specified on other entries that define mutually exclusive sets.
2. When JSET and/or JSET1 entries are present, all degrees of freedom not otherwise constrained will be placed on the o-set.
3. Use of JSET in dynamic reduction:
   a. JSET defines the structural/nonstructural j-set degrees of freedom (inertia relief shapes). An alternate input format is provided by the JSET1 entry.
   b. The SID is selected by the Solution Control Command BOUNDARY INERTIA = n.
   c. Use "0" as the grid point identification number to select the origin of the basic coordinate system as the j-set degrees of freedom.
4. Any number of continuations are allowed.
Input Data Entry: **JSET1** Select Coordinates for the j-set, Alternate Form

**Description:** Defines coordinates (degrees of freedom) that the user desires to use in the computation of inertia relief mode shape(s) in Dynamic Reduction.

**Format and Example:**

<table>
<thead>
<tr>
<th></th>
<th>JSET1</th>
<th>SETID</th>
<th>C</th>
<th>GID1</th>
<th>GID2</th>
<th>GID3</th>
<th>GID4</th>
<th>GID5</th>
<th>GID6</th>
<th>CONT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td></td>
<td></td>
<td>GID7</td>
<td>GID8</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>JSET1</th>
<th>SETID</th>
<th>C</th>
<th>GID1</th>
<th>GID2</th>
<th>GID3</th>
<th>GID4</th>
<th>GID5</th>
<th>GID6</th>
<th>GID7</th>
<th>GID8</th>
<th>-etc-</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+bc</td>
<td></td>
<td></td>
<td>7</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Alternate Form:**

<table>
<thead>
<tr>
<th></th>
<th>JSET1</th>
<th>SETID</th>
<th>C</th>
<th>GID1</th>
<th>GID2</th>
<th>GID3</th>
<th>GID4</th>
<th>GID5</th>
<th>GID6</th>
<th>GID7</th>
<th>GID8</th>
<th>-etc-</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field Contents**

<table>
<thead>
<tr>
<th>SetID</th>
<th>The INERTIA set identification number</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Component number (any unique combination of the digits 1 through 6 (with no embedded blanks) when point identification numbers are grid points; must be blank or zero if point identification numbers are scalar points.</td>
</tr>
<tr>
<td>GIDI</td>
<td>Grid or scalar point identification numbers (Integer &gt; 0).</td>
</tr>
</tbody>
</table>

**Remarks:**

1. Coordinates specified on this entry form members of a set that is exclusive from other sets defined by bulk data entries.
2. When **JSET** and/or **JSET1** entries are present, all degrees of freedom not otherwise constrained will be placed in the o-set.
3. If the alternate form is used, all points in the sequence ID1 through ID2 are required to exist and ID2 must be greater than or equal to ID1.
4. Use of **JSET1** in dynamic reduction:
   a. **JSET1** defines the structural and nonstructural j-set degrees of freedom (inertia relief shapes). An alternate input format is provided by the **JSET** entry.
   b. The **SID** is selected by Solution Control Command **BOUNDARY INERTIA = n**.
   c. Use "0" as the grid point identification number to select the origin of the basic coordinate system as the j-set degrees freedom.
Input Data Entry  LDVLIST  Local Design Variable List

Description: Defines a list of local design variables for which outputs are desired.

Format and Example:

```
1  2  3  4  5  6  7  8  9  10
LDVLIST  SID  ETYPE  LAYER  EID1  EID2  EID3  EID4  EID5  CONT
CONT   EID6  EID7  -etc-
```

Alternate Form

```
1  2  3  4  5  6  7  8  9  10
LDVLIST  SID  ETYPE  LAYER  EID1  THRU  EID2
```

Field  Contents

- **SID**: Set identification number referenced by Solution Control. (Integer > 0)
- **ETYPE**: Character input identifying the element type. One of the following:
  - **BAR**
  - **MASS**
  - **ROD**
  - **TRM**
  - **CONM2**
  - **QDMEM1**
  - **SHEAR**
  - **ELAS**
  - **QUAD4**
  - **TRIA3**
- **LAYER**: Layer number if element is composite laminate. (Integer > 0 or blank)
- **EIDi**: Element identification number. (Integer > 0 or blank)

Remarks:

1. In order to be used, the **SID** must be referenced by Solution Control.
2. If the alternate form is used **EID2** must be greater than or equal to **EID1**.
3. Nonexistent elements may be referenced and will result in no error message.
4. If a layer number is omitted for a composite laminate element then all layers in that element will be selected.
5. Any number of continuations is allowed.
**Input Data Entry:** LOAD  Static Load Combination (Superposition)

**Description:** Defines a static load as a linear combination of load sets defined using FORCE, MOMENT, FORCE1, MOMENT1, PLOAD, and GRAV entries.

**Format and Example:**

<table>
<thead>
<tr>
<th>LOAD</th>
<th>SID</th>
<th>S</th>
<th>Si</th>
<th>Li</th>
<th>S2</th>
<th>L2</th>
<th>S3</th>
<th>L3</th>
<th>CONT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CON1</td>
<td>S4</td>
<td>L4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| LOAD | 101 | -0.5 | 1.0 | 3   | 6.2 | 4   | 4.5 | 10  | ABC  |
| BC   | 2.3 | 115  |

**Field Contents**

<table>
<thead>
<tr>
<th>SID</th>
<th>Load set identification number (Integer &gt;0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>Scale factor (Real ≠ 0.0)</td>
</tr>
<tr>
<td>Si</td>
<td>Scale factors (Real ≠ 0.0)</td>
</tr>
<tr>
<td>Li</td>
<td>Load set identification numbers defined via data entry types enumerated above (Integer &gt; 0)</td>
</tr>
</tbody>
</table>

**Remarks:**

1. The load vector defined is given by

   \[ P = S \sum S_i P_{Li} \]

2. The Li must be unique. The remainder of the physical entry containing the last entry must be blank.

3. Load sets must be selected in the Solution Control if they are to be applied to the structural model.

4. A LOAD entry may not reference a set identification number defined by another LOAD entry.
Input Data Entry: **MAT1**  
Material Property Definition, Form 1

**Description:** Defines the material properties for linear, temperature-independent, isotropic materials

**Format and Example:**

```
  1  2  3  4  5  6  7  8  9  10
MAT1 MID  E  G  NU  RHO  A  TREF  GE  CONT
CONT ST   SC  SS  MCSID

MAT1  17  3.7  0.33  4.28  6.5-6  5.37-6  0.23  ABC
+B    20.4  15.4  12.4
```

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>MID</td>
<td>Material identification number (Integer &gt;0)</td>
</tr>
<tr>
<td>E</td>
<td>Young's modulus (Real &gt; 0.0, or blank)</td>
</tr>
<tr>
<td>G</td>
<td>Shear modulus (Real or blank)</td>
</tr>
<tr>
<td>NU</td>
<td>Poisson’s ratio (-1.0 &lt; Real ≤ 0.5 or blank)</td>
</tr>
<tr>
<td>RHO</td>
<td>Mass density (Real ≥ 0.0)</td>
</tr>
<tr>
<td>A</td>
<td>Thermal expansion coefficient (Real)</td>
</tr>
<tr>
<td>TREF</td>
<td>Thermal expansion reference temperature (Real)</td>
</tr>
<tr>
<td>GE</td>
<td>Structural element damping coefficient (Real)</td>
</tr>
<tr>
<td>ST, SC, SS</td>
<td>Stress limits for tension, compression, and shear (Real). (Used to compute margins of safety in certain elements).</td>
</tr>
<tr>
<td>MCSID</td>
<td>Material Coordinate System identification number (Integer &gt; 0 or blank).</td>
</tr>
</tbody>
</table>

**Remarks:**

1. The material identification number must be unique for all MAT1, MAT2, MATB, and MAT9 bulk data entries.
2. The mass density, RHO, will be used to automatically compute mass for all structural elements.
3. Weight density may be used in Field 6 if the value 1/g is entered on the CONVERT entry where g is the acceleration of gravity.
4. Either E or G must be specified (i.e., nonblank).
5. If any one of E, G, or NU is blank, it will be computed to satisfy the identity $E = 2 \times (1+NU) \times G$; otherwise, values supplied by the user will be used.
6. If E and NU or G and NU are both blank, they will both be given the values 0.0.
7. Implausible data on one or more MAT1 entries will result in a warning message. Implausible data is defined as any of $E < 0.0$ or $G < 0.0$ or $NU > 0.5$ or $NU < 0.0$ or $|1 - E/2(1+NU)G| > 0.01$ except for cases covered by Remark 6.
8. It is strongly recommended that only two of the three values E, G, and NU be input. The three values may be input independently on the MAT2 entry.
Input Data Entry: MAT2 Material Property Definition, Form 2

Description: Defines the material properties for linear, temperature-independent, anisotropic materials for two-dimensional elements.

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAT2</td>
<td>MID</td>
<td>G11</td>
<td>G12</td>
<td>G13</td>
<td>G22</td>
<td>G23</td>
<td>G33</td>
<td>RHO</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>A1</td>
<td>A2</td>
<td>A12</td>
<td>TO</td>
<td>GE</td>
<td>ST</td>
<td>SC</td>
<td>SS</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>MCSID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| MAT2 | 13 | 6.2+3 | 6.2+3 | 5.1+3 | 0.056 | ABC |
| +BC  | 6.5-6 | 6.5-6 | -500.0 | 0.002 | 20.+5 |    |

Field Contents

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>MID</td>
<td>Material identification number (Integer &gt; 0)</td>
</tr>
<tr>
<td>Gij</td>
<td>The material property matrix (Real)</td>
</tr>
<tr>
<td>RHO</td>
<td>Mass density (Real ≥ 0.0)</td>
</tr>
<tr>
<td>Ai</td>
<td>Thermal expansion coefficient vector (Real)</td>
</tr>
<tr>
<td>TO</td>
<td>Thermal expansion reference temperature (Real)</td>
</tr>
<tr>
<td>GE</td>
<td>Structural element damping coefficient (Real)</td>
</tr>
<tr>
<td>ST, SC, SS</td>
<td>Stress limits for tension, compression, and shear (Real). (Used to compute margins of safety in certain elements).</td>
</tr>
<tr>
<td>MCSID</td>
<td>Material Coordinate System identification number (Integer &gt; 0 or blank).</td>
</tr>
</tbody>
</table>

Remarks:

1. The material identification numbers must be unique for all MAT1, MAT2, MAT8, and MAT9 bulk data entries.
2. The mass density, RHO, will be used to automatically compute mass for all structural elements.
3. The convention for the Gij in Fields 3 through 8 are represented by the matrix relationship

\[
\begin{bmatrix}
\sigma_1 \\
\sigma_2 \\
\tau_{12}
\end{bmatrix} =
\begin{bmatrix}
G_{11} & G_{12} & G_{13} \\
G_{12} & G_{22} & G_{23} \\
G_{13} & G_{23} & G_{33}
\end{bmatrix}
\begin{bmatrix}
\epsilon_1 \\
\epsilon_2 \\
\gamma_{12}
\end{bmatrix} - (T - T_0)
\begin{bmatrix}
A_1 \\
A_2 \\
A_{12}
\end{bmatrix}
\]

4. 2x2 matrices (for example, transverse shear) use elements G11, G12, and G22. For this case, G33 must be blank.
5. If the MAT2 entry is referenced by the PCOMP entry, the transverse shear flexibility for the referenced laminae is zero.
6. Unlike the MAT1 entry, data from the MAT2 entry are used directly, without adjustment of equivalent E, G, or NU values.
**Input Data Entry:**  MAT8  
**Material Property Definition, Form 8**

**Description:** Defines the material property for an orthotropic material.

**Format and Example:**

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>MID</td>
<td>Material identification number (Integer &gt; 0)</td>
</tr>
<tr>
<td>E1</td>
<td>Modulus of elasticity in longitudinal direction (also defined as fiber direction or 1-direction) (Real ≠ 0.0)</td>
</tr>
<tr>
<td>E2</td>
<td>Modulus of elasticity in lateral direction (also defined as matrix direction or 2-direction) (Real ≠ 0.0)</td>
</tr>
<tr>
<td>NU12</td>
<td>Poisson's ratio ( \frac{\varepsilon_2}{\varepsilon_1} ) for uniaxial loading in 1-direction. Note that ( \nu_{21} = \frac{\varepsilon_1}{\varepsilon_2} ) for uniaxial loading in 2-direction is related to ( \nu_{12}, E_1, E_2 ) by the relation ( \nu_{12}E_2 = \nu_{21}E_1 ). (Real)</td>
</tr>
<tr>
<td>G12</td>
<td>In-plane shear modulus (Real &gt; 0.0)</td>
</tr>
<tr>
<td>G1, Z</td>
<td>Transverse shear modulus for shear in 1-Z plane (Real &gt; 0.0 or blank) (default implies infinity)</td>
</tr>
<tr>
<td>G2, Z</td>
<td>Transverse shear modulus for shear in 2-Z plane (Real &gt; 0.0 or blank) (default implies infinity)</td>
</tr>
<tr>
<td>RHO</td>
<td>Mass density (Real ≥ 0.0)</td>
</tr>
<tr>
<td>A1</td>
<td>Thermal expansion coefficient in the 1-direction (Real)</td>
</tr>
<tr>
<td>A2</td>
<td>Thermal expansion coefficient in the 2-direction (Real)</td>
</tr>
<tr>
<td>TREF</td>
<td>Thermal expansion reference temperature (Real)</td>
</tr>
<tr>
<td>Xt, Xc</td>
<td>Allowable stresses in tension and compression, respectively, in the longitudinal direction. Required if failure index is desired. (Real ≥ 0.0) (Default value for Xc is Xt)</td>
</tr>
<tr>
<td>Yt, Yc</td>
<td>Allowable stresses in tension and compression, respectively, in the transverse direction. Required if failure index is desired. (Real ≥ 0.0) (Default value for Yc is Yt)</td>
</tr>
<tr>
<td>S</td>
<td>Allowable stress for in-plane shear (Real ≥ 0.0)</td>
</tr>
<tr>
<td>GE</td>
<td>Structural damping coefficient (Real)</td>
</tr>
</tbody>
</table>
Interaction term in the tensor polynomial theory of Tsai-Wu (Real). Required if failure index or stress constraint by Tsai-Wu theory is desired and if value of $F_{12}$ is different from 0.0.

**Remarks:**

1. If $G_1$, $Z$ and $G_2$, $Z$ values specified as zero, or are not supplied, transverse shear flexibility calculations will not be performed.

2. An approximate value for $G_1$, $Z$ and $G_2$, $Z$ is the in-plane shear modulus $G_{12}$. If test data are not available to accurately determine $G_1$, $Z$ and $G_2$, $Z$ for the material and transverse shear calculations are deemed essential, the value of $G_{12}$ may be supplied for $G_1$, $Z$ and $G_2$, $Z$.

3. $x_t$, $x_c$, $x_e$, $y_e$ and $ss$ are used for composite element failure calculations when requested in the PT field of the PCOMP1 entry.
Input Data Entry:  MAT9  
Material Property Definition, Form 9

Description:  Defines the material properties for linear, temperature-independent, anisotropic materials for solid isoparametric elements

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAT9</td>
<td>MID</td>
<td>G11</td>
<td>G12</td>
<td>G13</td>
<td>G14</td>
<td>G15</td>
<td>G16</td>
<td>G22</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>G23</td>
<td>G24</td>
<td>G25</td>
<td>G26</td>
<td>G33</td>
<td>G34</td>
<td>G35</td>
<td>G36</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>G44</td>
<td>G45</td>
<td>G46</td>
<td>G55</td>
<td>G56</td>
<td>G66</td>
<td>RHO</td>
<td>A1</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>A2</td>
<td>A3</td>
<td>A4</td>
<td>A5</td>
<td>A6</td>
<td>TREF</td>
<td>GE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| MAT9 | 17 | 6.2+3 | | | 6.2+3 | ABC |
| +BC | | | | | | DEF |
| +EF | 5.1+3 | 5.1+3 | 5.1+3 | 3.2 | 6.6-6 | Gn. |
| +HI | 6.5-6 | 6.5-6 | | 125. |

Field  Contents

MID  Material identification number (Integer > 0)

G_{ij}  Elements of the 6x6 symmetric material property matrix (Real \geq 0.0)

RHO  Mass density (Real \geq 0.0)

A_i  Thermal expansion coefficient vector (Real)

TREF  Thermal expansion reference temperature (Real)

GE  Structural element damping coefficient (Real)

Remarks:

1. The material identification numbers must be unique for all MAT1, MAT2, MAT8, and MAT9 entries.
2. The mass density RHO will be used to automatically compute mass in a structural dynamics problem.
3. Continuation number 4 need not be used.
4. The subscripts 1 through 6 refer to x, y, z, xy, yz, zx, for example:

\[
\begin{bmatrix}
\sigma_x \\
\sigma_y \\
\sigma_z \\
\tau_{xy} \\
\tau_{yz} \\
\tau_{zx}
\end{bmatrix} =
\begin{bmatrix}
G_{11} & G_{12} & G_{13} \\
G_{12} & G_{22} & G_{23} \\
G_{13} & G_{23} & G_{33} \\
G_{14} & G_{14} & G_{44} \\
G_{15} & G_{15} & G_{55} \\
G_{16} & G_{16} & G_{66}
\end{bmatrix}
\begin{bmatrix}
\varepsilon_x \\
\varepsilon_y \\
\varepsilon_z \\
\gamma_{xy} \\
\gamma_{yz} \\
\gamma_{zx}
\end{bmatrix} + \begin{bmatrix}
A_1 \\
A_2 \\
A_3 \\
A_4 \\
A_5 \\
A_6
\end{bmatrix}
(\mathbf{T} - \mathbf{T}_n)
\]

5. The damping coefficient, GE is:

\[
GE = 2 \frac{C}{C_0}
\]
Input Data Entry:  MFORM    Mass Matrix Form

Description:   Defines the form of the mass matrix as consistent (coupled) or lumped.

Format and Example:

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>VALUE</td>
<td>A character string denoting the form of the mass matrix. The available forms are:</td>
</tr>
<tr>
<td></td>
<td>1) LUMPED</td>
</tr>
<tr>
<td></td>
<td>2) COUPLED</td>
</tr>
</tbody>
</table>

Remarks:
1. If more than one MFORM is included in the Bulk Data, any COUPLED value will result in coupled mass being used.
2. If no MFORM is indicated, the LUMPED formulation will be used.
Input Data Entry: **MKAERO1**  Mach Number - Frequency Table

**Description:** Provides a table of Mach numbers (\(m\)) and reduced frequencies (\(k\)) for unsteady aerodynamic matrix calculation.

**Format and Example:**

<table>
<thead>
<tr>
<th>m1</th>
<th>m2</th>
<th>m3</th>
<th>m4</th>
<th>m5</th>
<th>m6</th>
<th>CONT</th>
</tr>
</thead>
<tbody>
<tr>
<td>k1</td>
<td>k2</td>
<td>k3</td>
<td>k4</td>
<td>k5</td>
<td>k6</td>
<td>k7</td>
</tr>
</tbody>
</table>

**Field**

| m. | List of from 1 to 6 Mach numbers (Real \(\geq 0.0\) or blank) |
| k. | List of from 1 to 8 reduced frequencies (Real \(\geq 0.0\) or blank) |

**Remarks:**

1. All combinations of \((m, k)\) will be used.
2. The continuation entry is required.
3. Several **MKAERO1** entries may be in the input packet. If these data entries are in the packet, they will be used.
4. The symmetry flags have the following definition:
   - \(+1\) for symmetric
   - \(0\) for asymmetric
   - \(-1\) for antisymmetric

   The \(m-k\) pairs generated by this entry will generate aerodynamic matrices having the symmetries selected.
5. \(m-k\) pairs may be repeated with different symmetry options.
6. These are the following restrictions associated with the symmetry flags:
   a) Ground effect is limited to antisymmetric only, \(SYMXY = 0\) or \(-1\).
   b) Ground effect is not available at all for supersonic flow.
7. Reduced frequency is computed using:

\[
k = \frac{h \omega}{2v}
\]

where \(h\) is the reference chord defined by an **AERO** entry, \(\omega\) is the frequency in radians per sec, and \(v\) is the true velocity.
Input Data Entry: MKAERO2 Mach Number - Frequency Table

Description: Provides a list of Mach numbers (m) and reduced frequencies (k) for aerodynamic matrix calculation.

Format and Example:

<p>| | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>MKAERO2</td>
<td>SYMXZ</td>
<td>SYMXY</td>
<td>m₁</td>
<td>k₁</td>
<td>m₂</td>
<td>k₂</td>
<td>m₃</td>
<td>k₃</td>
<td></td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>m₄</td>
<td>k₄</td>
<td>m₅</td>
<td>k₅</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

MKAERO2 0 0 .10 .60 .70 .30 .70 1.0 ABC
+BC .8 .9 .8 1.0

Field Contents

SYMXZ, SYMXY Symmetry flags (Integer). See Remarks 4 and 5.

m₁, k₁ List of pairs of Mach numbers (Real ≥ 0.) and reduced frequencies (Real ≥ 0.)

