SAP2000®

Integrated
Finite Element Analysis
and
Design of Structures

INPUT FILE FORMAT



Computers and Structures, Inc.
Berkeley, California, USA

Version 7.0
Revised October 1998

COPYRIGHT

The computer program SAP2000 and all associated documentation are proprietary and copyrighted products. Worldwide rights of ownership rest with Computers and Structures, Inc. Unlicensed use of the program or reproduction of the documentation in any form, without prior written authorization from Computers and Structures, Inc., is explicitly prohibited.

Further information and copies of this documentation may be obtained from:

Computers and Structures, Inc.
1995 University Avenue
Berkeley, California 94704 USA

tel: (510) 845-2177
fax: (510) 845-4096
e-mail: *info@csiberkeley.com*
web: *www.csiberkeley.com*

© Copyright Computers and Structures, Inc., 1978–1998.
The CSI Logo is a registered trademark of Computers and Structures, Inc.
SAP2000 is a registered trademark of Computers and Structures, Inc.
Windows is a registered trademark of Microsoft Corporation

DISCLAIMER

CONSIDERABLE TIME, EFFORT AND EXPENSE HAVE GONE INTO THE DEVELOPMENT AND DOCUMENTATION OF SAP2000. THE PROGRAM HAS BEEN THOROUGHLY TESTED AND USED. IN USING THE PROGRAM, HOWEVER, THE USER ACCEPTS AND UNDERSTANDS THAT NO WARRANTY IS EXPRESSED OR IMPLIED BY THE DEVELOPERS OR THE DISTRIBUTORS ON THE ACCURACY OR THE RELIABILITY OF THE PROGRAM.

THE USER MUST EXPLICITLY UNDERSTAND THE ASSUMPTIONS OF THE PROGRAM AND MUST INDEPENDENTLY VERIFY THE RESULTS.

ACKNOWLEDGMENT

Thanks are due to all of the numerous structural engineers, who over the years have given valuable feedback that has contributed toward the enhancement of this product to its current state.

Special recognition is due Dr. Edward L. Wilson, Professor Emeritus, University of California at Berkeley, who was responsible for the conception and development of the original SAP series of programs and whose continued originality has produced many unique concepts that have been implemented in this version.

Table of Contents

Chapter I Introduction 1

About This Manual 1
Typographical Conventions 2
Bold for Definitions 2
Bold for Variable Data 2
Italics for Mathematical Variables 2
Italics for Emphasis 2
All Capitals for Literal Data 3
Capitalized Names 3

Chapter II The Input Data File 5

Overview 6
Input Data Files and the Graphical User Interface 6
Importing SAP90 Input Data Files 7
Units 8
Upward Direction 8
Characters 8
Data Blocks and Separators 9
Data Lines 11
Continuations, Comments, and Blank Lines 13
Arithmetic Operations 14
Regular Array Specification 15
Frequently Used Keywords 17
NAME Keyword 17
GEN Keyword 17
DEL Keyword 18

ADD Keyword 18
REM Keyword 19
ELEM Keyword 19
CSYS Keyword 19
UX, UY, UZ, RX, RY, and RZ Keywords 20
U1, U2, U3, R1, R2, and R3 Keywords 20
How to Prepare the Input Data File 21
Data Block Format 22
Data Line Formats 23
Description of Variables 24
Default Values 25
Units 25
The Title Line 27
Data Block Format 27
Data Line Format 27
Description of Variables 27
SYSTEM Data Block 28
COORDINATE Data Block 32
JOINT Data Block 37
LOCAL Data Block 44
RESTRAINT Data Block 48
CONSTRAINT Data Block 51
WELD Data Block. 58
PATTERN Data Block 62
SPRING Data Block 68
MASS Data Block 73
MATERIAL Data Block. 77
FRAME SECTION Data Block 81
SHELL SECTION Data Block 87
NLPROP Data Block 90
FRAME Data Block 96
SHELL Data Block. 101
PLANE Data Block 106
ASOLID Data Block 111
SOLID Data Block 116
NLLINK Data Block 120
MATTEMP Data Block 125
REFTEMP Data Block 128
PRESTRESS Data Block. 131
LOAD Data Block 134
PDFORCE Data Block 150

PDELTA Data Block 154
MODES Data Block 157
FUNCTION Data Block 161
SPEC Data Block. 165
HISTORY Data Block 169
LANE Data Block 176
VEHICLE Data Block 179
VEHICLE CLASS Data Block. 184
BRIDGE RESPONSE Data Block 186
MOVING LOAD Data Block 189
COMBO Data Block 194
OUTPUT Data Block 199
END Data Block 203

Chapter I

Introduction

This manual describes the use and the format of the input data text file. Most users can skip this manual.

Basic Topics for All Users

- About This Manual
- Typographical Conventions

About This Manual

This manual describes the format of the input data text file for the SAP2000 structural analysis program. The graphical user interface, analysis concepts, and the design modules are described in separate manuals. See the *SAP2000 Getting Started* manual for a description of all the manuals supplied with the program.

This manual will be of interest to users with specialized analysis needs that cannot yet be directly defined in the SAP2000 graphical user interface. All variables described in this manual are cross-referenced to the *SAP2000 Analysis Reference*.

Typographical Conventions

Throughout this manual the following typographic conventions are used.

Bold for Definitions

Bold roman type (e.g., **example**) is used whenever a new term or concept is defined. For example:

> The **global coordinate system** is a three-dimensional, right-handed, rectangular coordinate system.

This sentence begins the definition of the global coordinate system.

Bold for Variable Data

Bold roman type (e.g., **example**) is used to represent variable data items for which you must specify values when defining a structural model and its analysis. For example:

> The Frame element coordinate angle, **ang**, is used to define element orientations that are different from the default orientation.

Thus you will need to supply a numeric value for the variable **ang** if it is different from its default value of zero.

Italics for Mathematical Variables

Normal italic type (e.g., *example*) is used for scalar mathematical variables, and bold italic type (e.g., ***example***) is used for vectors and matrices. If a variable data item is used in an equation, bold roman type is used as discussed above. For example:

$$0 \le \mathbf{da} < \mathbf{db} \le L$$

Here **da** and **db** are variables that you specify, and L is a length calculated by the program.

Italics for Emphasis

Normal italic type (e.g., *example*) is used to emphasize an important point, or for the title of a book, manual, or journal.

All Capitals for Literal Data

All capital type (e.g., EXAMPLE) is used to represent data that you type at the keyboard exactly as it is shown, except that you may actually type lower-case if you prefer. For example:

SAP2000

indicates that you type “SAP2000” or “sap2000” at the keyboard.

Capitalized Names

Capitalized names (e.g., Example) are used for certain parts of the model and its analysis which have special meaning to SAP2000. Some examples:

Frame element

Diaphragm Constraint

Frame Section

Load Case

Common entities, such as “joint” or “element” are not capitalized.

Chapter II

The Input Data File

The input data file is a text file that you can prepare containing all the information required by SAP2000 to define the structural model and its analysis.

You do not need to read this chapter if you are using the SAP2000 graphical user interface to define your problem.

Basic Topics for All Users

- Overview

Advanced Topics

- Input Data Files and the Graphical User Interface
- Importing SAP90 Input Data Files
- Characters
- Data Blocks and Separators
- Data Lines
- Continuations, Comments, and Blank Lines
- Arithmetic Operations
- Regular Array Specification

- Frequently Used Keywords
- How to Prepare the Input Data File
- The Title Line
- SYSTEM Data Block ... *through* ... END Data Block

Overview

The input data file is a text file that contains all the information required by SAP2000 to define the structural model. Such information includes the geometry, properties, loading, and analysis parameters for the structure to be analyzed. It is an alternative to the model data base file created by the SAP2000 graphical user interface. The input data file does not, however, contain certain information used by the graphical user interface, such as the grids, groups, or design parameters.

The input data file can serve the following purposes:

- It can be edited to add advanced analysis options that are not currently available through the SAP2000 graphical user interface
- It is a readable text form of the analysis data

This chapter describes in detail how to prepare an input data file. Sample input data files are provided in subdirectory EXAMPLES and are discussed in the *SAP2000 Verification Manual*.

Most users will have no need of the input data file and can skip the rest of this chapter.

Input Data Files and the Graphical User Interface

You may use the SAP2000 graphical user interface to prepare input data files, and then use a text editor to modify the file. For example, you could define most of the geometry graphically, then add advanced features with the editor.

The complete procedure is as follows:

1. Create or modify the model using the SAP2000 graphical user interface
2. Write the SAP2000 input data file by selecting **Export** from the **File** menu
3. Make the desired changes to the input data file using a text editor

4. Read the modified input data file into the graphical user interface by selecting **Import** from the **File** menu
5. Perform the analysis
6. Review the results of the analysis
7. Check the design of the structure, if desired

This is usually an iterative process that may involve many cycles of the above sequence of steps.

All data present in the input data file can be imported into the graphical user interface, even data that cannot be created or changed within the interface itself. The only exception is comment data, which is discarded. All imported data can be:

- Saved in the model file (extension .SDB)
- Used by the analysis
- Exported to an input data file (.extension .S2K)

WARNING! The order and format of an input data file are not preserved when importing. All comments, generations, and deletions are lost! Only the model and analysis data *as interpreted during import* are saved. If you subsequently export to an input data file of the same name, your original file will be overwritten. Export to a new file if you want to preserve the original format of your input data file!

Importing SAP90 Input Data Files

Most modeling and analysis features available in SAP90 are also present in SAP2000, and many new features have been added. Only the SAP90 heat-transfer analysis features are not currently available in SAP2000.

SAP90 input data files (versions 5.4 and 5.5) can be imported directly into the SAP2000 graphical user interface and automatically converted to SAP2000 models. An imported model can then be used directly in the graphical user interface, or exported as a SAP2000 input data file for use as described in this chapter.

WARNING! Some imported data may be interpreted differently by SAP2000 than by SAP90. For example, the interaction between end offsets and end releases is different between the two programs, as is the interaction between prestress load and P-Delta analysis.

Be sure to check your imported model carefully! Compare the results of analyses using both SAP90 and SAP2000 before making further use of the imported SAP90 model!

Units

When you import a SAP90 input data file, you will be asked to specify what force and length units were used in the SAP90 file. These units then become the base units for the SAP2000 model. You may convert the model to other units after importing.

Upward Direction

When you import a SAP90 input data file, you will be asked to specify what direction was assumed to be upward in the SAP90 file. All coordinate-dependent quantities in the SAP90 model will be converted to conform with the SAP2000 convention that the +Z direction is upward.

The X coordinates will not be changed unless ±X is upward in the SAP90 model, in which case the Y coordinates will be left unchanged. The following table shows how the coordinates are changed for all six possible upward directions in SAP90:

SAP90 Upward Direction	SAP90 Direction for SAP2000 +X	SAP90 Direction for SAP2000 +Y	SAP90 Direction for SAP2000 +Z
+Z	+X	+Y	+Z
–Z	+X	–Y	–Z
+Y	+X	–Z	+Y
–Y	+X	+Z	–Y
+X	–Z	+Y	+X
–X	+Z	+Y	–X

Characters

The input data file must be a plain text file. The only characters permitted in the data file are the standard printable keyboard characters, including the space, and the Tab character, which is interpreted as a space.

Uppercase and lowercase letters are treated the same throughout the input data file.

If you use a word-processor to prepare the file, be sure to save the file in ASCII text format. Otherwise, the word-processor may insert special formatting characters in the file that cannot be interpreted by SAP2000.

Each line of text in the input data file may be up to 500 characters long.

Data Blocks and Separators

The first data line of the input data file will be used as a **Title Line** that is printed at the top of every page of the output files. Any separators or data placed on this first line will be ignored and will not contribute to the structural model.

All input data following the title line is organized into distinct **data blocks** by means of corresponding unique separator lines. The **separator line** identifies the data block and is always the first line in the data block. Each separator contains a prescribed title of one or two words that must be typed exactly as specified; uppercase and lowercase are treated the same. The separator may be singular or plural, e.g., FRAME is the same as FRAMES, and MASS is the same as MASSES. No other data may be placed on a separator line except comment data. Data associated with the data block immediately follows the separator line.

The input data blocks and their functions are summarized below. Only the JOINT data block is mandatory. The need for the other data blocks in the input data file depends on the problem being analyzed. For example, if the structure has no spring supports, you can skip the SPRING Data Block completely (including the separator line). Similarly, if the model consists only of Frame elements, you will not provide any data associated with the SHELL, PLANE or other element data blocks.

The order in which the data blocks occur in the input file is immaterial. Data lines within a data block are always processed by the program in the order in which they appear in the input data file. The Title Line must be the first line in the input file.

General Data Blocks

Data Block	Description
SYSTEM	Overall job control information
COORDINATE	Alternate Coordinate System definitions
END	End of SAP2000 input data

Joint Data Blocks

Data Block	Description
JOINT	Joint (node) coordinate definitions
LOCAL	Joint local coordinate system assignments
RESTRAINT	Joint restraint assignments
WELD	Weld definitions
CONSTRAINT	Constraint definitions
PATTERN	Joint Pattern definitions
SPRING	Joint spring assignments
MASS	Joint mass assignments

Element Data Blocks

Data Block	Description
MATERIAL	Material property definitions
FRAME SECTION	Section property definitions for Frame elements
SHELL SECTION	Section property definitions for Shell elements
NLPROP	Nonlinear property definitions for Nllink elements
FRAME	Frame element definitions
SHELL	Shell element definitions
PLANE	Plane-stress and plane-strain element definitions
ASOLID	Axisymmetric-solid element definitions
SOLID	Solid element definitions
NLLINK	Nonlinear link and spring element definitions
MATTEMP	Element material temperature assignments
REFTEMP	Element reference temperature assignments
PRESTRESS	Prestress cable assignments for Frame elements
PDFORCE	P-Delta force assignments for Frame elements

Load and Analysis Data Blocks

Data Block	Description
LOAD	Static Load Case definitions
PDELTA	P-delta analysis control
MODES	Modal analysis control
FUNCTION	Time and period Function definitions
SPEC	Response-spectrum analysis definitions
HISTORY	Time-history analysis definitions
LANE	Bridge Lane definitions

Load and Analysis Data Blocks (continued)

Data Block	Description
VEHICLE	Bridge Vehicle definitions
VEHICLE CLASS	Bridge Vehicle Class definitions
BRIDGE RESPONSE	Bridge response assignments for Frame elements
MOVING LOAD	Bridge Moving Load analysis definitions
COMBO	Analysis combination definitions
OUTPUT	Analysis output selection

The contents of a simple input data file is shown in Figure 1 (page 12).

Data Lines

All data in the data blocks is divided into **data lines**. Normally each data line corresponds to a line of text in the input data file. However, you may continue a single data line onto several lines of text as described in the next topic.

Data lines within a data block are always processed by the program in the order in which they appear in the input data file.

All SAP2000 input data is prepared in free format. In other words, data on a particular data line does not have to correspond with specific column locations. Each data line consists of one or more lists of data items separated by a comma and/or one or more spaces. The data items may be numbers or alpha-numeric strings. All alphabetic characters that appear in the input data may be uppercase or lowercase.

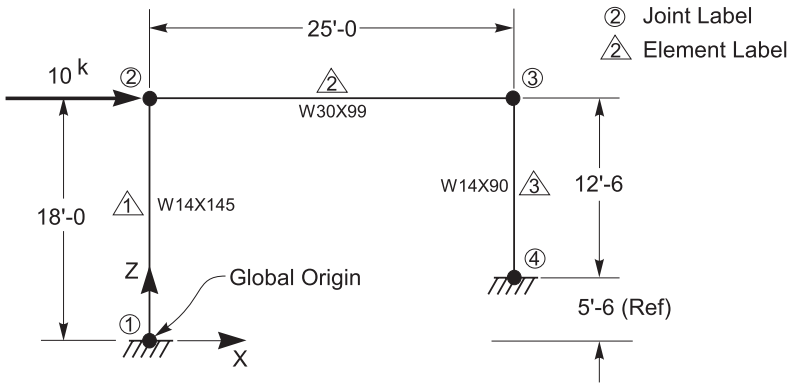
The lists of data items are of two types:

- Keyed data lists
- Unkeyed data lists

A keyed data list is a list of data items preceded by a specified keyword and an equal sign, such as:

```
X=0,10
```

Here the keyword is X. *No spaces may separate the keyword from the equal sign.* Spaces are permitted after the equal sign.



SAP2000 Structural Model

```
SIMPLE PORTAL FRAME EXAMPLE

SYSTEM                                  ; SYSTEM PARAMETERS
 DOF=UX,UZ,RY  LENGTH=IN  FORCE=KIP     ; PLANE FRAME IN X-Z PLANE

JOINT                                   ; DEFINE JOINT COORDINATES
 1  X=    0  Z=    0
 2  X=    0  Z=18*12
 3  X=25*12  Z=18*12
 4  X=25*12  Z= 5*12+6

RESTRAINT                               ; FIX BASE JOINTS
 ADD=1,4,3  DOF=ALL

MATERIAL                                ; DEFINE MATERIAL PROPERTIES
 NAME=STEEL
  E=29500

FRAME SECTION                           ; DEFINE SECTION PROPERTIES
 NAME=1  SH=W14X145  MAT=STEEL
 NAME=2  SH=W30X99   MAT=STEEL
 NAME=3  SH=W14X90   MAT=STEEL

FRAME                                   ; DEFINE FRAME ELEMENTS
 1  J=1,2  SEC=1
 2  J=2,3  SEC=2
 3  J=3,4  SEC=3

LOAD                                    ; APPLY FORCE AT JOINT 2
 NAME=LATERAL
  TYPE=FORCE  ELEM=JOINT
   ADD=2  UX=10
```

SAP2000 Input Data File

Figure 1
Typical SAP2000 Structural Model and Corresponding Input Data File

12 Data Lines

An unkeyed data list is just a list of data items without a preceding keyword, such as:

```
1,5,1
```

A typical data line may be a combination of keyed and unkeyed data lists, such as

```
1,5,1  X=0,10  Y=2,4  Z=0
```

Only one unkeyed data list is permitted on a data line, and it must be the first data list. The keyed data lists can appear in any sequence. In the above example the list 1,5,1 must be first, but the list X=0,10 can be before or after the list Y=2,4. If a data list is only partially entered, the trailing (omitted) items take on default values as specified in the later topics of this chapter.

In format specifications, variable data items are indicated by boldface type. For example, the format specification for the sample data line above might be given as:

j0, j1, ji1 X=**x0, x1** Y=**y0, y1** Z=**z0, z1**

You should substitute the appropriate values for these variables when entering a data line into the input data file. For the above example, "0" has been substituted for **z0**, but the value for **z1** has been omitted and allowed to default.

Decimal points for whole floating point numbers are not necessary. For example, the number 6.0 may just be entered as 6. Scientific exponential notation is also allowed. For example, the number 1.5×10^7 may be entered as 1.5E7.

Continuations, Comments, and Blank Lines

The ampersand (&) and semicolon (;) characters indicate the end of information on a line of text. All characters to the left of the first ampersand or semicolon on a line of text are treated as actual data for the program; the remaining characters are treated as comment data and are ignored.

The ampersand indicates that the data line continues onto the next line of text. The semicolon indicates the end of the data line (no continuation). The semicolon is not needed to end a data line having no comments.

Each *line of text* in the input data file, including spaces and comment data, may contain up to 500 characters.

Each *data line* may contain up to 500 characters of data, including spaces, but not counting comment data. Multiple continuation lines are allowed, but the sum of all

characters to the left of the comment data on all lines of text may not exceed 500 characters for a single data line.

For example, the three lines of text:

```
1,5,1  X=0,10  & Joint labels and X coordinates
Y=2,4          & Y coordinates
Z=0            ; Z coordinates
```

give the same data line as the single line of text:

```
1,5,1  X=0,10  Y=2,4          Z=0
```

Be sure to include a comma and/or spaces between data items across continuations. For example, the two lines of text:

```
NAME=SECT01  TYPE=B  T=10&
10
```

would be interpreted as:

```
NAME=SECT01  TYPE=B  T=1010
```

The ampersand and semicolon have no special meaning for the Title Line. These characters will become part of the title.

Blank lines may appear anywhere in the data file and are completely ignored, except that a blank line ends continuation. A text line containing only spaces to the left of a semicolon is considered to be a blank line. For example, the three lines of text:

```
ADD=101  UX=50  & Add UX load to joint 101
; Blank line
ADD=201  UX=25
```

gives the same two data lines as the two lines of text:

```
ADD=101  UX=50  ; Add UX load to joint 101
ADD=201  UX=25
```

Arithmetic Operations

Simple arithmetic statements are possible when entering floating-point real numbers in the data lists. The following types of operators can be used:

+ for addition
– for subtraction

/ for division
* for multiplication

The operators are applied as they are encountered in the scan from left to right.

The following are examples of data entries that are possible and how they are interpreted by the program:

Data entered as:	**Is evaluated as:**
11.92*12	11.92 (12)
7.63/386.4	$\frac{7.63}{386.4}$
6.66-1.11*7.66/12.2	$\frac{(6.66-1.11)\,7.66}{12.2}$

Regular Array Specification

A regular array is group of labels that increment in a regular fashion. A regular array is specified in the input data file as a data list consisting of the starting label, the ending labels, and the label increments. The data list may or may not be keyed, i.e., may or may not be associated with a keyword.

The format of the data list for specifying a regular array depends upon the dimension of the array as follows:

- Zero dimensions (a single label with no increments):

  a0

- One dimension:

  a0, a1, ai1

- Two dimensions:

  a0, a1, ai1, a2, ai2

- Three dimensions:

  a0, a1, ai1, a2, ai2, a3, ai3

where

- **a0** is the starting label
- **a1** is the ending label in the first direction
- **a2** is the ending label in the second direction
- **a3** is the ending label in the third direction
- **ai1** is the label increment in the first direction
- **ai2** is the label increment in the second direction
- **ai3** is the label increment in the third direction

Throughout the remainder of this chapter, the format of the data list for specifying an array of arbitrary dimension will be indicated as:

a0, a1, ai1...

This indicates that you should choose one of the formats above for an array of zero, one, two, or three dimensions. In some cases, the format for an individual data line may restrict the allowable dimensions of the array.

Although the labels and increments **a0, a1, ai1...** have been used here to illustrate the specification of regular arrays, other variable names may be used instead, such as **j0, j1, ji1...** or **e0, e1, ei1...**. No matter what variable names are used, the interpretation of the starting label, ending labels, and label increments in the data list is the same.

The following rules apply to the specification of regular arrays:

- The starting label is always required
- There may be zero, one, two, or three ending labels; the dimension of the array is determined by the number of ending labels specified
- There is no default for starting or ending labels
- For each ending label, a label increment must be specified
- There is no default for label increments

See Topic “Regular Arrays” in Chapter “Labels, Arrays, and Generation” of the *SAP2000 Analysis Reference* for more information.

Frequently Used Keywords

Many keywords and their associated data lists are used repeatedly throughout the different data blocks in the data file. Some of the most frequently used keywords are described here.

NAME Keyword

The specification:

NAME=**name**

is used to assign the label **name** to a new entity being defined. The type of entity being defined in a given data block is indicated by the separator. For example, **name** applies to a new Constraint in the CONSTRAINT data block, and to a new Load in the LOAD data block.

Joints and elements do not use the NAME keyword. The labels for new joints and elements are given at the beginning of the appropriate data lines without a keyword.

See Topic "Labels" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

GEN Keyword

The specification:

GEN=**a0, a1, ai1...**

is used to generate (create) new items in the specified array, **a0, a1, ai1...**, from the existing definition of the starting item, **a0**. These items may be elements, Constraints, or Welds. The type of item being generated in a given data block is indicated by the separator. For example, Constraints are being generated in the CONSTRAINT data block, and Frame elements are being generated in the FRAME data block.

Several similar specifications are used to generate joints in the Joint data block, such as:

LGEN=**j0, j1, ji1...**

See Topic "Generation" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

DEL Keyword

The specification:

DEL=**a0, a1, ai1...**

is used to delete (eliminate) all items in the specified array, **a0, a1, ai1...**, from the model. Nonexistent items may be included in the array. These items may be elements, Constraints, or Welds. The type of item being deleted in a given data block is indicated by the separator. For example, Welds are being deleted in the WELD data block, and Shell elements are being deleted in the SHELL data block.

See Topic "Deletion" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

ADD Keyword

The specification:

ADD=**a0, a1, ai1...**

is used to assign a load or property to all existing joints or elements in the specified array, **a0, a1, ai1...**. Nonexistent joints or elements may be included in the array.

Unlike the GEN keyword, the ADD keyword does not create any of the items in the array.

The type of load or property being assigned in a given data block is indicated by the separator and by other data on the same or previous data lines in the data block.

The type of array (joint, Frame, Shell, etc.) is determined by the type of load or property being assigned, and sometimes by the ELEM keyword (see below).

The specification:

ADD=∗

may be used to indicate an assignment to all of the joints or element of the appropriate type.

See Topic "Assignment" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

REM Keyword

The specification:

REM=**a0, a1, ai1...**

is used to remove (set to zero) a load or property from all existing joints or elements in the specified array, **a0, a1, ai1...**. Nonexistent joints or elements may be included in the array.

Unlike the DEL keyword, the REM keyword does not eliminate any of the items in the array.

The type of load or property being removed in a given data block is indicated by the separator and by other data on the same or previous data lines in the data block.

The type of array (joint, Frame, Shell, etc.) is determined by the type of load or property being removed, and sometimes by the ELEM keyword (see below).

See Topic "Assignment" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

ELEM Keyword

The specification:

ELEM=**elem**

is used to select an element type to which subsequent ADD and REM specifications in a data block apply. The valid values for **elem** depend upon the particular data block and context, but they must be from among JOINT, FRAME, SHELL, PLANE, ASOLID, SOLID, and NLLINK. Note that joints are treated as a type of element for this purpose.

CSYS Keyword

The specification:

CSYS=**csys**

is used to select a fixed coordinate system that applies to subsequent data lines in a data block until the next CSYS specification is given. The variable **csys** must be one of:

- The label of an Alternate Coordinate System
- Zero, which indicates the global coordinate system

A CSYS specification only applies to subsequent data lines in the current data block; it does not affect any other data block. The global coordinate system is used (CSYS=0) until the first CSYS specification is encountered in a data block.

See Topic "Alternate Coordinate Systems" in Chapter "Coordinate Systems" of the *SAP2000 Analysis Reference*.

UX, UY, UZ, RX, RY, and RZ Keywords

The specifications:

UX=**ux**, UY=**uy**, and UZ=**uz**

are used to specify numeric values for translations, forces, and translational properties that act parallel to the X, Y, and Z axes, respectively, of a fixed coordinate system.

Similarly, the specifications:

RX=**rx**, RY=**ry**, and RZ=**rz**

are used to specify numeric values for rotations, moments, and rotational properties that act parallel to the X, Y, and Z axes, respectively, of a fixed coordinate system.

The fixed coordinate system may be the global system or an Alternate Coordinate System, as indicated by the most recent CSYS specification. See the previous subtopic.

U1, U2, U3, R1, R2, and R3 Keywords

The specifications:

U1=**u1**, U2=**u2**, and U3=**u3**

are used to specify numeric values for translations, forces, and translational properties that act parallel to the 1, 2, and 3 axes, respectively, of the local coordinate system of the joint, element, or other entity to which they apply.

Similarly, the specifications:

RX=**rx**, RY=**ry**, and RZ=**rz**

are used to specify numeric values for rotations, moments, and rotational properties that act parallel to the 1, 2, and 3 axes, respectively, of the local coordinate system of the joint, element, or other entity to which they apply.

See Topic "Local Coordinate Systems" in Chapter "Coordinate Systems" of the *SAP2000 Analysis Reference*.

How to Prepare the Input Data File

You should read all the preceding topics in this chapter for general information about the structure and content of the input data file.

Use a text editor to create or modify the input data file. The input data filename should have an extension of .S2K (e.g., EXAMPLE.S2K). Enter the data required by your particular problem according to the format specifications presented in the remainder of this chapter.

Each of the remaining topics, from "The Title Line" through "END Data Block," gives the detailed format of a single data block. It is suggested, but not required, that you prepare the various data blocks in the order in which they are presented in this chapter.