Remarks:

1. This entry will cause the aerodynamic matrices to be computed for the given sets of parameter pairs.
2. Several MKAERO2 entries may be in the input packet. If these data entries are in the packet, they will be used.
3. Any number of continuations are allowed.
4. The symmetry flags have the following definition:
   +1 for symmetric (not available for SYMXY)
   0 for asymmetric
   -1 for antisymmetric

   The m-k pairs listed on the entry will generate aerodynamic matrices having the symmetries selected.
5. m-k pairs may be repeated with different symmetry options.
6. The following restrictions are imposed on the symmetry flags:
   a) Ground effect (if present) must be antisymmetric SYMXY = 0 or -1.
   b) Ground effect is not available at all for supersonic flow.
7. Reduced frequency is computed using:

\[ k = \frac{b \omega}{2v} \]

where b is the reference chord defined by an AERO entry, \( \omega \) is the frequency in radians per sec, and \( v \) is the true velocity.
**Input Data Entry:** **MODELIST**

**Description:** Defines a list of modes at which outputs are desired.

**Format and Example:**

<p>| | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>MODELIST</td>
<td>SID</td>
<td>MODE1</td>
<td>MODE2</td>
<td>MODE3</td>
<td>MODE4</td>
<td>MODE5</td>
<td>MODE6</td>
<td>MODE7</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>MODE8</td>
<td>MODE9</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>MODELIST</td>
<td>100</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Alternate Form:**

<p>| | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>MODELIST</td>
<td>SID</td>
<td>MODE1</td>
<td>THRU</td>
<td>MODE2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field** | **Contents**
---|---
SID | Set identification number referenced by Solution Control (Integer > 0)
MODE1 | Mode number of mode at which outputs are desired. (Integer > 0)

**Remarks:**

1. In order to be used, the **SID** must be referenced by Solution Control.
2. If the alternate form is used **MODE2** must be greater than or equal to **MODE1**.
3. Modes are numbered from 1 to n, starting at the lowest frequency for which a eigenvector was computed.
4. Nonexistent modes may be referenced and will result in no error message.
**Input Data Entry:**  
**MOMENT**  
**Static Moment**

**Description:** Defines a static moment at a grid point by specifying a vector.

**Format and Example:**

<p>| | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>MOMENT</td>
<td>SID</td>
<td>G</td>
<td>CID</td>
<td>M</td>
<td>N1</td>
<td>N2</td>
<td>N3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field**  
**Contents**

- SID: Load set identification number (Integer > 0)
- G: Grid point identification number (Integer > 0)
- CID: Coordinate system identification number (Integer ≥ 0)
- M: Scale factor (Real)
- Ni: Components of vector measured in coordinate system defined by CID (Real; at least one nonzero component)

**Remarks:**

1. The static moment applied to grid point G is given by
   \[ (m) = M \cdot (N1, N2, N3) \]
2. A CID of zero references the basic coordinate system.
**Input Data Entry:**  **MOMENT1**  Static Moment, Alternate Form 1

**Description:** Defines a static moment by specification of a value and two grid points which determine the direction.

**Format and Example:**

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOMENT</td>
<td>SID</td>
<td>G</td>
<td>M</td>
<td>G1</td>
<td>G2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOMENT</td>
<td>6</td>
<td>13</td>
<td>-2.93</td>
<td>16</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field** | **Contents**
---|---
SID | Load set identification number (Integer > 0)
G | Grid point identification number (Integer > 0)
M | Value of moment (Real)
G1 | Grid point identification numbers (Integer > 0; G1 ≠ G2)

**Remarks:**

1. The direction of the moment vector is determined by the vector from G1 and G2.
**Input Data Entry: MPC**  
**Multipoint Constraint**

**Description:** Defines a multipoint constraint equation of the form

\[ \sum_{j} A_j u_j = 0.0 \]

**Format and Example:**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPC</td>
<td>SID</td>
<td>G0</td>
<td>C0</td>
<td>A0</td>
<td>G</td>
<td>C</td>
<td>A</td>
<td>CONT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td></td>
<td>G</td>
<td>C</td>
<td>A</td>
<td>G</td>
<td>C</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MPC</td>
<td>3</td>
<td>28</td>
<td>3</td>
<td>6.2</td>
<td>2</td>
<td>3</td>
<td>4.29</td>
<td>+B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+B</td>
<td>1</td>
<td>4</td>
<td>-2.91</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field Contents**

- **SID**: Set identification (Integer > 0)
- **G0, G**: Identification number of grid or scalar point (Integer > 0)
- **C0, C**: Component number - any one of the digits 1 through 6 in the case of geometric grid points; blank or zero in the case of scalar points (Integer)
- **A0, A**: Coefficient (Real; A0 must be nonzero)

**Remarks:**

1. The first coordinate \((G0, C0)\) in the sequence is assumed to be the dependent coordinate. A dependent degree of freedom assigned by one **MPC** entry cannot be assigned dependent by another **MPC** entry or by a rigid element.

2. Forces of multipoint constraint are not recovered.

3. Multipoint constraint sets must be selected in Solution Control (MPC = SID) to be used.

4. The m-set coordinates specified on this entry may not be specified on other entries that define mutually exclusive sets.
Input Data Entry: **MPCADD**  
Multipoint Constraint Set Combination

**Description:** Defines a multipoint constraint set as a union of multipoint constraint sets defined via **MPC** entries.

**Format and Example:**

```
1 2 3 4 5 6 7 8 9 10
MPCADD SID  S1  S2  S3  S4  S5  S6  S7  CONT
CONT      S8  S9  etc-
```

```
MPCADD  101  2  3  1  6  4
```

**Field Contents**

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>Set identification number (Integer &gt; 0)</td>
</tr>
<tr>
<td>S</td>
<td>Set identification numbers of multipoint constraint sets defined via <strong>MPC</strong> entries (Integer &gt; 0)</td>
</tr>
</tbody>
</table>

**Remarks:**

1. The *Sj* must be unique.
2. Multipoint constraint sets must be selected in Solution Control (**MPC = SID**) to be used.
3. *Sj* may not be the identification number of a multipoint constraint set defined by another **MPCADD** entry.
4. **MPCADD** entries take precedence over **MPC** entries. If both have the same set identification number, only the **MPCADD** entry will be used.
Input Data Entry: **MPPARM**

**Description:** Identify values of user defined optimizer parameters that overrides the default values.

**Format and Example:**

<table>
<thead>
<tr>
<th>MPPARM</th>
<th>PARAM</th>
<th>VALUE</th>
<th>MPPARM</th>
<th>PARAM</th>
<th>VALUE</th>
<th>MPPARM</th>
<th>PARAM</th>
<th>VALUE</th>
<th>CONT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CONT</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ISCAL</td>
<td>0</td>
<td>STOL</td>
<td>0.005</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field Contents**

<table>
<thead>
<tr>
<th>PARAM</th>
<th>Name of parameter to be overridden ('Text)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VALUE</td>
<td>Integer or real value to be used for the parameter.</td>
</tr>
</tbody>
</table>

**Remarks:**

1. Any number of **PARAM-VALUE** combinations can be specified on an **MPPARM** entry.
2. See EDO software manual (ADS V 1.10) for a definition of parameters, but the most useful are shown below:

<table>
<thead>
<tr>
<th>REAL PARAMETER</th>
<th>DEFINITION</th>
<th>DEFAULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT</td>
<td>Constraint tolerance in the Method of Feasible Directions or the Modified Method of Feasible Directions. A constraint is active if its numerical value is more positive than CT.</td>
<td>-0.003</td>
</tr>
<tr>
<td>CTL</td>
<td>Same as CT, but for linear constraints.</td>
<td>-0.003</td>
</tr>
<tr>
<td>CTMIN</td>
<td>Same as CTMIN, but for linear constraints.</td>
<td>0.000</td>
</tr>
<tr>
<td>CTMIN</td>
<td>Minimum constraint tolerance for nonlinear constraints. If a constraint is more positive than CTMIN, it is considered to be violated.</td>
<td>0.0005</td>
</tr>
<tr>
<td>DABOBJ</td>
<td>Maximum absolute change in the objective between two consecutive iterations to indicate convergence in optimization.</td>
<td>(max. 0.001)</td>
</tr>
<tr>
<td>DABOBJM</td>
<td>Absolute convergence criterion for the optimization sub-problem when using sequential minimization techniques.</td>
<td>(Note 3)</td>
</tr>
<tr>
<td>DABSTR</td>
<td>Same as DABOBJ, but used at the strategy level.</td>
<td>(Note 3)</td>
</tr>
<tr>
<td>DELOBJ</td>
<td>Maximum relative change in the objective between two consecutive iterations to indicate convergence in optimization.</td>
<td>0.001</td>
</tr>
<tr>
<td>REAL PARAMETER</td>
<td>DEFINITION</td>
<td>DEFAULT</td>
</tr>
<tr>
<td>---------------</td>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>DELOBM</td>
<td>Relative convergence criterion for the optimization sub-problem when using sequential minimization techniques.</td>
<td>(Note 3)</td>
</tr>
<tr>
<td>DELSTR</td>
<td>Same as DELOBJ, but used at the strategy level.</td>
<td>(Note 3)</td>
</tr>
<tr>
<td>DOBJ1</td>
<td>Relative change in the objective function attempted on the first optimization iteration. Used to estimate initial move in the one-dimensional search. Updated as the optimization progresses.</td>
<td>0.1</td>
</tr>
<tr>
<td>DOBJ2</td>
<td>Absolute change in the objective function attempted on the first optimization iteration. Used to estimate initial move in the one-dimensional search. Updated as the optimization progresses.</td>
<td>0.2 max(Xi)</td>
</tr>
<tr>
<td>DX1</td>
<td>Maximum relative change in a design variable attempted on the first optimization iteration. Used to estimate initial move in the one-dimensional search. Updated as the optimization progresses.</td>
<td>0.01</td>
</tr>
<tr>
<td>DX2</td>
<td>Maximum absolute change in a design variable attempted on the first optimization iteration. Used to estimate initial move in the one-dimensional search. Updated as the optimization progresses.</td>
<td>0.02</td>
</tr>
<tr>
<td>EXTRAP</td>
<td>Maximum multiplier on the one-dimensional search parameter, ALPHA in the one-dimensional search using polynomial interpolation/extrapolation.</td>
<td>(Note 3)</td>
</tr>
<tr>
<td>SCFO</td>
<td>The user-simplified value of the scale factor for the objective function if the default or calculated value is to be overridden.</td>
<td>(Note 3)</td>
</tr>
<tr>
<td>SCIMIN</td>
<td>Maximum numerical value of any scale factor allowed.</td>
<td>(Note 3)</td>
</tr>
<tr>
<td>STOL</td>
<td>Tolerance on the components of the calculated search direction to indicate that the Kuhn-Tucker conditions are satisfied.</td>
<td>(Note 3)</td>
</tr>
<tr>
<td>THETAZ</td>
<td>Nominal value of the push-off factor in the Method of Feasible Directions.</td>
<td>(Note 3)</td>
</tr>
</tbody>
</table>
### REAL PARAMETER

<table>
<thead>
<tr>
<th>REAL PARAMETER</th>
<th>DEFINITION</th>
<th>DEFAULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMULT</td>
<td>Multiplier on the move parameter, ALPHA, in the one-dimensional search to find bounds on the solution.</td>
<td>(Note 3)</td>
</tr>
<tr>
<td>ZRO</td>
<td>Numerical estimate of zero on the computer. Usually the default value is adequate. If a computer with a short word length is used, ZRO = 1.0E-4 may be preferred.</td>
<td>(Note 3)</td>
</tr>
</tbody>
</table>

### INTEGER PARAMETER

<table>
<thead>
<tr>
<th>INTEGER PARAMETER</th>
<th>DEFINITION</th>
<th>DEFAULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISCAL</td>
<td>Scaling parameter. By default, scaling is done every NDV iterations, otherwise scaling is performed every ISCA iterations.</td>
<td>-1</td>
</tr>
<tr>
<td>ITMAX</td>
<td>Maximum number of iterations allowed at the optimizer level.</td>
<td>40</td>
</tr>
<tr>
<td>ITROMP</td>
<td>The number of consecutive iterations for which the absolute or relative convergence criteria must be met to indicate convergence at the optimizer level.</td>
<td>2</td>
</tr>
<tr>
<td>ITRMST</td>
<td>The number of consecutive iterations for which the absolute or relative convergence criteria must be met to indicate convergence at the optimizer level.</td>
<td>(Note 3)</td>
</tr>
<tr>
<td>JMAX</td>
<td>Maximum of iterations allowed at the strategy level.</td>
<td>(Note 3)</td>
</tr>
</tbody>
</table>

3. Some of these parameters, indicated in the tables, are used only with the original version of the ADS optimizer. They are not used in MicroDOT.
Input Data Entry  OCPARM  Optimality Criteria Parameters

Description:  Identifies values of user defined optimizer parameters that override default parameters

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>OCPARM</td>
<td>OCPARM</td>
<td>PARAM</td>
<td>VALUE</td>
<td>PARAM</td>
<td>VALUE</td>
<td>PARAM</td>
<td>VALUE</td>
<td>PARAM</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>VALUE</td>
<td>PARAM</td>
<td>VALUE</td>
<td>ETC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

OCPARM  OCPARM  ISCAL  1

Field  Contents

PARAM  Name of parameter to be overridden (Text)

VALUE  Integer or real value to be used for the parameter

Remarks:

1. Any number of PARAM/VALUE pairs may be specified

2. Legal values of PARAM are shown in the table below:

<table>
<thead>
<tr>
<th>REAL PARAMETERS</th>
<th>INTEGER PARAMETERS</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>PARAM</th>
<th>Default</th>
<th>PARAM</th>
<th>Default</th>
<th>PARAM</th>
<th>Default</th>
<th>PARAM</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPSRED</td>
<td>0.0001</td>
<td>EPSOC</td>
<td>0.01</td>
<td>IUNIT</td>
<td>Note 1</td>
<td>ITER</td>
<td>NONE</td>
</tr>
<tr>
<td>EPSBNI</td>
<td>0.01</td>
<td>EPSFIN</td>
<td>0.001</td>
<td>ISTAT</td>
<td>NONE</td>
<td>NLPP</td>
<td>Note 3</td>
</tr>
<tr>
<td>EPSBNF</td>
<td>0.03</td>
<td>EPSCTB</td>
<td>0.1</td>
<td>ALNCSC</td>
<td>3</td>
<td>IPRINT</td>
<td>0</td>
</tr>
<tr>
<td>EPSCON</td>
<td>0.1</td>
<td>EPSGLB</td>
<td>0.9</td>
<td>NCSC</td>
<td>NONE</td>
<td>NRTAIN</td>
<td>NONE</td>
</tr>
<tr>
<td>XPAND</td>
<td>2.0</td>
<td>REDUC</td>
<td>0.5</td>
<td>NCBRZ</td>
<td>1</td>
<td>IGRDST</td>
<td>NONE</td>
</tr>
<tr>
<td>THRESH</td>
<td>0.05</td>
<td>OBJRSZ</td>
<td>NONE</td>
<td>ALSCMN</td>
<td>3</td>
<td>HSCALE</td>
<td>1</td>
</tr>
<tr>
<td>EPSOBJ</td>
<td>0.005</td>
<td></td>
<td></td>
<td>MXRTAN</td>
<td>Note 2</td>
<td>MXITR8</td>
<td>30</td>
</tr>
</tbody>
</table>

1. ASTROS Output Unit

2. Minimum of 100 and the Number of Constraints

3. ASTROS Number of Lines per Page
**Input Data Entry:** OMIT

**Omitted Coordinates**

**Description:** Defines degrees of freedom that the user desires to omit from the problem through matrix partitioning. Used to reduce the number of independent degrees of freedom.

**Format and Example:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMIT</td>
<td>SETID</td>
<td>ID</td>
<td>C</td>
<td>ID</td>
<td>C</td>
<td>ID</td>
<td>C</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field**

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>SETID</td>
<td>The reduce set identification number (Integer &gt; 0).</td>
</tr>
<tr>
<td>ID</td>
<td>Grid or scalar point identification number (Integer &gt; 0).</td>
</tr>
<tr>
<td>C</td>
<td>Component number, zero or blank for scalar points, any unique combination of the digits 1 through 6 for grid points.</td>
</tr>
</tbody>
</table>

**Remarks:**

1. Coordinates specified on this entry form members of a mutually exclusive set. They may not be specified on other entries that define mutually exclusive sets.

2. In many cases it may be more convenient to use OMIT1, ASET or ASET1 entries.
Input Data Entry: OMIT1  Omitted Coordinates, Alternate Form

Description: Defines degrees of freedom that the user desires to omit from the problem through matrix partitioning. Used to reduce the number of independent degrees of freedom.

Format and Example:

<table>
<thead>
<tr>
<th>OMIT1</th>
<th>SETID</th>
<th>C</th>
<th>GID1</th>
<th>GID2</th>
<th>GID3</th>
<th>GID4</th>
<th>GID5</th>
<th>GID6</th>
<th>CONT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONT</td>
<td>GID7</td>
<td>GID8</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OMIT1</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>3</th>
<th>10</th>
<th>9</th>
<th>6</th>
<th>5</th>
<th>ABC</th>
</tr>
</thead>
<tbody>
<tr>
<td>+BC</td>
<td>7</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Alternate Form:

<table>
<thead>
<tr>
<th>OMIT1</th>
<th>SETID</th>
<th>C</th>
<th>GID1</th>
<th>THRU</th>
<th>GID2</th>
</tr>
</thead>
</table>

Field Contents

<table>
<thead>
<tr>
<th>SETID</th>
<th>The reduce set identification number (Integer &gt; 0).</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Component number (Any unique combination of the digits 1 through 6 (with no embedded blanks) when point identification numbers are grid points; must be null or zero if point identification numbers are scalar points).</td>
</tr>
<tr>
<td>GIDI</td>
<td>Grid or scalar point identification number (Integer &gt; 0).</td>
</tr>
</tbody>
</table>

Remarks:

1. Coordinates specified on this entry form members of a mutually exclusive set. They may not be specified on other entries that define mutually exclusive sets.

2. If the alternate form is used, points in the sequence ID1 through ID2 are required to exist and ID2 must be greater than or equal to ID1.
**Input Data Entry:** PAERO1  **Acrodyanic Panel Property**

**Description:** Gives associated bodies for the panels in the unsteady aerodynamic model.

**Format and Examples:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAERO1</td>
<td>PID</td>
<td>B1</td>
<td>B2</td>
<td>B3</td>
<td>B4</td>
<td>B5</td>
<td>B6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| PAERO1 | 1 | 3 |   |   |   |   |   |   | |

**Field** | **Contents**
--- | ---
PID | Property identification number (referenced by CAERO1) (Integer > 0)
Bi | Identification number of CAERO2 entries for associated bodies (Integer ≥ 0, or blank)

**Remarks:**

1. The associated bodies must be in the same aerodynamic group.
2. The Bi numbers above must appear on a CAERO2 entry to define these bodies completely.
Input Data Entry: **PAERO2**  
**Aerodynamic Body Properties**

**Description:** Defines the cross-section properties of unsteady aerodynamic bodies.

**Format and Examples:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAERO2</td>
<td>PID</td>
<td>ORIENT</td>
<td>WIDTH</td>
<td>AR</td>
<td>LRSB</td>
<td>LRIB</td>
<td>LTH1</td>
<td>LTH2</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>THI1</td>
<td>THN1</td>
<td>THI2</td>
<td>THN2</td>
<td>THI3</td>
<td>THN3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2</th>
<th>2</th>
<th>6.0</th>
<th>1.0</th>
<th>22</th>
<th>91</th>
<th>100</th>
<th>abc</th>
</tr>
</thead>
<tbody>
<tr>
<td>+bc</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field** | **Contents**

- **PID**: Property identification number (Integer > 0)
- **ORIENT**: Orientation flag "Z", "Y", or "XY". Type of motion allowed for bodies (Text). Refers to the aerodynamic coordinate system "y" and "z" directions (see AERO data entry)
- **WIDTH**: Reference half-width of body (Real > 0.0)
- **AR**: Aspect ratio (height/width) (Real > 0.0)
- **LRSB**: Identification number of an AEFACT data entry containing a list of slender body half-widths. If blank, the value of WIDTH will be used (Integer > 0 or blank)
- **LRIB**: Identification number of an AEFACT data entry containing a list of interference body half-widths. If blank, the value of WIDTH will be used (Integer > 0 or blank)
- **LTH1, LTH2**: Identification number of AEFACT data entries for defining theta arrays for interference calculations (Integer > 0)
- **THI1, THN1**: The first and last interference element of a body to use the θ array (Integer ≥ 0)

**Remarks:**

1. The **EID** of all CAERO2 elements in any IGID group must be ordered, so that their corresponding ORIENT values appear in the order Z, ZY, Y.

2. The half-widths (given on AEFACT data entries referenced in fields 6 and 7) are specified at division points. The number of entries on an AEFACT data entry used to specify half-widths must be one greater than the number of elements.

3. The half-width at the first point (i.e., the nose) on a slender body is usually 0.; thus, it is recommended (but not required) that the LRSB data is supplied with a zero first entry.

4. THI1 and THN1 are interference element locations on a body. The element numbering begins at one for each body.

5. A body is represented by a slender body surrounded by an interference body. The slender body creates the down wash due to the motion of the body, while the interference body represents the effects upon panels and other bodies.
Input Data Entry: PAERO6

Description: Defines body analysis parameters for steady aerodynamics.