The following information is provided for each data block topic:

- A brief description of the data block is given, and reference is made to background material that you should read before preparing the data
- A "Data Block Format" subtopic describes the types of data lines available and their ordering in the data block; see Subtopic "Data Block Format" below
- A "Data Line Format" subtopic describes the format of the individual data lines; see Subtopic "Data Line Format" below
- An "Examples" subtopic may be given
- A "Description of Variables" subtopic describes each of the variable data items; see Subtopic "Description of Variables" below
- A "Notes" subtopic gives additional details about the variable data items and provides cross-references to background material

Data Block Format

The “Data Block Format” subtopic for each data block begins with a schematic that shows the structure of the data block. For example, the schematic for the CONSTRAINT data block is:

CONSTRAINT	Separator
CSYS=	Coordinate System Data Lines
NAME=	Name Data Lines
ADD=	Add Data Lines
REM=	Remove Data Lines
GEN=	Generate Data Lines
DEL=	Delete Data Lines

Each line in this schematic represents one type of data line. The name of the data line and a typical keyword found on the data line are shown.

All data lines at a given level of indentation may be repeated and intermingled. All data lines that are more indented may only follow the preceding data line that is less indented. For example, Coordinate System, Name, Generate, and Delete data lines may be arbitrarily intermingled. Each Name data line may be followed by a group of arbitrarily intermingled Add and Remove data lines; this group ends with the next Coordinate System, Name, Generate, or Delete data line.

The following is sample data for the CONSTRAINT data block:

```
CONSTRAINT
 NAME=FLOOR01 TYPE=DIAPH
  ADD=1011,1099,1
  REM=1055,1056,1
  REM=1065,1066,1
  ADD=1111,1155,1
 GEN=FLOOR01,FLOOR10,1 JINC=1000
 DEL=FLOOR05
 NAME=FLOOR05 TYPE=DIAPH
  ADD=1011,1099,1
```

Indentation is not required in the input data file. It is used here for clarity.

A vertical bar to the left of a data line in the schematic indicates a required data line that cannot be repeated. For example, the schematic for the COORDINATE data block is:

COORDINATE		Separator
NAME=		Name Data Lines
	X=	Z Axis Data Line
		X= Z-X Plane Data Line

The Name data line may be repeated as often as needed. Every Name data line is followed by a single Z Axis data line, which in turn is followed by a single Z-X Plane data line.

The following is sample data for the COORDINATE data block:

```
COORDINATE
 NAME=45DEG
  Z=1
  X=1 Y=1
 NAME=60DEG
  Z=1
  CR=1 CA=60
```

Each schematic is followed by a general description of each of the data lines and how they function in the data block.

See Topic “Data Blocks and Separators” (page 9) in this chapter for more information.

Data Line Formats

The “Data Line Format” subtopic for each data block gives the detailed format specifications for each type of data line. For example, one of the data line format specifications from the JOINT data block is:

Definition Data Line — Single Joint in Rectangular Coordinates

j0 X=**x0** Y=**y0** Z=**z0**

In the format specifications, bold-faced items indicate variable data items which you will replace with specific values appropriate to the problem being analyzed.

Items not shown in bold face should be entered literally into the data file as shown in the format specifications.

The format specification for a given data line may sometimes be shown as several lines of text. However, it should be entered as a single data line in the input data file, using continuation as necessary.

For more information:

- See Topic “Data Lines” (page 11) in this chapter.
- See Topic “Continuations, Comments, and Blank Lines” (page 13) in this chapter.

Description of Variables

The “Description of Variables” subtopic for each data block contains a table that describes the variable data items that appear in the data line format specifications. For example, consider the following data line format specification from the SYSTEM data block:

System Data Line

DOF=**dofs** LENGTH=**length** FORCE=**force** UP=**up** CYC=**cyc** WARN=**warn** PAGE=**page** LINES=**lines**

The tabular description of the variable **length** looks like the following:

Variable	Note	Default	Description
length	(2)	[IN]	Length unit used throughout the input data file: = MM: millimeter (mm) = CM: centimeter (cm) = M: meter (m) = IN: inch (in) = FT: foot (ft)

The columns of the table are as follows:

Variable — The variable name

Notes — References to one or more notes in the “Notes” subtopic

Default — Default values, if applicable, are shown in square brackets; see Subtopic “Default Values” below

Description — A description of the variable, including allowable values and the units to be used; see Subtopic "Units" below

Default Values

In certain cases, the program will assign values to any variables that you do not specify. These default values, if applicable, are shown in square brackets.

A default value shown as "[pv]" indicates that the value of the variable on the current data line is set equal to what it was on the previous data line in that data block. The default value used if no previous value has been given is shown in parentheses; for example "[pv(0)]" indicates that "0" is used if no previous value was defined in the current data block.

Units

The data in a SAP2000 input data file may be prepared using any consistent set of units of your choice. For example, if you use meters to locate the joints and Newtons for the force loads, then you must use N/m^2 for modulus of elasticity.

It is important to note that mass and weight are not interchangeable. Weight has units of force, such and Newtons or pounds. The mass of an object can be computed by dividing its weight by, *g*, the acceleration due to gravity, expressed in consistent units of length and time.

Three types of angular units are used:

- Degrees are always used for geometry
- Radians are always used for specifying rotational displacements
- Cycles (per time) are always used for frequencies and rates of rotation; a cycle is a complete revolution (360°)

The description of each variable indicates the applicable units to be used. The following abbreviations for units are used in this chapter:

L = Length
T = Time
M = Mass
K = Temperature
F = Force, $F = ML / T^2$
cyc = Cycles

rad = Radians, rad = 2π cyc
deg = Degrees, deg = 360 cyc

If no units are indicated, the quantity is dimensionless.

The Title Line

Prepare one data line that identifies the contents of the input data file. This data line permits a descriptive title of up to 70 characters in length. This information will appear on every page of the output file created by SAP2000. This line *must* be the first line in the input data file.

This data block consists of only one data line and has no separator. This data line is always mandatory.

Data Block Format

The format of the data block is summarized in the table below:

title	Title Line

Data Line Format

Title Line

title

Description of Variables

Variable	Note	Default	Description
title			Title of up to 70 characters describing the contents of the input data file

SYSTEM Data Block

This data block defines the parameters that control the overall structural model and analysis.

This data block is optional. Prepare data according to the format described below.

Data Block Format

The format of the data block is summarized in the table below:

SYSTEM		Separator
	DOF=	System Data Line

Begin the data block with the **SYSTEM separator**.

Follow this by a single **System data line** that defines the system parameters.

Data Line Format

System Data Line

DOF=**dofs** LENGTH=**length** FORCE=**force** UP=**up** CYC=**cyc** WARN=**warn** PAGE=**page** LINES=**lines**

Example

```
SYSTEM
 DOF=UX,UY,RZ PAGE=SECTIONS
```

Description of Variables

Variable	Note	Default	Description
dofs	(1)	[ALL]	List of the global degrees of freedom that are available at every joint in the model. May be ALL, or any number of UX, UY, UZ, RX, RY and RZ
length	(2)	[IN]	Length unit used throughout the input data file: = MM: millimeter (mm) = CM: centimeter (cm) = M: meter (m) = IN: inch (in) = FT: foot (ft)
force	(2)	[KIP]	Force unit used throughout the input data file: = N: newton (N) = KN: kilonewton (kN = 1000 N) = KGF: kilogram-force (kgf) = TON: metric ton (1000 kgf) = LB: pound (lb) = KIP: kilopound (kip = 1000 lb)
up	(3)	[+Z]	Rectangular coordinate direction assumed to be upward that is to be converted to +Z upon import. May be any one of ±X, ±Y, or ±Z. The sign is required
cyc	(4)	[0]	Load frequency [cyc/T units] = 0: Static analysis > 0: Harmonic steady-state analysis
warn	(5)	[Y]	Warning output control parameter: = Y: Output all warnings = N: Suppress all warnings

Variable	Note	Default	Description
page	(6)	[LINES]	Output file page-eject control parameter: = LINES: Eject pages at new section headings and when **lines** exceeded = SECTIONS: Eject pages only at new section headings
lines	(6)	[59]	Maximum number of lines per page permitted in output files when **page**=LINES

Notes

1. **dofs** is a list of one or more global degrees of freedom that are permitted to be present at every joint in the model. Specifying ALL is the same as listing all six degrees of freedom. This is the default and should generally be used for all three-dimensional structures.

   See Topic “Degrees of Freedom” in Chapter “Joints and Degrees of Freedom” of the *SAP2000 Analysis Reference*.

2. The data in a SAP2000 input data file may be prepared using any consistent set of units of your choice. These units do not need to be specified in the SYSTEM Data Block except in the following cases:

   - Section properties are read from a property database file, in which case **length** is needed. See the SECTION Data Block (page 81).
   - Standard vehicle loads are used for moving-load analysis, in which case **length** and **force** are needed. See the VEHICLE Data Block (page 179).

   Section properties and standard vehicle loads are converted to the units specified in the SYSTEM Data Block.

3. This parameter is only used when the input data file is being imported into the SAP2000 graphical user interface. All coordinate-dependent quantities in the input data file will be converted upon import to conform with the SAP2000 convention that +Z is up. X coordinates will not be changed unless **up** = ±X, in which case the Y coordinates will be left unchanged.

4. If **cyc** is positive, the program is put into harmonic steady-state analysis mode; otherwise, static analysis is performed (the default).

P-delta, response-spectrum, time-history, and moving-load analyses may not be performed when the program is in harmonic steady-state analysis mode. As a result, the following data blocks will be ignored when **cyc** is positive: PDELTA, MODES, SPEC, HISTORY, LANE, VEHICLE, VEHICLE CLASS, BRIDGE RESPONSE, and MOVING LOAD.

See Topic "Harmonic Steady-State Analysis" in Chapter "Static and Dynamic Analysis" of the *SAP2000 Analysis Reference* for more information.

5. If **warn** is set to "N", all warning messages that are generated by the data check phase of the program will not appear in the echo output file (e.g., EXAMPLE.EKO). The messages, however, will always appear on the screen, irrespective of the value of **warn**.

   Warning messages generated during the execution of the analysis phase of the program will always be printed in the log file (e.g., EXAMPLE.LOG).

6. See Topic "Pagination Control" in Chapter "The Output Files" of the *SAP2000 Analysis Reference*.

COORDINATE Data Block

This data block defines Alternate Coordinate Systems that can be used for locating the joints; for defining local coordinate systems for joints, elements and constraints; and as a reference for other properties and loads.

Skip this data block if there are no Alternate Coordinate Systems to be defined. Otherwise, prepare data according to the format described below.

For More Information

See Topic "Alternate Coordinate Systems" in Chapter "Coordinate Systems" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

COORDINATE			Separator
	NAME=		Name Data Lines
		X=	Vertical Axis Data Line
			X= Vertical Plane Data Line

Begin the data block with the **COORDINATE separator**.

Follow this by as many Name, Vertical Axis, and Vertical Plane data lines as necessary to define all of the Alternate Coordinate Systems used in the model.

Each **Name data line** begins the definition of a new Alternate Coordinate System and locates the origin of the new system.

Each Name data line is followed by a single **Vertical Axis data line** that locates a point on the +Z half of the new Z axis.

Each Vertical Axis data line is followed by a single **Vertical Plane data line** that locates a point on the +X half of the new Z-X plane.

Data Line Formats

Name Data Line — Using Rectangular Coordinates

NAME=**name** X=**x0** Y=**y0** Z=**z0**

Name Data Line — Using Cylindrical Coordinates

NAME=**name** CR=**cr0** CA=**ca0** CZ=**cz0**

Name Data Line — Using Spherical Coordinates

NAME=**name** SB=**sb0** SA=**sa0** SR=**sr0**

Vertical Axis Data Line — Using Rectangular Coordinates

X=**x1** Y=**y1** Z=**z1**

Vertical Axis Data Line — Using Cylindrical Coordinates

CR=**cr1** CA=**ca1** CZ=**cz1**

Vertical Axis Data Line — Using Spherical Coordinates

SB=**sb1** SA=**sa1** SR=**sr1**

Vertical Plane Data Line — Using Rectangular Coordinates

X=**x2** Y=**y2** Z=**z2**

Vertical Plane Data Line — Using Cylindrical Coordinates

CR=**cr2** CA=**ca2** CZ=**cz2**

Vertical Plane Data Line — Using Spherical Coordinates

SB=**sb2** SA=**sa2** SR=**sr2**

Examples

(1) This example considers a two-dimensional problem in the horizontal X-Y plane. An Alternate Coordinate System can be defined that rotates the X and Y axes 45° about the Z axis as follows:

```
COORDINATE
 NAME=45DEG
  Z=1
  CR=1 CA=45
```

The same results could alternately be achieved using:

```
COORDINATE
 NAME=45DEG
  Z=1
  X=1 Y=1
```

(2) This example defines an Alternate Coordinate System located at a point on the surface of an cylinder centered on the global Z axis and of radius 10. The new X axis is normal to the cylinder, the new Y axis tangential to the circumferential direction, and the new Z axis parallel to the cylinder axis:

```
COORDINATE
 NAME=CYL CR=10   CA=30 CZ=5
          CR=10   CA=30 CZ=5+1
          CR=10+1 CA=30 CZ=5
```

(3) This example defines an Alternate Coordinate System located at a point on the surface of an origin-centered sphere of radius 10. The new X axis is normal to the sphere, the Y axis tangential to the latitude line, and the Z axis tangential to the longitude line:

```
COORDINATE
 NAME=SPH SB=45    SA=30 SR=10
          SB=45-60 SA=30 SR=2*10
          SB=45    SA=30 SR=10+1
```

Description of Variables

Variable	Note	Default	Description
Name Data Line			
name	(1, 2)		Label of an Alternate Coordinate System being defined
x0, y0, z0	(1, 3)	[0]	Global rectangular X, Y, and Z ordinates of the new origin [L, L, L units]
cr0, ca0, cz0	(1, 3)	[0]	Global cylindrical CR, CA, and CZ ordinates of the new origin [L, deg, L units]
sb0, sa0, sr0	(1, 3)	[0]	Global spherical SB, SA, and SR ordinates of the new origin [deg, deg, L units]
Vertical Axis Data Line			
x1, y1, z1	(1, 3)	[0]	Global rectangular X, Y, and Z ordinates of a point on the +Z half of the new vertical axis [L, L, L units]
cr1, ca1, cz1	(1, 3)	[0]	Global cylindrical CR, CA, and CZ ordinates of a point on the +Z half of the new vertical axis [L, deg, L units]
sb1, sa1, sr1	(1, 3)	[0]	Global spherical SB, SA, and SR ordinates of a point on the +Z half of the new vertical axis [deg, deg, L units]
Vertical Plane Data Line			
x2, y2, z2	(1, 3)	[0]	Global rectangular X, Y, and Z ordinates of a point on the +X half of the new Z-X plane [L, L, L units]
cr2, ca2, cz2	(1, 3)	[0]	Global cylindrical CR, CA, and CZ ordinates of a point on the +X half of the new Z-X plane [L, deg, L units]

Variable	Note	Default	Description
sb2, sa2, sr2	(1, 3)	[0]	Global spherical SB, SA, and SR ordinates of a point on the +X half of the new Z-X plane [deg, deg, L units]

Notes

1. See Topic "Alternate Coordinate Systems" in Chapter "Coordinate Systems" of the *SAP2000 Analysis Reference*.

2. Each Name data line defines a new Alternate Coordinate System. Alternate Coordinate System labels do not have to be consecutive and may be supplied in any order. Alternate Coordinate System labels may not be repeated in the data block.

   See Topic "Labels" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

3. The coordinates on each data line may be given in rectangular X-Y-Z coordinates, cylindrical CR-CA-CZ coordinates, or spherical SR-SA-SB coordinates, all measured in the global coordinate system. These coordinate types may not be mixed on a single data line, but can differ between data lines. The default value for all coordinates is zero.

JOINT Data Block

This data block defines the joints that describe the geometry of the structural model along with their associated coordinates.

This data block is mandatory. Prepare data according to the format described below.

For More Information

See Chapter "Joint Coordinates" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

JOINT	Separator
CSYS=	Coordinate System Data Lines
j0 V=	Definition Data Lines — Single Joint
j0, j1, ji1... V=	Definition Data Lines — Joint Array
LGEN=	Linear Generation Data Lines
FGEN=	Frontal Generation Data Lines
EGEN=	Edge Generation Data Lines
CGEN=	Cylindrical Generation Data Lines

Begin the data block with the **JOINT separator**.

Follow this by as many Coordinate System, Definition, and Generation data lines as necessary to define all of the joints in the model. The data is processed in the order it is supplied in the data block.

Each **Coordinate System data line** defines the fixed coordinate system and the scale factor used by all subsequent Definition data lines for the purpose of locating the joints. This fixed coordinate system and the scale factor are in effect until the next Coordinate System data line is encountered. Generation data lines are not affected by the coordinate system or the scale factor.

Each **Definition data line** defines a single joint or an array of joints. Each **Generation data line** generates an array of joints from previously defined or generated joints. Several types of generation are provided: Linear Generation, Frontal Generation, Edge Generation, and Cylindrical Generation.

Data Line Formats

Coordinate System Data Line

CSYS=**csys** SF=**sf**

Definition Data Line — Single Joint in Rectangular Coordinates

j0 X=**x0** Y=**y0** Z=**z0**

Definition Data Line — Single Joint in Cylindrical Coordinates

j0 CR=**cr0** CA=**ca0** CZ=**cz0**

Definition Data Line — Single Joint in Spherical Coordinates

j0 SB=**sb0** SA=**sa0** SR=**sr0**

Definition Data Line — Joint Array in Rectangular Coordinates

j0, j1, ji1... X=**x0, x1...** Y=**y0, y1...** Z=**z0, z1...** RATIO=**ratio1...**

Definition Data Line — Joint Array in Cylindrical Coordinates

j0, j1, ji1... CR=**cr0, cr1...** CA=**ca0, ca1...** CZ=**cz0, cz1...**
RATIO=**ratio1...**

Definition Data Line — Joint Array in Spherical Coordinates

j0, j1, ji1... SB=**sb0, sb1...** SA=**sa0, sa1...** SR=**sr0, sr1...** RATIO=**ratio1...**

Linear Generation Data Line

LGEN=**j0, j1, ji1...** RATIO=**ratio1**...

Frontal Generation Data Line

FGEN=**j0, j1, ji1, j2, ji2...**

Edge Generation Data Line

EGEN=**j0, j1, ji1, j2, ji2...**

Cylindrical Generation Data Line

CGEN=**j0, j1, ji1** AXVEC=**axveca, axvecb** DA=**da** DR=**dr** DL=**dl**

Examples

(1) Define a rectangular region of uniformly spaced joints:

```
JOINT
 1,10,1,51,10 X=0,8,0 Y=0,0,5 Z=0
```

(2) Define a trapezoidal region of uniformly spaced joints:

```
JOINT
 1  X=0 Y=0 Z=0
 10 X=8 Y=0
 51 X=1 Y=5
 60 X=6 Y=5
 LGEN=1,10,1,51,10
```

(3) Define a cylindrical helix of constant pitch, as for modeling a helical spring:

```
JOINT
 1,121,1 CR=10 CA=0,1800 CZ=0,20
```

(4) Define a grid of joints on the surface of a cylindrical shell:

```
JOINT
 1,37,1,801,100 CR=5 CA=0,360,0 CZ=0,0,15
```

(5) Define two layers of joints through the thickness of one quadrant of a hemispherical shell with an 18° opening at the top, using smaller elements near the opening:

```
JOINT
 100,109,1,170,10,200,100 SA=0,90,0,0 SB=90,90,18,90 &
  SR=150,150,150,160 RATIO=1,0.5,1
```

Description of Variables

Variable	Note	Default	Description
Coordinate System Data Line			
csys	(1, 4)	[pv(0)]	Fixed coordinate system for subsequent joint coordinates: = 0: Global coordinate system ≠ 0: Alternate coordinate system label
sf	(5)	[pv(1)]	Scale factor for subsequent lineal (not angular) joint coordinates, i.e., X, Y, Z, CR, SR
Definition Data Lines			
j0	(1, 2, 3)		Label of a single joint being defined, or of the starting joint in an array of joints being defined
j1...	(1, 2, 3)		Labels of ending joints along joint array axes 1, 2 and 3, respectively, up to the dimension of the array
ji1...	(1, 2, 3)		Label increments along joint array axes 1, 2 and 3, respectively, up to the dimension of the array
x0, x1...	(1, 6)	[pv(0)]	Rectangular X ordinates of joints **j0, j1...** [L units]
y0, y1...	(1, 6)	[pv(0)]	Rectangular Y ordinates of joints **j0, j1...** [L units]
z0, z1...	(1, 6)	[pv(0)]	Rectangular Z ordinates of joints **j0, j1...** [L units]
cr0, cr1...	(1, 6)	[pv(0)]	Cylindrical CR ordinates of joints **j0, j1...** [L units]
ca0, ca1...	(1, 6)	[pv(0)]	Cylindrical CA ordinates of joints **j0, j1...** [deg units]

Variable	Note	Default	Description
cz0, cz1...	(1, 6)	[pv(0)]	Cylindrical CZ ordinates of joints **j0, j1...** [L units]
sb0, sb1...	(1, 6)	[pv(0)]	Spherical SB ordinates of joints **j0, j1...** [deg units]
sa0, sa1...	(1, 6)	[pv(0)]	Spherical SA ordinates of joints **j0, j1...** [deg units]
sr0, sr1...	(1, 6)	[pv(0)]	Spherical SR ordinates of joints **j0, j1...** [L units]
ratio1...	(1)	[1]	For unequal spacing of joints, ratio of the last coordinate difference to the first coordinate difference along joint array axes 1, 2 and 3, respectively, up to the dimension of the array
Linear Generation Data Line			
j0, j1, ji1...	(1, 2, 3)		Labels and label increments for an array of joints having one, two or three dimensions
ratio1...	(1)	[1]	For unequal spacing of joints, ratio of the last coordinate difference to the first coordinate difference along joint array axes 1, 2 and 3, respectively, up to the dimension of the array
Frontal Generation Data Line			
j0, j1, ji1, j2, ji2...	(1, 2, 3)		Labels and label increments for an array of joints having two or three dimensions
Edge Generation Data Line			
j0, j1, ji1, j2, ji2...	(1, 2, 3)		Labels and label increments for an array of joints having two or three dimensions

Variable	Note	Default	Description
Cylindrical Generation Data Line			
j0, j1, ji1	(1, 2, 3)		Labels and label increments for a one-dimensional array of joints
axveca, axvecb	(1)		Labels of two previously-defined joints that define the axis of generation
da	(1)	[0]	Increment in angle (around axis) between generated joints [deg units]
dr	(1)	[0]	Increment in radius (away from axis) between generated joints [L units]
dz	(1)	[0]	Increment in height (along axis) between generated joints [L units]

Notes

1. See Chapter “Joint Coordinates” of the *SAP2000 Analysis Reference*.

2. Each Definition data line defines a single joint, **j0**, or an array of joints, **j0, j1, ji1...**, having one, two or three dimensions. Joint labels do not have to be consecutive and may be supplied in any order. Joints may be redefined or regenerated, in which case only the last definitions will be used.

   See Chapter “Labels, Arrays, and Generation” of the *SAP2000 Analysis Reference*.

3. See Topic “Regular Array Specification” (page 15) in this chapter.

4. All specified coordinates X, Y, Z, CR, CA, CZ, SB, SA, and SR are taken in the most recent coordinate system **csys** specified. If **csys**=0, the global system is used. Otherwise **csys** refers to an Alternate Coordinate System defined in the COORDINATE Data Block (page 32). If no **csys** is specified, the global system is used.

   See Chapter “Coordinate Systems” of the *SAP2000 Analysis Reference*.

5. The scale factor **sf** multiplies all lineal coordinate values specified on subsequent data lines, until the scale factor is redefined. The lineal coordinates are X,

Y, Z, CR, CZ, and SR. The angles CA, SB, and SA are not scaled. If no **sf** is specified, the default value of unity is used.

6. The location of the joints may be specified using rectangular X-Y-Z coordinates, cylindrical CR-CA-CZ coordinates, or spherical SB-SA-SR coordinates. These coordinate types may not be mixed on a single data line.

   At least one coordinate value must be specified on each Joint Definition or Joint Array Definition data line. The type of coordinate system (rectangular, cylindrical, or spherical) is determined from the specified coordinate value(s). Previous values are used for any unspecified coordinates. The previous value refers to the last explicit definition of that coordinate value for joint **j0** on a Joint Definition or Joint Array Definition data line.

   For example, if only X and Y are specified on a data line, the previous value of **z0** is used for Z. If only CR is specified on a data line, the previous values of **ca0** and **cz0** are used for CA and CZ.

   When a constant coordinate value is being assigned to an array of joints, it is not necessary to repeat that value on the data line; e.g., for a two-dimensional array of joints, specifying Z=10 is the same as specifying Z=10,10,10. Omitting Z altogether will assign the previous value of **z0** to all joints.

LOCAL Data Block

This data block defines the local coordinate systems associated with the degrees of freedom at the joints. The global coordinate system will be used for any joint local coordinate system not defined in this data block. The joint local coordinate system is *not* related to any coordinate system used to locate the joints in the JOINT Data Block (page 37).

Skip this data block if there are no joint local coordinate systems to be defined, i.e., if all joint degrees of freedom are in the global coordinate system. Otherwise, prepare data according to the format described below.

For More Information

See Topic "Local Coordinate System" in Chapter "Joints and Degrees of Freedom" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

LOCAL	Separator
CSYS=	Coordinate System Data Lines
ADD=	Add Data Lines
REM=	Remove Data Lines

Begin the data block with the **LOCAL separator**.

Follow this with as many Coordinate System, Add, and Remove data lines as necessary to define all of the joint local coordinate systems. The data is processed in the order it is supplied in the input data file.

Each **Coordinate System data line** defines the fixed coordinate system, the coordinate directions, and the local plane used by all subsequent Add data lines. These values are in effect until the next Coordinate System data line is encountered.

Each **Add data line** defines the local coordinate systems for an array of one or more joints. Each **Remove data line** removes the local coordinate systems from an array of one or more joints, returning them to the global coordinate system.

Data Line Formats

Coordinate System Data Line

CSYS=**csys** AXDIR=**axdir** PLDIR=**pldirp, pldirs** LOCAL=**local**

Add Data Line

ADD=**j0, j1, ji1...** AXVEC=**axveca, axvecb** PLVEC=**plveca, plvecb** ANG=**a, b, c**

Remove Data Line

REM=**j0, j1, ji1...**

Example

(1) This example applies a local coordinate system to all perimeter joints in a 5 x 6 array of joints (numbers 1 to 30). This local coordinate system has the local 1 and 2 axes rotated by 30° about the 3 (Z) axis. The local system is first applied to all thirty joints, then removed from the inner 3 x 4 array of joints:

```
LOCAL
 ADD=1,5,1,26,5 ANG=30
 REM=7,9,1,22,5
```

Alternatively, the same result could be obtained by specifying each edge separately as:

```
LOCAL
 ADD=1,5,1   ANG=30
 ADD=1,26,5  ANG=30
 ADD=26,30,1 ANG=30
 ADD=5,30,5  ANG=30
```

Description of Variables

Variable	Note	Default	Description
Coordinate System Data Line			
csys	(1, 3)	[pv(0)]	Fixed coordinate system used to define coordinate directions **axdir**, **pldirp**, and **pldirs**: = 0: Global coordinate system ≠ 0: Alternate coordinate system label
axdir	(1, 3)	[pv(+Z)]	Axial coordinate direction, taken at the joint in fixed coordinate system **csys**, used to determine the axis reference vector. May be one of ±X, ±Y, ±Z, ±CR, ±CA, ±CZ, ±SB, ±SA, or ±SR. The sign is required
pldirp, pldirs	(1, 3)	[pv(+X, +Y)]	Primary and secondary coordinate directions, taken at the joint in fixed coordinate system **csys**, used to determine the plane reference vector. Each may be one of ±X, ±Y, ±Z, ±CR, ±CA, ±CZ, ±SB, ±SA, or ±SR. The sign is required. If only **pldirp** is specified, **pldirs** is set equal to **pldirp**.
local	(1)	[pv(31)]	Local plane (and axis) parallel to the reference vectors: = 12: Plane 1-2 (axis 1) = 13: Plane 1-3 (axis 1) = 21: Plane 2-1 (axis 2) = 23: Plane 2-3 (axis 2) = 31: Plane 3-1 (axis 3) = 32: Plane 3-2 (axis 3)
Add Data Line			
j0, j1, ji1...	(1, 2)		Labels and label increments for an array of joints being assigned joint local coordinate systems

Variable	Note	Default	Description
axveca, axvecb	(1)	[0, 0]	Labels of two joints that define the axis reference vector. Either joint may be zero to indicate the current joint in the array. If both are zero, this option is not used
plveca, plvecb	(1)	[0, 0]	Labels of two joints that define the plane reference vector. Either joint may be zero to indicate the current joint in the array. If both are zero, this option is not used
a, b, c	(1)	[0, 0, 0]	Angles that the local coordinate system is rotated first about its 3 axis (**a**), then about its resulting 2 axis (**b**), and finally about its resulting 1 axis (**c**) [deg units]
Remove Data Line			
j0, j1, ji1...	(1, 2)		Labels and label increments for an array of joints being returned to global coordinate system

Notes

1. See Topic “Local Coordinate System” in Chapter “Joints and Degrees of Freedom” of the *SAP2000 Analysis Reference*.
2. See Topic “Regular Array Specification” (page 15) in this chapter.
3. The coordinate directions **axdir**, **pldirp** and **pldirs** are taken in the most recently specified coordinate system **csys**. If **csys** is zero, the global system is used. Otherwise **csys** refers to an alternate coordinate system defined in the COORDINATE Data Block (page 32). If no **csys** is specified, the global system is used.