Format and Examples:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
</table>

296
### Field Contents

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCID</td>
<td>Body component identification number (Integer &gt; 0)</td>
</tr>
<tr>
<td>CMPNT</td>
<td>Component type <em>(FUSEL for the fuselage and POD for a POD)</em></td>
</tr>
<tr>
<td>CP</td>
<td>Coordinate system of the geometry input (Integer ≥ 0, or blank)</td>
</tr>
<tr>
<td>IGRP</td>
<td>Group flag (Integer &gt; 0)</td>
</tr>
<tr>
<td>NRAD</td>
<td>Number of equal radial cuts used to define the body panels (Integer ≥ 0 or blank)</td>
</tr>
<tr>
<td>LRAD</td>
<td>Identification number of an <strong>AEFACT</strong> data entry which defines the angular locations in degrees of the body panels (Integer ≥ 0 or blank)</td>
</tr>
<tr>
<td>LAXIAL</td>
<td>Identification number of an <strong>AEFACT</strong> data entry which defines the axial locations in degrees of the body panels (Integer ≥ 0 or blank)</td>
</tr>
</tbody>
</table>

### Remarks:

1. **NRAD** and **LRAD** are mutually exclusive.
2. If **LRAD** and **NRAD** are zero or blank, the radial cuts specified by the BODY or AXSTA entries are used.
3. **LAXIAL** is used only for **FUSEL** components. Inputs on the **AEFACT** entry are the dimensional fuselage stations.
4. If **LAXIAL** is blank, the axial locations are the same as those given by **AXSTA** data entries for the given body component.
Input Data Entry: **PBAR** Simple Beam Property

**Description:** Defines the properties of a simple beam (bar) which is used to create bar elements via the CBAR entry.

**Format and Examples:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBAR</td>
<td>PID</td>
<td>MID</td>
<td>A</td>
<td>I1</td>
<td>I2</td>
<td>J</td>
<td>NSM</td>
<td>TMIN</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>C1</td>
<td>C2</td>
<td>D1</td>
<td>D2</td>
<td>E1</td>
<td>E2</td>
<td>F1</td>
<td>F2</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>K1</td>
<td>K2</td>
<td>I12</td>
<td>R12</td>
<td>R22</td>
<td>ALPHA</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field**

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID</td>
<td>Property identification number (Integer &gt; 0)</td>
</tr>
<tr>
<td>MID</td>
<td>Material identification number (Integer &gt; 0)</td>
</tr>
<tr>
<td>A</td>
<td>Area of bar cross-section (Real ≥ 0.0)</td>
</tr>
<tr>
<td>I1</td>
<td>Area moments of inertia (Real)</td>
</tr>
<tr>
<td>J</td>
<td>Torsional constant (Real ≥ 0.0)</td>
</tr>
<tr>
<td>NSM</td>
<td>Nonstructural mass per unit length (Real ≥ 0.0)</td>
</tr>
<tr>
<td>TMIN</td>
<td>The minimum cross-sectional area in design (Real, Default = 0.0001)</td>
</tr>
<tr>
<td>K1,K2</td>
<td>Area factor for shear (Real)</td>
</tr>
<tr>
<td>C1,D1,E1,F1</td>
<td>Stress recovery coefficients (Real)</td>
</tr>
<tr>
<td>R12,R22,ALPHA</td>
<td>Inertia linking terms for design (see Remarks 5 and 6)</td>
</tr>
</tbody>
</table>

**Remarks:**

1. The BAR element geometry and coordinate system is shown in the Figure on the following page.

2. PBAR entries may only reference MAT1 material entries.

3. The transverse shear stiffnesses in planes 1 and 2 are (K1)AG and (K2)AG, respectively. The default values for K1 and K2 are infinite. In other words, the transverse shear flexibilities are set equal to zero. K1 and K2 are ignored if I12 ≤ 0.

4. The stress recovery coefficients C1 and C2, etc., are the y and z coordinates in the BAR element coordinate system of a point at which stresses are computed. Stresses are computed at both ends of the BAR.

5. The TMIN value is used only for shape function design variable linking.
6. For design, the following applies to the R12 and R22 values. The moments of inertia are linked to the cross-sectional area by the following expressions:

\[ I_1 = R12 \times A^{\alpha} \]

\[ I_2 = R22 \times A^{\alpha} \]

(A) If \( R12 = 0.0 \) then the missing value is computed from \( R12 = I1 / (A^{\alpha}) \). The same is true for \( R22 \) and \( I2 \).

(B) The \( \alpha \) value defaults to 1.0 and must be \( \geq 1.0 \).

(C) If both \( I1 \) and \( R12 \) or \( I2 \) and \( R22 \) are given, the linking expression will override the input \( I_i \) values.

7. If the CBAR is to be designed, the following restrictions apply.

(A) \( J = NSM = R1 = R2 = I12 = 0.0 \)

If any of these values are not zero, a warning message will be issued and the value set to zero.
Input Data Entry:  PCOMP  Layered Composite Element Property

Description:  Defines the properties of an n-ply composite material laminate.

Format and Examples:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCOMP</td>
<td>PID</td>
<td>Z0</td>
<td>NSM</td>
<td>SBOND</td>
<td>F.T.</td>
<td>TMIN</td>
<td>LOPT</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>MID1</td>
<td>T1</td>
<td>TH1</td>
<td>SOUT1</td>
<td>MID2</td>
<td>T2</td>
<td>TH2</td>
<td>SOUT2</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>MID3</td>
<td>T3</td>
<td>TH3</td>
<td>SOUT3</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| PCOMP | 100 | -0.5 | 1.5 | 5.0+3 | HOFF |       | MEM | ABC |
| +BC   | 150 | 0.05  | 90. | YES   | -45. | DEF   |     |     |
| +EF   |     | 45.0  |     |       |      |       |     |     |

Field Contents

PID  Property identification number (Integer > 0).
Z0   Offset of the laminate lower surface from the element mean plane. A positive value means the +Z0 direction. (Real or blank, see Remark 2)
NSM  Nonstructural mass per unit area (Real ≥ 0.0).
SBOND Allowable shear stress of the bonding material. (Real ≥ 0.0)
F.T.  Failure theory, one of the strings HILL, HOFF, TSAT, STRESS, or STRAIN. See Remark 4.
TMIN Minimum ply thickness for design (Real > 0.0 or blank). (Default = 10^-4)
LOPT  Lamination generation option, MEM or blank. See Remark 5.
MIDi  Material identification number of the i(th) layer. (Integer > 0 or blank)
T_i   Thickness of the i(th) layer (Real > 0.0 or blank).
TH_i  Angle between the longitudinal direction of the fibers of the i(th) layer and the material X-axis. (Real or blank)
SOUTi Stress output request for i(th) layer, one of the strings YES or NO. (Default = NO)

Remarks:

1. For non-designed elements, the plies are numbered from 1 to n beginning with the bottom layer.
2. For composites there are two methods for specifying the offset of the element reference plane from the element mean plane: Zo on this entry and ZOFF on the CQUAD4 or CTRIA3 Bulk Data entries. The distinction is shown in the figure below:

![Diagram showing the offset of the element reference plane from the element mean plane]

You may only specify a Zo on this entry if the ZOFF field of any CQUAD4 or CTRIA3 referencing it is blank. The default value for Zo is \(-t/2\) where \(t\) is the overall thickness of the laminate.

3. SBOND is required if bonding material failure index calculations are desired.

4. The failure theory is used to determine the element failure on a ply-by-ply basis. The available theories are:
   - HILL - Hill Theory
   - HOFF - Hoffman Theory
   - TSAI - Tsai-Wu Theory
   - STRESS - For Maximum Stress Theory
   - STRAIN - For Maximum Strain Theory

5. MEM indicates a layup of membrane only plies.

6. The material properties, Midi, may reference only MAT1, MAT2, and MAT8 Bulk Data entries.

7. If any of the Midi, Ti or Thi are blank, then the last non-blank values specified for each will be used to define the values for the ply.

8. Tmin will be ignored unless the element is linked to design variables by SHAPE entries.
Input Data Entry:  PCOMP1  Layered Composite Element Property

Description:  Defines the properties of an n-ply laminated composite material where all plies are composed of the same material and are of equal thickness.

Format and Examples:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCOMP1</td>
<td>PID</td>
<td>Z0</td>
<td>NSM</td>
<td>SBOND</td>
<td>F.T.</td>
<td>TMIN</td>
<td>MID</td>
<td>LOPT</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>TPLY</td>
<td>TH1</td>
<td>TH2</td>
<td>TH3</td>
<td>TH4</td>
<td>TH5</td>
<td>TH6</td>
<td>TH7</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>TH8</td>
<td>TH9</td>
<td>TH10</td>
<td>etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PCOMP1  100  -0.5  1.7  5.0  3  STRAIN  200  ABC  
+BC  0.25  -45.0  45.0  90.0  90.0  45.0  |

Field  Contents

PID  Property identification number (1,000,000 > Integer > 0).

Z0  Offset of the laminate lower surface from the element mean plane. A positive value means the +z_e direction. (Real or blank, see Remark 2)

NSM  Nonstructural mass per unit area (Real ≥ 0.0).

SBOND  Allowable shear stress of the bonding material. (Real > 0.0)

F.T.  Failure theory, one of the strings HILL, HOFF, TSAI, STRESS or STRAIN. See Remark 4.

TMIN  Minimum ply thickness for design (Real > 0.0 or blank) (Default = 0.0001)

MID  Material identification number for all layers. (Integer > 0.0 or blank)

LOPT  Lamination generation option, new or blank. See Remark 5.

TPLY  Thickness of each layer. (Real > 0.0).

THi  Angle between the longitudinal direction of the fibers of the i(th) layer and the material X-axis. (Real or blank)

Remarks:

1. For nondesigned elements, the plies are numbered from 1 to n beginning with the bottom layer.
2. For composites there are two methods for specifying the offset of the element reference plane from the element mean plane: \( z_0 \) on this entry and \( \text{ZOFF} \) on the \text{CQUAD4} or \text{CTRIA3} Bulk Data entries. The distinction is shown in the figure below:

![Diagram showing element reference plane offsets](image)

You may only specify a \( z_0 \) on this entry if the \text{ZOFF} field of any \text{CQUAD4} or \text{CTRIA3} referencing it is blank. The default value for \( z_0 \) is \(-t/2\) where \( t \) is the overall thickness of the laminate.

3. \text{SBOND} is required if bonding material failure index calculations are desired.

4. The failure theory is used to determine the element failure on a ply-by-ply basis. The available theories are:
   - \text{HILL} - Hill Theory
   - \text{HOFF} - Hoffman Theory
   - \text{TSAI} - Tsai-Wu Theory
   - \text{STRESS} - For Maximum Stress Theory
   - \text{STRAIN} - For Maximum Strain Theory

5. \text{MEM} indicates a layup of membrane only plies.

6. The material properties, \text{MID}_i, may reference only \text{MAT1}, \text{MAT2}, and \text{MAT8} Bulk Data entries.

7. \text{TMIN} will be ignored unless the element is linked to design variables by \text{SHAPE} entries.
Input Data Entry:  PCOMP2  Layered Composite Element Property

Description:  Defines the properties of an n-ply laminated composite material where all plies are composed of the same material but are of different thickness.

Format and Examples:

<table>
<thead>
<tr>
<th>PCOMP2</th>
<th>PID</th>
<th>ZO</th>
<th>NSM</th>
<th>SBOND</th>
<th>F.T.</th>
<th>TMIN</th>
<th>MID</th>
<th>LOPT</th>
<th>CONT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONT</td>
<td>T1</td>
<td>TH1</td>
<td>T2</td>
<td>TH2</td>
<td>T3</td>
<td>TH3</td>
<td>-ect-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PCOMP2 100 -0.5 1.7 5. +3 TSAI 200 ABC +BC 0.25 -45.0 0.5 90.0 0.25 45.0

Field Contents

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID</td>
<td>Property identification number (Integer &gt; 0).</td>
</tr>
<tr>
<td>ZO</td>
<td>Offset of the laminate lower surface from the element mean plane. A positive value means the +ze direction. (Real or blank, see Remark 2)</td>
</tr>
<tr>
<td>NSM</td>
<td>Nonstructural mass per unit area (Real ≥ 0.0).</td>
</tr>
<tr>
<td>SBOND</td>
<td>Allowable shear stress of the bonding material. (Real ≥ 0.0)</td>
</tr>
<tr>
<td>F.T.</td>
<td>Failure theory, one of the strings HILL, HOFF, TSAI, STRESS, or STRAIN. See Remark 4.</td>
</tr>
<tr>
<td>TMIN</td>
<td>Minimum ply thickness for design (Real &gt; 0.0 or blank) (Default = 10^-4)</td>
</tr>
<tr>
<td>MID</td>
<td>Material identification number for all layers. (Integer &gt; 0 or blank)</td>
</tr>
<tr>
<td>LOPT</td>
<td>Lamination generation option, M66 or blank. See Remark 5.</td>
</tr>
<tr>
<td>Ti</td>
<td>Thickness of all layers. (Real &gt; 0.0 or blank).</td>
</tr>
<tr>
<td>THi</td>
<td>Angle between the longitudinal direction of the fibers of the i(th) layer and the material X-axis. (Real or blank)</td>
</tr>
</tbody>
</table>

Remarks:

1. For nondesigned elements, the plies are numbered from 1 to n beginning with the bottom layer.
2. For composites there are two methods for specifying the offset of the element reference plane from the element mean plane: \( z_0 \) on this entry and \( ZOFF \) on the \texttt{CQUAD4} or \texttt{CTRIA3} Bulk Data entries. The distinction is shown in the figure below:

![Element Reference Plane Diagram]

You may only specify a \( z_0 \) on this entry if the \( ZOFF \) field of any \texttt{CQUAD4} or \texttt{CTRIA3} referencing it is blank. The default value for \( z_0 \) is \(-t/2\) where \( t \) is the overall thickness of the laminate.

3. \texttt{SBOND} is required if bonding material failure index calculations are desired.

4. The failure theory is used to determine the element failure on a ply-by-ply basis. The available theories are:

- \texttt{HILL} - Hill Theory
- \texttt{HOFF} - Hoffman Theory
- \texttt{TSAI} - Tsai-Wu Theory
- \texttt{STRESS} - For Maximum Stress Theory
- \texttt{STRAIN} - For Maximum Strain Theory

5. \texttt{MEM} indicates a layup of membrane only plies.

6. The material properties, \( \texttt{MID}_i \), may reference only \texttt{MAT1}, \texttt{MAT2}, and \texttt{MAT8} Bulk Data entries.

7. If any of the \( T_i \) or \( TH_i \) are blank, then the last non-blank values specified for each will be used to define the values for the ply.

8. \texttt{TMIN} will be ignored unless the element is linked to design variables by \texttt{SHAPE} entries.
Input Data Entry: **PELAS**  Scalar Elastic Property

**Description:** Used to define the stiffness, damping coefficient, and stress coefficient of a scalar elastic element (spring) defined by means of the **CELAS1** entry.

**Format and Example:**

```
   1  2  3  4  5  6  7  8  9  10
PELAS  PID  K GE S  Tmin
PELAS  7  4.29 0.06 7.92
```

**Field** | **Contents**
--- | ---
PID | Property identification number (Integer > 0)
K | Elastic property value (Real)
GE | Damping coefficient (Real ≥ 0.0)
S | Stress coefficient (Real)
TMIN | Minimum value for design (Real > 0.0, or blank, Default = 0.0001)

**Remarks:**

1. The user is cautioned to be careful using negative spring values.
2. TMIN is ignored unless the element is designed using shape function design variable linking.
**Input Data Entry:** PIHEX Isoparametric Hexahedron Property

**Description:** Defines the properties of an isoparametric solid element, including a material reference and the number of integration points. Referenced by the CIHEX1, CIHEX2, and CIHEX3 entries.

**Format and Examples:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIHEX</td>
<td>PID</td>
<td>MID</td>
<td>CID</td>
<td>NIP</td>
<td>AR</td>
<td>ALFA</td>
<td>BETA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIHEX</td>
<td>15</td>
<td>3</td>
<td></td>
<td>3</td>
<td></td>
<td>5.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field** | **Contents**
--- | ---
PID | Property identification number (Integer > 0)
MID | Material identification number (Integer > 0)
CID | Identification number of the coordinate system in which the material referenced by MID is defined (Integer ≥ or blank)
NIP | Number of integration points along each edge of the element (Integer = 2, 3, 4, or blank)
AR | Maximum aspect ratio (ratio of longest to shortest edge) of the element (Real > 1.0 or blank)
ALFA | Maximum angle in degrees between the normals of two subtriangles comprising a quadrilateral face (Real, 0.0 ≤ ALFA ≤ 180.0 or blank) (Default = 45.0)
BETA | Maximum angle in degrees between the vector connecting a corner point to an adjacent midside point and the vector connecting that midside point and the other midside or corner point (Real, 0.0 < BETA < 180.0 or blank) (Default = 45.0)
Examples of Field Definitions:

Example of ALPHA

Example of BETA

Remarks:
1. All PIHEX cards must have unique identification numbers.
2. CID is not used for isotropic materials.
3. The default for CID is the basic coordinate system.
4. The default for NIP is 2 for IHEX and 3 for IHEX2 and IHEX3.
5. AR, ALFA, and BETA are used for checking the geometry of the element. The defaults are:

<table>
<thead>
<tr>
<th></th>
<th>AR</th>
<th>ALFA (degrees)</th>
<th>BETA (degrees)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIHEX1</td>
<td>5.0</td>
<td>45.0</td>
<td>--</td>
</tr>
<tr>
<td>CIHEX2</td>
<td>10.0</td>
<td>45.0</td>
<td>45.0</td>
</tr>
<tr>
<td>CIHEX3</td>
<td>15.0</td>
<td>45.0</td>
<td>45.0</td>
</tr>
</tbody>
</table>
Input Data Entry: PLIST

Description: Defines property entries associated with a design variable.

Format and Examples:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLIST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td></td>
<td>PID7</td>
<td>PID8</td>
<td>PID9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PID7</td>
<td>PID8</td>
<td>PID9</td>
<td></td>
<td>PID4</td>
<td>PID5</td>
<td>PID6</td>
</tr>
</tbody>
</table>

Alternate Form:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLIST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DVID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PID1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PID2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

- LINKID Property list identifier (Integer > 0).
- PTYPE Property type associated with this list (e.g., PROD).
- PID1, PID2, Property entry identifications. (Integer > 0, or blank)
- PID3

Remarks:

1. Allowable PTYPES are: PROD, PSHEAR, PCOMP, PCOMP1, PCOMP2, PELAS, PSELL, PMASS, PRMEN, PGMEM1, and PbAR.
2. If the alternate form is used, PID2 must be greater than or equal to PID1.
3. All elements using properties listed on PLIST entries for a particular LINKID will be designed by (linked to) that design variable that references the PLIST LINKID.
Input Data Entry: PLOAD Static Pressure Load

Description: Defines a static pressure load on a triangular or quadrilateral surface.

Format and Examples:

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>Load set identification number (Integer &gt; 0)</td>
</tr>
<tr>
<td>P</td>
<td>Pressure (Real)</td>
</tr>
<tr>
<td>G1</td>
<td>Grid point identification numbers (Integer &gt; 0; G4 may be zero)</td>
</tr>
</tbody>
</table>

Remarks:

1. The grid points define either a triangular or a quadrilateral surface to which a pressure is applied. If G4 is zero or blank, the surface is triangular.

2. In the case of a triangular surface, the assumed direction of the pressure is computed according to the right-hand rule using the sequence of grid points G1, G2, and G3 as illustrated below.

   ![Diagram of a triangle with grid points G1, G2, and G3 and a pressure P applied at point G3.]

   The total load on the surface, AP, is divided into three equal parts and applied to the grid points as concentrated loads. A minus sign in field 3 reverses the direction of the load.

3. In the case of a quadrilateral surface, the grid points G1, G2, G3, and G4 should form a consecutive sequence around the perimeter. The right-hand rule is applied to find the assumed direction of the pressure. Four concentrated loads are applied to the grid points in approximately the same manner as for a triangular surface. The following specific procedures are adopted to accommodate irregular and/or warped surfaces:
   a. The surface is divided into two sets of overlapping triangular surfaces. Each triangular surface is bounded by two of the sides and one of the diagonals of the quadrilateral.
   b. One-half of the pressure is applied to each triangle which is then treated in the manner described in Remark 2.

4. Load sets must be selected in Solution Control to be used.
**Input Data Entry:** PLYLIST  A list of composite element layer numbers.

**Description:** Defines a set of layers of composite elements by a list.

**Format and Examples:**

<p>| | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>PLYLIST</td>
<td>SID</td>
<td>P1</td>
<td>P2</td>
<td>P3</td>
<td>P4</td>
<td>P5</td>
<td>P6</td>
<td>P7</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>P8</td>
<td>etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PLYLIST SID P1 P2 P3 P4 P5 P6 P7 CONT
+BC 13

**Alternate Form:**

<p>| | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>PLYLIST</td>
<td>SID</td>
<td>P1</td>
<td>THRU</td>
<td>P2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field**  
**Contents**

SID  Set of identification numbers (Integer > 0)

Pi  List of ply numbers (Integer > 0)

**Remarks:**

1. These entries are referenced by the DESVAR, DESVARS, DCONLMN, DCONPMN, DCONLM and DCONTB2 data entries.

2. When using the THRU option, all intermediate plies will be assumed to exist.

3. When used by DESVARS and DESVARP, the entry refers to composite layer numbers to be linked together in the design model.

4. When used by DCONLMN, DCONPMN and DCONLM, the entry refers to composite layers that, together, define a "ply" or a "laminate" whose summed thicknesses will be contribute to the constraint.
Input Data Entry: **PMASS**  
Scalar Mass Property

**Description:** Used to define the mass value of a scalar mass element which is defined by means of the CMASS1 entries.

**Format and Examples:**

<p>| | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PMASS</td>
<td>PID</td>
<td>M</td>
<td>TMIN</td>
<td>PID</td>
<td>M</td>
<td>TMIN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PMASS</td>
<td>7</td>
<td>4.29</td>
<td>0.2</td>
<td>6</td>
<td>13.2</td>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field Contents**

- **PID**: Property identification number (Integer > 0).
- **M**: Value of scalar mass (Real).
- **TMIN**: The minimum mass value in design. Default = 0.0001

**Remarks:**

1. This entry defines a mass value.
2. Up to 2 mass values may be defined by this entry.
3. **TMIN** is ignored unless the mass element is linked to design variables through SHAPE entries.
Input Data Entry:  **PQDMEM1**  Quadrilateral Membrane Property

**Description:**  Used to define the properties of a quadrilateral membrane referenced by the **cgaimu** entry. No bending properties are included.

**Format and Examples:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>PQDMEM1</td>
<td>PID</td>
<td>MID</td>
<td>T</td>
<td>NSM</td>
<td>TMIN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PQDMEM1</td>
<td>235</td>
<td>2</td>
<td>0.5</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field**  
**Contents**

**PID**  
Property identification number (Integer > 0).

**MID**  
Material identification number (Integer > 0).

**T**  
Thickness of membrane (Real ≥ 0.0)

**NSM**  
Nonstructural mass per unit area (Real ≥ 0.0).

**TMIN**  
Minimum thickness for design (Real > 0.0 or blank) (Default = 0.0001)

**Remarks:**

1. All **PQDMEM1** entries must have unique property identification numbers.

2. **TMIN** is ignored unless the element is linked to the global design variables by a **SHAPE** entry.
Input Data Entry: PROD  Rod Property

Description: Defines the properties of a rod which is referenced by the CROD entry.

Format and Examples:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROD</td>
<td>PID</td>
<td>MID</td>
<td>A</td>
<td>J</td>
<td>C</td>
<td>NSM</td>
<td>TMIN</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PROD 17 23 42.6 17.92 4.236 0.5

Field Contents

<table>
<thead>
<tr>
<th>PID</th>
<th>Property identification number (Integer &gt; 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MID</td>
<td>Material identification number (Integer &gt; 0)</td>
</tr>
<tr>
<td>A</td>
<td>Area of rod (R &gt; 0, or blank)</td>
</tr>
<tr>
<td>J</td>
<td>Torsional const. (Real &gt; 0.0, or blank)</td>
</tr>
<tr>
<td>C</td>
<td>Coefficient to determine torsional stress (Real &gt; 0.0, or blank)</td>
</tr>
<tr>
<td>NSM</td>
<td>Nonstructural mass per unit length (Real &gt; 0.0, or blank)</td>
</tr>
<tr>
<td>TMIN</td>
<td>Minimum rod area for design (Real &gt; 0.0, or blank). Default = 0.0001</td>
</tr>
</tbody>
</table>

 Remarks:

1. PROD entries must all have unique property identification numbers.

2. For structural problems, PROD entries may only reference MAT1 material entries.

3. The formula used to compute torsional stress is:

\[ \tau = \frac{C M_t}{J} \]

where \( M_t \) is the torsional moment.

4. TMIN is ignored unless the rod element is linked to the design variables by SHAPE entries.
Input Data Entry: PSHEAR   Shear Panel Property

Description: Defines the elastic properties of a shear panel. Referenced by the CSHEAR entry.

Format and Examples:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSHEAR</td>
<td>PID</td>
<td>MID</td>
<td>T</td>
<td>NSM</td>
<td>TMIN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSHEAR</td>
<td>13</td>
<td>2</td>
<td>4.9</td>
<td>16.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field   

PID   Property identification number (Integer > 0)
MID   Material identification number (Integer > 0)
T     Thickness of shear panel (Real > 0.0)
NSM   Nonstructural mass per unit area (Real ≥ 0.0, or blank)
TMIN  Minimum panel thickness for design (Real ≥ 0.0, or blank). Default = 0.0001

Remarks:

1. All PSHEAR entries must have unique identification numbers.
2. PSHEAR entries may reference only MAT1 material entries.
3. TMIN is ignored unless the element is linked to global design variables by SHAPE entries.
Input Data Entry: PSHELL Shell Element Property

Description: Defines the membrane, bending, transverse shear, and coupling properties of the shell elements. (QUAD4 and TRIA3)

Format and Examples:

<table>
<thead>
<tr>
<th>PID</th>
<th>MID1</th>
<th>T</th>
<th>MID2</th>
<th>121/T3</th>
<th>MID3</th>
<th>TS/T</th>
<th>NSM</th>
<th>CONT</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSHELL</td>
<td>203</td>
<td>1.90</td>
<td>204</td>
<td>1.2</td>
<td>205</td>
<td>0.8</td>
<td>6.32</td>
<td>ABC</td>
</tr>
<tr>
<td>+BC</td>
<td>+.95</td>
<td>-.95</td>
<td>0</td>
<td>0</td>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID</td>
<td>Property identification number (Integer &gt; 0)</td>
</tr>
<tr>
<td>MID1</td>
<td>Material identification number for membrane (Integer &gt; 0 or blank)</td>
</tr>
<tr>
<td>T</td>
<td>Default value for membrane thickness (Real &gt; 0.0, or blank)</td>
</tr>
<tr>
<td>MID2</td>
<td>Material identification number for bending (Integer &gt; 0, or blank)</td>
</tr>
<tr>
<td>121/T3</td>
<td>Bending stiffness parameter (Real &gt; 0.0, or blank, Default = 1.0)</td>
</tr>
<tr>
<td>MID3</td>
<td>Material identification number for transverse shear (Integer &gt; 0, or blank), must be blank unless MID2 &gt; 0</td>
</tr>
<tr>
<td>TS/T</td>
<td>Transverse shear thickness divided by membrane thickness (Real &gt; 0.0 or blank, Default = .833333)</td>
</tr>
<tr>
<td>NSM</td>
<td>Nonstructural mass per unit area (Real &gt; 0.0, or blank)</td>
</tr>
<tr>
<td>Z1,Z2</td>
<td>Fiber distances for stress computation. The positive direction is determined by the right-hand rule and the order in which the grid points are listed on the connection entry. (Real or blank, defaults are -1/2 T for Z1 and 1/2 T for Z2.)</td>
</tr>
<tr>
<td>MID4</td>
<td>Material identification number for membrane-bending coupling (Integer &gt; 0 or blank, must be blank unless MID1 &gt; 0 and MID2 &gt; 0, may not equal MID1 or MID2)</td>
</tr>
<tr>
<td>MCSID</td>
<td>Identification number of material coordinate system (Real or blank, or Integer ≥ 0) (See Remark 9)</td>
</tr>
<tr>
<td>SCSID</td>
<td>Identification number of stress coordinate system (Real or blank, or Integer ≥ 0) (See Remark 9)</td>
</tr>
<tr>
<td>ZOFF</td>
<td>Offset of the element reference plane from the plane of grid points. A positive value means the +z_{ref} direction. (Real or blank, default = 0.0) (See Remark 10)</td>
</tr>
<tr>
<td>TMIN</td>
<td>Minimum thickness for design (Real &gt; 0.0 or blank) (Default = 0.0001)</td>
</tr>
</tbody>
</table>

Remarks:

1. All PSHELL property entries must have unique identification numbers.
2. The structural mass is computed from the density using the membrane material properties.
3. The results of leaving an MID field blank are:
   - MID1: No membrane or coupling stiffness.
   - MID2: No bending, coupling, or transverse shear stiffness.
   - MID3: No transverse shear flexibility.
   - MID4: No bending-membrane coupling.

4. The continuation entry is not required.

5. The MID4 field should be left blank if the material properties are symmetric with respect to the middle surface of the shell.

6. This entry is used only with the CQUAD4 elements.

7. For structural problems, PSHELL entries may reference MAT1, MAT2, or MAT8 material property entries.

8. If the transverse shear material, MID3, references MAT2 data, then G33 must be zero. If MID3 references MAT8 data, then G1, Z and G2, Z must not be zero.

9. If MCSID/SCSID is left blank (0.0) or is real, it is considered to be the angle of rotation of the X axis of the material/stress coordinate system with respect to the X axis of the element coordinate system in the XY plane of the latter. If Integer, the orientation of the material/stress x-axis is along the projection of the x-axis of the specified coordinate system onto the x-y plane of the element system. The value of MCSID is the default value for the TM field on CQUAD4 Bulk Data entries.

10. The offset ZOFF may also be provided on the CQUAD4 or CTRIA3 Bulk Data entry. The element reference plane is located at the mid-thickness of the element parallel to the element mean plane.

11. TMIN is ignored unless element is linked to global design variables by SHAPE entries.

12. The hierarchy of local coordinate systems is:
   - MCSID supplies the default value for the TM field on the element connectivity entry
   - TM overrides MCSID if TM is not blank
   - SCSID defaults to the material coordinate system if SCSID is blank
Input Data Entry:  **PTRMEM**

**Description:** Defines property data for **TRMEM** element.

**Format and Examples:**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PTRMEM</strong></td>
<td>PID</td>
<td>MID</td>
<td>T</td>
<td>NSM</td>
<td>TMIN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PTRMEM</td>
<td>500</td>
<td>1000</td>
<td>0.15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field**  | **Contents**
---|---
PID | Property entry identification number (Integer > 0)
MID | Material property identification (Integer > 0)
T | Thickness of membrane element (Real > 0.0)
NSM | Nonstructural mass associated with the element (Real > 0.0, or blank)
TMIN | Minimum thickness for design (Real > 0.0 or blank) (Default = 0.0001)

**Remarks:**

1. The **PTRMEM** entry can reference either **MAT1**, **MAT2** or **MAT8** entries.
2. **TMIN** is ignored unless the element is linked to global design variables by **SHAPE** entries.
Input Data Entry: **RANDPS**  Power Spectral Density Specification

Description: Defines load set power spectral density factors for use in Random Analysis having the frequency dependent form.

\[ S_{jk}(F) = (x + iy) \times G(F) \]

Format and Examples:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DPS</td>
<td>SID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3</td>
<td>7</td>
<td>2.0</td>
<td>2.5</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

- **SID**: Random analysis set identification number (Integer > 0)
- **J**: Subcase identification number of excited load set (Integer > 0)
- **K**: Subcase identification number of applied load set (Integer > 0; K > J)
- **X, Y**: Components of complex number (Real)
- **TID**: Identification number of a TABEMMi entry which defines \( G(F) \) (Integer ≥ 0)

Remarks:

1. If \( J = K \), then Y must be 0.0.
2. For **TID** = 0, \( G(F) = 1.0 \).
3. Set identification numbers must be selected in Solution Control to be used.
4. **RANDPS** can only reference subcases included within a single loop (change in direct matrix input is not allowed).
Input Data Entry  **RBAR**  Rigid Bar

**Description:** Defines a Rigid Bar element with six degrees of freedom at each end.

**Format and Example:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBAR</td>
<td>SETID</td>
<td>EID</td>
<td>GA</td>
<td>GB</td>
<td>CNA</td>
<td>CNB</td>
<td>CMA</td>
<td>CMB</td>
<td></td>
</tr>
<tr>
<td>RBAR</td>
<td>1001</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>234</td>
<td>123</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field**

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>SETID</td>
<td>Multipoint constraint set identification number specified in Solution Control. (Integer &gt; 0)</td>
</tr>
<tr>
<td>EID</td>
<td>Rigid Bar element identification number. (Integer &gt; 0)</td>
</tr>
<tr>
<td>GA, GB</td>
<td>Grid point identification numbers of connection points. (Integer &gt; 0)</td>
</tr>
<tr>
<td>CNA, CNB</td>
<td>Independent degrees of freedom in the global coordinate system for the elements at grid point GA and GB. Indicated by any of the digits 1 through 6 with no embedded blanks. (Integer &gt; 0, or blank) (Remark 2)</td>
</tr>
<tr>
<td>CMA, CMB</td>
<td>Component numbers of dependent degrees of freedom in the global coordinate system assigned by the element at grid point GA and GB. Indicated by any of the digits 1 through 6 with no embedded blanks. (Integer &gt; 0 or blank) (Remarks 3 and 4)</td>
</tr>
</tbody>
</table>

**Remarks:**

1. The **RBAR** entry is selected in the Solution Control with the **MPC-SETID** option of the **BOUNDARY** command. **THIS IS AN ENHANCEMENT TO THE NASTRAN METHOD, WHICH DOES NOT ALLOW RIGID CONNECTIONS TO BE CHANGED FOR DIFFERENT BOUNDARY CONDITIONS.**

2. The total number of components in **CNA** and **CNB** must be six; for example, **CNA=1236, CNB=34.** The components must jointly be capable of representing any general rigid body motion of the element.

3. If both **CMA** and **CMB** are zero or blank, all of the degrees of freedom not in **CNA** and **CNB** will be made dependent, i.e. they will be placed in the m-set.

4. The m-set degrees of freedom specified on this entry may not be specified on other entries that define mutually exclusive sets.

5. Rigid element identification numbers must be unique within each element type for each **MPC** set identification number.
Input Data Entry  RBE1  Rigid Body Element, Form 1

Description: Defines a rigid body connected to an arbitrary number of grid points.

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBE1</td>
<td>SETID</td>
<td>EID</td>
<td>GN1</td>
<td>CN1</td>
<td>GN2</td>
<td>CN2</td>
<td>GN3</td>
<td>CN3</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>GN4</td>
<td>CN4</td>
<td>GN5</td>
<td>CN5</td>
<td>GN6</td>
<td>CN6</td>
<td>CONT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>&quot;UM&quot;</td>
<td>GM1</td>
<td>CM1</td>
<td>GM2</td>
<td>CM2</td>
<td>GM3</td>
<td>CM3</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>GM4</td>
<td>CM4</td>
<td>GM5</td>
<td>CM5</td>
<td>CM5</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| RBE1 | 1001 | 11 | 1 | 2 | 2 | 134 | 3 | 5 | AB |
| AB | 4 | 2 | | | | | | | CD |
| CD | UM | 1 | 13 | 2 | 1 | 12 | 5 | | |

Field  Contents

| SETID | Multipoint constraint set identification number specified in Solution Control. (Integer > 0) |
| EID | Rigid body element identification number. (Integer > 0) |
| GN1 | Grid point identification numbers at which independent degrees of freedom are assigned. (Integer > 0) |
| CN1 | Component numbers of independent degrees of freedom in the global coordinate system at grid points GN1, indicated by any of the digits 1 through 6 with no embedded blanks. (Integer > 0) (Remark 2) |
| "UM" | Character string indicating the start of the list of dependent degrees of freedom. |
| GMj | Grid point identification numbers at which dependent degrees of freedom are assigned. (Integer > 0) |
| CMj | Component numbers of dependent degrees of freedom in the global coordinate system at grid points GMj, indicated by any of the digits 1 through 6 with no embedded blanks. (Integer > 0) (Remark 2) |

Remarks:

1. The RBE1 entry is selected in the Solution Control with the MPC=SETID option of the BOUNDARY command. THIS IS AN ENHANCEMENT TO THE NASTRAN METHOD, WHICH DOES NOT ALLOW RIGID CONNECTIONS TO BE CHANGED FOR DIFFERENT BOUNDARY CONDITIONS.

2. The total number of components in CNi must be six; for example, CN1=123, CN2=3, CN3=2 and CN4=3. The components must jointly be capable of representing any general rigid body motion of the element. The m-set degrees of freedom specified on this entry may not be specified on other entries that define mutually exclusive sets.

3. A degree-of-freedom cannot be both independent and dependent for the same element. However, both independent and dependent components may exist at the same grid point.

4. Rigid element identification numbers must be unique within each element type for each MPC set identification number.
## Input Data Entry

**RBE2**

**Rigid Body Element, Form 2**

### Description:
Defines a body whose independent degrees of freedom are specified at a single grid point and whose dependent degrees of freedom are specified at an arbitrary number of grid points.

### Format and Example:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBE2</td>
<td>SETID</td>
<td>EID</td>
<td>GN</td>
<td>CM</td>
<td>GM1</td>
<td>GM2</td>
<td>GM3</td>
<td>GM4</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>GM5</td>
<td>GM6</td>
<td>GM7</td>
<td>GM8</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RBE2</td>
<td>1001</td>
<td>9</td>
<td>8</td>
<td>12</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>15</td>
<td>AB</td>
<td></td>
</tr>
<tr>
<td>AB</td>
<td>16</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Field Contents

- **SETID**: Multipoint constraint set identification number specified in Solution Control. (Integer > 0)
- **EID**: Rigid body element identification number. (Integer > 0)
- **GN**: Grid point identification number at which all 6 independent degrees of freedom are assigned. (Integer > 0)
- **CM**: Component numbers of dependent degrees of freedom in the global coordinate system assigned by the element at grid points GM1, GM2, etc. Indicated by any of the digits 1 through 6 with no embedded blanks. (Integer > 0 or blank)
- **GMi**: Grid point identification number at which dependent degrees of freedom are assigned. (Integer > 0)

### Remarks:

1. The **RBE2** entry is selected in the Solution Control with the **MPC=SETID** option of the **BOUNDARY** command. **THIS IS AN ENHANCEMENT TO THE NASTRAN METHOD, WHICH DOES NOT ALLOW RIGID CONNECTIONS TO BE CHANGED FOR DIFFERENT BOUNDARY CONDITIONS.**

2. The components indicated by **CM** are made dependent at all grid points **GMi**.

3. The m-set degrees of freedom specified on this entry may not be specified on other entries that define mutually exclusive sets.

4. Rigid element identification numbers must be unique within each element type for each **MPC** set identification number.
**Input Data Entry**

**RBE3**

**Rigid Body Element, Form 3**

**Description:** Defines the motion of a reference grid point as the weighted average of motions at a set of other grid points.

**Format and Example:**

```
1 2 3 4 5 6 7 8 9 10
RBE3 SETID EID REFG REFC WT1 C1 G1,1 G1,2 CONT
CONT G1,3 WT2 C2 G2,1 G2,2 -etc- WT3 CONT
CONT C3 G3,1 -etc- -etc- WT4 C4 G4,1 CONT
CONT G4,2 -etc- CONT
CONT "UM" GM1 CM1 GM2 CM2 GM3 CM3 CONT
CONT GM4 CM4 -etc-
```

```
RBE3 1001 14 100 1234 1.0 123 1 3 AE
+4 5 4.7 1 2 4 6 5.2 AF
+4 2 7 8 9 5.1 1 15 AG
+G 16
+H UM 100 14 5 3 7 2
```

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>SETID</td>
<td>Multipoint constraint set identification number specified in Solution Control. (Integer &gt; 0)</td>
</tr>
<tr>
<td>EID</td>
<td>Rigid body element identification number. (Integer &gt; 0)</td>
</tr>
<tr>
<td>REFG</td>
<td>Reference grid point identification number. (Integer &gt; 0)</td>
</tr>
<tr>
<td>REFC</td>
<td>Component numbers of degrees of freedom in the global coordinate system that will be computed at REFG, Indicated by any of the digits 1 through 6 with no embedded blanks. (Integer &gt; 0)</td>
</tr>
<tr>
<td>WTi</td>
<td>Weighting factor for most common defined by Gi,j. (Real)</td>
</tr>
<tr>
<td>Ci</td>
<td>Component numbers of degrees of freedom in the global coordinate system which have weighting factor WTi, at grid points Gi,j. Indicated by any of the digits 1 through 6 with no embedded blanks. (Integer &gt; 0)</td>
</tr>
<tr>
<td>Gi,j</td>
<td>Grid point identification number whose components Ci have weighting factor WTi. (Integer &gt; 0)</td>
</tr>
<tr>
<td>&quot;UM&quot;</td>
<td>Character string indicating the start of the list of dependent degrees of freedom. The default is that all of the components in REFC at REFG, and no others, will be placed in the m-set.</td>
</tr>
<tr>
<td>GMi</td>
<td>Grid point identification numbers with components in the m-set. (Integer &gt; 0)</td>
</tr>
<tr>
<td>CMi</td>
<td>Component numbers in the global coordinate system at grid points GMi which are placed in the m-set. Indicated by any of the digits 1 through 6 with no embedded blanks. (Integer &gt; 0 or blank) (Remark 2)</td>
</tr>
</tbody>
</table>
Remarks:

1. The RBE3 entry is selected in the Solution Control with the MPC-SETID option of the BOUNDARY command. **THIS IS AN ENHANCEMENT TO THE NASTRAN METHOD, WHICH DOES NOT ALLOW RIGID CONNECTIONS TO BE CHANGED FOR DIFFERENT BOUNDARY CONDITIONS.**

2. The form of $g_{i,j}$ is different than NASTRAN. The first data field on the continuations has been reserved for the "UM" identifier. The $g_{i,j}$ list must be contained within data fields 3 through 9. Blanks may appear anywhere in the list.

3. The default for "UM" should be used except in cases where the user wishes to include some or all of the REFC components in displacement sets other that the $m$-set. If the default is not used for "UM" then:
   - the total number of components in "UM" must equal the number of components in REFC.
   - the components in "UM" must be a subset of the components specified in the (REFG,REFC) and ($g_{i,j},Ci$).
   - the $m$-set coefficient matrix in the constraint equation must be nonsingular.