   See Chapter “Coordinate Systems” of the *SAP2000 Analysis Reference*.

RESTRAINT Data Block

This data block defines all of the joint Restraints that are needed to support the structure. Restraints only apply to the available degrees of freedom, as specified in the SYSTEM Data Block (page 28). Unavailable degrees of freedom are automatically restrained. Displacements of the Restraints (e.g., support settlement) may be specified in the LOAD Data Block (page 134).

This data block is mandatory unless the model is adequately supported by springs. Prepare data according to the format described below.

For More Information

See Topics “Restraints and Reactions” and “Degrees of Freedom” in Chapter “Joints and Degrees of Freedom” of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

RESTRAINT	Separator
ADD=	Add Data Lines
REM=	Remove Data Lines

Begin the data block with the **RESTRAINT separator**.

Follow this with as many Add and Remove data lines as necessary to define all of the Restraints. The data is processed in the order it is supplied in the input data file.

Each **Add data line** adds Restraints to selected degrees of freedom for an array of one or more joints. Each **Remove data line** removes Restraints from selected degrees of freedom for an array of one or more joints.

Data Line Formats

Add Data Line

ADD=**j0, j1, ji1...** DOF=**dofs**

Remove Data Line

REM=**j0, j1, ji1...** DOF=**dofs**

Examples

(1) A rectangular plate in the X-Y plane is simply supported on all four sides. The Z displacement and the rotation about the axis normal to each edge is restrained. The corner joints, being included in the two adjacent edges, thus have both rotations restrained. All joints are in the global coordinate system.

```
RESTRAINT
 ADD= 1, 5,1 DOF=U3,R2
 ADD=21,25,1 DOF=U3,R2
 ADD= 1,21,5 DOF=U3,R1
 ADD= 5,25,5 DOF=U3,R1
```

(2) Another rectangular plate in the X-Y plane is fully clamped on all four sides. First the Z displacement and both rotations are fixed at all joints in the plate, then these degrees of freedom are released for the interior joints. This method permits a simpler specification. All joints are in the global coordinate system.

```
RESTRAINT
 ADD=1,5,1,21,5 DOF=UZ,RX,RY
 REM=7,9,1,17,5 DOF=UZ,RX,RY
```

For both examples, either local or global degree-of-freedom specifications may be used since all joints are in the global coordinate system.

Description of Variables

Variable	Note	Default	Description
Add and Remove Data Lines			
j0, j1, ji1...	(1)		Labels and label increments for an array of joints having restraints added or removed
dofs	(2)	[ALL]	List of degrees of freedom at the joints having restraints added or removed. May be ALL; or any number of U1, U2, U3, R1, R2 and R3; or any number of UX, UY, UZ, RX, RY and RZ

Notes

1. See Topic “Regular Array Specification” (page 15) in this chapter.

2. **dofs** is a list of one or more local or global degrees of freedom that are to have restraints added to (restrained) or removed from (unrestrained) each joint in the array. Local and global degrees of freedom may not be mixed on a single data line. Specifying ALL is the same as listing all six *local* degrees of freedom.

   Restraints are always applied to local degrees of freedom. If global degrees of freedom are specified, the restraints are added to or removed from the parallel local degrees of freedom at each joint. If no local degree of freedom can be found at a particular joint that is parallel to a listed global degree of freedom, no corresponding restraint is added or removed.

   Each available degree of freedom at each joint in the structure must be either restrained or unrestrained. Initially all available degrees of freedom are unrestrained. The data lines are processed in the order that they are given. Repeated joint degree-of-freedom specifications are allowed; the last specification (Add or Remove) will govern.

   See Topic “Restraints and Reactions” in Chapter “Joints and Degrees of Freedom” of the *SAP2000 Analysis Reference*.

CONSTRAINT Data Block

This data block defines the Constraints that are used to enforce certain types of rigid-body behavior, to connect together different parts of the model, and to impose certain types of symmetry conditions.

Skip this data block if there are no Constraints to be defined. Otherwise, prepare data according to the format described below.

For More Information

See Chapter "Constraints and Welds" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

CONSTRAINT	Separator
CSYS=	Coordinate System Data Lines
NAME=	Name Data Lines
ADD=	Add Data Lines
REM=	Remove Data Lines
GEN=	Generate Data Lines
DEL=	Delete Data Lines

Begin the data block with the **CONSTRAINT separator**.

Follow this with as many Coordinate System, Name, Add, Remove, Generate, and Delete data lines as necessary to define all the Constraints. The data is processed in the order in which it is given in the data file.

Each **Coordinate System data line** defines the fixed coordinate system to be used by the Constraints defined on subsequent Name data lines. The coordinate system is in effect until the next Coordinate System data line is encountered.

Each **Name data line** begins the definition of a new Constraint, and may be followed by any number of Add and Remove data lines; at least one Add data line is required.

Each **Add** or **Remove data line** lists an array of constrained joints that are to be added to or removed from the Constraint of the previous Name data line. No Add or Remove data lines may follow a Generate or Delete data line.

Each **Generate data line** generates an array of Constraints from a previously defined or generated Constraint. Each **Delete data line** deletes an array of unwanted Constraints.

Data Line Formats

Coordinate System Data Line

CSYS=**csys**

Name Data Line — Body Constraint

NAME=**name** TYPE=BODY

Name Data Line — Diaphragm, Plate, Rod, or Beam Constraint

NAME=**name** TYPE=**type** AXIS=**axis**

Name Data Line — Equal Constraint

NAME=**name** TYPE=EQUAL DOF=**cdofs**

Name Data Line — Local Constraint

NAME=**name** TYPE=LOCAL DOF=**ldofs**

Add Data Line

ADD=**j0, j1, ji1...**

Remove Data Line

REM=**j0, j1, ji1...**

Generate Data Line

GEN=**i0, i1, ii1...** JINC=**ji1...**

Delete Data Line

DEL=**i0, i1, ii1...**

Examples

(1) A ten-story building has an L-shaped floor plan. Each of the ten floors is to be modeled as a rigid diaphragm, i.e., no deformation is permitted in the plane of the floor. A DIAPHRAGM constraint is defined for the first floor, and then generated to the other nine floors. The joint label increment between floors is 1000. All joints on a given floor lie in the same plane.

```
CONSTRAINTS
 NAME=FLOOR01 TYPE=DIAPH
  ADD=1011,1015,1,1041,10
  ADD=1051,1059,1,1091,10
 GEN=FLOOR01,FLOOR10,1 JINC=1000
```

(2) Joints 101 to 125 are to be constrained on a one-to-one basis to have the same deflections as joints 201 to 225, respectively. This could be specified as:

```
CONSTRAINT
 NAME=1 TYPE=BODY
  ADD=101,201,100
 GEN=1,25,1
```

This creates 25 separate constraints. Each constraint has a pair of constrained joints. If the two joints in each constraint pair occupy the same spatial location, these 25 constraints could alternatively be defined using the WELD Data Block as:

```
WELD
 NAME=1
  ADD=101,125,1
  ADD=201,225,1
```

(3) A structure is symmetric with respect to the Y-Z plane and is loaded symmetrically; thus the deflections will be symmetric. This symmetry condition can be imposed with EQUAL constraints, thus halving the number of equations to be solved:

```
CONSTRAINT
 NAME=1 TYPE=EQUAL DOF=-UX,UY,UZ,RX,-RY,-RZ
  ADD=LEFT01
  ADD=RIGHT01
 GEN=1,25,1
```

(4) A structure is symmetric with respect to the Y-Z plane and is loaded *antisymmetrically*; thus the deflections will be antisymmetric. This antisymmetry condition can be imposed with EQUAL constraints, thus halving the number of equations to be solved:

```
CONSTRAINT
 NAME=1 TYPE=EQUAL DOF=UX,-UY,-UZ,-RX,RY,RZ
  ADD=LEFT01
  ADD=RIGHT01
 GEN=1,25,1
```

Description of Variables

Variable	Note	Default	Description
Coordinate System Data Line			
csys	(1, 4)	[pv(0)]	Fixed coordinate system used to define **axis** and **cdofs**: = 0: Global coordinate system ≠ 0: Alternate coordinate system label
Name Data Line			
name	(1, 2)		Label of a Constraint being defined
type	(1, 2)		Constraint type: = DIAPH: Rigid Diaphragm = PLATE: Rigid Plate = ROD: Rigid Rod = BEAM: Rigid Beam

Variable	Note	Default	Description
axis	(1)	[0]	Axis in coordinate system **csys** that is perpendicular to the plane of the Diaphragm or Plate Constraint, or parallel to the axis of the Rod or Beam Constraint. May be one of 0, X, Y or Z. If 0, the axis is automatically determined from the joints
cdofs	(1, 5)	[ALL]	List of degrees of freedom, in coordinate system **csys**, for the Equal Constraint. May be ALL, or any number of ±UX, ±UY, ±UZ, ±RX, ±RY, and ±RZ; the "+" sign is optional
ldofs	(1, 5)	[ALL]	List of degrees of freedom, in each joint local coordinate system, for the Local Constraint. May be ALL, or any number of ±U1, ±U2, ±U3, ±R1, ±R2, and ±R3; the "+" sign is optional
Add and Remove Data Lines			
j0, j1, ji1...	(1, 3, 6)		Labels and label increments for an array of constrained joints to be added to or removed from a Constraint
Generate Data Line			
i0, i1, ii1...	(3, 7)		Labels and label increments for an array of Constraints to be generated
ji1...	(3, 7)	[**ii1...**]	Secondary increments for the constrained joint labels
Delete Data Line			
i0, i1, ii1...	(3, 8)		Labels and label increments for an array of Constraints to be deleted

Notes

1. See Chapter "Constraints and Welds" of the *SAP2000 Analysis Reference*.

2. Each Name data line defines a single Constraint. Constraint labels do not have to be consecutive and may be supplied in any order. The type of Constraint being defined *must* be specified on the Name data line.

   Constraints may be redefined or regenerated, in which case only the last definition or generation will be used. The Constraint type may be changed upon redefinition or regeneration.

   See Topic "Labels" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

3. See Topic "Regular Array Specification" (page 15) in this chapter.

4. All specifications for **axis** and **cdofs** are taken in the most recently specified coordinate system **csys**. If **csys** is zero, the global system is used. Otherwise **csys** refers to an Alternate Coordinate System defined in the COORDINATE Data Block (page 32). If no **csys** is specified, the global system is used.

5. Specifying ALL is the same as listing all six positive degrees of freedom.

6. Each Add and Remove data line specifies an array of one or more constrained joints to be added to or removed from the Constraint being defined. The additions and removals are processed in the order that they are given in the data file. Nonexistent joints may be added or removed; if added, they are retained for generation purposes, but are ultimately removed by the program after all Constraints have been defined. A joint that is added more than once (e.g., in overlapping arrays) still counts as a single addition, and can be removed by a single, subsequent removal.

7. Each Generate data line defines an array of Constraints of the same type as the starting Constraint, **i0**, and having the same values as the starting Constraint for **csys**, **axis**, **cdofs**, and/or **ldofs**.

   Note that the values of **csys** used by the generated Constraints is that of the starting Constraint, **i0**, which is not necessarily the value on the most recent Coordinate System data line.

   Each generated Constraint will contain the same number of joints as the starting Constraint, but the joint labels will differ according to the secondary joint label increments. Even nonexistent joints from Constraint **i0** are generated;

they are ultimately eliminated by the program after all Constraints have been defined.

See Topic “Generation” in Chapter “Labels, Arrays, and Generation” of the *SAP2000 Analysis Reference*.

8. Each Delete data line defines an array of one or more elements to be deleted. Nonexistent elements may be included in the array.

   See Topic “Deletion” in Chapter “Labels, Arrays, and Generation” of the *SAP2000 Analysis Reference*.

WELD Data Block

This data block defines the Welds that are used to connect together different parts of the model.

Skip this data block if there are no Welds to be defined. Otherwise, prepare data according to the format described below.

For More Information

See Chapter "Constraints and Welds" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

WELD	Separator
NAME=	Name Data Lines
ADD=	Add Data Lines
REM=	Remove Data Lines
GEN=	Generate Data Lines
DEL=	Delete Data Lines

Begin the data block with the **WELD separator**.

Follow this with as many Name, Add, Remove, Generate, and Delete data lines as necessary to define all the Welds. The data is processed in the order in which it is given in the data file.

Each **Name data line** begins the definition of a new Weld, and may be followed by any number of Add and Remove data lines; at least one Add data line is required.

Each **Add** or **Remove data line** lists an array of joints that are to be added to or removed from the Weld of the previous Name data line. No Add or Remove data lines may follow a Generate or Delete data line.

Each **Generate data line** generates an array of Welds from a previously defined or generated Weld. Each **Delete data line** deletes an array of unwanted Welds.

Data Line Formats

Name Data Line

NAME=**name** TOL=**tol**

Add Data Line

ADD=**j0, j1, ji1...**

Remove Data Line

REM=**j0, j1, ji1...**

Generate Data Line

GEN=**i0, i1, ii1...** JINC=**ji1...**

Delete Data Line

DEL=**i0, i1, ii1...**

Example

(1) For most structures, a single Weld can be defined that encompasses all joints in the model. Any two or more joints in the same location will be constrained together. For example:

```
WELD
 NAME=ALL TOL=0.000001
  ADD=*
```

(2) Suppose that on the first floor of a structure, joints 1001 to 1010 are to be welded with joints 1101 to 1110, with any other coincident joints to remain unconnected. Furthermore, similar Welds are needed on the next four floors, and the joint numbers increment by 1000 from one floor to the next. This could be specified as:

```
WELD
 NAME=FLOOR1 TOL=0.000001
  ADD=1001,1010,1
  ADD=1101,1110,1
  GEN=FLOOR1,FLOOR4,1 JINC=1000
```

Description of Variables

Variable	Note	Default	Description
Name Data Line			
name	(1, 2)		Label of a Weld being defined
tol	(1)	[pv (10^{-6})]	Distance tolerance [L units]
Add and Remove Data Lines			
j0, j1, ji1...	(1, 3, 4)		Labels and label increments for an array of joints to be added to or removed from a Weld
Generate Data Line			
i0, i1, ii1...	(3, 5)		Labels and label increments for an array of Welds to be generated
ji1...	(3, 5)	[**ii1...**]	Secondary increments for the joint labels
Delete Data Line			
i0, i1, ii1...	(3, 6)		Labels and label increments for an array of Welds to be deleted

Notes

1. See Chapter “Constraints and Welds” of the *SAP2000 Analysis Reference*.
2. Each Name data line defines a single Weld. Weld labels do not have to be consecutive and may be supplied in any order. Constraints may be redefined or regenerated, in which case only the last definition or generation will be used.

   See Topic “Labels” in Chapter “Labels, Arrays, and Generation” of the *SAP2000 Analysis Reference*.
3. See Topic “Regular Array Specification” (page 15) in this chapter.
4. Each Add and Remove data line specifies an array of one or more joints to be added to or removed from the Weld being defined. The additions and removals are processed in the order that they are given in the data file. Nonexistent joints

may be added or removed; if added, they are retained for generation purposes, but are ultimately removed by the program after all Welds have been defined. A joint that is added to a Weld more than once (e.g., in overlapping arrays) still counts as a single addition, and can be removed by a single, subsequent removal.

5. Each Generate data line defines an array of Welds having the same values as the starting Weld, **i0**, for the distance tolerance.

   Each generated Weld will contain the same number of joints as the starting Weld, but the joint labels will differ according to the secondary joint label increments. Even nonexistent joints from Weld **i0** are generated; they are ultimately eliminated by the program after all Welds have been defined.

   See Topic "Generation" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

6. Each Delete data line defines an array of one or more elements to be deleted. Nonexistent elements may be included in the array.

   See Topic "Deletion" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

PATTERN Data Block

This data block defines one or more joint Patterns. Each Pattern consists of a set of numeric values, one for each joint in the structure. These Patterns can be used to assign properties to the joints in the MASS and SPRING Data Blocks (pages 73 and 68, respectively), or to assign loads to the joints and elements in the LOAD Data Block (page 134).

Skip this data block if there are no Patterns to be defined. Otherwise, prepare data according to the format described below.

For More Information

See Chapter "Joint Patterns" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

PATTERN	Separator
CSYS=	Coordinate System Data Lines
NAME=	Name Data Lines
CSYS=	Coordinate System Data Lines
ADD=	Add Data Lines
REM=	Remove Data Lines
LMAP=	Linear Mapping Data Lines
FMAP=	Frontal Mapping Data Lines
EMAP=	Edge Mapping Data Lines

Begin the data block with the **PATTERN separator**.

For each Pattern to be defined, prepare a data set beginning with a Name data line, and followed by as many Coordinate System, Add, Remove, and Mapping data lines as necessary. Data lines are processed in the order that they are supplied in the input data file.

Each **Name data line** begins the definition of a new Pattern and initializes the Pattern values to zero at every joint in the structure.

Each **Add data line** adds specified numeric values to the current Pattern values for a single joint or an array of joints. Each **Remove data line** resets the Pattern values to zero for a single joint or an array of joints.

Each **Mapping data line** interpolates or extrapolates Pattern values from some of the joints to the remaining joints in an array. Mapping is *not* additive, but overwrites previous Pattern values. Several types of mapping are provided: Linear Mapping, Frontal Mapping, and Edge Mapping.

Each **Coordinate System data line** defines the fixed coordinate system used to define the Pattern gradient and zero point on all subsequent Add data lines. This fixed coordinate system is in effect until the next Coordinate System data line is encountered. If this data line is omitted, the global coordinate system is assumed (e.g., CSYS=0).

Data Line Formats

Coordinate System Data Line

CSYS=**csys**

Name Data Line

NAME=**name**

Add Data Line — Value for a Single Joint

ADD=**j0** V=**v0**

Add Data Line — Values for a Joint Array

ADD=**j0, j1, ji1...** V=**v0, v1...** RATIO=**ratio1...**

Add Data Line — Gradient (Hydrostatic)

ADD=**j0, j1, ji1...** VX=**vx** VY=**vy** VZ=**vz** X=**x** Y=**y** Z=**z**
SETZERO=**setzero**

Remove Data Line — Single Joint

REM=**j0**

Remove Data Line — Joint Array

REM=**j0, j1, ji1...**

Linear Mapping Data Line

LMAP=**j0, j1, ji1...** RATIO=**ratio1...**

Frontal Mapping Data Line

FMAP=**j0, j1, ji1, j2, ji2...**

Edge Mapping Data Line

EMAP=**j0, j1, ji1, j2, ji2...**

Examples

(1) Define a Pattern for hydrostatic pressure caused by a fluid with a weight density of 62.4 lb/ft^3, with gravity acting in the –Z direction, and the free surface at elevation 50 ft:

```
PATTERN
 NAME=HYDRO
  ADD=1,10,1,51,10 VZ=-62.4 Z=50 SETZERO=NEG
```

(2) Define a Pattern that interpolates Pattern values from the corners of a quadrilateral region, but has zero values at four joints in the interior:

```
PATTERN
 NAME=QUAD2
  ADD=1  V=0
  ADD=10 V=8
  ADD=51 V=1
  ADD=60 V=6
  LMAP=1,10,1,51,10
  REM=4,5,1,14,10
```

Description of Variables

Variable	Note	Default	Description
Name Data Line			
name	(1, 2)		Label of a Pattern being defined
Coordinate System Data Line			
csys	(1, 4)	[pv(0)]	Fixed coordinate system used to define gradients on subsequent Add data lines: = 0: Global coordinate system ≠ 0: Alternate Coordinate System label
Add Data Line			
j0...	(1, 3, 5)		Labels and label increments for a single joint or an array of joints to which Pattern values are being added
v0...	(1, 5)	[0]	Pattern values being added to joints **j0, j1...**
ratio1...	(1)	[1]	For unequal increments in added Pattern values, ratio of last increment to first increment along joint array axes 1, 2 and 3, respectively, up to the dimension of the array
vx, vy, vz	(1, 4)	[0]	Gradient of Pattern values, in fixed coordinate system **csys**
x, y, z	(1, 4)	[0]	Coordinates of any point on the zero-value surface for Pattern values defined by a gradient, in fixed coordinate system **csys** [L units]
setzero	(1, 5)	[NO]	Key to indicate whether to set to zero any Pattern values defined by a gradient: = NEG: Set negative values to zero (e.g., hydrostatic pressure) = POS: Set positive values to zero = NO: Do not set any values to zero

Variable	Note	Default	Description
Remove Data Line			
j0...	(1, 3, 6)		Labels and label increments for a single joint or an array of joints at which Pattern values are being reset to zero
Linear Mapping Data Line			
j0, j1, ji1...	(1, 3)		Labels and label increments for an array of joints having one, two or three dimensions
ratio1...	(1)	[1]	For non-uniform interpolation, ratio of last increment to first increment along joint array axes 1, 2 and 3, respectively, up to the dimension of the array
Frontal Mapping Data Line			
j0, j1, ji1, j2, ji2...	(1, 3)		Labels and label increments for an array of joints having two or three dimensions
Edge Mapping Data Line			
j0, j1, ji1, j2, ji2...	(1, 3)		Labels and label increments for an array of joints having two or three dimensions

Notes

1. See Chapter “Joint Patterns” of the *SAP2000 Analysis Reference*.
2. Each Name data line begins the definition of a new Pattern. Pattern labels do not have to be consecutive and may be supplied in any order. Pattern labels may not be repeated in the data block.

   See Topic “Labels” in Chapter “Labels, Arrays, and Generation” of the *SAP2000 Analysis Reference*.
3. See Topic “Regular Array Specification” (page 15) in this chapter.
4. Pattern-value gradients and zero-surface coordinates are taken in the most recently specified coordinate system **csys**. If **csys** is zero, the global system is

used. Otherwise **csys** refers to an Alternate Coordinate System defined in the COORDINATE Data Block (page 32). If no **csys** is specified, the global system is used.

5. Each Add data line may refer to a single joint, **j0**, or to an array of joints, **j0, j1, ji1...** having one, two or three dimensions.

   When a constant value is being added to an array of joints, it is not necessary to repeat that value on the data line; e.g., for a two-dimensional array of joints, specifying V=10 is the same as specifying V=10,10,10.

   The parameter **setzero** only affects Pattern-values defined by a gradient on the current data line. It does not affect the previous values that are present at any joint to which Pattern values are being added.

6. Each Remove data line may refer to a single joint, **j0**, or to an array of joints **j0, j1, ji1...** having one, two or three dimensions.

SPRING Data Block

This data block defines all of the joint springs that support the structure. Displacements at the grounded end of the springs (e.g., support settlement) may be specified in the LOAD Data Block (page 134).

Skip this data block if there are no joint springs to be defined. Otherwise, prepare data according to the format described below.

For More Information

See Topic “Springs” in Chapter “Joints and Degrees of Freedom” of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

SPRING	Separator
CSYS=	Coordinate System Data Line
ADD=	Add Data Lines
REM=	Remove Data Lines

Begin the data block with the **SPRING separator**.

Follow this with as many Coordinate System, Add and Remove data lines as necessary to define all joint spring supports in the model. The data is processed in the order it is supplied in the input data file.

Each **Coordinate System data line** defines the fixed coordinate system used by all subsequent Add data lines until the next Coordinate System data line is encountered. This coordinate system does not affect springs specified in joint local coordinate systems. If this data line is omitted, the global coordinate system is assumed (e.g., CSYS=0).

Each **Add data line** adds spring supports to an array of joints. Each **Remove data line** removes *all* spring supports from an array of joints.

Data Line Formats

Coordinate System Data Line

CSYS=**csys**

Add Data Line — Uncoupled Springs in Fixed Coordinates

ADD=**j0, j1, ji1...** UX=**ux** UY=**uy** UZ=**uz** RX=**rx** RY=**ry** RZ=**rz** PAT=**pat**

Add Data Line — Uncoupled Springs in Joint Local Coordinates

ADD=**j0, j1, ji1...** U1=**u1** U2=**u2** U3=**u3** R1=**r1** R2=**r2** R3=**r3** PAT=**pat**

Add Data Line — Coupled Springs in Fixed Coordinates

ADD=**j0, j1, ji1...** URXYZ=**ux, uxuy, uy, uxuz, uyuz, uz, uxrx, uyrx, uzrx, rx, uxry, uyry, uzry, rxry, ry, uxrz, uyrz, uzrz, rxrz, ryrz, rz** PAT=**pat**

Add Data Line — Coupled Springs in Joint Local Coordinates

ADD=**j0, j1, ji1...** UR123=**u1, u1u2, u2, u1u3, u2u3, u3, u1r1, u2r1, u3r1, r1, u1r2, u2r2, u3r2, r1r2, r2, u1r3, u2r3, u3r3, r1r3, r2r3, r3** PAT=**pat**

Remove Data Line

REM=**j0, j1, ji1...**

Examples

(1) A flat plate in the X-Y plane is supported transversely (in the Z direction) by an elastic foundation. This can be represented by spring stiffness coefficients that are proportional to the tributary area surrounding each joint. Thus the ratio of the spring constants for joint at the corners, on the sides, and in the interior, respectively, is 1:2:4. This can be specified as:

```
SPRING
 ADD=1, 5,1,21,5 UZ=1
 ADD=2, 4,1,22,5 UZ=1
 ADD=6,10,1,16,5 UZ=1
 ADD=7, 9,1,17,5 UZ=1
```

(2) For the same plate, discrete rotational springs with linearly varying stiffness are placed along all four edges. The change in stiffness is the same along parallel edges. A Pattern is defined that interpolates values over the whole plate from three corner values, then the interior values are set back to zero. The value at joint 25 will be 0.6.

```
PATTERN
 NAME=SPAT
  ADD=1,5,1,21,5 V=1,2,5
  REM=7,9,1,17,5
SPRING
 ADD=1,5,1,21,5 RX=0.1 PAT=SPAT
```

Description of Variables

Variable	Note	Default	Description
Coordinate System Data Line			
csys	(3)	[pv(0)]	Fixed coordinate system for subsequent Add data lines: = 0: Global coordinate system ≠ 0: Alternate coordinate system label
Add Data Line			
j0, j1, ji1...	(2, 4)		Labels and label increments for an array of one or more joints to which springs are being added
ux, uy, uz	(1, 4)	[0]	Uncoupled spring force per unit translation in fixed coordinate system **csys** [F/L units]
u1, u2, u3	(1, 4)	[0]	Uncoupled spring force per unit translation in each joint local coordinate system [F/L units]
rx, ry, rz	(1, 4)	[0]	Uncoupled spring moment per unit rotation in fixed coordinate system **csys** [FL/rad units]

Variable	Note	Default	Description
r1, r2, r3	(1, 4)	[0]	Uncoupled spring moment per unit rotation in each joint local coordinate system [FL/rad units]
uxuy, uxuz, uyuz	(1, 4)	[0]	Coupled spring force per unit translation in fixed coordinate system **csys** [F/L units]
u1u2, u1u3, u2u3	(1, 4)	[0]	Coupled spring force per unit translation in each joint local coordinate system [F/L units]
uxrx, uyrx, ... uzrz	(1, 4)	[0]	Coupled spring force per unit rotation, or moment per unit translation, in fixed coordinate system **csys** [F/rad = FL/L units]
u1r1, u2r1, ... u3r3	(1, 4)	[0]	Coupled spring force per unit rotation, or moment per unit translation, in each joint local coordinate system [F/rad = FL/L units]
rxry, rxrz, ryrz	(1, 4)	[0]	Coupled spring moment per unit rotation in fixed coordinate system **csys** [FL/rad units]
r1r2, r1r3, r2r3	(1, 4)	[0]	Coupled spring moment per unit rotation in each joint local coordinate system [FL/rad units]
pat	(4, 5)		Label of a Pattern of scale factors multiplying spring stiffness coefficients on this data line. If omitted, a unit scale factor is assumed at every joint
Remove Data Line			
j0, j1, ji1...	(2, 6)		Labels and label increments for an array of one or more joints from which *all* previously-added springs are removed

Notes

1. See Topic “Springs” in Chapter “Joints and Degrees of Freedom” of the *SAP2000 Analysis Reference*.
2. See Topic “Regular Array Specification” (page 15) in this chapter.

3. Translations, rotations, forces and moments for all springs that are specified in fixed coordinates are taken in the most recent coordinate system **csys** specified. If csys is "0", the global system is used. Otherwise **csys** refers to an alternate coordinate system defined in the COORDINATE Data Block (page 32). If no **csys** is specified, the global system is used.

4. Initially, all spring coefficients at all joints are zero.

   Each Add data line may refer to a single joint **j0**, or an array of joints **j0, j1, ji1...** having one, two or three dimensions. For each joint, the specified spring stiffness coefficients are added to the current values at the joint in the following manner:

   - If a Pattern label **pat** is given, then all spring stiffness coefficients on the data line are multiplied by the Pattern value at that joint;
   - Spring stiffness coefficients given in joint local coordinates are added directly to the current values at that joint;
   - Spring stiffness coefficients given in fixed coordinates are transformed to the joint's local coordinate system and then added to the current values.

5. Patterns can be used to provide scale factors for spring stiffness coefficients that vary from joint to joint. Each Pattern applies to all spring coefficients on a given Add data line. If different spring coefficients vary according to different Patterns, they should be given on separate Add data lines.

6. Each Remove data line may refer to a single joint **j0**, or an array of joints **j0, j1, ji1...** having one, two or three dimensions. For each joint, the entire 6x6 spring stiffness matrix is set back to zero, overwriting the effect of any previous Add data lines.

MASS Data Block

This data block defines all of the joint masses and mass moments of inertia in the structure.

Skip this data block if there are no joint masses to be defined. Otherwise, prepare data according to the format described below.

For More Information

See Topic "Masses" in Chapter "Joints and Degrees of Freedom" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

MASS	Separator
ADD=	Add Data Lines
REM=	Remove Data Lines

Begin the data block with the **MASS separator**.

Follow this with as many Add and Remove data lines as necessary to define all joint masses and mass moments of inertia in the model. The data is processed in the order it is supplied in the input data file.

Each **Add data line** adds mass values to an array of joints. Each **Remove data line** removes *all* joint mass values from an array of joints (the mass contributed by the elements is not removed).

Data Line Formats

Add Data Line — in Global Coordinates

ADD=**j0, j1, ji1...** UX=**ux** UY=**uy** UZ=**uz** RX=**rx** RY=**ry** RZ=**rz** PAT=**pat**

Add Data Line — in Joint Local Coordinates

ADD=**j0, j1, ji1...** U1=**u1** U2=**u2** U3=**u3** R1=**r1** R2=**r2** R3=**r3** PAT=**pat**

Remove Data Line

REM=**j0, j1, ji1...**

Example

(1) Twenty-four FRAME elements are used represent the length of a bridge superstructure running in the global X direction. The torsional mass moment of inertia is considered to be important for this structure. The program automatically generates translational masses for the FRAME element, but not rotational inertia. The total torsional mass moment of inertia for each element is 10. Hence the torsional inertia is given in the MASS Data Block as follows:

```
MASS
 ADD=1,25,1 RX=5
 ADD=2,24,1 RX=5
```

Description of Variables

Variable	Note	Default	Description
Add Data Line			
j0, j1, ji1...	(2, 3)		Labels and label increments for an array of one or more joints to which mass values are being added
ux, uy, uz	(1, 3)	[0]	Translational mass in the global coordinate system [M units]
u1, u2, u3	(1, 3)	[0]	Translational mass in each joint local coordinate system [M units]
rx, ry, rz	(1, 3)	[0]	Rotational mass moment of inertia in the global coordinate system [ML^2 units]
r1, r2, r3	(1, 3)	[0]	Rotational mass moment of inertia in each joint local coordinate system [ML^2 units]
pat	(3, 4)		Label of a Pattern of scale factors multiplying mass values on this data line. If omitted, a unit scale factor is assumed at every joint
Remove Data Line			
j0, j1, ji1...	(2, 5)		Labels and label increments for an array of one or more joints from which *all* previously-added joint mass values are removed

Notes

1. See Topic “Masses” in Chapter “Joints and Degrees of Freedom” of the *SAP2000 Analysis Reference*.

2. See Topic “Regular Array Specification” (page 15) in this chapter.

3. Initially, all joint mass values at all joints are zero.

   Each Add data line may refer to a single joint **j0**, or an array of joints **j0, j1, ji1...** having one, two or three dimensions. For each joint, the specified mass values are added to the current values at the joint in the following manner:

   - If a Pattern label **pat** is given, then all mass values on the data line are multiplied by the Pattern value at that joint;
   - Mass values given in joint local coordinates are added directly to the current values at that joint;
   - Mass values given in global coordinates are transformed to the joint’s local coordinate system and then added to the current values; any coupling terms that may be generated are discarded.

   Mass values must be in consistent mass units (W/g) and mass moments of inertia must be in WL^2/g units. Here W is weight, L is length, and g is the acceleration due to gravity.

4. Patterns can be used to provide scale factors for mass values that vary from joint to joint. Each Pattern applies to all mass values on a given Add data line. If different mass values vary according to different Patterns, they should be given on separate Add data lines.

5. Each Remove data line may refer to a single joint **j0**, or an array of joints **j0, j1, ji1...** having one, two or three dimensions. For each joint, all six mass values are set back to zero, overwriting the effect of any previous Add data lines. This, however, only affects mass values defined in this data block, not the masses obtained from the elements.

MATERIAL Data Block

This data block defines the Material properties used by the Frame, Shell, Plane, Asolid and Solid elements. A given material defined in this data block may be used by more than one element type. For the Frame and Shell elements, the Materials are referenced indirectly through the FRAME SECTION and SHELL SECTION Data Blocks (pages 81 and 87, respectively).

This data block is mandatory if there are any Frame, Shell, Plane, Asolid, or Solid elements in the structure. Prepare data according to the format described below.

For More Information

See Chapter "Material Properties" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

MATERIAL	Separator
NAME=	Name Data Lines
E=	Property Data Lines

Begin the data block with the **MATERIAL separator**.

Follow this with as many Name and Property data lines as necessary to define all the Materials.

Each **Name data line** begins the definition of a new Material and defines the type of material and the temperature-independent properties.

Each **Property data line** specifies temperature-dependent material properties at a fixed material temperature. Only one of these data lines is needed in the case where properties are assumed not to vary with temperature.

Data Line Formats

Name Data Line

NAME=**name** TYPE=**type** IDES=**ides** M=**m** W=**w**

Property Data Line — Isotropic Material

T=**t** E=**e1** U=**u12** A=**a1**

Property Data Line — Orthotropic Material

T=**t** E=**e1, e2, e3** G=**g12, g13, g23** U=**u12, u13, u23** A=**a1, a2, a3**

Property Data Line — Anisotropic Material

T=**t** E=**e1, e2, e3** G=**g12, g13, g23** U=**u12, u13, u23, u14, u24, u34, u15, u25, u35, u45, u16, u26, u36, u46, u56** A=**a1, a2, a3, a12, a13, a23**

Example

(1) Two temperature-independent Materials are defined, one isotropic and the other orthotropic:

```
MATERIAL
 NAME=STEEL TYPE=ISO M=0.49/1728/386.4 W=0.49/1728
IDES=S
  E=29500 U=0.3 A=6.5E-6
 NAME=GRATE TYPE=ORTHO M=0.05/1728/386.4 W=0.05/1728
  E=3000,1000,100 U=0,0,0 G=50,1500,500
```

Description of Variables

Variable	Note	Default	Description
Name Data Line			
name	(1, 2)		Label of a material being defined
type	(1, 3)	[ISO]	Type of Material: = ISO: Isotropic = ORTHO: Orthotropic = ANISO: Anisotropic
ides	(1)	[N]	Design-type indicator: = S: Steel = C: Concrete = N: Neither (no design)
m	(1, 4)	[0]	Mass per unit volume [M/L^3 units]
w	(1, 4)	[0]	Weight per unit volume [F/L^3 units]
Property Data Line			
t	(1, 4, 5)	[0]	Material temperature associated with properties specified on this data line [K units]
e1, e2, e3	(1, 3, 4)	[0]	Moduli of elasticity in the Material 1, 2, and 3 directions, respectively. Must be positive [F/L^2 units]
g12, g13, g23	(1, 3, 4)	[0]	Shear moduli in the Material 1-2, 1-3, and 2-3 planes, respectively. Must be positive [F/L^2 units]
u12, u13, u23	(1, 3)	[0]	Standard Poisson's ratios
u14, u24..., u56	(1, 3)	[0]	Shear and coupling Poisson's ratios
a1, a2, a3	(1, 3, 4)	[0]	Coefficients of thermal expansion in the Material 1, 2, and 3 directions, respectively [1/K units]

Variable	Note	Default	Description
a12, a13, a23	(1, 3, 4)	[0]	Coefficients of thermal shear in the Material 1-2, 1-3, and 2-3 planes, respectively [1/K units]

Notes

1. See Chapter “Material Properties” of the *SAP2000 Analysis Reference*.
2. Each Name data line defines a new Material. Material labels do not have to be consecutive and may be supplied in any order. Material labels may not be repeated in the data block.

   See Topic “Labels” in Chapter “Labels, Arrays, and Generation” of the *SAP2000 Analysis Reference*.
3. The format of the Property data line differs according to whether the Material is Isotropic, Orthotropic, or Anisotropic.
4. All properties must be given in force, length, mass, and/or temperature units that are consistent with the rest of the data file
5. The values of temperature **t** on consecutive Property data lines for any particular Material must be in numerically ascending order. Only a single data line is needed, at an arbitrary temperature, if the properties are not actually temperature dependent.

FRAME SECTION Data Block

This data block defines the Frame Section properties associated with the three-dimensional Frame elements that are present in the structure. For each Section that is defined, various geometric properties are specified, and a Material is selected from the MATERIAL Data Block (page 77). Any Section defined in this data block may be assigned to one or more elements defined in the FRAME Data Block (page 96).

Skip this data block if there are no Frame elements in the model. Otherwise, prepare data according to the format described below.

For More Information

See Topic "Section Properties" in Chapter "The Frame Element" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

FRAME SECTION	Separator
FILE=	File Data Lines
NAME=	Name Data Lines
SEC=	Additional Segment Data Lines

Begin the data block with the **FRAME SECTION separator**.

Follow this with as many File and Name data lines as necessary to define all of the Sections used by the Frame elements.

Each **File data line** defines the name of a Section property database file that may be used by subsequent Name data lines to extract geometric section properties. This database file will be used until the next File data line is encountered.

Each **Name data line** defines all of the properties for a single prismatic Section, or for the first segment of a non-prismatic Section. For non-prismatic Sections, the Name data line may be followed by zero or more Additional Segment data lines.

Data Line Formats

File Data Line

FILE=**filename**

Name Data Line — Prismatic Section

The general format is as follows:

NAME=**name** TYPE=PRISM MAT=**mat** A=**a** J=**j** I=**i33, i22** AS=**as2, as3** MPL=**mpl** WPL=**wpl** SH=**sh** T=**...**

The specific form of the entry "T=**...**" depends on the value of the shape type **sh** as follows. Note that the entry "T=**...**" is *not* permitted for the General section or the Database section:

Name Data Line — Prismatic Rectangular Section

NAME=**name** TYPE=PRISM ... SH=R T=**t3, t2**

Name Data Line — Prismatic Pipe Section

NAME=**name** TYPE=PRISM ... SH=P T=**t3, tw**

Name Data Line — Prismatic Solid Circular Section

NAME=**name** TYPE=PRISM ... SH=P T=**t3**

Name Data Line — Prismatic Box Section

NAME=**name** TYPE=PRISM ... SH=B T=**t3, t2, tf, tw**

Name Data Line — Prismatic I-Section

NAME=**name** TYPE=PRISM ... SH=I T=**t3, t2t, tft, tw, t2b, tfb**

Name Data Line — Prismatic Channel Section

NAME=**name** TYPE=PRISM ... SH=C T=**t3, t2, tf, tw**

Name Data Line — Prismatic T-Section

NAME=**name** TYPE=PRISM ... SH=T T=**t3, t2, tf, tw**

Name Data Line — Prismatic Angle Section

NAME=**name** TYPE=PRISM ... SH=L T=**t3, t2, tf, tw**

Name Data Line — Prismatic Double-Angle Section

NAME=**name** TYPE=PRISM ... SH=2L T=**t3, t2, tf, tw, dis**

Name Data Line — Prismatic General Section

NAME=**name** TYPE=PRISM ... SH=G

Name Data Line — Prismatic Database Section

NAME=**name** TYPE=PRISM ... SH=**sh**

Name Data Line — Non-prismatic Section, Variable Length Segment

NAME=**name** TYPE=NONPR SEC=**seci, secj** EIVAR=**eivar33, eivar22** VL=**vl**

Name Data Line — Non-prismatic Section, Fixed Length Segment

NAME=**name** TYPE=NONPR SEC=**seci, secj** EIVAR=**eivar33, eivar22** L=**l**

Additional Segment Data Line — Non-prismatic Section, Variable Length Segment

SEC=**seci, secj** EIVAR=**eivar33, eivar22** VL=**vl**

Additional Segment Data Line — Non-prismatic Section, Fixed Length Segment

SEC=**seci, secj** EIVAR=**eivar33, eivar22** L=**l**

Description of Variables

Variable	Note	Default	Description
File Data Line			
filename	(1, 3)		Name of a Section Property database file
Name Data Line			
name	(1, 2)		Label of a Section being defined
Name Data Line — Prismatic Section			
mat	(1, 4)	[pv]	Label of a Material assigned to this Section
sh	(1)	[G]	Shape type: = G: General section = R: Rectangular section = P: Pipe section or Solid circular section = B: Box section = I: I-section = C: Channel section = T: T-section = L: Angle section = 2L: Double-angle section Otherwise: Shape name in the database file
t3	(1)		Section depth in 2 direction [L units]
t2	(1)		Section width in 3 direction [L units]
tf	(1)		Flange thickness in 2 direction [L units]
tw	(1)		Web thickness in 3 direction, or wall thickness for Pipe section (**tw**=0 or omitted for a Solid circular section) [L units]
dis	(1)		Spacing between angles in Double-angle section [L units]
t2t	(1)		Top flange width in 3 direction for I-section [L units]

Variable	Note	Default	Description
t2b	(1)	[**t2t**]	Bottom flange width in 3 direction for I-section [L units]
tft	(1)		Top flange thickness in 2 direction for I-section [L units]
tfb	(1)	[**tft**]	Bottom flange thickness in 2 direction for I-section [L units]
a	(1, 5)		Cross-section (axial) area [L^2 units]
j	(1, 5)		Torsional constant [L^4 units]
i33	(1, 5)		Moment of inertia about 3 axis [L^4 units]
i22	(1, 5)		Moment of inertia about 2 axis [L^4 units]
as2	(1, 5)		Shear area in 2 direction [L^2 units]
as3	(1, 5)		Shear area in 3 direction [L^2 units]
mpl	(1)	[0]	Additional mass per unit length [M/L units]
wpl	(1)	[0]	Additional weight per unit length [F/L units]
Name and Additional Segment Data Lines — Non-prismatic Section			
seci	(1, 6)		Label of prismatic starting Section
secj	(1, 6)	[**seci**]	Label of prismatic ending Section
eivar33	(1)	[pv(2)]	Variation of **i33** · **e1** along the segment: = 1: Linear = 2: Parabolic = 3: Cubic
eivar22	(1)	[pv(1)]	Variation of **i22** · **e1** along the segment: = 1: Linear = 2: Parabolic = 3: Cubic
vl	(1, 7)	[1]	Variable segment length: **vl** > 0
l	(1, 7)		Fixed segment length [L units]

Notes

1. See Topic “Section Properties” in Chapter “The Frame Element” of the *SAP2000 Analysis Reference*.

2. Each Name data line defines a new Section. Section labels do not have to be consecutive and may be supplied in any order. Section labels may not be repeated in the data block.

   If the type is omitted, then a prismatic Section (TYPE=PRISM) is assumed.

   See Topic “Labels” in Chapter “Labels, Arrays, and Generation” of the *SAP2000 Analysis Reference*.

3. Each File data line specifies the Section property database file, filename, to be used by all subsequent Name data lines until the next File data line is encountered. If no File data line is given, the SAP2000 database file SECTIONS.PRO is used. The filename must be a standard Windows filename including extension; drive and directory names are not permitted.

4. The label **mat** refers to a Material defined in the MATERIAL Data Block (page 77). A Material must be specified for the first Name data line.

5. The defaults for **a**, **j**, **i33**, **i22**, **as2**, and **as3** are zero if **sh**=G. The defaults are the values recovered for database shapes, or the values automatically calculated from the Section dimensions for other shape types. If an explicit value for **a**, **j**, **i33**, **i22**, **as2** or **as3** is provided in the latter two cases, it overwrites the corresponding recovered or calculated property value.

6. Each non-prismatic Section may have one or more segments. For a Section with a single segment, only the Name data line is required. Use Additional Segment data lines if the Section has more that one segment.

   The labels **seci** and **secj** refer to prismatic Sections that were previously defined in this data block. If **seci** and **secj** are the same, only **seci** need be specified.

7. For each segment, you may specify either a fixed length or a variable length, but not both. If neither is specified, the default is **vl** = 1.

SHELL SECTION Data Block

This data block defines the Shell Section properties associated with the Shell elements that are present in the structure. For each Section that is defined, the thickness and type of behavior are specified, and a Material is selected from the MATERIAL Data Block (page 77). Any Section defined in this data block may be assigned to one or more elements defined in the SHELL Data Block (page 101).

Skip this data block if there are no Shell elements in the model. Otherwise, prepare data according to the format described below.

For More Information

See Topic "Section Properties" in Chapter "The Shell Element" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

SHELL SECTION	Separator
NAME=	Name Data Lines

Begin the data block with the **SHELL SECTION separator**.

Follow this with as many Name data lines as necessary to define all of the Sections used by the Shell elements. Each **Name data line** defines all the properties for a single Shell Section.

Data Line Format

Name Data Line

NAME=**name** TYPE=**type, thicktype** MAT=**mat** MATANG=**a** TH=**th** THB=**thb**

Description of Variables

Variable	Note	Default	Description
Name Data Line			
type	(1, 2)	[pv (SHELL)]	Element type: = SHELL: Shell (Membrane plus Plate) = MEMBR: Membrane behavior only = PLATE: Plate-bending behavior only
thicktype	(1, 2)	[pv (THICK)]	Thickness formulation type: = THICK: Include transverse shearing deformations = THIN: Neglect transverse shearing deformations
mat	(1, 3)	[pv]	Label of Material for element
a	(1)	[pv(0)]	Material angle [deg units]
th	(1)	[pv(1)]	Thickness used for membrane behavior, self-weight, and mass [L units]
thb	(1)	[pv(**th**)]	Thickness used for plate-bending behavior. The default value is reset to **th** whenever **th** is specified [L units]

Notes

1. See Topic “Section Properties” in Chapter “The Shell Element” of the *SAP2000 Analysis Reference*.

2. Each Name data line defines a new Section. Section labels do not have to be consecutive and may be supplied in any order. Section labels may not be repeated in the data block.

   If the types are omitted, the previous values (or SHELL and THICK) are assumed.

3. The label **mat** refers to a Material defined in the MATERIAL Data Block (page 77). A Material must be specified for the first Name data line.

NLPROP Data Block

This data block defines the structural properties associated with the Nllink elements that are present in the structure. For each Nlprop that is defined, various linear and nonlinear force-deformation relationships are specified. Any Nlprop defined in this data block may be referenced by one or more elements defined in the NLLINK Data Block (page 120).

Skip this data block if there are no Nllink elements in the model. Otherwise, prepare data according to the format described below.

For More Information

See Topic “Nlprop Properties” in Chapter “The Nllink Element” of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

NLPROP	Separator
NAME=	Name Data Lines
DOF=	Property Data Lines

Begin the data block with the **NLPROP separator**.

Follow this with as many Name and Property data lines as necessary to define all of the Nlprops used by the Nllink elements.

Each **Name data line** begins the definition of a new Nlprop and defines the type of nonlinear property and the mass and weight. It may be followed by up to six Property data lines

Each **Property data line** specifies the effective stiffness, the effective damping, and the nonlinear force-deformation relationship for one of the six internal deformations. Data lines are not needed for internal deformations having zero properties.

Data Line Formats

Name Data Line

NAME=**name** TYPE=**type** M=**m** MR1=**mr1** MR2=**mr2** MR3=**mr3** W=**w**

Property Data Line — Linear Property for Non-shear Deformations

DOF=**dof1** KE=**ke** CE=**ce**

Property Data Line — Linear Property for Shear Deformations

DOF=**dof2** KE=**ke** CE=**ce** DJ=**dj**

Property Data Line — Damper Property for Non-shear Deformations

DOF=**dof1** KE=**ke** CE=**ce** K=**k** C=**c** CEXP=**cexp**

Property Data Line — Damper Property for Shear Deformations

DOF=**dof2** KE=**ke** CE=**ce** K=**k** C=**c** CEXP=**cexp** DJ=**dj**

Property Data Line — Gap and Hook Properties for Non-shear Deformations

DOF=**dof1** KE=**ke** CE=**ce** K=**k** OPEN=**open**

Property Data Line — Gap and Hook Properties for Shear Deformations

DOF=**dof2** KE=**ke** CE=**ce** K=**k** OPEN=**open** DJ=**dj**

Property Data Line — Plastic1 Property for Non-shear Deformations

DOF=**dof1** KE=**ke** CE=**ce** K=**k** YIELD=**yield** RATIO=**ratio** EXP=**exp**

Property Data Line — Plastic1 Property for Shear Deformations

DOF=**dof2** KE=**ke** CE=**ce** K=**k** YIELD=**yield** RATIO=**ratio** EXP=**exp** DJ=**dj**

Property Data Line — Isolator1 Property for Non-shear Deformations

DOF=**dof1** KE=**ke** CE=**ce**

Property Data Line — Isolator1 Property for Shear Deformations

DOF=**dof2** KE=**ke** CE=**ce** K=**k** YIELD=**yield** RATIO=**ratio** DJ=**dj**

Property Data Line — Isolator2 Property for Axial Deformations

DOF=U1 KE=**ke** CE=**ce** K=**k**

Property Data Line — Isolator2 Property for Rotational Deformations

DOF=**dof1** KE=**ke** CE=**ce**

Property Data Line — Isolator2 Property for Shear Deformations

DOF=**dof3** KE=**ke** CE=**ce** K=**k** FRICT=**slow, fast** RATE=**rate** RADIUS=**radius** DJ=**dj**

Description of Variables

Variable	Note	Default	Description
Name Data Line			
name	(1, 2)		Label of an Nlprop being defined
type	(1, 3)		Type of nonlinear behavior: = Damper: Nonlinear viscous damper = Gap: No positive force or moment = Hook: No negative force or moment = Plastic1: Uniaxial plasticity = Isolator1: Biaxial shear plasticity = Isolator2: Biaxial shear friction-pendulum
m	(1)	[0]	Total mass of element [M units]
mr1, mr2, mr3	(1)	[0]	Total element rotational mass moments of inertia about the element local 1, 2, and 3 axes, respectively [ML^2 units]
w	(1)	[0]	Total weight of element [F units]

Variable	Note	Default	Description
Property Data Line			
dof1	(1)		Non-shear deformation: = U1: Axial = R1: Torsion = R2: Pure-bending in 1-3 plane = R3: Pure-bending in 1-2 plane
dof2	(1)		Shear deformation: = U2: Shear in 1-2 plane = U3: Shear in 1-3 plane
dof3	(1)		Non-shear deformation: = R1: Torsion = R2: Pure-bending in 1-3 plane = R3: Pure-bending in 1-2 plane
ke	(1, 8)		Linear effective stiffness for linear degress of freedom and for linear analyses of nonlinear degrees of freedom. *Required* on all Property data lines: **ke** ≥ 0 [F/L or FL/rad units]
ce	(1)	[0]	Linear effective-damping coefficient for linear analyses: **ce** ≥ 0 [FT/L or FLT/rad units]
k	(1, 8)		Stiffness for nonlinear force-deformation relationship. *Required* on all nonlinear Property data lines. *Not permitted* on linear Property data lines. **k** ≥ 0 [F/L or FL/rad units]
dj	(1)	[0]	Distance from joint **j** to shear spring: **dj** ≥ 0 [L units]
Property Data Line — Nonlinear Damper Property			
c	(1, 4)		Nonlinear damping coefficient: **c** > 0 [$F(T/L)^{\mathbf{cexp}}$ or $FL(T/rad)^{\mathbf{cexp}}$ units]
cexp	(1)	[1]	Nonlinear damping exponent: **cexp** > 0

Variable	Note	Default	Description
Property Data Line — Nonlinear Gap and Hook Properties			
open	(1, 5)		Initial gap or hook opening: **open** ≥ 0 [L units]
Property Data Line — Plastic1 and Isolator1 Properties			
yield	(1, 6)		Yield force or moment: **yield** > 0 [F or FL units]
ratio	(1)	[0]	Ratio of post-yield stiffness to elastic stiffness (**k**): 0 ≤ **ratio** < 1
Property Data Line — Plastic1 Property			
exp	(1)	[2]	Yielding exponent: **exp** ≥ 1
Property Data Line — Isolator2 Property			
slow	(1, 7)		Friction coefficient at zero velocity: **slow** > 0
fast	(1, 7)	[**slow**]	Friction coefficient at fast velocity: **fast** > 0
rate	(1)	[0]	Inverse of the characteristic sliding velocity: **rate** ≥ 0 [T/L units]
radius	(1)	[0]	Radius of sliding contact surfaces [L units]: = 0:Flat, infinite radius > 0:Curved, finite radius

Notes

1. See Topic “Nlprop Properties” in Chapter “The Nllink Element” of the *SAP2000 Analysis Reference*.

2. Each Name data line defines a new Nlprop. Nlprop labels do not have to be consecutive and may be supplied in any order. Nlprop labels may not be repeated in the data block.

   See Topic “Labels” in Chapter “Labels, Arrays, and Generation” of the *SAP2000 Analysis Reference*.

3. The type of Nlprop must be specified on the Name data line. If strictly linear behavior is desired, any type may be specified and all corresponding nonlinear parameters omitted from the Property data lines.

4. There is no default value for **c**. If present, damping behavior is modeled for the specified deformation during nonlinear time-history analysis. If absent, a linear spring of stiffness **ke** is assumed.

5. There is no default value for **open**. If present, Gap or Hook behavior is modeled for the specified deformation during nonlinear time-history analysis. If absent, a linear spring of stiffness **ke** is assumed.

6. There is no default value for **yield**. If present, plasticity behavior is modeled for the specified deformation during nonlinear time-history analysis. If absent, a linear spring of stiffness **ke** is assumed.

7. There is no default value for **slow**. If present, frictional behavior is modeled for the specified deformation during nonlinear time-history analysis. If **slow** and **radius** are both absent, a linear spring of stiffness **ke** is assumed. Only **slow** need be specified if the friction coefficient is independent of velocity.

8. Linear effective stiffness **ke** must *always* be specified, even if zero. For linear degrees of freedom, only **ke**, **ce**, and **dj** (for shear deformations) may be specified.