4. The $m$-set degrees of freedom specified on this entry may not be specified on other entries that define mutually exclusive sets.

5. Rigid element identification numbers must be unique within each element type for each MPC set identification number.
**Input Data Entry:** RLOAD1

**Description:** Defines a frequency dependent dynamic load of the form.

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

**Format and Examples:**

<table>
<thead>
<tr>
<th>SID</th>
<th>DLAGID</th>
<th>TC</th>
<th>TD</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

**Field Contents**

<table>
<thead>
<tr>
<th>SID</th>
<th>Set identification number (Integer &gt; 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLAGID</td>
<td>Identification number of a DLAGS set which defines A, ( \theta ) and ( \tau ) (Integer &gt; 0)</td>
</tr>
<tr>
<td>TC</td>
<td>Set identification number of TABLED1 entry which gives ( C(f) ) (Integer ( \geq 0 ); ( TC + TD &gt; 0 ))</td>
</tr>
<tr>
<td>TD</td>
<td>Set identification number of TABLED1 entry which gives ( D(f) ) (Integer ( \geq 0 ); ( TC + TD &gt; 0 ))</td>
</tr>
</tbody>
</table>

**Remarks:**

1. **RLOAD1** loads may be combined with **RLOAD2** loads only by specification on a DLOAD entry.
2. **SID** must be unique for all **RLOAD1**, **RLOAD2**, **TLOAD1** and **TLOAD2** entries.
Input Data Entry: RLOAD2

Description: Defines a frequency dependent dynamic load of the form.

\[ P(f) = AB(f) e^{i(\varphi f) + 2\pi f t} \]

Format and Examples:

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>RLOAD2</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>101</td>
</tr>
</tbody>
</table>

Field  Contents

SID     Set identification number (Integer > 0)
DLAGID  Identification of a DLAGS entry which defines A, B, and \( \tau \) (Integer > 0)
TB      Set identification number of TABLEDI entry which gives B(f) (Integer > 0)
TP      Set identification number of TABLEDI entry which gives \( \varphi(f) \) in degrees (Integer \( \geq 0 \))

Remarks:

1. RLOAD2 loads may be combined with RLOAD1 loads only by specification on a DLOAD entry. That is, the SID on a RLOAD2 entry may not be the same as that on a RLOAD1 entry.

2. SID must be unique for all RLOAD1, RLOAD2, TLOAD1 and TLOAD2 entries.
Input Data Entry  RROD  Rigid Rod

Description: Defines a pin-ended rod that is rigid in extension.

Format and Example:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>RROD</td>
<td>SETID</td>
<td>EID</td>
<td>GA</td>
<td>GB</td>
<td>CMA</td>
<td>CMB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RROD</td>
<td>1001</td>
<td></td>
<td>14</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field  Contents

SETID  Multipoint constraint set identification number specified in Solution Control. (Integer > 0)
EID  Rigid Rod element identification number. (Integer > 0)
GA, GB  Grid point identification numbers of connection points. (Integer > 0)
CMA, CMB  Component number of one, and only one, dependent degree-of-freedom in the global coordinate system assigned by the element at either grid point GA or GB. (Integer 1, 2 or 3, either CMA or CMB may contain the digit and the other must be blank)

Remarks:

1. The RROD entry is selected in the Solution Control with the MPC-SETID option of the BOUNDARY command. THIS IS AN ENHANCEMENT TO THE NASTRAN METHOD, WHICH DOES NOT ALLOW RIGID CONNECTIONS TO BE CHANGED FOR DIFFERENT BOUNDARY CONDITIONS.

2. The degree-of-freedom selected to be dependent must have a nonzero component along the axis of the rod; which also implies that the rod must have a finite length.

3. The m-set degrees of freedom specified on this entry may not be specified on other entries that define mutually exclusive sets.

4. Rigid element identification numbers must be unique within each element type for each MPC set identification number.
Input Data Entry:  SAVE

Description:  Defines a list of data base entities that are not to be purged.

Format and Examples:

```
  1  2  3  4  5  6  7  8  9  10
SAVE NAME1 NAME2 NAME3 NAME4 NAME5 NAME6 NAME7 NAME8 CONT
CONT NAME9 NAME10 NAME11 -etc-
SAVE DVCT
```

Field                  Contents
NAMEi                  The name of a data base entity whose contents are not to be purged.

Remarks:
1. Any number of continuations are allowed.
2. This data entry is used by the UTPURG utility to determine if a requested purge of an entity will take place.
Input Data Entry: **SEQGP**  
Grid and Scalar Point Resequencing

**Description:** Used to manually order the grid points and scalar points of the problem. The purpose of this card is to allow the user to reidentify the formation sequence of the grid and scalar points of the structural model in such a way as to optimize bandwidth.

**Format and Examples:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEQGP</td>
<td>ID</td>
<td>SEQID</td>
<td>ID</td>
<td>SEQID</td>
<td>ID</td>
<td>SEQID</td>
<td>ID</td>
<td>SEQID</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>ID</td>
<td>SEQID</td>
<td>etc-</td>
<td>etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SEQGP** 5392 15.6 596 0.2 2 1.9.2.6 3

**Field**                        | **Contents**
---                             | ---
ID                              | Grid point identification number (Integer > 0)
SEQID                           | Sequenced identification number (a special number described below)

**Remarks:**

1. **ID** is any grid or scalar point identification number which is to be reidentified for sequencing purposes. The sequence number identifies a special number which may have any of the following forms where X is a decimal integer digit - XXXX.X.X.X, XXXX.X.X, XXXX.X or XXXX where any of the leading Xes may be omitted. This number must contain no embedded blanks. The leading character must not be a decimal point.

2. If the user wishes to insert a point between two already existing grid or scalar points, such as 15 and 16, for example, he would define it as, say 5392, and then use this card to insert extra point number 5392 between them by equivalencing it to, say, 15.6. All output referencing this point will refer to 5392.3. The **SEQID** numbers must be unique and may not be the same as a point **ID** which is not being changed. No extra point **ID** may be referenced more than once.

3. The **SEQID** numbers must be unique and may not be the same as a point **ID** which is not being changed. No extra point **ID** may be referenced more than once.

4. If a point **ID** is referenced more than once, the last reference will determine its sequence.
**Input Data Entry:**  
*SET1*  
Set definition for aerodynamic analysis.

**Description:**  
Defines a set of integers by a list.

**Format and Examples:**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET1</td>
<td>SID</td>
<td>G1</td>
<td>G2</td>
<td>G3</td>
<td>G4</td>
<td>G5</td>
<td>G6</td>
<td>G7</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>G8</td>
<td>etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET1</td>
<td>3</td>
<td>31</td>
<td>62</td>
<td>93</td>
<td>124</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>ABC</td>
<td></td>
</tr>
<tr>
<td>+BC</td>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Alternate Form:**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET1</td>
<td>SID</td>
<td>G1</td>
<td>THRU</td>
<td>G2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field** | **Contents**
---|---
SID | Set of identification numbers (Integer > 0)
Gi | List of integers (Integer > 0)

**Remarks:**

1. These entries are referenced by the *SPLINE1* and *FLUTTER* data entries.
2. When using the *THRU* option, all intermediate quantities will be assumed to exist.
3. When used by *SPLINE1*, the entry refers to a list of structural grid points.
4. When used by *FLUTTER*, the entry refers to mode numbers to be omitted in the flutter analysis.
Input Data Entry: SET2 Grid Point List

Description: Defines a set of structural grid points in terms of aerodynamic macro elements.

Format and Examples:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET2</td>
<td>SID</td>
<td>SP1</td>
<td>SP2</td>
<td>CH1</td>
<td>CH2</td>
<td>ZMAX</td>
<td>ZMIN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SET2</td>
<td>3.00</td>
<td>.73</td>
<td>0.00</td>
<td>.667</td>
<td>1.00</td>
<td>-3.51</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

- **SID**: Set identification number (Integer > 0)
- **SP1, SP2**: Lower and higher span division points defining prism containing set 
  
  \[ 1.01 > \text{Real} > -.01 \]
- **CH1, CH2**: Lower and higher chord division points defining prism containing set 
  
  \[ 1.01 > \text{Real} > -.01 \]
- **ZMAX, ZMIN**: Z-coordinates of top and bottom (using right-hand rule with the order of the corners as listed on a CAEROi entry) of the prism containing set (Real). Usually \( ZMAX > 0.0, ZMIN < 0.0 \)

Remarks:

1. These entries are referenced by the SPLINEI data entries.

2. Every grid point, within the defined prism and within the height range, will be in the set. For example,

   ![Diagram](image)

   The shaded area in the figure defines the cross-section of the prism for the sample data given above. Points exactly on the boundary may be missed, hence, to get all the grid points within the area of the macro element, use \( SP1 = -.01, SP2 = 1.01 \), etc.

3. A zero value for \( ZMAX \) or \( ZMIN \) implies infinity is to be used.
**Input Data Entry:** SHAPE

**Description:** Defines element connectivity entries associated with a design variable.

**Format and Examples:**

<table>
<thead>
<tr>
<th>SHAPE</th>
<th>SHAPEID</th>
<th>ETYPE</th>
<th>EID1</th>
<th>PREF1</th>
<th>EID2</th>
<th>PREF2</th>
<th>EID3</th>
<th>PREF3</th>
<th>CONT</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHAPE</td>
<td></td>
<td>EID4</td>
<td></td>
<td>PREF4</td>
<td>EID5</td>
<td>PREF5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
1 2 3 4 5 6 7 8 9 10
SHAPE SHAPEID ETYPE EID1 PREF1 EID2 PREF2 EID3 PREF3 CONT
10 CROD 12 12.0 22 1.0
```

**Field** | **Contents**
--- | ---
SID | Shape function identification (Integer > 0)
ETYPE | Element type associated with this list (e.g., CROD)
EID1, EID2, EID3 | Element identification numbers (Integer > 0, or blank)
PREF1, PREF2, PREF3, PREF4 | Linking factor for the associated EID (Real)

**Remarks:**

1. Allowable ETYPEs are: CROD, CONROD, CSHEAR, CQDMEM1, CQUAD4, CTRIA3, CTRMEM, CBAR, CELAS1, CELAS2, CMASS1, CMASS2 and CMEM.

2. The shape function identification is referenced by the DESVARS entry to connect the global variable to the shape.

3. The linking factors define a shape function to be used as the global design variable.

4. Designed properties (e.g., thicknesses) of elements listed on SHAPE entries will be set to unity to ensure proper shape function definition; that is, the PREF values define the shape to be applied to a uniform property distribution.
Input Data Entry: SPC Single-Point Constraint

Description: Defines sets of single-point constraints and enforced displacements.

Format and Examples:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPC</td>
<td>SID</td>
<td>G</td>
<td>C</td>
<td>D</td>
<td>G</td>
<td>C</td>
<td>D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPC</td>
<td>2</td>
<td>32</td>
<td>436</td>
<td>-2.6</td>
<td>5</td>
<td></td>
<td>+2.9</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

- **SID**: Identification number of single-point constraint set (Integer > 0)
- **G**: Grid or scalar point identification number (Integer > 0)
- **C**: Component number of global coordinate (6 >= Integer >= 0; up to six unique digits may be placed in the field with no embedded blanks.)
- **D**: Value of enforced displacement for all coordinates designed by G and C (Real)

Remarks:

1. Degrees of freedom specified on this entry form members of a mutually exclusive set. They may not be specified on other entries that define mutually exclusive sets.
2. Single-point forces of constraint are recovered during stress data recovery.
3. Single-point constraint sets must be selected in Solution Control (SPC= SID) to be used.
4. SPC degrees of freedom may be redundantly specified as permanent constraints on the GRID entry.
**Input Data Entry:** SPCADD  
**Description:** Defines a single-point constraint set as a union of single-point constraint sets defined via SPC or SPC1 entries.

### Formal and Examples:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPCADD</td>
<td>SID</td>
<td>S1</td>
<td>S2</td>
<td>S3</td>
<td>S4</td>
<td>S5</td>
<td>S6</td>
<td>S7</td>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>S8</td>
<td>S9</td>
<td>-etc-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Field Contents

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>Identification number for new single-point constraint set (Integer &gt; 0)</td>
</tr>
<tr>
<td>Si</td>
<td>Identification numbers of single-point constraint sets defined via SPC or by SPC1 entries (Integer &gt; 0; SID ≠ Si)</td>
</tr>
</tbody>
</table>

### Remarks:

1. Single-point constraint sets must be selected in Solution Control (SPC = SID) to be used.
2. No Si may be the identification number of a single-point constraint set defined by another SPCADD entry.
3. The Si values must be unique.
4. SPCADD entries take precedence over SPC or SPC1 entries. If both have the same set ID, only the SPCADD entry will be used.
Input Data Entry:  

**SPC1**  
Single-Point Constraint, Alternate Form 1

**Description:** Defines sets of single-point constraints

**Format and Examples:**

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>Identification number of single-point constraint set (Integer &gt; 0)</td>
</tr>
<tr>
<td>C</td>
<td>Component number of global coordinate (any unique combination of the digits 1 through 6 (with no embedded blanks) when point identification numbers are grid points; must be null if point identification numbers are scalar points)</td>
</tr>
<tr>
<td>Gi, GIDi</td>
<td>Grid or scalar point identification numbers (Integer &gt; 0)</td>
</tr>
</tbody>
</table>

**Remarks:**

1. Note that enforced displacements are not available via this entry. As many continuation entries as desired may appear.
2. Coordinates specified on this entry form members of a mutually exclusive set. They may not be specified on other entries that define mutually exclusive sets.
3. Single-point constraint sets must be selected in Solution Control (**SPC = SID**) to be used.
4. **SPC** degrees of freedom may be redundantly specified as permanent constraints on the **GRID** entry.
5. If the alternate form is used, points in the sequence **GID1** through **GID2** are required to exist.
Input Data Entry: SPLINE1 Surface Spline

Description: Defines a surface spline for interpolating out-of-plane motion for aeroelastic problems.

Format and Examples:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPLINE1</td>
<td>EID</td>
<td>CP</td>
<td>MACROID</td>
<td>BOX1</td>
<td>BOX2</td>
<td>SETG</td>
<td>DZ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPLINE1</td>
<td>3</td>
<td>111</td>
<td>111</td>
<td>118</td>
<td>14</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field | Contents
--- | ---
EID | Element identification number (Integer > 0)
CP | Coordinate system defining the spline plane (Integer ≥ 0, or blank)
MACROID | Identification number of a CAERO entry which defines plane of spline (Integer > 0)
BOX1, BOX2 | First and last box whose motions are interpolated using this spline (Integer > 0)
SETG | Refers to a SET entry which lists the structural grid points to which the spline is attached (Integer > 0)
DZ | Linear attachment flexibility (Real ≥ 0.0)

Remarks:

1. The interpolated points (k-set) will be defined by aero-cells. The sketch shows the cells for which $u_x$ is interpolated if BOX1 = 111 and BOX2 = 118.

2. The attachment flexibility (units of area) is used for smoothing the interpolation. If DZ = 0.0, the spline will pass through all deflected grid points. If DZ >> (area of spline), a least squares plane fit will occur. Intermediate values will provide smoothing.

3. If no CP is specified, the spline plane is assumed to be the CAERO macro element plane.

4. The SPLINE EID is used only for error messages and need not be related to the macroelement identification number.
Input Data Entry:  SPLINE2

Description: Defines a beam spline for interpolating panels and bodies for steady and unsteady aeroelastic analyses.

Format and Examples:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPLINE1</td>
<td>EID</td>
<td>MACROID</td>
<td>BOX1</td>
<td>BOX2</td>
<td>SETG</td>
<td>DZ</td>
<td>DTOR</td>
<td>CID</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>DTHX</td>
<td>DTHY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPLINE1</td>
<td>1000</td>
<td>5000</td>
<td>5000</td>
<td>5100</td>
<td>10</td>
<td>0.</td>
<td>1.0</td>
<td>4</td>
<td>+SPL</td>
</tr>
<tr>
<td>+SPL</td>
<td>-1.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EID</td>
<td>Element identification number (Integer &gt; 0)</td>
</tr>
<tr>
<td>MACROID</td>
<td>The identification of a CAERO1, CAERO2, CAERO6 or PAERO6 aerodynamic macroelement to be splined (Integer &gt; 0)</td>
</tr>
<tr>
<td>BOX1, BOX2</td>
<td>The identification numbers of the first and last boxes on the macroelement to be interpolated using this spline (Integer &gt; 0)</td>
</tr>
<tr>
<td>SETG</td>
<td>The identification of a SETi entry which lists the structural grid points to which the spline is attached (Integer &gt; 0)</td>
</tr>
<tr>
<td>DZ</td>
<td>Linear attachment flexibility (Real ≥ 0.0)</td>
</tr>
<tr>
<td>DTOR</td>
<td>Torsional flexibility, $\frac{EI}{GJ}$ (Real ≥ 0.0; use 1.0 for bodies)</td>
</tr>
<tr>
<td>CID</td>
<td>Rectangular coordinate system which defines the y-axis of the spline (Integer ≥ 0 or blank; not used for bodies)</td>
</tr>
<tr>
<td>DTHX, DTHY</td>
<td>Rotational attachment flexibility. DTEX is for rotation about the x-axis; not used for bodies. DTHY is for rotation about the y-axis; used for slope of bodies. (Real)</td>
</tr>
</tbody>
</table>

Remarks:

1. The interpolation points (k-set) will be defined by aero-cells.
2. For panels, the spline axis is the projection of the y-axis of coordinate system CID, projected onto the plane of the panel. For bodies, the spline axis is parallel to the x-axis of the aerodynamic coordinate system.
3. The flexibilities are used for smoothing. Zero attachment flexibilities will imply rigid attachment, i.e., no smoothing. Negative values of DTEX and/or DTHY will imply no attachment.
4. The continuation card is optional.
5. The SPLINE2 EID must be unique with respect to all other SPLINEi data entries, it is used only for error messages.
Input Data Entry: **SPOINT** Scalar Point List

**Description:** Defines scalar points of the structural model

**Format and Examples:**

```
  1  2  3  4  5  6  7  8  9  10
```

<table>
<thead>
<tr>
<th>SPOINT</th>
<th>ID1</th>
<th>ID2</th>
<th>ID3</th>
<th>ID4</th>
<th>ID5</th>
<th>ID6</th>
<th>ID7</th>
<th>ID8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SPOINT</th>
<th>3</th>
<th>18</th>
<th>1</th>
<th>4</th>
<th>16</th>
<th>2</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Alternate Form:**