   Stiffness **k** is only used for nonlinear deformational degrees of freedom and then only during nonlinear time-history analysis. Stiffness **k** must *always* be specified for nonlinear degrees of freedom, *never* for linear degrees of freedom. Note that DOF=U1 for the Isolator2 property is always nonlinear.

FRAME Data Block

This data block defines all of the general three-dimensional Frame elements that exist in the model. All elements defined in this data block reference Frame Sections defined in the FRAME SECTION Data Block (page 81).

Skip this data block if there are no Frame elements to be defined. Otherwise, prepare data according to the format described below.

For More Information

See Chapter "The Frame Element" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

FRAME	Separator
CSYS=	Coordinate System Data Lines
e J=	Definition Data Lines
GEN=	Generation Data Lines
DEL=	Delete Data Lines

Begin the data block with the **FRAME separator**.

Follow this with as many Coordinate System, Definition, Generate and Delete data lines as necessary to define all of the Frame elements in the model. The data is processed in the order it is supplied in the input data file.

Each **Coordinate System data line** defines the fixed coordinate system and the coordinate directions used by all subsequent Definition data lines for the purpose of defining the element local coordinate systems. The coordinate system and directions are in effect until the next Coordinate System data line is encountered.

Each **Definition data line** defines a new element. Each **Generate data line** generates an array of elements from a previously defined or generated element. Each **Delete data line** deletes an array of unwanted elements.

Data Line Formats

Coordinate System Data Line

CSYS=**csys** PLDIR=**pldirp, pldirs** LOCAL=**local**

Definition Data Line

e J=**i, j** SEC=**sec** NSEG=**nseg** PLVEC=**plveca, plvecb** ANG=**ang** IOFF=**ioff** JOFF=**joff** RIGID=**rigid** IREL=**irels** JREL=**jrels**

Generate Data Line

GEN=**e0, e1, ei1...** IINC=**ii1...** JINC=**ji1...**

Delete Data Line

DEL=**e0, e1, ei1...**

Description of Variables

Variable	Note	Default	Description
Coordinate System Data Line			
csys	(1, 4)	[pv(0)]	Fixed coordinate system used to define coordinate directions **pldirp** and **pldirs**: = 0: Global coordinate system ≠ 0: Alternate coordinate system label
pldirp, pldirs	(1, 4)	[pv(+Z, +X)]	Primary and secondary coordinate directions, taken at the element center in fixed coordinate system **csys**, used to determine the reference vector. Each may be one of ±X, ±Y, ±Z, ±CR, ±CA, ±CZ, ±SB, ±SA, or ±SR. The sign is required. If only **pldirp** is specified, **pldirs** is set equal to **pldirp**
local	(1)	[pv(12)]	Local plane parallel to the reference vector: = 12: Plane 1-2 = 13: Plane 1-3

Variable	Note	Default	Description
Definition Data Line			
e	(1, 2)		Label of an element being defined
i	(1, 5)		Label of joint at end I
j	(1, 5)		Label of joint at end J
sec	(1, 6)	[pv]	Label of Frame Section for element
nseg	(1)	[pv(2)]	Number of output segments, i.e., the number of spaces between internal-force output points
plveca, plvecb	(1)	[0, 0]	Labels of two joints that define the reference vector. Either joint may be zero to indicate the element center. If both are zero, this option is not used
ang	(1)	[0]	Angle that the local 2 and 3 axes are rotated about the positive local 1 axis to determine the local coordinate system [deg units]
ioff	(1)	[0]	End offset length for end I [L units]
joff	(1)	[0]	End offset length for end J [L units]
rigid	(1)	[pv(0)]	Rigid-end factor
irels	(1)		List of released degrees of freedom at end I. May be any number of U1, U2, U3, R1, R2 and R3
jrels	(1)		List of released degrees of freedom at end J. May be any number of U1, U2, U3, R1, R2 and R3
Generate Data Line			
e0, e1, ei1...	(3, 7)		Labels and label increments for an array of elements to be generated
ii1...	(3, 7)	[**ji1...**]	Secondary increments for joints at end I
ji1...	(3, 7)	[**ei1...**]	Secondary increments for joints at end J

Variable	Note	Default	Description
Delete Data Line			
e0, e1, ei1...	(3, 8)		Labels and label increments for an array of elements to be deleted

Notes

1. See Chapter "The Frame Element" of the *SAP2000 Analysis Reference*.

2. Each Definition data line defines a single element. Element labels do not have to be consecutive and may be supplied in any order. Elements may be redefined or regenerated, in which case only the last definition or generation will be used.

   When an element is redefined the previous definition is completely lost; all unspecified variables use the standard default values, and "previous-value" defaults refer to values on the previous Definition data line, not to the previous values for the element being redefined.

   See Topic "Labels" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

3. See Topic "Regular Array Specification" (page 15) in this chapter.

4. The coordinate directions **pldirp** and **pldirs** are taken in the most recently specified coordinate system **csys**. If **csys** is zero, the global system is used. Otherwise **csys** refers to an alternate coordinate system defined in the COORDINATE Data Block (page 32). If no **csys** is specified, the global system is used.

   See Chapter "Coordinate Systems" of the *SAP2000 Analysis Reference* for the definition of the various coordinate directions.

5. Joints **i** and **j** must have been defined in the JOINT Data Block (page 37). The two joints must not share the same location in space.

6. The label **sec** refers to a Frame Section defined in the FRAME SECTION Data Block (page 81). A Section must be specified for the first Definition data line.

7. Each Generate data line defines an array of elements having the same values as the starting element, **e0**, for Frame Section, number of segments, coordinate-system specifications, end offsets, and end releases. Only the joints **i** and **j** will differ according to the secondary joint label increments.

Note that the values of **csys**, **pldirp**, **pldirs**, and **local** used by the generated elements are those of the starting element **e0**, which are not necessarily the values on the most recent Coordinate System data line. The values of **plveca**, **plvecb**, and **ang** for the starting element are also used by all generated elements. This does not mean, however, that all generated elements will have the same local coordinate system as the starting element, since the axes depend upon the spatial location of the joints **i** and **j**.

See Topic "Generation" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

8. Each Delete data line defines an array of one or more elements to be deleted. Nonexistent elements may be included in the array.

   See Topic "Deletion" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

SHELL Data Block

This data block defines all of the general three-dimensional Shell elements that exist in the model. Three-dimensional plate-bending or membrane elements are considered as special cases of this general element. All elements defined in this data block reference Shell Sections defined in the SHELL SECTION Data Block (page 87).

Skip this data block if there are no Shell elements to be defined. Otherwise, prepare data according to the format described below.

For More Information

See Chapter "The Shell Element" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

SHELL	Separator
CSYS=	Coordinate System Data Lines
e J=	Definition Data Lines
GEN=	Generation Data Lines
DEL=	Delete Data Lines

Begin the data block with the **SHELL separator**.

Follow this with as many Coordinate System, Definition, Generate and Delete data lines as necessary to define all of the Shell elements in the model. The data is processed in the order it is supplied in the input data file.

Each **Coordinate System data line** defines the fixed coordinate system and the coordinate directions used by all subsequent Definition data lines for the purpose of defining the element local coordinate systems. The coordinate system and directions are in effect until the next Coordinate System data line is encountered.

Each **Definition data line** defines a new element. Each **Generate data line** generates an array of elements from a previously defined or generated element. Each **Delete data line** deletes an array of unwanted elements.

Data Line Formats

Coordinate System Data Line

CSYS=**csys** PLDIR=**pldirp, pldirs** LOCAL=**local**

Definition Data Line — Quadrilateral

e J=**j1, j2, j3, j4** SEC=**sec** PLVEC=**plveca, plvecb** ANG=**ang**

Definition Data Line — Triangle

e J=**j1, j2, j3** SEC=**sec** PLVEC=**plveca, plvecb** ANG=**ang**

Generate Data Line

GEN=**e0, e1, ei1...** JINC=**ji1...**

Delete Data Line

DEL=**e0, e1, ei1...**

Description of Variables

Variable	Note	Default	Description
Coordinate System Data Line			
csys	(1, 4)	[pv(0)]	Fixed coordinate system used to define coordinate directions **pldirp** and **pldirs**: = 0: Global coordinate system ≠ 0: Alternate coordinate system label
pldirp, pldirs	(1, 4)	[pv(+Z, +Y)]	Primary and secondary coordinate directions, taken at the element center in fixed coordinate system **csys**, used to determine the reference vector. Each may be one of ±X, ±Y, ±Z, ±CR, ±CA, ±CZ, ±SB, ±SA, or ±SR. The sign is required. If only **pldirp** is specified, **pldirs** is set equal to **pldirp**. Setting **pldirp** to zero instead activates the special backward compatibility option

Variable	Note	Default	Description
local	(1)	[pv(32)]	Local plane parallel to the reference vector: = 31: Plane 3-1 = 32: Plane 3-2
Definition Data Line			
e	(1, 2)		Label of an element being defined
j1, j2, j3, j4	(1, 5)		Labels of 3 or 4 joints defining the element
sec	(1, 6)	[pv]	Label of Shell Section for element
plveca, plvecb	(1)	[0, 0]	Labels of two joints that define the reference vector. Either joint may be zero to indicate the element center. If both are zero, this option is not used
ang	(1)	[0]	Angle that the local 1 and 2 axes are rotated about the positive local 3 axis to determine the local coordinate system [deg units]
Generate Data Line			
e0, e1, ei1...	(3, 7)		Labels and label increments for an array of elements to be generated
ji1...	(3, 7)	[**ei1...**]	Secondary increments for element joints
Delete Data Line			
e0, e1, ei1...	(3, 8)		Labels and label increments for an array of elements to be deleted

Notes

1. See Chapter “The Shell Element” of the *SAP2000 Analysis Reference*.
2. Each Definition data line defines a single element. Element labels do not have to be consecutive and may be supplied in any order. Elements may be redefined or regenerated, in which case only the last definition or generation will be used.

When an element is redefined the previous definition is completely lost; all unspecified variables use the standard default values, and "previous-value" defaults refer to values on the previous Definition data line, not to the previous values for the element being redefined.

See Topic "Labels" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

3. See Topic "Regular Array Specification" (page 15) in this chapter.

4. The coordinate directions **pldirp** and **pldirs** are taken in the most recently specified coordinate system **csys**. If **csys** is zero, the global system is used. Otherwise **csys** refers to an alternate coordinate system defined in the COORDINATE Data Block (page 32). If no **csys** is specified, the global system is used.

   See Chapter "Coordinate Systems" of the *SAP2000 Analysis Reference* for the definition of the various coordinate directions.

5. Joints **j1**, **j2**, **j3**, and **j4** must have been defined in the JOINT Data Block (page 37).

6. The label **sec** refers to a Shell Section defined in the SHELL SECTION Data Block (page 87). A Section must be specified for the first Definition data line.

7. Each Generate data line defines an array of elements having the same values as the starting element, **e0**, for Shell Section and coordinate-system specifications. Only the joints will differ according to the secondary joint label increments.

   Note that the values of **csys**, **pldirp**, **pldirs**, and **local** used by the generated elements are those of the starting element **e0**, which are not necessarily the values on the most recent Coordinate System data line. The values of **plveca**, **plvecb**, and **ang** for the starting element are also used by all generated elements. This does not mean, however, that all generated elements will have the same local coordinate system as the starting element, since the axes depend upon the spatial location of the element joints.

   See Topic "Generation" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

8. Each Delete data line defines an array of one or more elements to be deleted. Nonexistent elements may be included in the array.

See Topic “Deletion” in Chapter “Labels, Arrays, and Generation” of the *SAP2000 Analysis Reference*.

PLANE Data Block

This data block defines all of the plane-stress and plane-strain elements that exist in the model. All elements defined in this data block reference Materials defined in the MATERIAL Data Block (page 77).

Skip this data block if there are no Plane elements to be defined. Otherwise, prepare data according to the format described below.

For More Information

See Chapter “The Plane Element” of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

PLANE	Separator
e J=	Definition Data Lines
GEN=	Generation Data Lines
DEL=	Delete Data Lines

Begin the data block with the **PLANE separator**.

Follow this with as many Definition, Generate and Delete data lines as necessary to define all of the Plane elements in the model. The data is processed in the order it is supplied in the input data file.

Each **Definition data line** defines a new element. Each **Generate data line** generates an array of elements from a previously defined or generated element. Each **Delete data line** deletes an array of unwanted elements.

Data Line Formats

Definition Data Line — Three-node Triangle

e J=**j1, j3, j7** TYPE=**type** MAT=**mat** MATANG=**a** TH=**th**

Definition Data Line — Four-Node Quadrilateral

e J=**j1, j3, j7, j9** TYPE=**type** MAT=**mat** MATANG=**a** TH=**th**

Definition Data Line — Four- to Nine-Node Quadrilateral

e J9=**j1, j2, j3, j4, j5, j6, j7, j8, j9** TYPE=**type** MAT=**mat** MATANG=**a** TH=**th**

Definition Data Line — Regular Nine-Node Quadrilateral

e J9R=**j1, j2, j4** TYPE=**type** MAT=**mat** MATANG=**a** TH=**th**

Generate Data Line

GEN=**e0, e1, ei1...** JINC=**ji1...**

Delete Data Line

DEL=**e0, e1, ei1...**

Description of Variables

Variable	Note	Default	Description
Definition Data Line			
e	(1, 2)		Label of an element being defined
j1, j3, j7, j9	(1, 4)		Labels of all joints for a 3- or 4-node element
j1, j2, j3, j4, j5, j6, j7, j8, j9	(1, 4)		Labels of all joints for a 4- to 9-node element
j1, j2, j4	(1, 4)		Labels of 3 representative joints for a 9-node element with regular joint increments
type	(1)	[pv (STR-AIN)]	Element type: = STRAIN: Plane-strain = STRESS: Plane-stress
mat	(1, 5)	[pv]	Label of Material for element
a	(1)	[pv(0)]	Material angle [deg units]
th	(1)	[pv(1)]	Element thickness [L units]
Generate Data Line			
e0, e1, ei1...	(3, 6)		Labels and label increments for an array of elements to be generated
ji1...	(3, 6)	[**ei1...**]	Secondary increments for element joints
Delete Data Line			
e0, e1, ei1...	(3, 7)		Labels and label increments for an array of elements to be deleted

Notes

1. See Chapter "The Plane Element" of the *SAP2000 Analysis Reference*.

2. Each Definition data line defines a single element. Element labels do not have to be consecutive and may be supplied in any order. Elements may be redefined or regenerated, in which case only the last definition or generation will be used.

   When an element is redefined the previous definition is completely lost; all unspecified variables use the standard default values, and "previous-value" defaults refer to values on the previous Definition data line, not to the previous values for the element being redefined.

   See Topic "Labels" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

3. See Topic "Regular Array Specification" (page 15) in this chapter.

4. In general, a Plane element is defined by specifying nine joints using the J9 identifier as:

   J9=**j1, j2, j3, j4, j5, j6, j7, j8, j9**

   These joints must have been defined in the JOINT Data Block (page 37). The four corner joints **j1**, **j3**, **j7** and **j9** are mandatory when using the J9 identifier. The other five joints are optional; a value of zero should be given to indicate an omitted joint. For example, a four-node quadrilateral may be specified as:

   J9=**j1**, 0, **j3**, 0, 0, 0, **j7**, 0, **j9**

   A simplified input option for defining four-node elements is available by using the J identifier in place of J9. By this method a four-node quadrilateral is specified as:

   J=**j1, j3, j7, j9**

   Three-node triangular elements *cannot* be specified using the J9 option but must be specified as:

   J=**j1, j3, j7**

   For the best accuracy, the use of the nine-node quadrilateral is recommended. Full nine-node elements that have regular joint increments in both directions can be defined easily by using the J9R identifier. By this method a nine-node element can be defined as:

J9R=**j1, j2, j4**

The remaining joint labels are assumed to be as follows:

j3 = **j2** + *j12*
j5 = **j4** + *j12*
j6 = **j5** + *j12*
j7 = **j4** + *j14*
j8 = **j7** + *j12*
j9 = **j8** + *j12*

where *j12* = **j2** – **j1** and *j14* = **j4** – **j1**.

Only one of the J, J9 or J9R identifiers may exist on a single Definition data line.

5. The label **mat** refers to a Material defined in the MATERIAL Data Block (page 77). The Material must be specified for the first Definition data line.

6. Each Generate data line defines an array of elements having the same values as the starting element, **e0**, for element type, material properties, material angle, and thickness. Only the joints will differ according to the secondary joint label increments.

   See Topic "Generation" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

7. Each Delete data line defines an array of one or more elements to be deleted. Nonexistent elements may be included in the array.

   See Topic "Deletion" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

ASOLID Data Block

This data block defines all of the axisymmetric solid elements that exist in the model. All elements defined in this data block reference Materials defined in the MATERIAL Data Block (page 77).

Skip this data block if there are no Asolid elements to be defined. Otherwise, prepare data according to the format described below.

For More Information

See Chapter "The Asolid Element" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

ASOLID	Separator
e J=	Definition Data Lines
GEN=	Generation Data Lines
DEL=	Delete Data Lines

Begin the data block with the **ASOLID separator**.

Follow this with as many Definition, Generate and Delete data lines as necessary to define all of the Asolid elements in the model. The data is processed in the order it is supplied in the input data file.

Each **Definition data line** defines a new element. Each **Generate data line** generates an array of elements from a previously defined or generated element. Each **Delete data line** deletes an array of unwanted elements.

Data Line Formats

Definition Data Line — Three-node Triangle

e J=**j1, j3, j7** MAT=**mat** MATANG=**a** ARC=**arc**

Definition Data Line — Four-Node Quadrilateral

e J=**j1, j3, j7, j9** MAT=**mat** MATANG=**a** ARC=**arc**

Definition Data Line — Four- to Nine-Node Quadrilateral

e J9=**j1, j2, j3, j4, j5, j6, j7, j8, j9** MAT=**mat** MATANG=**a** ARC=**arc**

Definition Data Line — Nine-Node Quadrilateral, Regular Joint Increments

e J9R=**j1, j2, j4** MAT=**mat** MATANG=**a** ARC=**arc**

Generate Data Line

GEN=**e0, e1, ei1...** JINC=**ji1...**

Delete Data Line

DEL=**e0, e1, ei1...**

Description of Variables

Variable	Note	Default	Description
Definition Data Line			
e	(1, 2)		Label of an element being defined
j1, j3, j7, j9	(1, 4)		Labels of all joints for a 3- or 4-node element
j1, j2, j3, j4, j5, j6, j7, j8, j9	(1, 4)		Labels of all joints for a 4- to 9-node element
j1, j2, j4	(1, 4)		Labels of 3 representative joints for a 9-node element with regular joint increments
mat	(1, 5)	[pv]	Label of Material for element
a	(1)	[pv(0)]	Material angle [deg units]
arc	(1)	[pv(0)]	Element arc. Zero indicates one radian, i.e., **arc**=0 is the same as **arc**=180/π [deg units]
Generate Data Line			
e0, e1, ei1...	(3, 6)		Labels and label increments for an array of elements to be generated
ji1...	(3, 6)	[**ei1...**]	Secondary increments for element joints
Delete Data Line			
e0, e1, ei1...	(3, 7)		Labels and label increments for an array of elements to be deleted

Notes

1. See Chapter “The Asolid Element” of the *SAP2000 Analysis Reference*.

2. Each Definition data line defines a single element. Element labels do not have to be consecutive and may be supplied in any order. Elements may be redefined or regenerated, in which case only the last definition or generation will be used.

   When an element is redefined the previous definition is completely lost; all unspecified variables use the standard default values, and “previous-value” defaults refer to values on the previous Definition data line, not to the previous values for the element being redefined.

   See Topic “Labels” in Chapter “Labels, Arrays, and Generation” of the *SAP2000 Analysis Reference*.

3. See Topic “Regular Array Specification” (page 15) in this chapter.

4. In general, a Asolid element is defined by specifying nine joints using the J9 identifier as:

   J9=**j1, j2, j3, j4, j5, j6, j7, j8, j9**

   These joints must have been defined in the JOINT Data Block (page 37). The four corner joints **j1**, **j3**, **j7** and **j9** are mandatory when using the J9 identifier. The other five joints are optional; a value of zero should be given to indicate an omitted joint. For example, a four-node quadrilateral may be specified as:

   J9=**j1**, 0, **j3**, 0, 0, 0, **j7**, 0, **j9**

   A simplified input option for defining four-node elements is available by using the J identifier in place of J9. By this method a four-node quadrilateral is specified as:

   J=**j1, j3, j7, j9**

   Three-node triangular elements *cannot* be specified using the J9 option but must be specified as:

   J=**j1, j3, j7**

   For the best accuracy, the use of the nine-node quadrilateral is recommended. Full nine-node elements that have regular joint increments in both directions can be defined easily by using the J9R identifier. By this method a nine-node element can be defined as:

J9R=**j1, j2, j4**

The remaining joint labels are assumed to be as follows:

j3 = **j2** + *j12*
j5 = **j4** + *j12*
j6 = **j5** + *j12*
j7 = **j4** + *j14*
j8 = **j7** + *j12*
j9 = **j8** + *j12*

where *j12* = **j2** – **j1** and *j14* = **j4** – **j1**.

Only one of the J, J9 or J9R identifiers may exist on a single Definition data line.

5. The label **mat** refers to a Material defined in the MATERIAL Data Block (page 77). The Material must be specified for the first Definition data line.

6. Each Generate data line defines an array of elements having the same values as the starting element, **e0**, for material properties, material angle, and arc. Only the joints will differ according to the secondary joint label increments.

   See Topic "Generation" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

7. Each Delete data line defines an array of one or more elements to be deleted. Nonexistent elements may be included in the array.

   See Topic "Deletion" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

SOLID Data Block

This data block defines all of the three-dimensional Solid elements that exist in the model. All elements defined in this data block reference Materials defined in the MATERIAL Data Block (page 77).

Skip this data block if there are no Solid elements to be defined. Otherwise, prepare data according to the format described below.

For More Information

See Chapter "The Solid Element" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

SOLID	Separator
e J=	Definition Data Lines
GEN=	Generation Data Lines
DEL=	Delete Data Lines

Begin the data block with the **SOLID separator**.

Follow this with as many Definition, Generate and Delete data lines as necessary to define all of the Solid elements in the model. The data is processed in the order it is supplied in the input data file.

Each **Definition data line** defines a new element. Each **Generate data line** generates an array of elements from a previously defined or generated element. Each **Delete data line** deletes an array of unwanted elements.

Data Line Formats

Definition Data Line — General Joint Increments

e J=**j1, j2, j3, j4, j5, j6, j7, j8** MAT=**mat** MATANG=**a, b, c** I=**i**

Definition Data Line — Regular Joint Increments

e JR=**j1, j2, j3, j5** MAT=**mat** MATANG=**a, b, c** I=**i**

Generate Data Line

GEN=**e0, e1, ei1...** JINC=**ji1...**

Delete Data Line

DEL=**e0, e1, ei1...**

Description of Variables

Variable	Note	Default	Description
Definition Data Line			
e	(1, 2)		Label of an element being defined
j1, j2, j3, j4, j5, j6, j7, j8	(1, 4)		Labels of all joints for an element
j1, j2, j3, j5	(1, 4)		Labels of 4 representative joints for an element with regular joint increments
mat	(1, 5)	[pv]	Label of Material for element
a, b, c	(1)	[pv(0)]	Material angles [deg units]
i	(1)	[pv(Y)]	Incompatible-mode flag: = Y: Include incompatible modes = N: Do not include incompatible modes

Variable	Note	Default	Description
Generate Data Line			
e0, e1, ei1...	(3, 6)		Labels and label increments for an array of elements to be generated
ji1...	(3, 6)	[**ei1...**]	Secondary increments for element joints
Delete Data Line			
e0, e1, ei1...	(3, 7)		Labels and label increments for an array of elements to be deleted

Notes

1. See Chapter “The Solid Element” of the *SAP2000 Analysis Reference*.

2. Each Definition data line defines a single element. Element labels do not have to be consecutive and may be supplied in any order. Elements may be redefined or regenerated, in which case only the last definition or generation will be used.

   When an element is redefined the previous definition is completely lost; all unspecified variables use the standard default values, and “previous-value” defaults refer to values on the previous Definition data line, not to the previous values for the element being redefined.

   See Topic “Labels” in Chapter “Labels, Arrays, and Generation” of the *SAP2000 Analysis Reference*.

3. See Topic “Regular Array Specification” (page 15) in this chapter.

4. In general, a Solid element is defined by specifying all eight joints using the J identifier as:

   J=**j1, j2, j3, j4, j5, j6, j7, j8**

   These joints must have been defined in the JOINT Data Block (page 37).

   Elements that have regular joint increments in all three directions can be defined easily by using the JR identifier. By this method an element can be defined as:

   JR=**j1, j2, j3, j5**

The remaining joint labels are assumed to be as follows:

j4 = **j2** + *j13*
j6 = **j2** + *j15*
j7 = **j3** + *j15*
j8 = **j4** + *j15*

where *j13* = **j3** – **j1** and *j15* = **j5** – **j1**.

Only one of the J or JR identifiers may exist on a single Definition data line.

5. The label **mat** refers to a Material defined in the MATERIAL Data Block (page 77). The Material must be specified for the first Definition data line.

6. Each Generate data line defines an array of elements having the same values as the starting element, **e0**, for material properties, material angles, and incompatible-mode flag. Only the joints will differ according to the secondary joint label increments.

   See Topic "Generation" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

7. Each Delete data line defines an array of one or more elements to be deleted. Nonexistent elements may be included in the array.

   See Topic "Deletion" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

NLLINK Data Block

This data block defines all of the nonlinear Nllink elements that exist in the model. All elements defined in this data block reference Nlprops defined in the NLPROP Data Block (page 90).

Skip this data block if there are no Nllink elements to be defined. Otherwise, prepare data according to the format described below.

For More Information

See Chapter "The Nllink Element" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

NLLINK	Separator
CSYS=	Coordinate System Data Lines
e J=	Definition Data Lines
GEN=	Generation Data Lines
DEL=	Delete Data Lines

Begin the data block with the **NLLINK separator**.

Follow this with as many Coordinate System, Definition, Generate and Delete data lines as necessary to define all of the Nllink elements in the model. The data is processed in the order it is supplied in the input data file.

Each **Coordinate System data line** defines the fixed coordinate system and the coordinate directions used by all subsequent Definition data lines for the purpose of defining the element local coordinate systems. The coordinate system and directions are in effect until the next Coordinate System data line is encountered.

Each **Definition data line** defines a new element. Each **Generate data line** generates an array of elements from a previously defined or generated element. Each **Delete data line** deletes an array of unwanted elements.

Data Line Formats

Coordinate System Data Line

CSYS=**csys** AXDIR=**axdir** PLDIR=**pldirp, pldirs** LOCAL=**local**

Definition Data Line — Single-Joint Elements (Grounded Springs)

e J=**j** NLP=**nlp** AXVEC=**axveca, axvecb** PLVEC=**plveca, plvecb** ANG=**ang**

Definition Data Line — Two-Joint Elements (Links)

e J=**i, j** NLP=**nlp** AXVEC=**axveca, axvecb** PLVEC=**plveca, plvecb** ANG=**ang** ZERO=**zero**

Generate Data Line

GEN=**e0, e1, ei1...** IINC=**ii1...** JINC=**ji1...**

Delete Data Line

DEL=**e0, e1, ei1...**

Description of Variables

Variable	Note	Default	Description
Coordinate System Data Line			
csys	(1, 4)	[pv(0)]	Fixed coordinate system used to define coordinate directions **axdir**, **pldirp** and **pldirs**: = 0: Global coordinate system ≠ 0: Alternate coordinate system label
axdir	(1, 4)	[pv(+Z)]	Axial coordinate direction, taken at the element center in fixed coordinate system **csys**, used to determine the axis reference vector. May be one of ±X, ±Y, ±Z, ±CR, ±CA, ±CZ, ±SB, ±SA, or ±SR. The sign is required

Variable	Note	Default	Description
pldirp, pldirs	(1, 4)	[pv(+Z, +X)]	Primary and secondary coordinate directions, taken at the element center in fixed coordinate system **csys**, used to determine the plane reference vector. Each may be one of ±X, ±Y, ±Z, ±CR, ±CA, ±CZ, ±SB, ±SA, or ±SR. The sign is required. If only **pldirp** is specified, **pldirs** is set equal to **pldirp**
local	(1)	[pv(12)]	Local plane parallel to the reference vector: = 12: Plane 1-2 = 13: Plane 1-3
Definition Data Line			
e	(1, 2)		Label of an element being defined
i	(1, 5)		Label of joint at end I of a two-joint link
j	(1, 5)		Label of joint at end J of a two-joint link, or the only joint of a one-joint grounded spring
nlp	(1, 6)	[pv]	Label of an Nlprop property
axveca, axvecb	(1)	[0, 0]	Labels of two joints that define the axis reference vector. Either joint may be zero to indicate the element center. If both are zero, this option is not used
plveca, plvecb	(1)	[0, 0]	Labels of two joints that define the plane reference vector. Either joint may be zero to indicate the element center. If both are zero, this option is not used
ang	(1)	[0]	Angle that the local 2 and 3 axes are rotated about the positive local 1 axis to determine the local coordinate system [deg units]
zero	(1)	[0.001]	Length tolerance for determining if two-joint elements are considered to have zero length [L units]

Variable	Note	Default	Description
Generate Data Line			
e0, e1, ei1...	(3, 7)		Labels and label increments for an array of elements to be generated
ii1...	(3, 7)	[**ji1...**]	Secondary increments for joints at end I
ji1...	(3, 7)	[**ei1...**]	Secondary increments for joints at end J
Delete Data Line			
e0, e1, ei1...	(3, 8)		Labels and label increments for an array of elements to be deleted

Notes

1. See Chapter “The Nllink Element” of the *SAP2000 Analysis Reference*.
2. Each Definition data line defines a single element. Element labels do not have to be consecutive and may be supplied in any order. Elements may be redefined or regenerated, in which case only the last definition or generation will be used.

   When an element is redefined the previous definition is completely lost; all unspecified variables use the standard default values, and “previous-value” defaults refer to values on the previous Definition data line, not to the previous values for the element being redefined.

   See Topic “Labels” in Chapter “Labels, Arrays, and Generation” of the *SAP2000 Analysis Reference*.
3. See Topic “Regular Array Specification” (page 15) in this chapter.
4. The coordinate directions **axdir**, **pldirp**, and **pldirs** are taken in the most recently specified coordinate system **csys**. If **csys** is zero, the global system is used. Otherwise **csys** refers to an alternate coordinate system defined in the COORDINATE Data Block (page 32). If no **csys** is specified, the global system is used.

   See Chapter “Coordinate Systems” of the *SAP2000 Analysis Reference*.
5. Joints **i** and **j** must have been defined in the JOINT Data Block (page 37).

6. Property **nlp** must have been defined in the NLPROP Data Block (page 90). A value for **nlp** must be specified on the first Definition data line.

   See Topic "Nlprop Properties" in Chapter "The Nllink Element" of the *SAP2000 Analysis Reference*.

7. Each Generate data line defines an array of elements having the same values as the starting element, **e0**, for Nlprop properties and coordinate-system specifications. Only the joints **i** and **j** will differ according to the secondary joint label increments.

   Note that the values of **csys**, **axdir**, **pldirp**, **pldirs**, and **local** used by the generated elements are those of the starting element **e0**, which are not necessarily the values on the most recent Coordinate System data line. The values of **axveca**, **axvecb**, **plveca**, **plvecb**, and **ang** for the starting element are also used by all generated elements. This does not mean, however, that all generated elements will have the same local coordinate system as the starting element, since the axes may depend upon the spatial location of the joints **i** and **j**.

   See Topic "Generation" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

8. Each Delete data line defines an array of one or more elements to be deleted. Nonexistent elements may be included in the array.

   See Topic "Deletion" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

MATTEMP Data Block

This data block assigns material temperatures to the elements. These are the temperature at which temperature-dependent material properties are evaluated for the elements (see the MATERIAL Data Block, page 77). Elements not included in this data block will be assigned a material temperature of zero.

Skip this data block if all Materials are temperature-independent, or if all element material temperatures are zero. Otherwise, prepare data according to the format described below.

For More Information

See Topic "Element Material Temperature" in Chapter "Material Properties" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

MATTEMP	Separator
ELEM=	Element Data Lines
ADD=	Add Data Lines
REM=	Remove Data Lines

Begin the data block with the **MATTEMP separator**.

Follow this with as many Element, Add, and Remove data lines as necessary to define the material temperatures for all elements in the model. The data is processed in the order in which it is given in the data file.

Each **Element data line** defines the type of element to which subsequent Add and Remove data lines apply, until the next Element data line is encountered.

Each **Add data line** adds specified temperature values to the current material temperatures for a single element or an array of elements. Each **Remove data line** resets the material temperatures to zero for a single element or an array of elements.

Data Line Formats

Element Data Line

ELEM=**elem**

Add Data Line

ADD=**e0, e1, ei1...** T=**t** PAT=**pat**

Remove Data Line

REM=**e0, e1, ei1...**

Example

(1) All Shell elements in a model are made of a temperature-dependent material. For a given analysis, the reference temperature is 20°C and the load temperature in one Load Case is 40°C. The load temperature is unspecified in the other Load Cases (i.e., it is equal to the reference temperature). A material temperature of 30°C is chosen to be representative for all Load Cases:

```
MATTEMP
 ELEM=SHELL
  ADD=* T=30
```

Description of Variables

Variable	Note	Default	Description
Element Data Line			
elem			Type of element to which subsequent Add and Remove data lines apply. May be any one of: FRAME, SHELL, PLANE, ASOLID, or SOLID.
Add Data Line			
e0 e1 ei1...	(2, 3)		Labels and label increments for an array of one or more elements to which temperatures are being added

Variable	Note	Default	Description
t	(1, 3)	[0]	Temperature value [K units]
pat	(3)		Label of a Pattern of scale factors multiplying temperature values. If omitted, a unit scale factor is assumed at every joint
Remove Data Line			
e0, e1, ei1...	(2, 4)		Labels and label increments for an array of one or more elements for which temperatures are being reset to zero

Notes

1. See Topic “Element Material Temperature” in Chapter “Material Properties” of the *SAP2000 Analysis Reference*.
2. See Topic “Regular Array Specification” (page 15) in this chapter.
3. The material temperatures defined on this data line are added to the current values at the joints of each element in the array. If no Pattern is specified, then the temperature added to each joint is just **t**. If a Pattern label **pat** is given, then the temperature added to a joint is equal to **t** multiplied by the Pattern value at that joint. A single material temperature for the element is computed as the average of the joint temperatures.
4. For each element in the array, the material temperature is set back to zero. This overwrites the effect of any previous Add data lines.

REFTEMP Data Block

This data block assigns reference temperatures to the elements. These are the temperatures at which the unloaded elements are assumed to be stress-free. Elements not included in this data block will be assigned a reference temperature of zero.

Skip this data block if no Temperature Load is to be applied, or if all element reference temperatures are zero. Otherwise, prepare data according to the format described below.

For More Information

See Topic "Reference Temperature" in Chapter "Load Cases" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

REFTEMP	Separator
ELEM=	Element Data Lines
ADD=	Add Data Lines
REM=	Remove Data Lines

Begin the data block with the **REFTEMP separator**.

Follow this with as many Element, Add, and Remove data lines as necessary to define the reference temperatures for all elements in the model. The data is processed in the order in which it is given in the data file.

Each **Element data line** defines the type of element to which subsequent Add and Remove data lines apply, until the next Element data line is encountered.

Each **Add data line** adds specified temperature values to the current reference temperatures for a single element or an array of elements. Each **Remove data line** resets the reference temperatures to zero for a single element or an array of elements.

Data Line Formats

Element Data Line

ELEM=**elem**

Add Data Line

ADD=**e0, e1, ei1...** T=**t** PAT=**pat**

Remove Data Line

REM=**e0, e1, ei1...**

Example

(1) All SHELL elements in a model have a reference temperature of 20°C:

```
REFTEMP
 ELEM=SHELL
  ADD=* T=20
```

Description of Variables

Variable	Note	Default	Description
Element Data Line			
elem			Type of element to which subsequent Add and Remove data lines apply. May be any one of: FRAME, SHELL, PLANE, ASOLID, or SOLID.
Add Data Line			
e0 e1 ei1...	(2, 3)		Labels and label increments for an array of one or more elements to which temperatures are being added
t	(1, 3)	[0]	Temperature value [K units]

Variable	Note	Default	Description
pat	(3)		Label of a Pattern of scale factors multiplying temperature values. If omitted, a unit scale factor is assumed at every joint
Remove Data Line			
e0, e1, ei1...	(2, 4)		Labels and label increments for an array of one or more elements for which temperatures are being reset to zero

Notes

1. See Topic “Reference Temperature” in Chapter “Load Cases” of the *SAP2000 Analysis Reference*.
2. See Topic “Regular Array Specification” (page 15) in this chapter.
3. The reference temperatures defined on this data line are added to the current values at the joints of each element in the array. If no Pattern is specified, then the temperature added to each joint is just **t**. If a Pattern label **pat** is given, then the temperature added to a joint is equal to **t** multiplied by the Pattern value at that joint. The reference temperature field over the element is interpolated from the values at the joints.
4. For each element in the array, the reference temperature is set back to zero. This overwrites the effect of any previous Add data lines.

PRESTRESS Data Block

This data block defines the prestressing cables that act on each Frame element. The actual application of the resulting loads is specified in the LOAD Data Block (page 134).

Skip this data block if there are no prestressing cables acting on any of the Frame elements in the structure. Otherwise, prepare data according to the format described below.

For More Information

See Topic "Prestress Load" in Chapter "The Frame Element" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

PRESTRESS	Separator
ADD=	Add Data Lines
REM=	Remove Data Lines

Begin the data block with the **PRESTRESS separator**.

Follow this with as many Add and Remove data lines as necessary to define all of the prestressing cables that act on the Frame elements in the structure. The data is processed in the order in which it is given in the data file.

Each **Add data line** adds a prestressing cable to each element in an array of one or more Frame elements. Each **Remove data line** removes all prestressing cables from an array of one or more Frame elements.

Data Line Formats

Add Data Line

ADD=**e0, e1, ei1...** D=**di, dc, dj** T=**t**

Remove Data Line

REM=**e0, e1, ei1...**

Example

Each of twelve FRAME elements has a single prestressing cable of the same geometry and tension.

```
PRESTRESS
 ADD=1,12,1  D=0.25,0.50,0.5  T=100
```

Description of Variables

Variable	Note	Default	Description
Add Data Line			
e0, e1, ei1...	(1, 2, 3)		Labels and label increments for an array of one or more Frame elements to which a prestressing cable is being added
di	(1)	[0]	Upward (+2 direction) cable drape at element end I [L units]
dc	(1)	[0]	Downward (–2 direction) cable drape at element center [L units]
dj	(1)	[0]	Upward (+2 direction) cable drape at element end J [L units]
t	(1)	[0]	Cable tension [F units]
Remove Data Line			
e0, e1, ei1...	(2, 4)		Labels and label increments for an array of one or more Frame elements from which all prestressing cables are being removed

Notes

1. See Topic “Prestress Load” in Chapter “The Frame Element” of the *SAP2000 Analysis Reference*.
2. See Topic “Regular Array Specification” (page 15) in this chapter.
3. For each element in the array, the specified cable is applied in addition to any other cables that may already be acting on the element.
4. For each element in the array, all previously-added prestressing cables are removed. This overwrites the effect of any previous Add data lines.

LOAD Data Block

This data block defines the basic Load Cases used for the analysis. Each Load Case is a spatial distribution of forces, displacements, temperatures and other effects that act upon the structure.

Skip this data block if there are no Loads to be defined. Otherwise, prepare data according to the format described below.

For More Information

See Chapter "Load Cases" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

LOAD	Separator
CSYS=	Coordinate System Data Lines
NAME=	Name Data Lines
CSYS=	Coordinate System Data Lines
TYPE=	Type Data Lines
CSYS=	Coordinate System Data Lines
ADD=	Add Data Lines
REM=	Remove Data Lines

Begin the data block with the **LOAD separator**.

Follow this with as many Coordinate System, Name, Type, Add, and Remove data lines as necessary to define all the Loads. Data lines are processed in the order that they are supplied in the input data file.

Each **Name data line** begins the definition of a new Load Case, and may be followed by as many Type, Add, and Remove data lines as necessary to define the Load Case. Self-Weight Load, if any, is specified on the Name data line.

Each **Type data line** begins the application of a specified type of load on a specified type of joint or element. Type data lines may be given in any order, and may be

omitted if no loads of that type are present for the Load Case being defined. The type of load is specified using one, two, or three parameters as follows:

- The general load type, such as GRAVITY or TEMPERATURE. This parameter is always present;
- The element type to which the loads apply: JOINT, FRAME, PLANE, SHELL, ASOLID, SOLID, or NLLINK. Note that for loading purposes, JOINTS are treated as a type of element. This parameter is not used when the general load type can only apply to a single element type;
- The part of the element being loaded, such as Face 1. This parameter is not used when the load type applies to a whole element.

Thus, for example, Gravity Load applied to Frame elements is considered to be different from Gravity Load applied to Shells, and these would be defined using two separate Type data lines.

The general load-type names may be quite long. Only the first four characters need be supplied to identify the load; these characters are shown as underlined in the data line formats that follow. Thus GRAVITY may be abbreviated as GRAV, and RESTRAINT DISPLACEMENT may be abbreviated as REST. Additional characters may be supplied if desired, but they will not be read by the program. The element type and element part must be fully supplied.

Each Type data line is followed by as many Add and Remove data lines as necessary to apply all loads of a given type acting in given Load Case.

Each **Add data line** adds loads to a regular array of one or more elements. Each **Remove data line** removes all load of the given type in the given Load Case from a regular array of one or more elements.

Coordinate System data lines may be used *anywhere* in the LOAD Data Block. Each Coordinate System data line defines the fixed coordinate system used by all subsequent Add data lines until the next Coordinate System data line is encountered. If this data line is omitted, the global coordinate system is assumed (e.g., CSYS=0).

Data Line Formats

Coordinate System Data Line

CSYS=**csys**

Name Data Line

NAME=**name** SW=**sw**

Force Load on Joints

Type Data Line

TYPE=FORCE

Add Data Line — Fixed Coordinates

ADD=**j0, j1, ji1...** UX=**ux** UY=**uy** UZ=**uz** RX=**rx** RY=**ry** RZ=**rz** PAT=**pat**

Add Data Line — Joint Local Coordinates

ADD=**j0, j1, ji1...** U1=**u1** U2=**u2** U3=**u3** R1=**r1** R2=**r2** R3=**r3** PAT=**pat**

Remove Data Line

REM=**j0, j1, ji1...**

Restraint Displacement Load on Joints

Type Data Line

TYPE=RESTRAINT DISPLACEMENT

Add Data Line — Fixed Coordinates

ADD=**j0, j1, ji1...** UX=**ux** UY=**uy** UZ=**uz** RX=**rx** RY=**ry** RZ=**rz** PAT=**pat**

Add Data Line — Joint Local Coordinates

ADD=**j0, j1, ji1...** U1=**u1** U2=**u2** U3=**u3** R1=**r1** R2=**r2** R3=**r3** PAT=**pat**

Remove Data Line

REM=**j0, j1, ji1...**

Spring Displacement Load on Joints

Type Data Line

TYPE=SPRING DISPLACEMENT

Add Data Line — Fixed Coordinates

ADD=**e0, e1, ei1...** UX=**ux** UY=**uy** UZ=**uz** RX=**rx** RY=**ry** RZ=**rz** PAT=**pat**

Add Data Line — Joint Local Coordinates

ADD=**e0, e1, ei1...** U1=**u1** U2=**u2** U3=**u3** R1=**r1** R2=**r2** R3=**r3** PAT=**pat**

Remove Data Line

REM=**e0, e1, ei1...**

Gravity Load

Type Data Line

TYPE=GRAVITY ELEM=**elem**

Add Data Line

ADD=**e0, e1, ei1...** UX=**ux** UY=**uy** UZ=**uz**

Remove Data Line

REM=**e0, e1, ei1...**

Concentrated Span Load on Frame Elements

Type Data Line

TYPE=CONCENTRATED SPAN

Add Data Line — Fixed Coordinates

ADD=**e0, e1, ei1...** RD=**rd** (*or* D=**d**) UX=**ux** UY=**uy** UZ=**uz** RX=**rx** RY=**ry** RZ=**rz**

Add Data Line — Element Local Coordinates

ADD=**e0, e1, ei1...** RD=**rd** (*or* D=**d**) U1=**u1** U2=**u2** U3=**u3** R1=**r1** R2=**r2** R3=**r3**

Remove Data Line

REM=**e0, e1, ei1...**

Distributed Span Load on Frame Elements

Type Data Line

TYPE=DISTRIBUTED SPAN

Add Data Line — Fixed Coordinates

ADD=**e0, e1, ei1...** RD=**rda, rdb** (*or* D=**da, db**) UX=**uxa, uxb** UY=**uya, uyb** UZ=**uza, uzb** RX=**rxa, rxb** RY=**rya, ryb** RZ=**rza, rzb**

Add Data Line — Fixed Coordinates, upon Projected Length

ADD=**e0, e1, ei1...** RD=**rda, rdb** (*or* D=**da, db**) UXP=**uxpa, uxpb** UYP=**uypa, uypb** UZP=**uzpa, uzpb** RXP=**rxpa, rxpb** RYP=**rypa, rypb** RZP=**rzpa, rzpb**

Add Data Line — Element Local Coordinates

ADD=**e0, e1, ei1...** RD=**rda, rdb** (*or* D=**da, db**) U1=**u1a, u1b** U2=**u2a, u2b** U3=**u3a, u3b** R1=**r1a, r1b** R2=**r2a, r2b** R3=**r3a, r3b**

Remove Data Line

REM=**e0, e1, ei1...**

Prestress Load on Frame Elements

Type Data Line

TYPE=PRESTRESS

Add Data Line

ADD=**e0, e1, ei1...** P=**p**

Remove Data Line

REM=**e0, e1, ei1...**

Uniform Load on Shell Elements

Type Data Line

TYPE=UNIFORM

Add Data Line — Fixed Coordinates

ADD=**j0, j1, ji1...** UX=**ux** UY=**uy** UZ=**uz**

Add Data Line — Fixed Coordinates, upon Projected Area

ADD=**j0, j1, ji1...** UXP=**uxp** UYP=**uyp** UZP=**uzp**

Add Data Line — Element Local Coordinates

ADD=**j0, j1, ji1...** U1=**u1** U2=**u2** U3=**u3**

Remove Data Line

REM=**j0, j1, ji1...**

Surface Pressure Load

Type Data Line

TYPE=SURFACE PRESSURE ELEM=**elem** FACE=**face**

Add Data Line

ADD=**e0, e1, ei1...** P=**p** PAT=**pat**

Remove Data Line

REM=**e0, e1, ei1...**

Pore Pressure Load

Type Data Line

TYPE=PORE PRESSURE ELEM=**elem**

Add Data Line

ADD=**e0, e1, ei1...** P=**p** PAT=**pat**

Remove Data Line

REM=**e0, e1, ei1...**

Temperature Load

Type Data Line

TYPE=TEMPERATURE ELEM=**elem**

Add Data Line — Frame Elements

ADD=**e0, e1, ei1...** T=**t** T2=**t2** T3=**t3** PAT=**pat**

Add Data Line — Shell Elements

ADD=**e0, e1, ei1...** T=**t** T3=**t3** PAT=**pat**

Add Data Line — Plane, Asolid, and Solid Elements

ADD=**e0, e1, ei1...** T=**t** PAT=**pat**

Remove Data Line

REM=**e0, e1, ei1...**

Rotate Load on Asolid Elements

Type Data Line

TYPE=ROTATE

Add Data Line

ADD=**e0, e1, ei1...** R=**r**

Remove Data Line

REM=**e0, e1, ei1...**

Example

A structure modeled entirely with Frame elements is loaded by dead load in one Load Case, by a series of Concentrated Span Loads in a second Load Case, and by support settlement in a third Load Case:

```
LOAD
 NAME=DL  SW=1
 NAME=CONC
  TYPE=CONCENTRATED SPAN
   ADD=101,125,1 RD=0.25 UZ=-2.34
   ADD=101,125,1 RD=0.75 UZ=-2.34
   ADD=101,125,1 RD=0.5  UZ=-4.68
 NAME=SETTLE
  TYPE=RESTRAINT DISPLACEMENT
   ADD=3 UZ=-0.67
```

Description of Variables

Variable	Note	Default	Description
Name Data Line			
name	(1, 2)		Label of a Load Case being defined
sw	(1)	[0]	Scale factor that multiplies the self-weight of every element in the structure, applied in the global downward direction
Coordinate System Data Line			
csys	(1, 4)	[pv(0)]	Fixed coordinate system used by subsequent Add data lines: = 0: Global coordinate system ≠ 0: Alternate coordinate system label
Type Data Line			
elem	(1, 5)		Type of element that is to be loaded. See Note 5 for allowable values
face	(1, 6)		Face of an element that is to be loaded. See Note 6 for allowable values
Add Data Line			
j0 j1 ji1...	(3, 7)		Labels and label increments for an array of one or more joints to which loads are being added
e0 e1 ei1...	(3, 7)		Labels and label increments for an array of one or more elements to which loads are being added
Add Data Line — Force Load			
ux, uy, uz	(1, 8)	[0]	Forces, in fixed coordinate system **csys** [F units]

Variable	Note	Default	Description
rx, ry, rz	(1, 8)	[0]	Moments, in fixed coordinate system **csys** [FL units]
u1, u2, u3	(1, 8)	[0]	Forces, in each joint local coordinate system [F units]
r1, r2, r3	(1, 8)	[0]	Moments, in each joint local coordinate system [FL units]
pat	(1, 8)		Label of a Pattern of scale factors multiplying force and moment values. If omitted, a unit scale factor is assumed at every joint
Add Data Line — Restraint Displacement Load and Spring Displacement Load			
ux, uy, uz	(1, 9)	[0]	Translations, in fixed coordinate system **csys** [L units]
rx, ry, rz	(1, 9)	[0]	Rotations, in fixed coordinate system **csys** [rad units]
u1, u2, u3	(1, 9)	[0]	Translations, in each joint local coordinate system [L units]
r1, r2, r3	(1, 9)	[0]	Rotations, in each joint local coordinate system [rad units]
pat	(1, 9)		Label of a Pattern of scale factors multiplying translation and rotation values. If omitted, a unit scale factor is assumed at every joint
Add Data Line — Gravity Load			
ux, uy, uz	(1)	[0]	Scale factors that multiply the self-weight of each element, in fixed coordinate system **csys**
Add Data Line — Concentrated Span Load			
d	(1)		Distance from element end I to loads [L units]
rd	(1)		Relative distance from element end I to loads. Range is 0 (at end I) to 1 (at end J)

Variable	Note	Default	Description
ux, uy, uz	(1)	[0]	Forces, in fixed coordinate system **csys** [F units]
rx, ry, rz	(1)	[0]	Moments, in fixed coordinate system **csys** [FL units]
u1, u2, u3	(1)	[0]	Forces, in each element local coordinate system [F units]
r1, r2, r3	(1)	[0]	Moments, in each element local coordinate system [FL units]
Add Data Line — Distributed Span Load			
da, db	(1, 10)		Distances from element end I to beginning and end of loads [L units]
rda, rdb	(1, 10)	[0, 1]	Relative distances from element end I to beginning and end of loads. Range is 0 (at end I) to 1 (at end J)
uxa, uxb, uya, uyb, uza, uzb	(1, 10)	[0, **uxa**, 0, **uya**, 0, **uza**]	Force intensities at beginning and end of load, in fixed coordinate system **csys** [F/L units]
rxa, rxb, rya, ryb, rza, rzb	(1, 10)	[0, **rxa**, 0, **rya**, 0, **rza**]	Moment intensities at beginning and end of load, in fixed coordinate system **csys** [FL/L units]
uxpa, uxpb, uypa, uypb, uzpa, uzpb	(1, 10)	[0, **uxpa**, 0, **uypa**, 0, **uzpa**]	Force intensities at beginning and end of load, in fixed coordinate system **csys**. Will be scaled by the sine of the angle between the element and the direction of load [F/L units]
rxpa, rxpb, rypa, rypb, rzpa, rzpb	(1, 10)	[0, **rxpa**, 0, **rypa**, 0, **rzpa**]	Moment intensities at beginning and end of load, in fixed coordinate system **csys**. Will be scaled by the cosine of the angle between the element and the direction of load [FL/L units]

Variable	Note	Default	Description
u1a, u1b, u2a, u2b, u3a, u3b	(1, 10)	[0, **u1a**, 0, **u2a**, 0, **u3a**]	Force intensities at beginning and end of load, in each element local coordinate system [F/L units]
r1a, r1b, r2a, r2b, r3a, r3b	(1, 10)	[0, **r1a**, 0, **r2a**, 0, **r3a**]	Moment intensities at beginning and end of load, in each element local coordinate system [FL/L units]
Add Data Line — Prestress Load			
p	(1, 11)	[0]	Prestress scale factor
Add Data Line — Uniform Load			
ux, uy, uz	(1)	[0]	Force intensities, in fixed coordinate system **csys** [F/L^2 units]
uxp, uyp, uzp	(1)	[0]	Force intensities, in fixed coordinate system **csys**. Will be scaled by the cosine of the angle between the Shell element normal and the direction of load [F/L^2 units]
u1, u2, u3	(1)	[0]	Force intensities, in each Shell element local coordinate system [F/L^2 units]
Add Data Line — Pore Pressure Load and Surface Pressure Load			
p	(1, 12)	[0]	Pressure value [F/L^2 units]
pat	(1, 12)		Label of a Pattern of scale factors multiplying the pressure value. If omitted, a unit scale factor is assumed at every joint
Add Data Line — Temperature Load			
t	(1, 13)	[See Note]	Temperature value [K units]
t2	(1, 14)	[0]	Temperature gradient in local 2 direction, Frame elements only [K/L units]

Variable	Note	Default	Description
t3	(1, 14)	[0]	Temperature gradient in local 3 direction, Frame and Shell elements only [K/L units]
pat	(13, 14)		Label of a Pattern of scale factors multiplying temperature and temperature gradient values. If omitted, a unit scale factor is assumed at every joint
Add Data Line — Rotate Load			
r	(15)	[0]	Angular velocity for rotation about the axis of symmetry of the element [cyc/T units]
Remove Data Line			
j0, j1, ji1...	(3, 16)		Labels and label increments for an array of one or more joints from which previously-added loads are being removed
e0, e1, ei1...	(3, 16)		Labels and label increments for an array of one or more elements from which previously-added loads are being removed

Notes

1. See Chapter “Load Cases” of the *SAP2000 Analysis Reference*.
2. Each Name data line begins the definition of a new Load Case. Load labels do not have to be consecutive and may be supplied in any order. Load labels may not be repeated in the data block.

   See Topic “Labels” in Chapter “Labels, Arrays, and Generation” of the *SAP2000 Analysis Reference*.
3. See Topic “Regular Array Specification” (page 15) in this chapter.
4. All load components that are specified in fixed coordinates are taken in the most recently specified coordinate system **csys**. If **csys** is zero, the global system is used. Otherwise **csys** refers to an Alternate Coordinate System defined in the COORDINATE Data Block (page 32). If no **csys** is specified, the global system is used.

5. The available combinations of load type and element type are indicated by the entries “Yes” in the table below:

Load Type	Frame	Shell	Plane	Asolid	Solid	Nllink
Gravity	Yes	Yes	Yes	Yes	Yes	Yes
Temperature	Yes	Yes	Yes	Yes	Yes	
Surface Pressure		Yes	Yes	Yes	Yes	
Pore Pressure			Yes	Yes	Yes	

6. For Surface Pressure Loading, the parameter **face** indicates the element face upon which the pressure is acting. Legal values for Shell and Solid elements are from 1 to 6. Legal values for Plane and Asolid elements are from 1 to 4.

7. Each Add data line may refer to a single joint, **j0**, or an array of joints, **j0, j1, ji1...**, having one, two or three dimensions; or to a single element, **e0**, or an array of elements, **e0, e1, ei1...**, having one, two or three dimensions.

   Additional parameters on the Add data line define a load that acts upon the elements. This load is added to the load defined on previous Add data lines for the current load type, in the current Load Case. The current Load Case is that defined on the most recent Name data line. The current load type is that defined on the most recent Type data line.