```
  1  2  3  4  5  6  7  8  9  10
```

<table>
<thead>
<tr>
<th>SPOINT</th>
<th>ID1</th>
<th>&quot;THRU&quot;</th>
<th>ID2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field Contents**

- ID1, ID2: Scalar point identification number (Integer > 0; ID1 < ID2)

**Remarks:**

1. If the alternate form is used, all scalar points ID1 through ID2 are defined.
Input Data Entry: **SUPORT**  Fictitious Support

**Description:** Defines coordinates at which the user desires determinate reactions to be applied to a free body during analysis.

**Format and Examples:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUPORT</td>
<td>SETID</td>
<td>ID</td>
<td>C</td>
<td>ID</td>
<td>C</td>
<td>ID</td>
<td>C</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| SUPORT | 1000 | 16 | 215 | | | | | |

**Field** | **Contents**
---|---
SETID | Solution control **SUPORT** set identification (Integer > 0)
ID | Grid or scalar point identification number (Integer > 0)
C | Component number (zero or blank for scalar points; any unique combination of the digits 1 through 6 for grid points)

**Remarks:**

1. Coordinates specified on this entry form members of a mutually exclusive set. They may not be specified on other entries that define mutually exclusive sets.
2. From one to three support coordinates may be defined on a single entry.
3. Continuation entries are not allowed.
Input Data Entry: **TABDMP1** Modal Damping Table

**Description:** Defines modal damping as a tabular function of frequency.

**Format and Examples:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABDMP1</td>
<td>ID</td>
<td>TYPE</td>
<td>F1</td>
<td>G1</td>
<td>F2</td>
<td>G2</td>
<td>F3</td>
<td>G3</td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>F4</td>
<td>G4</td>
<td>F5</td>
<td>G5</td>
<td>F6</td>
<td>G6</td>
<td>-etc-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TABDMP1</td>
<td>3</td>
<td>G</td>
<td>0.0</td>
<td>0.005</td>
<td>1.0</td>
<td>0.008</td>
<td>2.0</td>
<td>0.001</td>
<td>ABC</td>
</tr>
<tr>
<td>+BC</td>
<td>2.5</td>
<td>.01057</td>
<td>2.6</td>
<td>.01362</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field Contents**

<table>
<thead>
<tr>
<th>ID</th>
<th>Table identification number (Integer &gt; 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE</td>
<td>Data word which indicates the type of damping units, G, CRIT, Q, or blank. Default is G.</td>
</tr>
<tr>
<td>F1</td>
<td>Frequency value in cycles per unit time (Real ≥ 0.0).</td>
</tr>
<tr>
<td>G1</td>
<td>Damping value (Real).</td>
</tr>
</tbody>
</table>

**Remarks:**

1. The F1 must be in either ascending or descending order but not both.
2. Jumps between two points (Fi = Fi+1) are allowed, but not at the end points.
3. At least two entries must be present.
4. Any F1, G1 entry may be ignored by placing the BCD string SKIP in either of two fields used for that entry.
5. The TABDMP1 mnemonic infers the use of the algorithm:

   \[ g = g_e (F) \]

   where F is input to the table and g is returned. The table look-up \( g_T (F) \) is performed using linear interpolation within the table and linear extrapolation outside the table using the last two end points at the appropriate table end. At jump points the average \( g_T (F) \) is used. There are no error returns from this table look-up procedure.

6. If TYPE is G or blank, the damping values are in structural damping units, that is, the value of g in \((1+ig)K\). If TYPE is CRIT, the damping values are in the units of fraction of critical damping, C/CO. If TYPE is Q, the damping values are in the units of the amplification or quality factor, Q. These constants are related by the following equations:

   \[ \frac{C}{C_0} = \frac{g}{2}, \]

   \[ Q = \left\{ \begin{array}{cl}
   \frac{1}{2C} & (C_0) \\
   \frac{1}{C_0} & \\
   \frac{1}{g} & \\
   \end{array} \right\} \]
**Input Data Entry:**  **TABLED1**

**Description:**  Defines a tabular function for use in generating frequency-dependent and time-dependent dynamic loads.

**Format and Examples:**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLED1</td>
<td>ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>x1</td>
<td>y1</td>
<td>x2</td>
<td>y2</td>
<td>x3</td>
<td>y3</td>
<td>etc</td>
<td></td>
<td></td>
<td>CONT</td>
</tr>
<tr>
<td>TABDMP1</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ABC</td>
</tr>
<tr>
<td>+BC</td>
<td>-3.0</td>
<td>6.9</td>
<td>2.0</td>
<td>5.6</td>
<td>3.0</td>
<td>5.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field** | **Contents**
--- | ---
ID | Table identification number (Integer > 0)
x_i, y_i | Tabular entries (Real)

**Remarks:**

1. The x_i must be in either ascending or descending order but not both.
2. Jumps between two points (x_i = x_i+1) are allowed, but not at the end points.
3. At least two entries must be present.
4. Any x-y entry may be ignored by placing the string *SKIP* in either of the two fields used for that entry.
5. The generated function is:

   \[ y = y_T(x) \]

   where X is input to the table and Y is returned. The table look-up \( y_T(x) \) is performed using linear interpolation within the table and linear extrapolation outside the table using the last two end points at the appropriate table end. At jump points the average \( y_T(x) \) is used. There are no error returns from this table look-up procedure.
Input Data Entry: **TABRND1** Power Spectral Density Table

**Description:** Defines Power Spectral density as a tabular function of frequency for use in Random Analysis. Referenced on the RANDPS entry.

**Format and Examples:**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>CONT</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABRND1</td>
<td>ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CONT</td>
</tr>
<tr>
<td>CONT</td>
<td>F1</td>
<td>G1</td>
<td>F2</td>
<td>G2</td>
<td>F3</td>
<td>G3</td>
<td>F4</td>
<td>G4</td>
<td></td>
<td></td>
<td>-etc-</td>
</tr>
<tr>
<td>TABRND1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ABC</td>
</tr>
<tr>
<td>+BC</td>
<td>2.5</td>
<td>.01057</td>
<td>2.6</td>
<td>.01362</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field** | **Contents**
---|---
ID | Table identification number (Integer > 0)
fi | Frequency value in cycles per unit time (Real > 0.0)
gi | Power Spectral Density (Real)

**Remarks:**
1. The $f_i$ must be in either ascending or descending order but not both.
2. Jumps between two points ($f_i = f_{i+1}$) are allowed, but not at the end points.
3. At least two entries must be present.
4. Any $f$-$g$ entry may be ignored by placing the BCD string **SKIP** in either of the two fields used for that entry.
5. The **TABRND1** mnemonic infers the use of the algorithm

$$g = g_T (F)$$

where $F$ is input to the table and $G$ is returned. The table look-up $g_T (F)$ is performed using linear interpolation within the table and linear extrapolation outside the table using the last two end points at the appropriate table end. At jump points the average $g_T (F)$ is used. There are no error returns from this table look-up procedure.
Input Data Entry: TEMP Grid Point Temperature Field

Description: Defines temperature at grid points for determination of (1) Thermal Loading; and (2) data recovery.

Format and Examples:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEMP</td>
<td>SID</td>
<td>G</td>
<td>T</td>
<td>G</td>
<td>T</td>
<td>G</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEMP</td>
<td>3</td>
<td>94</td>
<td>316.2</td>
<td>49</td>
<td>219.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>Temperature set identification number (Integer &gt; 0)</td>
</tr>
<tr>
<td>G</td>
<td>Grid point identification number (Integer &gt; 0)</td>
</tr>
<tr>
<td>T</td>
<td>Temperature (Real)</td>
</tr>
</tbody>
</table>

Remarks:

1. From one to three grid point temperatures may be defined on a single entry.

2. Average element temperatures are obtained as a simple average of the connecting grid point temperatures when no element temperature data are defined.

3. For each thermal load, temperatures must be specified for all grid points using either TEMP or TEMPD entries.
Input Data Entry: **TEMPD**  
Grid Point Temperature Field Default

**Description:** Defines a temperature value for all grid points of the structural model which have not been given a temperature on a **TEMP** entry.

**Format and Examples:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEMPD</td>
<td>SID</td>
<td>T</td>
<td>SID</td>
<td>T</td>
<td>SID</td>
<td>T</td>
<td>SID</td>
<td>T</td>
<td></td>
</tr>
</tbody>
</table>

| TEMPD | 1 | 215.3 |

**Field** | **Contents**
--- | ---
SID | Temperature set identification number (Integer > 0)
T | Default temperature value (Real)

**Remarks:**

1. From one to four default temperatures may be defined on a single entry.
2. Average element temperatures are obtained as a simple average of the connecting grid point temperatures when no element temperature data are defined.
3. For each thermal load, temperatures must be specified for all grid points using either **TEMP** or **TEMPD** entries.
Input Data Entry: TF  Dynamic Transfer Function

Description:
1. Used to define a transfer function of the form

\[ (B_0 + B_1 p + B_2 p^2) u_d + \sum_i \left( A_0(i) + A_1(i) p + A_2(i) p^2 \right) u_i = 0 \]

2. May also be used as a means of direct matrix input. See Remark 3.

Format and Examples:

<table>
<thead>
<tr>
<th>TF</th>
<th>SID</th>
<th>GD</th>
<th>CD</th>
<th>B0</th>
<th>B1</th>
<th>B2</th>
<th>CONT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONT</td>
<td>G(1)</td>
<td>C(1)</td>
<td>A0(1)</td>
<td>A1(1)</td>
<td>A2(1)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TF 1 2 3 4.0 5.0 6.0 +ABC
+ABC

Field Contents

SID  Set identification (Integer > 0).
GD, G(i) Grid, scalar or extra point identification numbers (Integer > 0).
CD, C(i) Component numbers (null or zero for scalar or extra points, any one of the digits 1 through 6 for a grid point).
B0, B1, B2, Transfer function coefficients (Real).
A0(i), A1(i), A2(i)

Remarks:
1. The matrix elements defined by this entry are added to the dynamic matrices for the problem.
2. Transfer function sets must be selected in Solution Control (TFL = SID) to be used.
3. The constraint relation given in Equation 1 will hold only if no structural elements or other matrix elements are connected to the dependent coordinate, u_d. In fact, the terms on the left side of Equation 1 are simply added to the terms from all other sources in the row for u_d.
4. Any number of continuations are allowed.
Input Data Entry: TIMELIST

Description: Defines a list of times at which outputs are desired.

Format and Examples:

```
1  2  3  4  5  6  7  8  9  10
TIMELIST   SID   TIME   TIME   TIME   TIME   TIME   TIME   TIME   CONT
CONT   TIME   TIME   -etc-

TIMELIST  100   0.1   0.2   0.5   1.0
```

Field Contents

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>Set identification number referenced by Solution Control (Integer &gt; 0)</td>
</tr>
<tr>
<td>TIME</td>
<td>Time, (in consistent time unit) at which outputs are desired. (Real)</td>
</tr>
</tbody>
</table>

Remarks:

1. In order to be used, the SID must be referenced by Solution Control.
2. The nearest time to TIME, either above or below, which was used in the Transient Response analysis will be used to satisfy the output requests.
3. Any number of continuations is allowed.
Input Data Entry:  TLOAD1

Description: Defines a time dependent function of the form:

\[ P(t) = AF(t - \tau) \]

for use in a transient response problem.

Format and Examples:

<p>| | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TLOAD1</td>
<td>SID</td>
<td>DLAGID</td>
<td>TID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TLOAD1</td>
<td>10</td>
<td>8</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Contents

- **SID**: Set identification number (Integer > 0)
- **DLAGID**: Identification number of DLAGS set which defines A and \( \tau \) (Integer > 0)
- **TID**: Identification number of a TABLE1 entry which gives \( F(t-\tau) \) (Integer > 0)

Remarks:

1. SID must be unique for all TLOAD1, TLOAD2, RLOAD1, and RLOAD2 entries.
Input Data Entry: TLOAD2

Description: Defines a time-dependent dynamic load of the form:

\[ P(t) = \begin{cases} 0 & \text{when } \bar{t} < 0 \text{ or } \bar{t} > \omega - \alpha \\ A \bar{t}^n e^{-\pi t} \cos(2\pi \bar{t} + \theta) & \text{when } 0 \leq \bar{t} \leq \omega - \alpha \end{cases} \]

where \( \bar{t} = t - \omega - \tau \)

Format and Examples:

<table>
<thead>
<tr>
<th>TLOAD2</th>
<th>SID</th>
<th>DLAGID</th>
<th>T1</th>
<th>T2</th>
<th>FREQ</th>
<th>PHASE</th>
<th>CTEXP</th>
<th>GROWTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLOAD2</td>
<td>10</td>
<td>6</td>
<td>2.1</td>
<td>4.7</td>
<td>12.0</td>
<td>30.0</td>
<td>2.0</td>
<td>3.0</td>
</tr>
</tbody>
</table>

Field Contents

- **SID**: Set identification number (Integer > 0)
- **DLAGID**: Identification number of the DLAGS entry set which define the time invariant load A and the time delay (Integer > 0)
- **T1**: Time constant (Real \( \geq 0.0 \))
- **T2**: Time constant (Real, \( T2 > T1 \))
- **FREQ**: Frequency in cycles per unit time (Real \( \geq 0.0 \))
- **PHASE**: Phase angle in degrees (Real)
- **CTEXP**: Exponential coefficient (Real)
- **GROWTH**: Growth coefficient (Real)

Remarks:

1. TLOAD2 loads may be combined with TLOAD1 loads only by specification on a DLOAD entry.
2. SID must be unique for all TLOAD1, TLOAD2, RLOAD1 and RLOAD2 entries.
**Input Data Entry**  TRIM  Trim Variable Specification

**Description:** Specifies conditions for steady aeroelastic trim or nonplanar steady aerodynamic analysis.

**Format and Example:**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIM</td>
<td>TRIMID</td>
<td>MACH</td>
<td>QDP</td>
<td>TRMTYP</td>
<td>EFFID</td>
<td>VO</td>
<td>CONT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONT</td>
<td>LABEL1</td>
<td>VAL1</td>
<td>LABEL2</td>
<td>VAL2</td>
<td>LABEL3</td>
<td>VAL3</td>
<td>LABEL4</td>
<td>VAL4</td>
<td>-etc-</td>
<td></td>
</tr>
</tbody>
</table>

TRIM

<table>
<thead>
<tr>
<th></th>
<th>1001</th>
<th>0.90</th>
<th>1200.</th>
<th>LIFT</th>
<th>100</th>
<th>926.3</th>
<th>+ABC</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ABC</td>
<td>NZ</td>
<td>8.0</td>
<td>QRATE</td>
<td>0.243</td>
<td>ELEV</td>
<td>FREE</td>
<td>ALPHA</td>
</tr>
</tbody>
</table>

**Field**

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIMID</td>
<td>Trim set identification number (Integer&gt;0)</td>
</tr>
<tr>
<td>MACH</td>
<td>Mach number (Real ≥ 0.0)</td>
</tr>
<tr>
<td>QDP</td>
<td>Dynamic pressure (Real&gt;0.0)</td>
</tr>
<tr>
<td>TRMTYP</td>
<td>Type of trim required (Character or blank) (See Remark 3)</td>
</tr>
<tr>
<td></td>
<td>blank</td>
</tr>
<tr>
<td></td>
<td>ROLL</td>
</tr>
<tr>
<td></td>
<td>LIFT</td>
</tr>
<tr>
<td></td>
<td>PITCH</td>
</tr>
<tr>
<td>EFFID</td>
<td>Identification number of CONEFFECTS Bulk Data entries which modify control surface effectiveness values (Integer ≥ 0, or blank)(Remark 2)</td>
</tr>
<tr>
<td>VO</td>
<td>True velocity (Real&gt;0.0, or blank) (See Remark 12)</td>
</tr>
<tr>
<td>LABEL1</td>
<td>Label defining aerodynamic trim parameters.</td>
</tr>
<tr>
<td>VAL1</td>
<td>Magnitude of the specified trim parameter (Real) or the character string FREE.</td>
</tr>
</tbody>
</table>

**Remarks:**

1. The **TRIM** entry is selected in Solution Control in the **SAERO** and **NPSAERO** disciplines with the **TRIM** option.

2. All aerodynamic forces created by the control surface will be reduced to the referenced amount. For example, an **EFFI** of 0.70 indicates a 30% reduction in the forces.

3. The **TRMTYP** field has the following interpretation:

   **LIFT**
   
   Implies that the vertical acceleration will be trimmed by one **FREE** symmetric control parameter or surface, or, the acceleration computed for some set of symmetric parameters/surfaces.
ROLL implies that the roll acceleration, $P_{ACC\ell}$, will be trimmed by some one FREE antisymmetric control parameter or surface — OR — the acceleration computed for some set of antisymmetric parameters/surfaces. Any number of antisymmetric parameters may be fixed, but the FREE parameters are limited to $P_{ACC\ell}$ — OR — any one antisymmetric parameter or surface. For example, $P_{ACC\ell}=0.0; \text{AILERON}=1.0; \text{PRATE}=\text{FREE}$

PITCH implies that the vertical acceleration, $N_{Z}$, and the pitch acceleration, $Q_{ACC\ell}$, will be trimmed by no more than two FREE symmetric control parameters or surfaces — OR — the accelerations computed for some set of symmetric parameters/surfaces. Any number of symmetric parameters may be fixed, but the FREE parameters are limited to $Q_{ACC\ell}$ and $N_{Z}$ — OR — up to two symmetric parameters or surfaces — OR — some combination. For example, $N_{Z}=9.0g's; Q_{ACC\ell}=0.0; \text{ALPHA}=\text{FREE}; \text{ELEV}=\text{FREE}$

blank implies that the support DOFs are equal to the number of free parameters. Appropriate trim equations are assembled and solved.

4. Units for $QDP$ are force per unit area.

5. Allowable options for $LABEL_{i}$ for symmetric trim (the symmetry option is selected in Solution Control) are:

   **Structural Accelerations**

   - $NX$ Longitudinal acceleration (Remark 10)
   - $NZ$ Vertical acceleration (load factor) (Remark 10)
   - $Q_{ACC\ell}$ Pitch acceleration (Remark 11)

   **Aerodynamic Parameters**

   - $\text{ALPHA}$ Angle of attack in degrees
   - $\text{QRATE}$ Pitch rate (Remark 12)
   - $\text{THICKCAM}$ Thickness and camber (Remark 13)
   and Control surfaces in degrees

6. Allowable options for $LABEL_{i}$ for antisymmetric trim (the symmetry option is selected in Solution Control) are:

   **Structural Accelerations**

   - $NY$ Side-slip acceleration (Remark 10)
   - $P_{ACC\ell}$ Roll acceleration (Remark 11)
   - $R_{ACC\ell}$ Yaw acceleration (Remark 11)

   **Aerodynamic Parameters**

   - $\text{BETA}$ Yaw angle in degrees (Remark 14)
   - $\text{PRATE}$ Roll rate (Remark 12)
   - $\text{RRATE}$ Yaw rate (Remark 14)
   and Antisymmetric control surfaces in degrees

7. If $VALUE_{i}$ is a real number, the associated aerodynamic parameter or structural acceleration is set to that value. If $VALUE_{i}$ is the character string FREE, then the associated parameter will be determined as part of the trim analysis.
8. The number of **FREE** Values of **VALUE** must correspond exactly to the number of unknowns in the trim analysis. If **TRMTYP** is blank, the number of **SUPORT** DOF.

9. If **TRMID** is referenced by an **NPSAERO** discipline, **TRMTYP** must be blank and **FREE** is not allowed for **VALUE**.

10. For **NX**, **NY** and **NZ**, units are length per second in consistent units unless a **CONVERT/MASS** Bulk Data entry is provided. In this case, the values are dimensionless.

11. The angular accelerations, **QACCEL**, **PACCEL**, and **RACCEL**, are entered in units of radians per second per second.

12. **QRATE**, **PRATE**, and **RRATE**, are entered in units of radians per second. The velocity must be input if any of the "rate" parameters are given since its value is needed to dimensionalize the forces computed for a unit rate per velocity in the aerodynamic preface.

13. The **THKCAM** label refers to thickness and camber effects and its corresponding value is usually set to 1.0. Non-unit values of the **THKCAM** parameter are available only to provide added generality.

14. Any control surfaces, trim parameters, or structural accelerations not specified on the **TRIM** entry will not participate in the analysis: they will be given fixed values of 0.0. This includes **THKCAM**.

15. Refer to the **STATIC AEROELASTIC TRIM** Application Note for more information.
**Input Data Entry:** TSTEP

**Description:** Defines time step intervals at which a solution will be generated and output in transient analysis.

**Format and Examples:**