8. For each joint, the specified forces and moments are added to the current values at the joint in the following manner:

   - If a Pattern label **pat** is given, then all force and moment values on the data line are multiplied by the Pattern value at that joint;
   - Force and moment values given in joint local coordinates are added directly to the current values at that joint;
   - Force and moment values given in fixed coordinates are transformed to the joint’s local coordinate system and then added to the current values.

9. For each joint, the specified translations and rotations are added to the current values at the joint in the following manner:

   - If a Pattern label **pat** is given, then all translation and rotation values on the data line are multiplied by the Pattern value at that joint;

- Translation and rotation values given in joint local coordinates are added directly to the current values at that joint;
- Translation and rotation values given in fixed coordinates are transformed to the joint's local coordinate system and then added to the current values.

10. For each force or moment component, a single intensity value may be given if the load is uniformly distributed. Two values are needed if the load intensity varies linearly over its range of application.

11. The prestress scale factor **p** on an Add data line multiplies the prestress load created by all prestressing cables that act on an element. These cables are defined in the PRESTRESS Data Block (page 131).

12. The pressure field is interpolated over each element (for Pore Pressure) or element face (for Surface Pressure) from the specified pressure values at the element joints. If no Pattern is specified, then the pressure at each joint is just **p**. If a Pattern label **pat** is given, then the pressure at a joint is equal to **p** multiplied by the Pattern value at that joint. This interpolated pressure field is added to the current field for the element.

13. The load temperature field is interpolated over each element from the specified temperature values at the element joints. If no Pattern is specified, then the temperature at each joint is just **t**. If a Pattern label **pat** is given, then the temperature at a joint is equal to **t** multiplied by the Pattern value at that joint. This interpolated load temperature field is added to the current field for the element.

    The default load temperature for each element is the reference temperature. A Remove data line returns the element to its reference temperature. Thus a load is produced for a given element only if load temperature is added to the element following the last Remove data line (if any) for the element. However, note that the load temperatures add from zero, not from the reference temperature.

14. The load temperature-gradient field is interpolated over each element from the specified temperature-gradient values at the element joints. If no Pattern is specified, then the temperature gradient at each joint is just **t3** (or **t2**). If a Pattern label **pat** is given, then the temperature gradient at a joint is equal to **t3** (or **t2**) multiplied by the Pattern value at that joint. This interpolated load temperature-gradient field is added to the current field for the element.

    The reference temperature gradient is always taken to be zero. A Remove data line returns the element to its reference temperature gradient of zero.

15. The angular velocities from all Add data lines that refer to a given element are added together. The load on the element is computed from this total angular velocity.

16. Each Remove data line may refer to a single joint, **j0**, or an array of joints, **j0, j1, ji1...**, having one, two or three dimensions; or to a single element, **e0**, or an array of elements, **e0, e1, ei1...**, having one, two or three dimensions.

    For each joint or element in the array, all loads of the current type, in the current Load Case, are set back to zero. This overwrites the effect of any previous Add data lines. The current Load Case is that defined on the most recent Name data line. The current type is that defined on the most recent Type data line.

PDFORCE Data Block

This data block defines the directly specified P-Delta axial forces acting on the Frame elements. The PDELTA Data Block (page 154) is not needed if these are the only P-Delta axial forces present in the structure.

Skip this data block if there are no directly specified P-Delta axial forces in the Frame elements. Otherwise, prepare data according to the format described below.

For More Information

See Chapter "P-Delta Analysis" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

PDFORCE	Separator
CSYS=	Coordinate System Data Line
ADD=	Add Data Lines
REM=	Remove Data Lines

Begin the data block with the **PDFORCE separator**.

Follow this with as many Coordinate System, Add and Remove data lines as necessary to define all the directly specified P-Delta axial forces in the model. The data is processed in the order it is supplied in the input data file.

Each **Coordinate System data line** defines the fixed coordinate system used by all subsequent Add data lines until the next Coordinate System data line is encountered. If this data line is omitted, the global coordinate system is assumed (e.g., CSYS=0).

Each **Add data line** adds P-Delta axial forces to an array of Frame Elements. Each **Remove data line** removes P-Delta axial forces from an array of Frame Elements.

Data Line Formats

Coordinate System Data Line

CSYS=**csys**

Add Data Line

ADD=**e0, e1, ei1...** P=**p** PX=**px** PY=**py** PZ=**pz**

Remove Data Line

REM=**e0, e1, ei1...**

Example

(1) The horizontal (X) component of the tension in the two main cables of a suspension bridge is known to be 10 000 kips. This is specified to be a P-Delta axial force as:

```
PDFORCE
 ADD=101,125,1 PX=10000
 ADD=201,225,1 PX=10000
```

Description of Variables

Variable	Note	Default	Description
Coordinate System Data Line			
csys	(1, 3)	[pv(0)]	Fixed coordinate system for subsequent Add data lines: = 0: Global coordinate system ≠ 0: Alternate coordinate system label
Add Data Line			
e0, e1, ei1...	(1, 2, 4)		Labels and label increments for an array of one or more Frame elements to which P-Delta axial force is being added
p	(1, 4)	[0]	P-Delta axial force [F units]
px	(1, 3, 4)	[0]	Projection of the P-Delta axial force upon the X axis of coordinate system **csys** [F units]
py	(1, 3, 4)	[0]	Projection of the P-Delta axial force upon the Y axis of coordinate system **csys** [F units]
pz	(1, 3, 4)	[0]	Projection of the P-Delta axial force upon the Z axis of coordinate system **csys** [F units]
Remove Data Line			
e0, e1, ei1...	(2, 5)		Labels and label increments for an array of one or more Frame elements from which P-Delta axial force is being removed

Notes

1. See Chapter "P-Delta Analysis" of the *SAP2000 Analysis Reference*.

2. See Topic "Regular Array Specification" (page 15) in this chapter.

3. Projections **px**, **py**, and **pz** are taken in the most recently specified coordinate system **csys**. If **csys** is zero, the global system is used. Otherwise **csys** refers to an Alternate Coordinate System defined in the COORDINATE Data Block (page 32). If no **csys** is specified, the global system is used.

   You must not specify a projection upon an axis that is perpendicular to the local 1 axis of the element. For example, if the element is parallel to the Z axis in coordinate system **csys**, you may not specify values for **px** or **py**.

4. For each element in the array, the specified P-Delta axial forces are added to the existing values. Normally only one of **p**, **px**, **py**, or **pz** is applied to each element, but this is not required.

5. For each element in the array, the specified P-Delta axial forces are set to zero.

PDELTA Data Block

This data block defines the parameters that control an iterative P-Delta analysis. This data block is not needed if all of the P-Delta axial forces are directly specified in the PDFORCE Data Block (page 150). It is needed if any P-Delta axial forces are to be computed from the P-Delta load combination.

Skip this data block if no iterative P-Delta analysis is to be performed. Otherwise, prepare data according to the format described below.

For More Information

See Chapter "P-Delta Analysis" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

PDELTA		Separator
	ITMAX=	Iteration Data Line
	LOAD=	Load Data Lines

Begin the data block with the **PDELTA separator**.

Follow this with a single **Iteration data line** that specifies the control parameters for the iterative analysis.

Follow the Iteration data line with as many Load data lines as necessary to define the P-Delta load combination. Each **Load data line** specifies a single Load Case that is to be included in the combination and the scale factor by which it is to be multiplied.

Data Line Formats

Iteration Data Line

ITMAX=**itmax** TOLD=**told**

Load Data Line

LOAD=**load** SF=**sf**

Example

Suppose that Load Cases DL and LL are dead load and live load, respectively. The following data specifies the P-Delta load combination to be 1.2 times the dead load plus 0.5 times the live load. No other Load Cases are included.

```
PDELTA
 ITMAX=5 TOLD=0.0001
  LOAD=DL SF=1.2
  LOAD=LL SF=0.5
```

Description of Variables

Variable	Note	Default	Description
Control Data Line			
itmax	(1)	[1]	Maximum number of additional iterations
told	(1)	[.001]	Relative displacement convergence tolerance
Load Data Line			
load	(1, 2)		Label of a Load Case
sf	(1, 2)	[1]	Scale factor multiplying Load Case **load**

Notes

1. See Chapter “P-Delta Analysis” of the *SAP2000 Analysis Reference*.

2. The label **load** refers to a Load Case defined in the LOAD Data Block (page 134). The P-Delta load combination is defined as the sum of the specified Load Cases, each multiplied by the specified scale factor **sf**.

   Each Load Case should be specified at most once. If a Load Case is repeatedly specified, the scale factor from the last specification is used. Any Load Case that is omitted here is not added into the P-Delta load combination.

MODES Data Block

This data block defines the parameters that control the calculation of the Vibration Modes of the model. The Modes may be computed either by eigenvector analysis or Ritz-vector analysis, but not both.

This data block is required if response-spectrum and/or time-history analyses are to be performed, and is optional otherwise.

Skip this data block if no Vibration Modes are to be calculated. Otherwise, prepare data according to the format described below.

For More Information

See Topics "Eigenvector Analysis" and "Ritz-vector Analysis" in Chapter "Static and Dynamic Analysis" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

MODES		Separator
	TYPE=	Type Data Line
		ACC= Acceleration Data Lines
		LOAD= Load Data Lines
		NLLINK= Nonlinear Deformation Load Data Line

Begin the data block with the **MODES separator**.

Follow this with a single **Type data line** that specifies the type of analysis to be performed (eigen or Ritz) and the analysis parameters.

If Ritz-vector analysis is specified, follow the Type data line with as many Acceleration, Load, and Nonlinear Deformation Load data lines as necessary to specify all starting load vectors to be used. Each **Acceleration data line** specifies an Acceleration Load in the global coordinate system. Each **Load data line** specifies a Load Case. A single **Nonlinear Deformation Load data line** is permitted; it specifies all nonlinear deformation loads. Starting load vectors may not be repeated.

If no starting load vectors are specified, the program will automatically use the three Acceleration Loads as the starting load vectors.

Data Line Formats

Type Data Line — Eigenvector Analysis

TYPE=EIGEN N=**n** RESMASS=**resmass** SHIFT=**shift** CUT=**cut** TOL=**tol**

Type Data Line — Ritz-vector Analysis

TYPE=RITZ N=**n**

Acceleration Data Line

ACC=**acc** NCYC=**ncyc**

Load Data Line

LOAD=**load** NCYC=**ncyc**

Nonlinear Deformation Load Data Line

NLLINK=∗ NCYC=**ncyc**

Example

(1) Twenty eigen-modes are requested with frequencies not to exceed 30 Hz:

```
MODES
 TYPE=EIGEN N=20 CUT=30
```

(2) Twenty Ritz modes are requested for a seismic analysis. The three ground accelerations are automatically used as starting load vectors:

```
MODES
 TYPE=RITZ N=20
```

(3) Twenty Ritz modes are requested for a Time-History analysis. The two lateral ground accelerations and a Load Case named “2” are used as starting vectors:

```
MODES
 TYPE=RITZ N=20
  ACC=UX
  ACC=UY
  LOAD=2
```

Description of Variables

Variable	Note	Default	Description
Type Data Line			
n	(1, 2)	[1]	Number of Modes requested
resmass	(1)	[N]	Whether to calculate residual-mass modes = Y: Calculate residual-mass modes = N: Do not calculate residual-mass modes
shift	(1)	[0]	Eigenvalue shift frequency [cyc/T units]
cut	(1)	[0]	Eigenvalue cutoff frequency radius [cyc/T units] = 0: Infinite frequency radius — no limit > 0: Finite frequency radius
tol	(1)	[10^{-5}]	Relative convergence tolerance on eigenvalues
Acceleration Data Line			
acc	(2)		Direction, in global coordinates, of an Acceleration Load to be used as a Ritz starting load vector. May be UX, UY, or UZ
Load Data Line			
load	(2)		Label of a Load Case to be used as a Ritz starting load vector
Acceleration, Load, and Nonlinear Deformation Load Data Lines			
ncyc	(2)	[0]	Maximum number of generation cycles to be performed for the specified starting vector(s): = 0: Unlimited > 0: Maximum number

Notes

1. See Topic “Eigenvector Analysis” in Chapter “Static and Dynamic Analysis” of the *SAP2000 Analysis Reference*.

2. See Topic “Ritz-vector Analysis” in Chapter “Static and Dynamic Analysis” of the *SAP2000 Analysis Reference*.

FUNCTION Data Block

This data block defines the Functions used by the SPEC Data Block (page 165) and the HISTORY Data Block (page 169). Although the definition of the Functions is the same for both cases, the use of the Functions differs in some details. See the two data blocks for more information.

Skip this data block if there are no Functions to be defined. Otherwise, prepare data according to the format described below.

For More Information

See Topic "Functions" in Chapter "Static and Dynamic Analysis" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

FUNCTION	Separator
NAME=	Name Data Lines
t_0 f_0 ...	Function Value Data Lines

Begin the data block with the **FUNCTION separator**.

Follow this with as many Name and Function Value data lines as necessary to define all the Functions.

Each **Name data line** begins the definition of a new Function and indicates whether Function Value data lines follow the Name data line or are to be read from a separate file, and how the Function Value data lines are formatted.

Each **Function Value data line** specifies the value of the function at one or more time points. The format is the same whether the data lines follow the Name data line or are in a separate file.

Data Line Formats

Name Data Line — Function Values Follow the Name Data Line

NAME=**name** DT=**dt** NPL=**npl** PRINT=**print**

Name Data Line — Function Values are in a Separate File

NAME=**name** DT=**dt** NPL=**npl** PRINT=**print** FILE=**filename**

Function Value Data Lines — Function Values at Equal Intervals: **dt > 0**

$\mathbf{f_0}$ $\mathbf{f_1}$ $\mathbf{f_2}$ **...** $\mathbf{f_{npl-1}}$
$\mathbf{f_{npl}}$ **...**
...

Function Value Data Lines — Function Values at Unequal Intervals: **dt = 0**

$\mathbf{t_0}$ $\mathbf{f_0}$ $\mathbf{t_1}$ $\mathbf{f_1}$ $\mathbf{t_2}$ $\mathbf{f_2}$ **...** $\mathbf{t_{npl-1}}$ $\mathbf{f_{npl-1}}$
$\mathbf{t_{npl}}$ $\mathbf{f_{npl}}$ **...**
...

Examples

(1) A response-spectrum curve with unequal period intervals and one pair of values per data line is specified in the input data file:

```
FUNCTION
 NAME=ACCSPEC NPL=1
    .0  .30
    .1  .35
    .2  .70
    .5  .90
    .6  .90
   1.0  .60
   2.0  .50
  100.  .00
```

(2) An acceleration record with equal time steps and three values per data line is stored in file named EQVERT:

```
FUNCTION
 NAME=ELCENT DT=0.002 NPL=3 FILE=EQVERT
```

Description of Variables

Variable	Note	Default	Description
Name Data Line			
name	(1, 2)		Label of a Function being defined
dt	(1, 3)	[0]	Time or period interval spacing for function values [T units]: = 0: Arbitrary time or period values supplied with function values > 0: Equal spacing, starting at t=0
npl	(4)		Number of function values, or time and function-value pairs, defined per Function Value data line; **npl** > 0
print	(5)	[N]	Print flag for function values: = Y: Print values = N: Do not print values
filename	(6)		Optional name of a file containing Function Value data lines
Function Value Data Line			
f_0, f_1, f_2...	(1)		Function values at time or period **t_0, t_1, t_2...**
t_0, t_1, t_2...	(1)		Time or period values

Notes

1. See Topic “Functions” in Chapter “Static and Dynamic Analysis” of the *SAP2000 Analysis Reference*.

2. Each Name data line begins the definition of a new Function. Function labels do not have to be consecutive and may be supplied in any order. Function labels may not be repeated in the data block.

   See Topic “Labels” in Chapter “Labels, Arrays, and Generation” of the *SAP2000 Analysis Reference*.

3. If **dt** > 0 the program reads only function values. If **dt** = 0 the program reads pairs of time and function values; each function value must be specified on the same data line as the corresponding time value.

4. Parameter **npl** must be specified and must be positive.

5. The function values and corresponding times are not echoed in the output file if **print** is left as "N".

6. If **filename** is not specified the program expects to read the Function Value data lines from the input data file immediately after the Name data line.

   If **filename** is specified, it must be a standard Windows filename including extension; drive and directory names are *not* permitted.

SPEC Data Block

This data block defines the response-spectrum analyses to be performed. Multiple analyses may be requested. Each analysis is called a Spec.

Skip this data block if there are no Specs to be defined. Otherwise, prepare data according to the format described below.

For More Information

See Topic "Response Spectrum Analyses" in Chapter "Static and Dynamic Analysis" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

SPEC	Separator
CSYS=	Coordinate System Data Lines
NAME=	Name Data Lines
ACC=	Acceleration Data Lines

Begin the data block with the **SPEC separator**.

Follow this with as many Coordinate System, Name, and Acceleration data lines as necessary to define all the response-spectrum analysis cases.

Each **Coordinate System data line** defines the coordinate system used by all subsequent response-spectrum analysis cases until the next Coordinate System data line is encountered. A Coordinate System data line may only precede a Name data line, not an Acceleration data line.

Each **Name data line** begins the definition of a response-spectrum analysis and indicates the type of modal and directional combinations to be performed. It may be followed by one, two, or three Acceleration data lines.

Each **Acceleration data line** defines the response-spectrum curve to be used in one of the three directions of ground motion.

Data Line Formats

Coordinate System Data Line

CSYS=**csys**

Name Data Line

NAME=**name** ANG=**ang** MODC=**modc** DAMP=**damp** F1=**f1** F2=**f2** DIRF=**dirf**

Acceleration Data Line

ACC=**acc** FUNC=**func** SF=**sf**

Examples

(1) A Response-Spectrum Case uses the same response spectrum for both lateral directions (U1 and U2), and two-thirds as much for the vertical (U3) acceleration. The default CQC method with 5% damping is used for modal combination, and default SRSS method is used for directional combination:

```
SPEC
 NAME=RESPEC1 DAMP=0.05
  ACC=U1 FUNC=ACCSPEC SF=386.4
  ACC=U2 FUNC=ACCSPEC SF=386.4
  ACC=U3 FUNC=ACCSPEC SF=386.4*0.67
```

(2) A Response-Spectrum Case uses the same response spectrum for both lateral directions (U1 and U2). The CQC method with 5% damping is used for modal combination. The directional combination uses the maximum of: 100% of the local 1 response plus 30% of the local 2 response, and 100% of the local 2 response plus 30% of the local 1 response:

```
SPEC
 NAME=RESPEC1 ANG=30 DAMP=0.05 DIRF=0.3
  ACC=U1 FUNC=ACCSPEC SF=386.4
  ACC=U2 FUNC=ACCSPEC SF=386.4
```

Description of Variables

Variable	Note	Default	Description
Coordinate System Data Line			
csys	(1, 3)	[pv(0)]	Coordinate system used to define acceleration directions: = 0: Global coordinate system ≠ 0: Alternate coordinate system label
Name Data Line			
name	(1, 2)		Label of a Response-Spectrum Case being defined
ang	(1)	[0]	Coordinate angle between the response-spectrum local 1 axis and the +X axis of **csys**
modc	(1)	[CQC]	Modal combination type: = CQC: Complete quadratic combination = GMC: General modal combination = SRSS: Square root of the sum of the squares = ABS: Sum of the absolute values
damp	(1)	[0]	Damping value for CQC and GMC modal combinations: 0 ≤ **damp** < 1
f1	(1)	[1]	First rigid-response frequency for GMC modal combination: **f1** > 0 [cyc/T units]
f2	(1)	[0]	Second rigid-response frequency for GMC modal combination: **f2** > **f1** [cyc/T units]: = 0: Infinite frequency > **f1**: Actual frequency
dirf	(1)	[0]	Directional combination scale factor: = 0: Square root of the sum of the squares > 0: Scale factor for secondary directions using sum of the absolute values: 0 < **dirf** ≤ 1

Variable	Note	Default	Description
Acceleration Data Line			
acc	(1)		Direction of ground Acceleration Load, in response-spectrum local coordinates. May be U1, U2, or U3
func	(1, 4)	[0]	Function defining response spectrum curve: = 0: Constant unit acceleration response ≠ 0: Function label
sf	(1)	[1]	Positive scale factor multiplying acceleration (ordinate) values of Function [L/T^2 units]

Notes

1. See Topic "Response Spectrum Analyses" in Chapter "Static and Dynamic Analysis" of the *SAP2000 Analysis Reference*.

2. Each Name data line begins the definition of a new response-spectrum analysis case. Spec labels do not have to be consecutive and may be supplied in any order. Spec labels may not be repeated in the data block.

   See Topic "Labels" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

3. The response-spectrum local coordinate system is defined with respect to the most recently specified coordinate system **csys**. If **csys** is zero, the global system is used. Otherwise **csys** refers to an Alternate Coordinate System defined in the COORDINATE Data Block (page 32). If no **csys** is specified, the global system is used.

4. The shape of the response-spectrum curve is given by a Function defined in the FUNCTION Data Block (page 161). All values for the abscissa and ordinate of this function must be zero or positive.

   If no function is specified, a constant function of unit value for all periods is assumed.

HISTORY Data Block

This data block defines the time-history analyses to be performed. Multiple analyses may be requested. Each analysis is called a History.

Skip this data block if there are no Histories to be defined. Otherwise, prepare data according to the format described below.

For More Information

See Topics "Time-History Analyses" and "Nonlinear Time-History Analyses" in Chapter "Static and Dynamic Analysis" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

HISTORY	Separator
CSYS=	Coordinate System Data Lines
NAME=	Name Data Lines
MODE=	Damping Data Lines
CSYS=	Coordinate System Data Lines
ACC=	Acceleration Data Lines
LOAD=	Load Data Lines

Begin the data block with the **HISTORY separator**.

Follow this with as many Name, Damping, Coordinate System, Acceleration, and Load data lines as necessary to define all the time-history analysis cases.

Each **Name data line** begins the definition of a History analysis case and indicates the type of analysis, the time steps, and overall modal damping to be used.

Each **Damping data line** specifies any modal damping that may differ from the overall modal damping given on the Name data line.

Each **Coordinate System data line** defines the coordinate system used by all subsequent Acceleration data lines until the next Coordinate System data line is encountered.

Each **Acceleration data line** defines the time history of an Acceleration Load acting in a single direction upon the structure for the current History being defined.

Each **Load data line** defines the time history of a Load Case acting upon the structure for the current History being defined.

Data Line Formats

Name Data Line — Linear Transient Analysis

NAME=**name** TYPE=LIN NSTEP=**nstep** DT=**dt** DAMP=**damp**
ENVE=**enve** PREV=**prev**

Name Data Line — Periodic Analysis

NAME=**name** TYPE=PER NSTEP=**nstep** DT=**dt** DAMP=**damp**
ENVE=**enve**

Name Data Line — Nonlinear Transient Analysis

NAME=**name** TYPE=NON NSTEP=**nstep** DT=**dt** DAMP=**damp**
ENVE=**enve** PREV=**prev** FTOL=**ftol** ETOL=**etol** DTMAX=**dtmax**
DTMIN=**dtmin** ITMAX=**itmax** ITMIN=**itmin** CF=**cf** TSTAT=**tstat**

Damping Data Line

MODE=**m0, m1, mi1** DAMP=**damp**

Coordinate System Data Line

CSYS=**csys**

Acceleration Data Line

ACC=**acc** ANG=**ang** FUNC=**func** SF=**sf** TF=**tf** AT=**at**

Load Data Line

LOAD=**load** FUNC=**func** SF=**sf** TF=**tf** AT=**at**

Examples

(1) A structure is subjected to to 30 seconds of seismic ground acceleration in three directions:

```
HISTORY
 NAME=LPRIET TYPE=LIN NSTEP=30*200 DT=1/200 DAMP=0.05
  ACC=U1 FUNC=LPRNS   SF=386.4
  ACC=U2 FUNC=LPREW   SF=386.4
  ACC=U3 FUNC=LPRVERT SF=386.4
```

(2) Load Case "1" is applied to the structure in a triangular pulse with a half-second duration. This is done by using the built-in unit ramp function twice, first to ramp up and then to ramp down:

```
HISTORY
 NAME=TPULSE TYPE=LIN NSTEP=100 DT=0.05 DAMP=0.05
  LOAD=1 FUNC=0 SF=10  TF=0.25
  LOAD=1 FUNC=0 SF=-10 TF=0.25 AT=0.25
```

Description of Variables

Variable	Note	Default	Description
Name Data Line			
name	(1, 3)		Label of a History being defined
nstep	(1)	[1]	Number of output time steps
dt	(1)	[1]	Output time-step size [T units]: **dt** > 0
damp	(1, 4)	[0]	Modal damping ratio for all Modes: $0 \le$ **damp** < 1
enve	(1)	[N]	Whether or not to calculate response envelopes for this History: = Y: Calculate envelopes = N: Do not calculate envelopes

Variable	Note	Default	Description
prev	(1, 5)	[0]	Previously-defined History to use for initial conditions: = 0: Zero initial conditions ≠ 0: Label of a previously-defined History of the same type (LIN or NON)
ftol	(2)	[10^{-5}]	Relative force convergence tolerance for nonlinear analysis: **ftol** > 0
etol	(2)	[10^{-5}]	Relative energy convergence tolerance for nonlinear analysis: **etol** > 0
dtmax	(2)	[**dt**]	Maximum allowed substep size for nonlinear analysis [T units]: 0< **dtmax** ≤ **dt**
dtmin	(2)	[10^{-9} × **dtmax**]	Minimum allowed substep size for nonlinear analysis [T units]: 0< **dtmin** ≤ **dtmax**
itmax	(2)	[100]	Maximum number of force iterations for small substeps in nonlinear analysis: **itmax** ≥ 2
itmin	(2)	[2]	Maximum number of force iterations for large substeps in nonlinear analysis: 2 ≤ **itmin** ≤ **itmax**
cf	(2)	[1]	Convergence factor for nonlinear analysis: **cf** > 0
tstat	(2)	[0]	Period at which and below which modes are treated as static [T units]: **tstat** ≥ 0
Damping Data Line			
m0, m1, mi1	(1, 4)		First Mode number, last Mode number, and Mode number increment
damp	(1, 4)	[0]	Modal damping ratio for Modes **m0** to **m1** by **mi1**: 0 ≤ **damp**< 1

Variable	Note	Default	Description
Coordinate System Data Line			
csys	(1, 6)	[0]	Coordinate system used to define Acceleration Load directions: = 0: Global coordinate system ≠ 0: Alternate Coordinate System label
Acceleration Data Line			
acc	(1, 6)		Direction of ground Acceleration Load, in acceleration local coordinates. May be U1, U2, or U3
ang	(1, 6)	[0]	Coordinate angle between the acceleration local 1 axis and the +X axis of **csys**
func	(1, 8)		Function defining time variation of ground acceleration: = 0: Built-in unit ramp function ≠ 0: Function label
sf	(1)	[1]	Scale factor multiplying ordinate values of Function [L/T^2 units]
tf	(1)	[1]	Scale factor multiplying time (abscissa) values of Function [T units]: **tf** >0
at	(1)	[0]	Arrival time for Function [T units]
Load Data Line			
load	(1, 7)		Label of a Load Case
func	(1, 8)	[0]	Function defining time variation of Load Case: = 0: Built-in unit ramp function ≠ 0: Function label
sf	(1)	[1]	Scale factor multiplying ordinate values of Function

Variable	Note	Default	Description
tf	(1)	[1]	Positive scale factor multiplying time (abscissa) values of Function [T units]
at	(1)	[0]	Arrival time for Function [T units]

Notes

1. See Topic "Time-History Analyses" in Chapter "Static and Dynamic Analysis" of the *SAP2000 Analysis Reference*.

2. See Topic "Nonlinear Time-History Analyses" in Chapter "Static and Dynamic Analysis" of the *SAP2000 Analysis Reference*.

3. Each Name data line begins the definition of a new time-history analysis case. History labels do not have to be consecutive and may be supplied in any order. History labels may not be repeated in the data block.

   The type of History being defined *must* be specified on the Name data line.

   See Topic "Labels" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

4. The damping ratio specified on the Name data line applies to all modes unless overridden on a subsequent Damping data line. If the damping for a particular Mode is specified on more than one Damping data line, the last specification governs.

5. The parameter **prev** must be the label of a History defined earlier in this data block. Both Histories **prev** and **name** must be of the same type (LIN or NON).

6. For each Acceleration data line, the angle **ang** is used to define a separate acceleration local coordinate system with respect to the most recently specified coordinate system **csys**. If **csys** is zero, the global system is used. Otherwise **csys** refers to an Alternate Coordinate System defined in the COORDINATE Data Block (page 32). If no **csys** is specified, the global system is used.

   It is generally recommended, but not required, that the same coordinate system be used for all Acceleration data lines in a given History.

7. The label **load** refers to a Load Case defined in the LOAD Data Block (page 134).

8. The time-variation of each applied Load Case or Acceleration Load is given by a Function defined in the FUNCTION Data Block (page 161). If no function is specified, the built-in ramp function is used.

LANE Data Block

This data block defines the traffic Lanes that are required for bridge moving-load analysis. This data block alone is sufficient to produce influence lines.

Skip this data block if there are no Lanes to be defined. Otherwise, prepare data according to the format described below.

For More Information

See Topic “Roadways and Lanes” in Chapter “Bridge Analysis” of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

LANE	Separator
NAME=	Name Data Lines
PATH=	Path Data Lines

Begin the data block with the **LANE separator**.

Follow this with as many Name and Path data lines as necessary to define all the traffic Lanes. Each **Name data line** begins the definition of a Lane. Each **Path data line** specifies one or more Frame elements that contribute to the Lane. The Path data lines and the elements specified on them must be given in the sequence that a Vehicle travels along the Lane.

Data Line Formats

Name Data Line

NAME=**name**

Path Data Line

PATH=**e0, e1, ei1** ECC=**ecc**

Example

Two Lanes are defined. The element sequence for the first Lane is 1-2-3-4-5-6-7-8. For the second Lane, the sequence is 11-10-9-5-4-3:

```
LANE
 NAME=1
  PATH=1        ECC=0
  PATH=2        ECC=3
  PATH=3,5,1    ECC=6
  PATH=6,8,1    ECC=0
 NAME=2
  PATH=11,9,-1 ECC=0
  PATH=5,3,-1   ECC=-6
```

Description

Variable	Note	Default	Description
Name Data Line			
name	(1, 2)		Label of a Lane being defined
Path Data Line			
e0, e1, ei1	(1, 3, 5)		Labels and label increment for a single Frame element or a one-dimensional array of Frame elements that are part of the Lane
ecc	(1, 5)	[0]	Constant eccentricity [L units]

Notes

1. See Topic “Roadways and Lanes” in Chapter “Bridge Analysis” of the *SAP2000 Analysis Reference*.

2. Each Name data line begins the definition of a new Lane. Lane labels do not have to be consecutive and may be supplied in any order. Lane labels may not be repeated in the data block.

   See Topic “Labels” in Chapter “Labels, Arrays, and Generation” of the *SAP2000 Analysis Reference*.

3. This array may be one-dimensional at most.

   See Topic “Regular Array Specification” (page 15) in this chapter.

4. The order in which the elements are specified on the Path data lines is important. The Lane begins with element **e0** on the first Path data line and ends with element **e1** on the last Path data line, giving the following path:

e0, {**e0** +**ei1**, **e0** + 2 **ei1**, ..., **e1**,}	First Path data line
e0, {**e0** +**ei1**, **e0** + 2 **ei1**, ..., **e1**,}...	Second Path data line
e0, {**e0** +**ei1**, **e0** + 2 **ei1**, ..., **e1**,}...	Intermediate Path data lines
e0, {**e0** +**ei1**, **e0** + 2 **ei1**, ..., **e1**,}	Last Path data line

   The elements shown in braces ({}) are optional on each Path data line. This path should be nearly contiguous and progress in a consistent direction.

5. Each element in the array is assigned the same eccentricity.

VEHICLE Data Block

This data block defines the Vehicle loads that are required for bridge moving-load analysis. Vehicles are always referenced using Vehicle Classes defined in the VEHICLE CLASS Data Block (page 184).

Skip this data block if there are no Vehicles to be defined. Otherwise, prepare data according to the format described below.

For More Information

See Topic "Vehicles" in Chapter "Bridge Analysis" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

VEHICLE	Separator
NAME=	Name Data Lines
W=	Wheel Data Lines

Begin the data block with the **VEHICLE separator**.

Follow this with as many Name and Wheel data lines as necessary to define all the Vehicles.

Each **Name data line** begins the definition of a Vehicle. General-type Vehicles are followed by as many **Wheel data lines** as necessary to define the concentrated and uniform loads that make up the vehicle. Standard-type Vehicles do not require Wheel data lines.

Data Line Formats

Name Data Line — Standard-type

NAME=**name** TYPE=**type** IM=**im**

Name Data Line — General-type

NAME=**name** TYPE=GEN SUPMOM=**supmom** INTSUP=**intsup** OTHER=**other**

Wheel Data Line — Leading Uniform Load and First (Front) Axle

W=**w** P=**p**

Wheel Data Line — Intermediate Uniform Loads and Subsequent Axles

W=**w** D=**dmin, dmax** P=**p**

Wheel Data Line — Trailing Uniform Load and Single Floating Axle

W=**w** PX=**px**

Wheel Data Line — Trailing Uniform Load and Pair of Floating Axles

W=**w** PM=**pm** PXM=**pxm**

Examples

(1) Two identical vehicles are defined. The first uses a built-in standard type, the second defines the same loading explicitly using the general type. Units are kips and feet.

```
VEHICLE
 NAME=HL93K1 TYPE=HL-93K
 NAME=HL93K2 TYPE=GEN
  W=0.64          P=8
  W=0.64 D=14     P=32
  W=0.64 D=14,30 P=32
  W=0.64
```

(2) Two British Standard train loads are defined explicitly using the general type. Units are kilonewtons and meters.

```
VEHICLE
 NAME=RU TYPE=GEN
  W=80
  D=0.8 P=250
  D=1.6 P=250
  D=1.6 P=250
  D=1.6 P=250
  D=0.8
  W=80
 NAME=RL TYPE=GEN
  W=25
  W=50 D=100
  W=25 PX=200
```

Description

Variable	Note	Default	Description
Name Data Line			
name	(1, 2)		Label of a Vehicle being defined
type	(1)	[GEN]	Vehicle type: = GEN: General, to be explicitly defined by Wheel data lines ≠ GEN: Built-in standard type
im	(1)	[0]	Dynamic load allowance for HL-93 standard Vehicle types only [percentage]
Wheel Data Line			
w	(1, 3)	[0]	Uniform load [F/L units]
dmin	(1, 3)		Minimum distance between the current and preceding axles. Required on all Intermediate Wheel data lines. **dmin** > 0 [L units]
dmax	(1, 3)	[**dmin**]	Maximum distance between the current and preceding axles. *Only one Intermediate Wheel data line may have* **dmax** > **dmin**. All others must have **dmax** = **dmin**. Use **dmax** = 0 to indicate infinite maximum distance [L units]

Variable	Note	Default	Description
p	(1, 3)	[0]	Concentrated axle weight [F units]
px	(1, 3)	[0]	Floating concentrated axle weight [F units]
pm	(1, 3)	[0]	Floating concentrated axle weight for span moments in Lane elements only [F units]
pxm	(1, 3)	[0]	Floating concentrated axle weight for all response quantities except span moments in Lane elements [F units]
supmom	(1)	[Y]	Whether or not this Vehicle is to be used for negative span moments in Lane elements over the supports: = Y: Yes = N: No
intsup	(1)	[Y]	Whether or not this Vehicle is to be used for vertical forces at interior piers in Frame elements, reactions, and/or spring supports: = Y: Yes = N: No
other	(1)	[Y]	Whether or not this Vehicle is to be used for response quantities other than those listed for **supmom** and **intsup**: = Y: Yes = N: No

Notes

1. See Topic “Vehicles” in Chapter “Bridge Analysis” of the *SAP2000 Analysis Reference*.

2. Each Name data line begins the definition of a new Vehicle. Vehicle labels do not have to be consecutive and may be supplied in any order. Vehicle labels may not be repeated in the data block.

   See Topic “Labels” in Chapter “Labels, Arrays, and Generation” of the *SAP2000 Analysis Reference*.

3. An axle is the location of a concentrated load or a change in uniform load, or both. Axles are specified in order from the front to the rear of the vehicle. The floating axles have no fixed position with respect to the other axles.

   The number of Wheel data lines permitted is:

   - Zero or one Leading data line
   - Zero or more Intermediate data lines
   - Zero or one Trailing data line

   You must specify **dmin** on all Intermediate Wheel data lines. You may not specify **dmax** > **dmin** on more than one Intermediate Wheel data lines, i.e., at most one pair of axles may have a variable distance between them.

VEHICLE CLASS Data Block

This data block defines the Classes, or groups, of Vehicle loads that are required for bridge moving-load analysis.

Skip this data block if there are no Vehicle Classes to be defined. Otherwise, prepare data according to the format described below.

For More Information

See Topic "Vehicle Classes" in Chapter "Bridge Analysis" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

VEHICLE CLASS	Separator
NAME=	Name Data Lines
VEHI=	Vehicle Data Lines

Begin the data block with the **VEHICLE CLASS separator**.

Follow this with as many Name and Vehicle data lines as necessary to define all the Vehicle Classes.

Each **Name data line** begins the definition of a Vehicle Class. Each **Vehicle data line** specifies a Vehicle that belongs to the Class being defined by the most recent Name data line.

Data Line Formats

Name Data Line

NAME=**name**

Vehicle Data Line

VEHI=**vehi** SF=**sf**

Example

A Class is defined that contains the AASHTO HS20-44 Truck and Lane Loads and the Alternate Military Load. These Vehicles must have been previously defined in the VEHICLE data block:

```
VEHICLE CLASS
 NAME=HS2044
  VEHI=HS2044
  VEHI=HS2044L
  VEHI=AML
```

Description

Variable	Note	Default	Description
Name Data Line			
name	(1, 2)		Label of a Vehicle Class being defined
Vehicle Data Line			
vehi	(1, 3)		Label of a Vehicle being added to the Class
sf	(1)	[1]	Scale factor multiplying Vehicle load **vehi**

Notes

1. See Topic “Vehicle Classes” in Chapter “Bridge Analysis” of the *SAP2000 Analysis Reference*.

2. Each Name data line begins the definition of a new Vehicle Class. Class labels do not have to be consecutive and may be supplied in any order. Class labels may not be repeated in the data block.

   See Topic “Labels” in Chapter “Labels, Arrays, and Generation” of the *SAP2000 Analysis Reference*.

3. The label **vehi** refers to a Vehicle defined in the VEHICLE Data Block (page 179). A Vehicle may be included in more than one Class.

BRIDGE RESPONSE Data Block

This data block allows you to selectively control for which joints and elements the computationally-intensive moving-load analysis is to be performed. Only the results specifically requested in this data block will be calculated.

Skip this data block no moving-load analysis results are desired or if no Moving Load cases have been defined. Otherwise, prepare data according to the format described below.

For More Information

See Topic "Moving Load Response Control" in Chapter "Bridge Analysis" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

BRIDGE RESPONSE	Separator
ELEM=	Element Data Lines
ADD=	Add Data Lines
REM=	Remove Data Lines

Begin the data block with the **BRIDGE RESPONSE separator**.

Follow this with as many Element, Add, and Remove data lines as necessary to specify all of the analysis results desired.

Each **Element data line** defines the element type and the response types that apply to the subsequent Add and Remove data lines until the next Element data line is encountered. For the purposes of this data block, joints are treated as a type of element.

Each **Add data line** lists a regular array of joints or elements for which the selected response types are to be calculated. Each **Remove data line** lists a regular array of joints or elements for which the selected response types are *not* to be calculated. Data lines are processed in the order they are supplied in the input data file.

Data Line Formats

Element Data Line — Joints

ELEM=JOINT TYPE=**jtypes**

Element Data Line — Frame Elements

ELEM=FRAME

Add Data Line

ADD=**e0, e1, ei1...**

Remove Data Line

REM=**e0, e1, ei1...**

Examples

(1) To calculate all possible moving-load response, specify:

```
BRIDGE RESPONSE
 ELEM=JOINT TYPE=DISP,REAC,SPRING
  ADD=*
 ELEM=FRAME
  ADD=*
```

(2) To get only the Frame element forces, specify:

```
BRIDGE RESPONSE
 ELEM=FRAME
  ADD=*
```

Description of Variables

Variable	Note	Default	Description
Element Data Line			
jtypes	(1)		One or more response types at the joints. May be any of the following: = DISP: Displacements = REAC: Reactions = SPRING: Spring forces
Add Data Line			
e0, e1, ei1...	(2, 3)		Labels and label increments for an array of one or more joints or elements for which the selected output is to be output
Remove Data Line			
e0, e1, ei1...	(2, 4)		Labels and label increments for an array of one or more joints or elements for which the selected output is not to be output

Notes

1. See Topic “Moving Load Response Control” in Chapter “Bridge Analysis” of the *SAP2000 Analysis Reference*.
2. See Topic “Regular Array Specification” (page 15) in this chapter.
3. Each Add data line may refer to a single element **e0**, or an array of elements **e0, e1, ei1...** having one, two or three dimensions. All elements are of the type specified on the most recent Element data line. For the purposes of this data block, joints are treated as being a type of element.
4. Each Remove data line may refer to a single element **e0**, or an array of elements **e0, e1, ei1...** having one, two or three dimensions. All elements are of the type specified on the most recent Element data line. For the purposes of this data block, joints are treated as being a type of element.

MOVING LOAD Data Block

This data block defines the Moving Load cases that determine the response to the Vehicles in the Vehicle Classes moving along the traffic Lanes.

Skip this data block if there are no Moving Loads to be defined. Otherwise, prepare data according to the format described below.

For More Information

See Topic "Moving Load Cases" in Chapter "Bridge Analysis" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

MOVING LOAD		Separator
	CORR=	Control Data Line
	NAME=	Name Data Lines
	CLASS=	Assignment Data Lines

Begin the data block with the **MOVING LOAD separator**.

Follow this with a single Control data line, and then as many Name and Class data lines as necessary to define all the Moving Load cases.

The **Control data line** is optional, and specifies parameters that control the moving of Vehicles along the Lanes.

Each **Name data line** begins the definition of a Moving Load. This is followed by one or more **Assignment data lines** that each assign a Class to one or more Lanes.

Data Line Formats

Control Data Line

CORR=**corr** QUICK=**quick** TOL=**tol**

Name Data Line

NAME=**name** RF=**rf1, rf2, rf3...**

Assignment Data Line

CLASS=**class** LANE=**lanes** SF=**sf** LMIN=**lmin** LMAX=**lmax**

Examples

All of the following examples assume a four-Lane bridge.

(1) A single Moving Load is defined that assigns a single Vehicle Class to any number of the Lanes

```
MOVING LOAD
 NAME=HS20 RF=1,1,0.9,0.75
  CLASS=HS20
```

(2) A single Moving Load is defined that assigns an overload Vehicle Class to any one Lane, and an ordinary Vehicle Class to zero or one other Lane:

```
MOVING LOAD
 NAME=OVER RF=1,1,0.9,0.75
  CLASS=OVER LANE=* LMIN=1 LMAX=1
  CLASS=HS20 LANE=* LMIN=0 LMAX=1
```

(3) Two Moving Loads are defined. The first assigns an overload Vehicle Class to Lane 1, the second to Lane 4. In both cases, the Lane adjacent to the overload is empty, and ordinary Vehicles may occupy the remaining two Lanes:

```
MOVING LOAD
 CORR=Y QUICK=0
 NAME=OVER1 RF=1
  CLASS=OVER LANE=1 LMIN=1
  CLASS=HS20 LANE=3,4
 NAME=OVER4 RF=1
  CLASS=OVER LANE=4 LMIN=1
  CLASS=HS20 LANE=1,2
```

Description

Variable	Note	Default	Description
Control Data Line			
corr	(3)	[N]	Whether or not to calculate the other force components corresponding to the maximum and minimum Frame element forces: = Y: Calculate correspondence = N: Do not calculate correspondence
quick	(4)	[0]	Parameter controlling the Quick or "Exact" methods of response calculation: = 0: Use the "Exact" method > 0: Degree of approximation for the Quick method
tol	(5)	[.0001]	Relative tolerance for simplifying influence lines
Name Data Line			
name	(1, 2)		Label of a Moving Load being defined
rf1, rf2, rf3...	(1, 6)	[1, **rf1**, **rf2**...]	Multiple-lane scale factors applied to the Moving Load case if the number of loaded Lanes is one (**rf1**), two (**rf2**), three (**rf3**), and so on.
Assignment Data Line			
class	(1, 7)		Vehicle Class that loads Lanes **lanes**
lanes	(1, 8)	[∗]	List of one or more Lanes loaded by **class**. Use ∗ to indicate all Lanes (the default)
sf	(1)	[1]	Scale factor multiplying the Vehicle loads in **class**
lmin	(1, 9)	[0]	Minimum number of Lanes to be loaded by **class** for this assignment

Variable	Note	Default	Description
lmax	(1, 9)	[0]	Maximum number of Lanes to be loaded by **class** for this assignment: = 0: All of Lanes **lanes** ≥ **lmin**: Specified maximum **lmax**

Notes

1. See Topic "Moving Load Cases" in Chapter "Bridge Analysis" of the *SAP2000 Analysis Reference*.

2. Each Name data line begins the definition of a new Moving Load case. Moving Load labels do not have to be consecutive and may be supplied in any order. Moving Load labels may not be repeated in the data block.

   See Topic "Labels" in Chapter "Labels, Arrays, and Generation" of the *SAP2000 Analysis Reference*.

3. See Topic "Correspondence" in Chapter "Bridge Analysis" of the *SAP2000 Analysis Reference* for more information on **corr**.

4. See Topic "Exact and Quick Response Calculation" in Chapter "Bridge Analysis" of the *SAP2000 Analysis Reference* for more information on **quick**.

5. See Topic "Influence Line Tolerance" in Chapter "Bridge Analysis" of the *SAP2000 Analysis Reference* for more information on **tol**.

6. All omitted values default to the last specified multiple-lane scale factor on the current data line. If none are specified, all reduction factors default to unity.

7. The label **class** refers to a Vehicle Class defined in the VEHICLE CLASS Data Block (page 184).

8. Any number of Lanes defined in the LANE Data Block (page 176) may be listed. No Lane should be listed more than once. If no Lanes are listed, the default is to consider all Lanes.

   A maximum of ten Lanes may be listed on a single assignment data line. If more than ten Lanes are needed for a single assignment, you may use additional data lines that contain only the specification LANE=**lanes**. Each of these additional data lines may list up to ten Lanes. The CLASS=, SF=, LMAX=, and LMIN= specifications are not permitted.

9. Parameter **lmax** must be greater than or equal to **lmin**, unless **lmax** is zero which indicates that all listed lanes may be loaded.

COMBO Data Block

This data block defines various types of combinations of Load Cases, Vibration Modes, Specs, Histories, Moving Load Cases, and other Combos. Multiple Combos may be defined.

Skip this data block if there are no Combos to be defined. Otherwise, prepare data according to the format described below.

For More Information

See Topic "Combos" in Chapter "Static and Dynamic Analysis" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

COMBO	Separator
NAME=	Name Data Lines
LOAD=	Load Data Lines
MODE=	Mode Data Lines
SPEC=	Spec Data Lines
HIST=	History Data Lines
MOVE=	Moving Load Data Lines
COMB=	Combo Data Lines

Begin the data block with the **COMBO separator**.

Follow this with as many Name, Load, Mode, Spec, History, Moving Load, and Combo data lines as necessary to define all the Combos.

Each **Name data line** identifies the Combo being defined and its type.

Each **Load data line** specifies the contribution of a Load Case to the Combo. Each **Mode data line** specifies the contribution of a single Vibration Mode to the Combo. Each **Spec data line** specifies the contribution of a Response-Spectrum Case to the Combo. Each **History data line** specifies the contribution of a Time-

History Case to the Combo. Each **Moving Load data line** specifies the contribution of a Moving-Load Case to the Combo. Each **Combo data line** specifies the contribution of a previously-defined Combo to the Combo being defined.

Data Line Formats

Name Data Line

NAME=**name** TYPE=**type**

Load Data Line

LOAD=**load** SF=**sf**

Mode Data Line

MODE=**mode** SF=**sf**

Spec Data Line

SPEC=**spec** SF=**sf**

History Data Line

HIST=**hist** SF=**sf**

Move Data Line

MOVE=**move** SF=**sf**

Combo Data Line

COMB=**comb** SF=**sf**

Example

Suppose that Load Cases DL and WIND are dead load and transverse wind load, respectively, and that a response-spectrum analysis named EQ has been performed. The first Combo combines the dead load with the Response Spectrum Case, automatically accounting for both positive and negative senses of the Spec. The second and third Combos combine the dead load with the wind load acting in two opposite directions. The fourth Combo takes the envelope of the first three Combos to find the most severe response:

```
COMBO
 NAME=DLEQ  TYPE=ADD
  LOAD=DL    SF=1
  SPEC=EQ    SF=1
 NAME=DLWIND1  TYPE=ADD
  LOAD=DL    SF=1
  LOAD=WIND  SF=1
 NAME=DLWIND2  TYPE=ADD
  LOAD=DL    SF=1
  LOAD=WIND  SF=-1
 NAME=WORST  TYPE=ENVE
  COMB=DLEQ     SF=1
  COMB=DLWIND1  SF=1
  COMD=DLWIND2  SF=1
```

Description of Variables

Variable	Note	Default	Description
Name Data Line			
name	(1, 2)		Label of a Combo being defined
type	(1)	[ADD]	Type of Combo: = ADD: Algebraic sum of the contributing cases = ABS: Sum of the absolute values of the contributing cases = SRSS: Square root of the sum of the squares of the contributing cases = ENVE: Envelope of the contributing cases

Variable	Note	Default	Description
Load Data Line			
load	(1, 3)		Label of a Load Case
sf	(1)	[1]	Scale factor multiplying Load Case **load**
Mode Data Line			
mode	(1, 4)		Vibration Mode number
sf	(1)	[1]	Scale factor multiplying Vibration Mode **mode**
Spec Data Line			
spec	(1, 5)		Label of a Spec
sf	(1)	[1]	Scale factor multiplying Response-Spectrum Case **spec**
History Data Line			
hist	(1, 6)		Label of a History
sf	(1)	[1]	Scale factor multiplying Time-History Case **hist**
Moving Load Data Line			
move	(1, 7)		Label of a Moving Load
sf	(1)	[1]	Positive scale factor multiplying Moving Load Case **move**
Combo Data Line			
comb	(1, 8)		Label of a previously-defined Combo
sf	(1)	[1]	Positive scale factor multiplying Combo **comb**

Notes

1. See Topic “Combos” in Chapter “Static and Dynamic Analysis” of the *SAP2000 Analysis Reference*.

2. Each Name data line begins the definition of a new Combo. Combo labels do not have to be consecutive and may be supplied in any order. Combo labels may not be repeated in the data block.

   See Topic “Labels” in Chapter “Labels, Arrays, and Generation” of the *SAP2000 Analysis Reference*.

3. The label **load** refers to a Load Case defined in the LOAD Data Block (page 134).

4. The Vibration Mode number **mode** may be any number from 1 to the number of modes requested in the MODES Data Block (page 157). If **mode** is greater than the number of modes that were actually calculated, the contribution to the Combo is zero.

5. The label **spec** refers to a response-spectrum analysis defined in the SPEC Data Block (page 165).

6. The label **hist** refers to a time-history analysis defined in the HISTORY Data Block (page 169).

7. The label **move** refers to a Moving Load analysis defined in the MOVING LOAD Data Block (page 189).

8. The label **comb** refers to a Combo that was previously defined in this data block.

OUTPUT Data Block

This data block allows you to specify the joint and element results to be written to the results output (.OUT) file.

Skip this data block if no joint or element results are to be written out. Otherwise, prepare data according to the format described below.

For More Information

See Topic "Joint and Element Output Control" in Chapter "The Output Files" of the *SAP2000 Analysis Reference*.

Data Block Format

The format of the data block is summarized in the table below:

OUTPUT	Separator
ELEM=	Element Data Lines

Begin the data block with the **OUTPUT separator**.

Follow this with as many Element data lines as necessary to specify all of the analysis results desired.

Each **Element data line** defines the response types and analysis cases to be printed for a given element type. For the purposes of this data block, joints are treated as a type of element. A given element type may be repeated on different Element data lines in order to specify different combinations of response types and analysis cases.

Data Line Formats

Element Data Line — Joints

ELEM=JOINT TYPE=**jtypes** LOAD=**loads** MODE=**modes** SPEC=**specs** HIST=**hists** MOVE=**moves** COMB=**combs**

Element Data Line — Frame Elements

ELEM=FRAME TYPE=**frtypes** LOAD=**loads** MODE=**modes**
SPEC=**specs** HIST=**hists** MOVE=**moves** COMB=**combs**

Element Data Line — Shell Elements

ELEM=SHELL TYPE=**shtypes** LOAD=**loads** MODE=**modes**
SPEC=**specs** HIST=**hists** COMB=**combs**

Element Data Line — Plane Elements

ELEM=PLANE TYPE=**pltypes** LOAD=**loads** MODE=**modes**
SPEC=**specs** HIST=**hists** COMB=**combs**

Element Data Line — Asolid Elements

ELEM=ASOLID TYPE=**pltypes** LOAD=**loads** MODE=**modes**
SPEC=**specs** HIST=**hists** COMB=**combs**

Element Data Line — Solid Elements

ELEM=SOLID TYPE=**pltypes** LOAD=**loads** MODE=**modes**
SPEC=**specs** HIST=**hists** COMB=**combs**

Element Data Line — Nllink Elements

ELEM=NLLINK TYPE=**frtypes** LOAD=**loads** MODE=**modes**
SPEC=**specs** HIST=**hists** COMB=**combs**

Example

Suppose Load Cases “DL”, “LL”, and “WIND” have been defined, as well as Response-Spectrum case “EQ”. Joints displacements are requested for all Load Cases. Joint reactions and Frame element internal forces are requested for the Spec:

```
OUTPUT
 ELEM=JOINT TYPE=DISP LOAD=*
 ELEM=JOINT TYPE=REAC SPEC=EQ
 ELEM=FRAME TYPE=FORCE SPEC=EQ
```

Description of Variables

Variable	Note	Default	Description
Element Data Line			
jtypes	(1)		One or more response types at the joints. May be any of the following: = DISP: Displacements = APPL: Applied and Inertial loads = REAC: Restraint, Constraint, Spring, and Nllink forces
frtypes	(1)		One or more response types for the Frame and Nllink elements. May be any of the following: = FORCE: Internal forces = JOINTF: Joint forces
shtypes	(1)		One or more response types for the Shell elements. May be any of the following: = FORCE: Internal forces = STRESS: Stresses = JOINTF: Joint forces
pltypes	(1)		One or more response types for the Plane, Asolid, and Solid elements. May be any of the following: = STRESS: Stresses = JOINTF: Joint forces
loads	(1, 2)		Labels of one or more Load Cases, or "∗" for all Load Cases
modes	(1, 2, 3)		Enter "∗" for all Modes
specs	(1, 2)		Labels of one or more Specs, or "∗" for all Specs
hists	(1, 2)		Labels of one or more Histories, or "∗" for all Histories
moves	(1, 2)		Labels of one or more Moving Loads, or "∗" for all Moving-Loads

Variable	Note	Default	Description
combs	(1, 2)		Labels of one or more Combos, or "∗" for all Combos

Notes

1. See Topic "Joint and Element Output Control" in Chapter "The Output Files" of the *SAP2000 Analysis Reference*.

2. There is a maximum of 10 entries per case-type keyword (e.g., LOAD= or SPEC=) permitted on a single Element data line. If more than 10 entries are needed for a particular case type, use multiple Element data lines.

3. Only MODE=∗ is permitted. Individual modes cannot be selected. If individual modes are desired, place them in a Combo and select the Combo.

END Data Block

This data block indicates the end of data to be read by SAP2000 from the input data file. All data lines following the END Data Block are ignored by the program. This may be used to place extensive comment data at the end of the file, or to place data that is to be read by other programs that use a SAP2000 input data file.

This data block is not needed in the usual case where SAP2000 is to read all data in the file.

Data Block Format

The format of the data block is summarized in the table below:

END	Separator

This data block consists only of the END separator.