```
1 2 3 4 5 6 7 8 9 10
TSTEP SID N(1) DT(1) NO(1) CONT
CONT N(2) DT(2) NO(2) CONT
```

```
TSTEP 2 10 .001 5 +ABC
+ABC 9 0.01 1
```

<table>
<thead>
<tr>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID</td>
<td>Set identification number (Integer &gt; 0)</td>
</tr>
<tr>
<td>N(i)</td>
<td>Number of time steps of value DT (i) (Integer ≥ 2)</td>
</tr>
<tr>
<td>DT(i)</td>
<td>Time increment (Real &gt; 0.0)</td>
</tr>
<tr>
<td>NO(i)</td>
<td>Skip factor for output (every NO (i) th step will be saved for output) (Integer &gt; 0)</td>
</tr>
</tbody>
</table>

**Remarks:**

1. TSTEP entries must be selected in the Solution Control (TSTEP=SID).

2. Note that the entry permits changes in the size of the time step during the course of the solution. Thus, in the example shown, there are 10 time steps of value .001 followed by 9 time steps of value .01. Also, the user has requested that output be recorded for t = 0.0, .005, .01, .02, .03, etc.
Input Data Entry: **VSDAMP**

**Description:** Specifies values of g and/or \( \omega_3 \) to generate either viscous damping that has the same damping forces as structural damping of magnitude g at the frequency \( \omega_3 \) or to specify the structural damping g (see Remarks 3 and 4).

**Format and Examples:**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>VSDAMP</td>
<td>SID</td>
<td>G</td>
<td>( \omega_3 )</td>
<td>SID</td>
<td>G</td>
<td>( \omega_3 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VSDAMP</td>
<td>100</td>
<td>0.005</td>
<td>15.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field Contents**

- **SID**: Set identification number (Integer > 0)
- **G**: Damping value (Real)
- **\( \omega_3 \)**: Frequency value in Hertz (Real \( \geq 0.0 \))

**Remarks:**

1. The setid is selected by the **DAMPING=n** command in Solution Control.
2. Up to two values of g and \( \omega_3 \) can be defined on a single entry.
3. If \( \omega_3 \) is zero, g will be used to generate a complex stiffness matrix \( K = (1 + ig)K \).
4. If \( \omega_3 \) is nonzero, a viscous damping matrix of the form \( B = \frac{g}{2\pi\omega_3}K \) is generated.
5. ASTROS OUTPUT FEATURES

In a software system the magnitude of ASTROS, the amount of data that may be of interest to the user is very large. In multidisciplinary optimization, the quantity of data is even larger and the expense involved in its computation even more critical. It is worthwhile, therefore, to limit the amount of output to a minimum and to provide a mechanism for the user to select those data that are of importance in each particular case. Chapter 3 of this report documented one mechanism provided in ASTROS to select particular iterations, disciplines, subcases and response quantities: that of the Solution Control output request. This section endeavors to present the totality of output options available to the ASTROS user. The system controlled outputs from the engineering modules are described in order to establish a familiarity with an ASTROS output listing. This is followed by a more complete description of output from each Solution Control request than is contained in Chapter 3, with different disciplines, elements, design constraints and node types accounted for in some detail. These represent the outputs that are fully supported by the ASTROS software and require little or no user intervention to obtain. The presentation of these features assumes that the standard executive sequence is used. If the user substantially modifies the standard sequence (to the point where certain modules are not called), some or all of the presented output features may no longer be available.

The more advanced forms of user output requests are also presented in this section. The most basic of these forms involve changing the engineering module print control levels through the use of the DEBUG packet. Then, the MAPOL addressable print utilities are presented. The use of these utilities, in conjunction with the general versatility of the MAPOL language, provides the user with the capacity both to look at existing data and to compute and view additional data. In fact, these options enable the user to obtain virtually any data that reside on the data base or that can be computed and stored on the data base. Finally, a quick overview of the Interactive CADDB Environment (ICE) is given. The ICE program provides for complete Standard Query Language interactive queries on the CADDB entities.

5.1. SYSTEM CONTROLLED OUTPUT

Many of the engineering and executive system program units write data to the ASTROS output listing automatically. As enumerated in the introduction to this section, output of this nature in ASTROS is very limited, but sufficient amounts exist to justify a brief presentation of the data and their formats. It is also useful to present the basic ASTROS listing in order to facilitate contrasting it to listings containing user selected output quantities. The first page of ASTROS output is the title page showing the version number, date and host machine of the user's system. Each page of output following the solution control listing will be labeled with six lines of header information including the user selected title, subtitle and label. The version number, date and, if applicable, the design iteration number will also appear in the header of each page.
5.1.1. Default Output Printed by Modules

The DEBUG packet echo and the ASSIGN DATABASE entries, shown in Table 18, are the first output following the title page. Immediately following these, the solution control commands will be echoed to the output listing. This listing is helpful in identifying the particular disciplines and cases selected in the run. The multidisciplinary nature of ASTROS requires further output labeling; therefore, in addition to the solution control summary, the BOUND module writes a summary of selected disciplines for each boundary condition at the top of the boundary condition loop, presented in Table 19. It indicates all disciplines and most discipline options in the current boundary condition to assist the user in determining the particular path that will be taken through the standard MAPOL sequence. A similar print, Table 20, from the ABOUND module appears at the top of the sensitivity phase boundary condition loop to indicate the nature of the active boundary conditions and active design constraints.

The next set of output, Table 21, comes from the bandwidth minimizer. It details the method selected, numbers of grids and elements in the model and the values of the measures of merit in the resequencing of grid points.

Active constraint information is provided in the Active Constraint Summary from the ACTCON module. It indicates the total number of constraints considered active according to the current constraint deletion criteria. The user may select a complete listing of the active constraints through the PRINT DCONSTRAINT solution control option, but may not suppress the table header indicating the number of constraints retained of the total number applied. This number is computed even if the current design is considered to be the converged optimum. A summary of the convergence criteria and of the critical constraint value is included in the Active Constraint Summary header, illustrated in Table 22, if the approximate problem was considered converged following the preceding redesign step.

Each redesign step is summarized in a small table, shown in Table 23, entitled the Approximate Optimization Summary. It indicates the optimization method used in resizing and the changes in three measures of convergence. The first measure is the change in the value of the objective function during the solution of the approximate optimization problem. The second is the change in the Euclidean norm of the design variable vector and finally, the maximum absolute change in any component of the design variable vector. Each of the values are computed as an absolute change and a percentage change. These values are then printed. You may compare the first two percentage values against your input convergence limit, denoted UPPER BOUND PERCENT MOVE, to determine which (if either) is greater than the limit. If either value is greater, the approximate problem will not be considered converged, otherwise it will be. A message indicating the state of convergence closes the Approximate Optimization Summary.

The last default design print, Table 24, is generated by the ACTCON module on the final design iteration. The ACTCON module prints out the design iteration history. The iteration history includes statistics summarizing each approximate optimization problem and shows the increments in the objective function. All values in this table are associated with the approximate problem. Since weight in ASTROS is explicitly linear in the design variables, the objective function values are exact.

The final default outputs are a trailer indicating the status of the termination (either with or without errors), the date and the time the run was completed and an execution timing summary. The timing summary, shown in Table 25, indicates the CPU time spent in each phase of the execution. The
Table 18. DEBUG and ASSIGN DATABASE Output

```
... ...
AUTOMATED STRUCTURAL OPTIMIZATION SYSTEM ...

*** DEBUG KEY "LOGBEGIN" HAS BEEN SELECTED
*** DEBUG KEY "LOGMODULE" HAS BEEN SELECTED
*** DEBUG KEY "MATRIX" HAS BEEN SELECTED

*** AUTOASSIGN DATABASE COMMAND ECHO ******

ASSIGN DATABASE NAME SHAZAM NEW

DATA BASE NAME = SHAZAM
DATA BASE PASSWORD = SHAZAM
DATA BASE STATUS = NEW
USER PARAMETERS ARE:
** NONE GIVEN **
```

Table 19. Boundary Condition Summary

```
BOUNDARY CONDITION SUMMARY FOR BOUNDARY CONDITION 2

MATRIX SUMMARY:
THE PHYSICAL SET CONTAINS 1948 DEGREES OF FREEDOM (DOFS)
1948 PHYSICAL IXES ARE STRUCTURAL
AND 0 PHYSICAL IXES ARE EXTRA POINTS

DOFS ARE ATTACHED MULTIPLE CONSTRAINT IN SYZ LEAVING
*** INDEP. DOFS

THE 1948 IXES ARE USING STATIC CONVENTION
THERE ARE 2,984 HÍTTLE IXE LEAVING
7% ANALYSIS SET IXES
*** WHICH ARE SUPPORTED LEAVING
1% IXES THAT OVER

DISCLAIMER SUMMARY:
*** STATICS HAS BEEN SELECTED ***
** SUPERSAT. UNITS DEFINED BY SOLUTION CONTROL
*** SUPERSAT. UNITS SELECTED ***
A PLUS MECHANICAL WAYS WILL ALSO BE DONE IF N'T ALREADY SELECTED
1 SUPERSAT. UNITS DEFINED BY SOLUTION CONTROL
```

357
### Table 20. Active Boundary and Constraint Summary

**SENSITIVITY SUMMARY FOR BOUNDARY CONDITION 2**

- 4 FLUTTER CONSTRAINTS
- 6 TOTAL ACTIVE CONSTRAINTS FOR THIS BOUNDARY CONDITION.

**SENSITIVITY SUMMARY FOR BOUNDARY CONDITION 2**

- 2 TSAI-WU STRESS CONSTRAINTS ON STATIC SUBCASE 1
- 6 TSAI-WU STRESS CONSTRAINTS ON STATIC SUBCASE 2
- 4 FLUTTER CONSTRAINTS

- 12 TOTAL ACTIVE CONSTRAINTS FOR THIS BOUNDARY CONDITION.

### Table 21. Resequecing Summary

<table>
<thead>
<tr>
<th>Method Selected</th>
<th>CM Criterion</th>
<th>RMS Wavefront</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BEFORE</td>
<td>AFTER</td>
</tr>
<tr>
<td>BANDWIDTH</td>
<td>648</td>
<td>640</td>
</tr>
<tr>
<td>PROFILE</td>
<td>2.9660</td>
<td>2.9640</td>
</tr>
<tr>
<td>MAXIMUM WAVEFRONT</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>AVERAGE WAVEFRONT</td>
<td>35.45</td>
<td>36.41</td>
</tr>
<tr>
<td>RMS WAVEFRONT</td>
<td>40.612</td>
<td>38.168</td>
</tr>
</tbody>
</table>

- NUMBER OF GRID POINTS 488
- MAXIMUM NODAL DEGREE 14
- NUMBER OF MPC EQUATIONS PROCESSED 12

<table>
<thead>
<tr>
<th>ELEMENTS PROCESSED</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SBEAR</td>
<td>168</td>
</tr>
<tr>
<td>Q45</td>
<td>40</td>
</tr>
<tr>
<td>DRAW</td>
<td>105</td>
</tr>
<tr>
<td>DRAW2</td>
<td>15</td>
</tr>
<tr>
<td>TMA</td>
<td>152</td>
</tr>
<tr>
<td>QUAD4</td>
<td>662</td>
</tr>
</tbody>
</table>

- TOTAL ELEMENTS 3769
**Table 22. Active Constraint Summary**

<table>
<thead>
<tr>
<th>COUNT</th>
<th>CONSTRAINT VALUE</th>
<th>CONSTRAINT TYPE</th>
<th>TYPE</th>
<th>COUNT</th>
<th>BOUNDARY ID</th>
<th>SUBCASE</th>
<th>ELEMENT TYPE</th>
<th>EID/LAYR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7.5598E+02</td>
<td>UPPER INCL. LTF.</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>2</td>
<td>4.13E+00</td>
<td>LIM. LENGTH</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>3</td>
<td>4.13E+00</td>
<td>MIN. MISE:STRESS</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>COMPL1</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>3.10E+00</td>
<td>MIN. MISE:STRESS</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>COMPL1</td>
<td>14</td>
</tr>
<tr>
<td>5</td>
<td>4.10E+00</td>
<td>MIN. MISE:STRESS</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>COMPL1</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>4.10E+00</td>
<td>MIN. MISE:STRESS</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>COMPL1</td>
<td>16</td>
</tr>
<tr>
<td>7</td>
<td>4.10E+00</td>
<td>MIN. MISE:STRESS</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>COMPL1</td>
<td>17</td>
</tr>
<tr>
<td>8</td>
<td>4.10E+00</td>
<td>MIN. MISE:STRESS</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>COMPL1</td>
<td>18</td>
</tr>
<tr>
<td>9</td>
<td>4.10E+00</td>
<td>MIN. MISE:STRESS</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>COMPL1</td>
<td>19</td>
</tr>
<tr>
<td>10</td>
<td>4.10E+00</td>
<td>MIN. MISE:STRESS</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>COMPL1</td>
<td>20</td>
</tr>
<tr>
<td>11</td>
<td>4.10E+00</td>
<td>MIN. MISE:STRESS</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>COMPL1</td>
<td>21</td>
</tr>
<tr>
<td>12</td>
<td>4.10E+00</td>
<td>MIN. MISE:STRESS</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>COMPL1</td>
<td>22</td>
</tr>
<tr>
<td>13</td>
<td>4.10E+00</td>
<td>MIN. MISE:STRESS</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>COMPL1</td>
<td>23</td>
</tr>
<tr>
<td>14</td>
<td>4.10E+00</td>
<td>MIN. MISE:STRESS</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>COMPL1</td>
<td>24</td>
</tr>
<tr>
<td>15</td>
<td>4.10E+00</td>
<td>MIN. MISE:STRESS</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>COMPL1</td>
<td>25</td>
</tr>
<tr>
<td>16</td>
<td>4.10E+00</td>
<td>MIN. MISE:STRESS</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>COMPL1</td>
<td>26</td>
</tr>
<tr>
<td>17</td>
<td>4.10E+00</td>
<td>MIN. MISE:STRESS</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>COMPL1</td>
<td>27</td>
</tr>
<tr>
<td>18</td>
<td>4.10E+00</td>
<td>MIN. MISE:STRESS</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>COMPL1</td>
<td>28</td>
</tr>
<tr>
<td>19</td>
<td>4.10E+00</td>
<td>MIN. MISE:STRESS</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>COMPL1</td>
<td>29</td>
</tr>
</tbody>
</table>

**Table 23. Approximate Optimization Summary**

<table>
<thead>
<tr>
<th>ITERATION</th>
<th>ASTROS APPROXIMATE OPTIMIZATION SUMMARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MATHEMATICAL PROGRAMMING</td>
</tr>
<tr>
<td></td>
<td>DESIGN VAR. MOVE LIMIT = 2.000000</td>
</tr>
<tr>
<td></td>
<td>PERCENT FUTURE MOVE = 1.000000 PERCENT</td>
</tr>
<tr>
<td></td>
<td>CURRENT VALUE = 4.3531E+01</td>
</tr>
<tr>
<td></td>
<td>ITER 2: OBJECTIVE CHANGE</td>
</tr>
<tr>
<td></td>
<td>PREVIOUS VALUE = 2.7840E+01</td>
</tr>
<tr>
<td></td>
<td>PERCENT MOVE = 56.3600</td>
</tr>
<tr>
<td></td>
<td>DESIGN VECTOR MOVE</td>
</tr>
<tr>
<td></td>
<td>NORM OF X = 1.3200E+06</td>
</tr>
<tr>
<td></td>
<td>PERCENT MOVE = 56.3600</td>
</tr>
<tr>
<td></td>
<td>DESIGN VECTOR MOVE</td>
</tr>
<tr>
<td></td>
<td>MAXIMUM MOV. = 3.6300E+01</td>
</tr>
<tr>
<td></td>
<td>AT DESIGN VARIABLES</td>
</tr>
<tr>
<td></td>
<td>CURRENT VALUE = 3.6300E+01</td>
</tr>
<tr>
<td></td>
<td>PERCENT MOVE = 100.0000</td>
</tr>
</tbody>
</table>

The approximate optimization problem was converged with feasible constraint criteria (CTLMIN) = 0.0000E-04 and active constraint criteria (CTLMAX) = 0.0000E-04.
### Table 24. Design Iteration History

| Iteration | Objective Function Value | Number of Function Evaluations | Number of Gradient Evaluations | Number of Retained Constraints | Number of Active Constraints | Number of Violated Constraints | Number of Infeasible Bound Violations | Approximate Problem
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.25994E+00</td>
<td>(Initial Function Value)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>7.12705E+03</td>
<td>31</td>
<td>4</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Not Converged</td>
</tr>
<tr>
<td>3</td>
<td>6.38273E+03</td>
<td>30</td>
<td>6</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Not Converged</td>
</tr>
<tr>
<td>4</td>
<td>6.08681E+03</td>
<td>39</td>
<td>8</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Not Converged</td>
</tr>
<tr>
<td>5</td>
<td>5.89348E+03</td>
<td>47</td>
<td>11</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Not Converged</td>
</tr>
<tr>
<td>6</td>
<td>5.74943E+03</td>
<td>56</td>
<td>13</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Not Converged</td>
</tr>
<tr>
<td>7</td>
<td>5.62564E+03</td>
<td>58</td>
<td>9</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Not Converged</td>
</tr>
<tr>
<td>8</td>
<td>5.50224E+03</td>
<td>60</td>
<td>10</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Not Converged</td>
</tr>
<tr>
<td>9</td>
<td>5.38496E+03</td>
<td>24</td>
<td>7</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Not Converged</td>
</tr>
<tr>
<td>10</td>
<td>5.27604E+03</td>
<td>36</td>
<td>8</td>
<td>10</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>Not Converged</td>
</tr>
<tr>
<td>11</td>
<td>5.18694E+03</td>
<td>43</td>
<td>11</td>
<td>10</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>Not Converged</td>
</tr>
<tr>
<td>12</td>
<td>5.14224E+03</td>
<td>33</td>
<td>5</td>
<td>10</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>Not Converged</td>
</tr>
<tr>
<td>13</td>
<td>5.13961E+03</td>
<td>10</td>
<td>2</td>
<td>10</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>Not Converged</td>
</tr>
<tr>
<td>14</td>
<td>5.13618E+03</td>
<td>10</td>
<td>1</td>
<td>10</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>Not Converged</td>
</tr>
<tr>
<td>15</td>
<td>5.11049E+03</td>
<td>18</td>
<td>3</td>
<td>10</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>Converged</td>
</tr>
</tbody>
</table>

The final objective function value is:

- Fixed = 0.00000E+00
- Designed = 5.11049E+03
- Total = 5.11049E+03

Elapsed clock time is shown upon leaving each phase of the MAPOL execution. This summary is useful in determining where a problem may have occurred and in confirming that the proper path was taken through the MAPOL sequence. It is, of course, also useful as an indication of the relative CPU costs of each phase of execution.

### 5.1.2. Error Message Output

Error messages can be printed from virtually all the modules of the ASTROS system as well as from the data base management software. Data base errors should not occur unless the user has modified or otherwise written a special MAPOL sequence, incorrectly assigned file names or used other incorrect or inconsistent data base information. Typically, data base errors cause immediate termination of the execution. The system administrator should be able to assist in solving these errors which, it is felt, will most likely be caused by incorrect use of the system or by incorrect system installation. The ASTROS Programmer's Manual contains further information on the causes of particular data base errors.

The standard ASTROS error messages are printed by the utmgerror utility module and represent error checks that the modules are coded to perform or errors that may cause problems in the current module's algorithm. As much as possible, these error messages are intended to be standalone in that the user should be able to interpret the message without referring to the Programmer's Manual. There are four different levels of errors that can occur in ASTROS, each labeled differently when printed:

1. **System Fatal Message**
   
   These messages come about due to errors or inconsistencies in the system definitions. Usually, these relate to erroneous input to the system generation utility, SYSGEN, or are a result of using an outdated system database. You should contact your system adminis-
Table 25. ASTROS Execution Summary

<table>
<thead>
<tr>
<th>Design Iteration</th>
<th>Boundary Condition</th>
<th><strong>NEW</strong></th>
<th><strong>REPLACE</strong></th>
<th><strong>REPLACE</strong></th>
<th><strong>REPLACE</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>BEGIN OPTIMIZATION</td>
<td>00:00:00</td>
<td>00:00:00</td>
<td>00:00:00</td>
<td>00:00:00</td>
</tr>
<tr>
<td>1</td>
<td>DESIGN ITERATION 1</td>
<td>00:00:00</td>
<td>00:00:00</td>
<td>00:00:00</td>
<td>00:00:00</td>
</tr>
<tr>
<td>2</td>
<td>BOUNDARY CONDITION 1</td>
<td>00:00:00</td>
<td>00:00:00</td>
<td>00:00:00</td>
<td>00:00:00</td>
</tr>
<tr>
<td>3</td>
<td>NEW TERMINATION</td>
<td>00:00:00</td>
<td>00:00:00</td>
<td>00:00:00</td>
<td>00:00:00</td>
</tr>
<tr>
<td>4</td>
<td>REPLACE TERMINATION</td>
<td>00:00:00</td>
<td>00:00:00</td>
<td>00:00:00</td>
<td>00:00:00</td>
</tr>
<tr>
<td>5</td>
<td>REPLACE TERMINATION</td>
<td>00:00:00</td>
<td>00:00:00</td>
<td>00:00:00</td>
<td>00:00:00</td>
</tr>
</tbody>
</table>

**NEW** means that the solution process must be repeated.
**REPLACE** means that the process is completed.

(2) User Information Message

These messages are written when the system encounters data that may represent an input error or may later generate a problem but that may only be a special user input that falls outside the expected range. Usually, these messages can be ignored. This is the least serious type of user message in ASTROS.

(3) User Warning Message

These messages are written when the system encounters data that are incorrect but which may not cause termination. In some cases, this level of error is issued to signify that the system will continue to search for errors but will terminate abnormally following the search.
(4) User Fatal Message

These messages are written when the system encounters data that are in error to the extent that continuation is impossible. The system will terminate execution either immediately or after some minor clean up.

If the user is unable to decipher the error message, the following steps can be helpful in determining the source of the error:

1. Check the timing summary with the LOGBEGIN and LOGMODULE options in the DEBUG packet against the MAPOL sequence path to determine which module generated the error message. Also, check the SYSGEN output to determine the module that wrote the message. Note that the "message number" is included in the error message print if the message is a standard one and the message number can be used to trace the module that uses the message.

2. Check the Programmer's Manual documentation for the relevant module to determine the error checks it performs and to get further information on the source of the error.

5.2. SOLUTION CONTROL OUTPUT OPTIONS

This subsection presents a detailed description of the output quantities that can be selected through the solution control packet. These quantities fall into five categories: (1) element; (2) nodal; (3) design; (4) eigenvalues for flutter and normal modes; and (5) aeroelastic trim quantities. Each of these categories is presented in the separate subsections that follow.

The PRINT and PUNCH solution control commands are used to request the desired output quantities. These commands have three groups of options: subset options, quantity options and form options. These options are fully described in Chapter 3 of this report, but one point must be stressed: subset options play an extremely important role in ASTROS output requests. Subset options allow the user to identify the set of iterations or subcases to which the print selection will apply. This selection is necessary because many disciplines (MODES, for example) generate more than one subcase (eigenvector) with a single solution control directive. The critical point is that the default selection for subcases is that there be no output. In other words, if there are no subcases selected, ASTROS will, by default, print nothing.

Unlike NASTRAN, ASTROS has no options to reorder the output. The multidisciplinary nature of ASTROS completely negates the utility of NASTRAN's SORT1 and SORT2 options, and any other sort options become impossibly complex very quickly. Instead, a reasonable, fixed sort was established in which each boundary condition is treated separately and in the order given in the solution control packet. If the standard sequence is used, the response quantities will appear in the following order within each optimize or analyze boundary condition:

1. Non-planar steady aerodynamics output.
2. Steady aerodynamic trim parameters.
3. Flutter roots and flutter mode shape modal participation factors -- note that the mode shape is only available if flutter has occurred and if the FLUTTER discipline is within an ANALYZE boundary condition.
4. Applied LOAD print requests.
5. The "displacement" nodal response quantities: DISPLACEMENTS, VELOCITIES, and ACCELERATIONS.
Element response quantities in the order STRESS, STRAIN, FORCE and STRAIN ENERG for each subcase, elements are processed alphabetically within each quantity type.

In the OPTIMIZE subpacket, these data are followed by the selected design and resizing prints in following order:

1. Active constraint summary (either the default abbreviated print or the full print if the DCONSTRAINT print option is selected).
2. The print of the global and then local design variables representing the current design peding on the GDESIGN and LDESIGN PRINT requests. On the final design iteration, the iteration history precedes the design variable output by default.

Within each response quantity's print module, the disciplines are not treated in the order given in solution control packet; instead, they are treated, where applicable, in the following order:

(A) NPSAERO
(B) STATICS
(C) MODES
(D) SAERO
(E) TRANSIENT
(F) FREQUENCY
(G) BLAST

The subcases within each discipline are treated in the order given in the solution control packet. In case of MODES, the eigenvectors are ordered in increasing eigenvalue order. TRANSIENT, FREQUENCY, BLAST subcases are ordered in increasing time or frequency step.

5.2.1. Element Response Quantities

ASTROS has two basic forms of elements: aerodynamic elements and structural elements. Aerodynamic element is defined as a "box" of an aerodynamic macroelement, e.g., wing component or fuselage segment. The nature of the macroelement varies among both aerodynamic models and aerodynamic components within each model. In general, however, a box is the smallest subdivision of an aerodynamic component for which data (e.g., pressures, forces, and moments) are computed. Structural elements are either metric elements, which connect structural node points (grids); scalar elements, which connect pairs of degrees of freedom or pairs of scalar points; or mass elements. Table 26 shows the aerodynamic and structural elements in ASTROS for which element output exist. The following solutions document the quantities that are available as output for each of these elements. The structure mass elements are not included in this table since they have no element response quantities. NASTRAN User's Manual (Reference 2) was used as a major resource in writing this section and the is referred to that for additional information on the structural elements.

Structural element output is available for all disciplines that result in a real displacement. This includes STATICS, MODES, TRANSIENT, BLAST, and SAERO analyses. Complex displacement result from FLUTTER and FREQUENCY analyses, but their formatted print is not available except through executive sequence utilities described in Subsection 3.4. For all disciplines in ASTROS, the solution control print options STRESS, STRAIN, FORCE, and ENERGY are used to select print of the structural element quantities. AIRDISP and TPRESSURE options are used for aerodynamic element quantities. Each of these
Table 26. Aerodynamic and Structural Elements in ASTROS

<table>
<thead>
<tr>
<th>AERODYNAMIC</th>
<th>STRUCTURAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAERO1</td>
<td>CAER02</td>
</tr>
<tr>
<td>CAERO6</td>
<td>CELASI, CELAS2</td>
</tr>
<tr>
<td>PAERO6</td>
<td>CIHEXI, CIHEX2, CIHEX3</td>
</tr>
<tr>
<td></td>
<td>CR0D, CONROD</td>
</tr>
<tr>
<td></td>
<td>CQDMEM1, CTRNME</td>
</tr>
<tr>
<td></td>
<td>CTRIA3, CQUAD4</td>
</tr>
</tbody>
</table>

options selects either ALL, NONE or an integer set identification number that refers to one or more ELEMnIST bulk data entries specifying which elements are to have output computed and printed. Chapter 3 contains the complete description of the solution control print command. Each output is carefully labeled as to its boundary condition number, discipline generating the response field and load condition, mode number, time step, frequency step or flight condition represented by the output.

5.2.1.1. Aerodynamic Element Output

The solution control PRINT option TPRESSURE provides the trimmed pressures on the aerodynamic boxes for SAERO and NPSAERO. The NPSAERO trimmed pressures are printed to the output file. For SAERO, the trimmed pressures are computed and stored in the relational entity OAGRDL0D. The AIRDISP print option is available for the SAERO discipline and provides the out-of-plane displacements and streamwise slopes of the aerodynamic boxes that correspond to the structural displacements. These data are computed and stored on the relational entity OAGRDDSP.

Aerodynamic geometry data are computed and stored by default to a set of relational entities that parallel the structural model. These data forms are designed primarily for model checkout of the SAERO and NPSAERO models using existing FE preprocessors that support NASTRAN style input data. These relations are: AEROGEOM which supplies the GRID-like data and CAREGEOm which provides connectivity data for the boxes in a ROD or QUAD4 form. The ROD is used to model the outline of the airfoils and QUAD4's are used to model the boxes.

For unsteady aerodynamics, the box-on-box aerodynamic forces are only available through the DEBUG/UNSTADY and DEBUG/AMP options (see Chapter 1). The geometry data are not available.

5.2.1.2. Bar Element Output

The BAR element includes extension, torsion, bending in two perpendicular planes and the associated shears. The shear center is assumed to coincide with the elastic axis. The BAR element coordinate system is shown in Figure 5. The orientation of the BAR element is described in terms of two reference planes defined through the use of the orientation vector(v) as shown in that figure. The positive directions for the element forces are shown in Figure 6. Additional information on the structural element is contained in Section 5 of the ASTROS Theoretical Manual.

364
Figure 5. BAR Element Coordinate System

Stresses, strains, forces and strain energies are available as output for the BAR element through the STRESS, STRAIN, FORCE, and ENERGY solution control print options. The following element forces are output on request:

1. Bending moments at each end in both reference planes.
2. Shear forces in each reference plane.
3. Average axial force.
4. Torque about the bar axis.

The following element stresses and strains in the element coordinate system are output on request:

1. Average axial stress or strain.
2. Extensional stress or strain due to bending at 4 points on the cross-section at each end.
3. Maximum and minimum stress or strain at each end.
4. Stress margins of safety for the element in both tension and compression.

Tensile stresses and strains are given a positive value while compressive stresses and strains are given a negative value. The bending contribution to the stresses are always computed at the four points on the element cross-section that were specified on the connectivity entry for the BAR element. This means that the safety margins are computed using all eight stress values even if all 4 stress points at each end are the same and/or coincide with the element axis. Also, margins of safety are printed even if no stress limits were given on the material entry. In these cases, a very large value for the margin of safety is used to indicate that no limits were specified. In addition, ASTROS fully supports strain output for the BAR element. Strain energies may also be requested for the BAR element. The strain energy print (which is identical for all ASTROS structural elements) is patterned after that in NASTRAN. It shows the total strain energy for the given displacement field, the strain energy in each selected element and the total strain energy for all the elements of a given type, e.g., all the BAR elements. Examples of these outputs are shown in Table 27.
Figure 6. BAR Element Forces Sign Conventions

Table 27. BAR Element Output Quantities

<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>S1A</th>
<th>S2A</th>
<th>S3A</th>
<th>S4A</th>
<th>AXIAL</th>
<th>S MAX</th>
<th>S MIN</th>
<th>M.P.C.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID.</td>
<td>SBI</td>
<td>SB2</td>
<td>SB3</td>
<td>SB4</td>
<td>STRESS</td>
<td>S MAX</td>
<td>S MIN</td>
<td>M.P.C.</td>
</tr>
<tr>
<td>106</td>
<td>-2.816E+03</td>
<td>0.0000E+00</td>
<td>-2.816E+03</td>
<td>2.816E+03</td>
<td>0.6000E+00</td>
<td>2.816E+03</td>
<td>7.408E-02</td>
<td>1.704E+03</td>
</tr>
<tr>
<td></td>
<td>7.498E-02</td>
<td>0.0000E+00</td>
<td>7.498E-02</td>
<td>7.498E-02</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>7.498E-02</td>
<td>1.704E+03</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>S1A</th>
<th>S2A</th>
<th>S3A</th>
<th>S4A</th>
<th>AXIAL</th>
<th>S MAX</th>
<th>S MIN</th>
<th>M.P.C.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID.</td>
<td>SBI</td>
<td>SB2</td>
<td>SB3</td>
<td>SB4</td>
<td>STRAIN</td>
<td>S MAX</td>
<td>S MIN</td>
<td>M.P.C.</td>
</tr>
<tr>
<td>106</td>
<td>-1.960E-04</td>
<td>0.0000E+00</td>
<td>-1.960E-04</td>
<td>1.960E-04</td>
<td>0.0000E+00</td>
<td>1.960E-04</td>
<td>5.859E-10</td>
<td>1.463E-04</td>
</tr>
<tr>
<td></td>
<td>5.859E-10</td>
<td>0.0000E+00</td>
<td>5.859E-10</td>
<td>5.859E-10</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>5.859E-10</td>
<td>1.463E-04</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>BEND MOMENT END-A</th>
<th>BEND MOMENT END-B</th>
<th>SHEAR</th>
<th>AXIAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID.</td>
<td>PLANE 1</td>
<td>PLANE 2</td>
<td>PLANE 1</td>
<td>PLANE 2</td>
</tr>
<tr>
<td>106</td>
<td>0.0000E+00</td>
<td>3.1980E+00</td>
<td>0.0000E+00</td>
<td>3.1980E+00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ELEMENT STRAIN ENERGIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAR ELEMENTS</td>
</tr>
<tr>
<td>ELEMENT ID</td>
</tr>
<tr>
<td>103</td>
</tr>
<tr>
<td>162</td>
</tr>
<tr>
<td>103</td>
</tr>
<tr>
<td>104</td>
</tr>
<tr>
<td>105</td>
</tr>
<tr>
<td>106</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BAR ELEMENTS</th>
<th>SUBTOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOTAL</td>
<td>4.186E+01</td>
</tr>
</tbody>
</table>
5.2.1.3. **ELAS Element Output**

The ELAS element is a scalar spring element which relates the displacements at a pair of scale points or degrees of freedom or that relates a single degree of freedom to a ground state. The element force and strain energy are directly available for the element and the user can, if desired, input a scalar quantity that relates the "stress" in the element to the displacement(s) of the connected degree(s) of freedom. On output, these values will be printed for each output request for each selected ELAS element. Strains have no meaning for the scalar spring element and any such requests will be ignored without warning. Element strain energies, however, are available for the element and are computed from the spring constant and the nodal displacement(s). The strain energy print for the ELAS is identical to that for the BAR element and includes a breakdown by element and by element type. If no scalar value is given for the element stress but the stress value is requested, a value of zero will be computed and printed for the response quantity with no warnings given.

5.2.1.4. **IHEX1 Element Output**

The IHEX1 element is a linear isoparametric solid hexahedron element with three extensional degrees of freedom for each of its eight nodes.

Stresses, strains, and strain energies are available as output for the IHEX1 element through the `STRESS`, `STRAIN`, and `ENERGY` solution control print command options. Force output is not available for the IHEX1 element. On request, the following stresses and strains are output in the basic coordinate system at the center and at each corner grid point:

1. Normal stresses or strains in all three directions.
2. Shear stresses or strains in all three planes.
3. Principal stresses or strains in all three directions with associated direction cosines.
4. Mean stress or strain.
5. Octahedral shear stress or strain.

![IHEX1 Element Geometry](image_url)

**Figure 7. IHEX1 Element Geometry**
The stress and strain output at each of the nine points is identified by a stress or strain point ID. The stress and strain point IDs are numbered 1 through 9, with the first eight ordered as on the associated CIHEX1 input data entry, and the ninth located at the element center, as illustrated in Figure 7. All output is provided in the basic coordinate system, since there is no naturally occurring element coordinate system for the IHEX1. An example of the output for the IHEX family of elements is shown in Table 28. The HEX1 element is shown with the HEX2 and HEX3 elements differing only in the number of data recovery points.

Strain energy output may be requested for the IHEX1 element. The strain energy print for the IHEX1 is identical to that for the BAR element and includes a breakdown by element and by element type.

### Table 28. IHEX1 Element Solution Quantities

<table>
<thead>
<tr>
<th>ID</th>
<th>POINT</th>
<th>STRESS</th>
<th>STRESS</th>
<th>ELEMENT</th>
<th>STRAIN</th>
<th>STRAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>1</td>
<td>X 1.617462E-04 Y 0.982344E-04 Z 0.101964E-04</td>
<td>A 0.4589 E-04</td>
<td>1.1085</td>
<td>E-04</td>
<td>1.1085</td>
</tr>
<tr>
<td>123</td>
<td>2</td>
<td>X 7.238568E-04 Y 0.090240E-03 Z 0.000000E+00</td>
<td>A 0.9554 E-04</td>
<td>0.9554</td>
<td>0.9554</td>
<td>0.9554</td>
</tr>
<tr>
<td>123</td>
<td>3</td>
<td>X 0.000000E+00 Y 0.000000E+00 Z 0.000000E+00</td>
<td>A 0.0000 E-04</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>
5.2.1.5. IHEX2 Element Output

The IHEX2 element is a quadratic isoparametric solid hexahedron element with three extensional degrees of freedom for each of its twenty nodes.

Stresses, strains, and strain energies are available as output for the IHEX2 element through the STRESS, STRAIN, and ENERGY solution control print command options. Force output is not available for the IHEX2 element. On request, the following stresses and strains are output in the basic coordinate system at the twenty-one points located at the center, corners, and mid-edges of the element:

1) Normal stresses or strains in all three directions.
2) Shear stresses or strains in all three planes.
3) Principal stresses or strains in all three directions with associated direction cosines.
4) Mean stress or strain.
5) Octahedral shear stress or strain.

The stress and strain output at each of the twenty-one points is identified by a stress or strain point ID. The stress and strain point IDs are numbered 1 through 21, with the first twenty ordered as on the associated IHEX2 input data entry, and the twenty-first located at the element center. Although the corner stress and strain points are located at the corner grid points of the element, the mid-edge stress and strain points may or may not be located at the mid-edge grid points, depending on the location of those grid points. The stress/strain points for the IHEX2 are illustrated in Figure 8. All output occurring element coordinate system for the IHEX2.

Strain energy output may be requested for the IHEX2 element. The strain energy print for the IHEX2 is identical to that for the BAR element and includes a breakdown by element and by element type.

(a) ELEMENT GRID POINTS
(b) ELEMENT STRESS POINTS

Figure 8. IHEX2 Element Geometry
5.2.1.6. IHEX3 Element Output

The IHEX3 element is a cubic isoparametric solid hexahedron element with three extensional degrees of freedom for each of its thirty-two nodes.

Stresses, strains, and strain energies are available as output for the IHEX3 element through the STRESS, STRAIN, and ENERGY solution control print command options. Force output is not available for the IHEX3 element. On request, the following stresses and strains are output in the basic coordinate system at the twenty-one points located at the center, corners, and mid-edges of the element:

1. Normal stresses or strains in all three directions.
2. Shear stresses or strains in all three planes.
3. Principal stresses or strains in all three directions with associated direction cosines.
4. Mean stress or strain.
5. Octahedral shear stress or strain.

The stress and strain output at each of the twenty-one points is identified by a stress or strain point ID. The stress and strain point IDs are numbered 1 through 21. The first twenty points are ordered as on the associated CIX3 input data entry, except that there is only one mid-edge point per edge, instead of two, and the twenty-first point is located at the element center. Although the corner stress and strain points are located at the corner grid points of the element, the mid-edge stress and strain points may or may not be located at a grid point, depending on the location of the mid-edge grid points. The stress/strain points for the IHEX3 are illustrated in Figure 9. All output is provided in the basic coordinate system, since there is no naturally occurring element coordinate system for the IHEX3.

Strain energy output may be requested for the IHEX3 element. The strain energy print for the IHEX3 is identical to that for the BAR element and includes a breakdown by element and by element type.

Figure 9. IHEX3 Element
5.2.1.7. Rod Element Output

The ASTROS ROD element supports both extensional and rotational properties. The element coordinate system and sign conventions are shown in Figure 10. ASTROS supports stress, strain, force and strain energy output for the ROD. The forces that are computed are:

1. Axial force.
2. Torque about the element axis.

The torque and force are both computed even if the particular element does not support torsional or extensional forces, respectively. In these cases, a value of zero will be printed for the appropriate response quantity. The stresses and/or strains that are available are:

1. Axial stress or strain.
2. Torsional stress or strain.

The margins of safety for strain are not available and the stress margins are computed even if there are no limits specified on the material property entry. In these cases, a large safety margin value is used to signify that no limits were imposed. An example of the ROD element output prints is shown in Table 29. The strain energy print for the ROD is identical to that for the BAR element and includes a breakdown by element and by element type.

5.2.1.8. QDMEM1/TRMEM Element Output

The QDMEM1 isoparametric element and the TRMEM constant strain triangular element are membrane elements which support isotropic, orthotropic and composite membrane properties. If the element is composite, the individual layers are treated as independent, stacked elements in which each "layer," as defined on the PCOMP bulk data entry, represents an element. In the case of composite elements, the layers are numbered sequentially starting with the first layer appearing on the PCOMP entry. Non-composite elements will show a layer number of zero.
Table 29. ROD Element Solution Quantities

<table>
<thead>
<tr>
<th>Id.</th>
<th>Element</th>
<th>Axial Stress (MPa)</th>
<th>Axial Strain</th>
<th>Torsional Strain</th>
<th>Axial Force (N)</th>
<th>Axial Torque (N.m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.512166E+03</td>
<td>2.8E+00</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>1.953650E+05</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>6.821167E+03</td>
<td>2.7E+00</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>2.046350E+05</td>
<td>0.000000E+00</td>
</tr>
</tbody>
</table>

Stresses, strains, forces and strain energies are available for each element or layer of a composite element. Since the stresses, strains, and forces vary within a QDMEM1 element, the intersection point of the diagonals projected onto the mean plane of a warped element has been chosen as the point at which the stresses, strains, forces and strain energies for the element are computed. The stresses, strains and element forces are computed in the element coordinate system. The element coordinate system and the stress computation point for the QDMEM1 element are shown in Figure 11 and those for the TRMEM in Figure 12.
ASTROS computes the running loads associated with the stresses for the QDMEM1 element. These forces are:

1. The force components in the element coordinate system at the stress computation point.

The QDMEM1 stress and strain print includes the following:

2. The normal stresses or strains at the stress point in the element x- and y-directions.
3. The shear stress or strain on the element x face in the element y-direction.

Table 30. QDMEM1 Solution Quantities
(4) The angle in degrees between the element x-axis and the major principal axis.
(5) The major and minor principal (zero shear) stresses or strains.
(6) The maximum shear stress or strain.

An example of the printed output for the QDMEM1 is shown in Table 30. The output for the TRMEM is identical except for the titling.

The strain energy print for the QDMEM1 is identical to that for the BAR element and includes a breakdown by element and by element type.

5.2.1.9. QUAD4/TRIA3 Element Output

The QUAD4 and TRIA4 isoparametric quadrilateral and triangular plate elements include both membrane and bending behavior. Transverse shear flexibility may be requested, as can the coupling of membrane and bending behavior. The QUAD4 element coordinate system and node numbering are shown in Figure 13. The TRIA3 element coordinate system and node numbering are shown in Figure 14. These elements may be assigned general anisotropic or composite material properties. For designed composites, the layers are treated as stacked membrane elements similar to the QDMEM1 element. In this case, the layers are identified by number in the order specified on the \texttt{PCOMP}, \texttt{PCOMP1} or \texttt{PCOMP2} entry. For design invariant composite laminates, the output always refers to the aggregate laminate properties and refers to layer number zero. The reference plane of the QUAD4/TRIA3 elements may be offset from the plane of the grid points and variation in the element thickness may be modeled by assigning different element thicknesses at each of the grid points. The reader is referred to Appendix A of the ASTROS Theoretical Manual for additional information on these plate bending elements.

![Figure 13. QUAD4 Element Coordinate System](image1)

![Figure 14. TRIA3 Element Coordinate System](image2)
Table 31. QUAD4 and TRIA3 Solution Quantities

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5.2.1.10. Shear Panel Output

The shear panel is an element which resists the action of tangential forces applied to its edges. In ASTROS, the shear panel supports only isotropic material properties and makes use of the shear flow distribution approximation of Garvey (Reference 4) with special handling for warped, parallel edge and general trapezoidal geometries. The element force sign convention is shown in Figure 15. The stresses, strains, forces and strain energies are available for the shear panel. The element forces that are computed include the following:

1. The eight forces between each pair of nodes; each force is directed along the line connecting the adjacent nodes (the element edge).
2. The four "kick" forces at each node, normal to the plane formed by the two adjacent element edges.
3. The shear flows (forces/unit length) along each edge.

When stresses or strains are requested, they are computed at the 

When stresses or strains are requested, they are computed at the points in skewed coordinates parallel to the adjacent edges. Both the average and maximum shear stress or strain are then printed. A safety margin based on the maximum stress value is computed for stress output. A large safety margin is printed if not limits were specified on the material property entry. Table 32 contains a sample of the SHEAR panel element output.

The strain energy print for the SHEAR panel is identical to that for the BAR element and includes a breakdown by element and by element type.

5.2.2. Nodal Response Quantities

ASTROS has two basic forms of node point: the structural node and the extra point. The structural node is defined as either a "grid" point having six degrees of freedom (three translations and three rotations) or a "scalar" point having a single degree of freedom. These node points can be used to connect metric and scalar structural elements. The extra point is similar to the scalar point in that it has a single degree of freedom, but differs in that extra points included in the model are selected in the boundary condition rather than being implicitly included in the model. Further, they cannot be connected directly to either metric or scalar structural elements; instead, these elements are connected through terms introduced by direct matrix input or by transfer functions. Extra points are used in dynamic analyses for modeling control systems and other nonstructural mechanisms in the system under analysis. These degrees of freedom do not appear in the system matrices until after the dynamic matrix assembly and do not appear in any but the dynamic response disciplines (FLUTTER, TRANSIENT, FREQUENCY and BLAST). When nodal output is requested for dynamic analyses, any extra point results may be selected using the GRIDLIST entry just as are grid and scalar point results.

Nodal output is available for all disciplines in ASTROS, although particular nodal response quantities may not be available for all disciplines. The solution control print options VELOCITY, DISPLACEMENT, GIPFORCE, LOAD, SPCCFORCE, and ACCELERATION are used to select print of the nodal response quantities. Each of these print options selects either ALL, NONE or an integer set identification number that refers to one or more GRIDLIST bulk data entries. Chapter 3 contains the complete description of the solution control print command. Each output is carefully labeled as to its boundary condition number, which discipline generated the output quantities and which load condition, mode shape, time step, frequency step or flight condition is represented by the output.